

**EXPLORING GAME DESIGN
THROUGH HUMAN-AI COLLABORATION**

Malmö University,
Studies in Computer Science No 20,
Doctoral Dissertation

© Alberto Alvarez, 2022
ISBN 978-91-7877-307-7 (print)
ISBN 978-91-7877-308-4 (pdf)
Media-Tryck, Lund 2022

ALBERTO ALVAREZ
**EXPLORING GAME
DESIGN THROUGH
HUMAN-AI
COLLABORATION**

Malmö University, 2022
Faculty of Technology and Society
Department of Computer Science and Media Technology

Studies in Computer Science

Faculty of Technology and Society
Malmö University

1. Jevinger, Åse. Toward intelligent goods: characteristics, architectures and applications, 2014, Doctoral dissertation.
2. Dahlskog, Steve. Patterns and procedural content generation in digital games: automatic level generation for digital games using game design patterns, 2016, Doctoral dissertation.
3. Fabijan, Aleksander. Developing the right features: the role and impact of customer and product data in software product development, 2016, Licentiate thesis.
4. Paraschakis, Dimitris. Algorithmic and ethical aspects of recommender systems in e-commerce, 2018, Licentiate thesis.
5. Hajinasab, Banafsheh. A Dynamic Approach to Multi Agent Based Simulation in Urban Transportation Planning, 2018, Doctoral dissertation.
6. Fabijan, Aleksander. Data-Driven Software Development at Large Scale, 2018, Doctoral dissertation.
7. Bugeja, Joseph. Smart Connected Homes: Concepts, Risks, and Challenges, 2018, Licentiate thesis.
8. Alkhabbas, Fahed. Towards Emergent Configurations in the Internet of Things, 2018, Licentiate thesis.
9. Paraschakis, Dimitris. Sociotechnical Aspects of Automated Recommendations: Algorithms, Ethics, and Evaluation, 2020, Doctoral dissertation.
10. Tegen, Agnes. Approaches to Interactive Online Machine Learning, 2020, Licentiate thesis.
11. Alvarez, Alberto. Exploring the Dynamic Properties of Interaction in Mixed-Initiative Procedural Content Generation, 2020, Licentiate thesis.
12. Alkhabbas, Fahed. Realizing Emergent Configurations in the Internet of Things, 2020, Doctoral Dissertation.
13. Ashouri, Majid. Towards Supporting IoT System Designers in Edge Computing, 2021, Licentiate Thesis.
14. Bugeja, Joseph. On Privacy and Security in Smart Connected Homes, 2021, Doctoral Dissertation.
15. Azadvar, Ahmad. Predictive Psychological Player Profiling, 2021, Licentiate Thesis.
16. Holmberg, Lars. Human In Command Machine Learning, 2021, Licentiate Thesis.
17. John, Meenu Mary. Design Methods and Processes for ML/DL models, 2021, Licentiate thesis.

18. Tegen, Agnes. Interactive Online Machine Learning, 2022, Doctoral Dissertation.
19. Ouhaichi, Hamza. Towards designing a flexible multimodal learning analytics system, 2022, Licentiate thesis.
20. Alvarez, Alberto. Exploring Game Design through Human-AI Collaboration, 2022, Doctoral Dissertation.

The publication is also available at:

<http://doi.org/10.24834/isbn.9789178773084>

To the brave Venezuelans; the sun will shine again.

ABSTRACT

Game design is a hard and multi-faceted task that intertwines different gameplay mechanics, audio, level, graphic, and narrative facets. Games' facets are developed in conjunction with others with a common goal that makes games coherent and interesting. These combinations result in plenty of games in diverse genres, which usually require a collaboration of a diverse group of designers. Collaborators can take different roles and support each other with their strengths resulting in games with unique characteristics. The multi-faceted nature of games and their collaborative properties and requirements make it an exciting task to use Artificial Intelligence (AI). The generation of these facets together requires a holistic approach, which is one of the most challenging tasks within computational creativity. Given the collaborative aspect of games, this thesis approaches their generation through Human-AI collaboration, specifically using a mixed-initiative co-creative (MI-CC) paradigm. This paradigm creates an interactive and collaborative scenario that leverages AI and human strengths with an alternating and proactive initiative to approach a task. However, this paradigm introduces several challenges, such as Human and AI goal alignment or competing properties.

In this thesis, game design and the generation of game facets by themselves and intertwined are explored through Human-AI collaboration. The AI takes a colleague's role with the designer, arising multiple dynamics, challenges, and opportunities. The main hypothesis is that AI can be incorporated into systems as a collaborator, enhancing design tools, fostering human creativity, and reducing workload. The challenges and opportunities that arise from this are explored, discussed, and approached throughout the thesis. As a result, multiple approaches and methods such as quality-diversity algorithms and designer modeling are proposed to generate game facets in tandem with humans, create a better workflow, enhance the interaction, and establish adaptive experiences.

Keywords. Mixed-Initiative, Procedural Content Generation, Quality Diversity, Computer Games, Computational Creativity

LIST OF PUBLICATIONS

Included Papers

- [1] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018.
- [2] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018.
- [3] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, 2019.
- [4] A. Alvarez and J. Font, “Learning the designer’s preferences to drive evolution,” in *Proceedings of the 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation1*, EvoApplications ’20, 2020.
- [5] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *IEEE Transactions on Games*, pp. 1–1, 2020. Preprint available: <https://arxiv.org/abs/2003.03377>.
- [6] S. Tolinsson, A. Flodhag, A. Alvarez, and J. Font, “To make sense of procedurally generated dungeons,” in *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY ’20, (New York, NY, USA), pp. 384—387, Association for Computing Machinery, 2020.

- [7] A. Alvarez, E. Grevillius, E. Olsson, and J. Font, “Questgram [qg]: Toward a mixed-initiative quest generation tool,” in *Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG’21*, (New York, NY, USA), Association for Computing Machinery, 2021.
- [8] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, “Assessing the effects of interacting with map-elites,” in *The 17th AAAI conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE’21*, pp. 1–8, 2021.
- [9] A. Alvarez, J. M. Font Fernandez, and J. Togelius, “Designer modeling through design style clustering,” *IEEE Transactions on Games*, pp. 1–10, 2022.
- [10] A. Alvarez and J. Font, “Tropetwist: Trope-based narrative structure generation,” in *the 13th Workshop on Procedural Content Generation, FDG*, pp. 1–8, 2022.
- [11] A. Alvarez, J. Font, and J. Togelius, “Story designer: Towards a mixed-initiative tool to create narrative structures,” in *the 17th International Conference on the Foundations of Digital Games, FDG’22*, pp. 1–8, 2022.
- [12] T. Larsson, J. Font, and A. Alvarez, “Towards ai as a creative colleague in game level design,” in *the 18th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE’22*, 2022.

Personal Contribution and Clarification

Publication [5] is an extended version of [3], adding new features to the proposed algorithm, experiments, and evaluations.

For all publications above except for [1], [2], and [7], the first author was the main contributor with regards to inception, planning, execution and writing of the research.

For [1], [2], and [7] the first author contributed to inception, planning, and execution of key parts of the publications, and was the main contributor regarding the writing of the research. For [12], the last author was the main contributor with regards to inception and planning. For papers [1], [2], [6], [7], and [12], the research originated from the following theses for which I was [co]supervisor, respectively:

- C. Nolasco, and A. Österman, "A Study on Mixed-Initiative for Fostering Creativity in Game Design," M.S. Thesis, DVMT, TS, Malmö University, Malmö, 2018, <http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-20051>
- S. Johansson, "A Study on Fitness Functions and Their Impact in PCG," M.S. Thesis, DVMT, TS, Malmö University, Malmö, 2018. <http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-20829>
- S. Tolinsson, and A. Flodhag, "To make sense of dungeons," B.S. Thesis, DVMT, TS, Malmö University, Malmö, 2020. <http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-20865>
- E. Grevilius, and E. Olsson, "Mixed-initiative quest generation," B.S. Thesis, DVMT, TS, Malmö University, Malmö, 2020. <http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-20600>
- T. Larsson, "AI as a Creative Colleague in Game Level Design," B.S. Thesis, DVMT, TS, Malmö University, Malmö, 2022. <http://urn.kb.se/resolve?urn=urn:nbn:se:mau:diva-53097>

ACKNOWLEDGEMENT

I want to thank all my supervisors, Jose Font, Julian Togelius, Steve Dahlskog, and Nancy L. Russo; your advice, uncountable conversations, mentorship, and friendship is paramount to me and has contributed enormously to the conception of this thesis and my research path. I would like to thank especially my main supervisor, Jose; your teaching on AI back in Spain during my undergraduate really sparked the curiosity on me, which would then become the roadmap I took to be where I am now. I hope for many more years of collaboration!

Likewise, I would like to thank my PhD examiner Paul Davidsson, and the follow-up group, Carl Magnus Olsson and Arezoo Sarkheily-Hägele, for their academic guidance. Your suggestions and discussions on my study plan helped shape several of the decisions on this thesis. I would like to thank Sebastian Risi, opponent at my Licentiate Seminar, Marco Scirea, opponent at the Final Seminar, and Georgios N. Yannakakis, opponent at the PhD Defence, for their participation, guidance, and discussions with me. I do feel that I have leveled up for every one of you!

I would like to thank all my PhD colleagues at MAU (and outside) for uncountable conversations, discussions, fikas, and random encounters. Your support and ears to discuss much more than research have made this an extraordinary journey! I want to also thank the faculty for your support, friendship, and guidance in my research and teaching. I cannot leave this without thanking all my students throughout the years, those that I supervised (whose works I proudly incorporate as part of this thesis), and those that I briefly taught. I have learned so much from all of you!

Finally, I would like to thank my friends outside of academia and family for always being there no matter what and supporting me every step of the way. This would not be possible without all of you!

PREFACE

Since I can remember, games have always been part of my life. It was a few days before the start of the new millennium when my brother and I got a Nintendo 64 with *Zelda: Ocarina of Time*. I played tirelessly until 06:00 am, which was probably the first time I stayed awake so long. Likewise, I remember playing *Metal Gear Solid 1* on my PS1, the thrill when passing each boss and scenario, sneaking around hoping to don't get caught and the exclamation mark appearing was just incredible. There was a caveat though, I did not have a memory card; thus, I had to finish the game without turning off my PS1. I hope that you, as a reader, can imagine the challenge that meant for a 10 years old. Passed 3 days of non-stop playing, and I finished the game. And with just the press of a button, the game restarted as if nothing had happened. Yet, something happened; these experiences and the ones I left out shaped my path.

I studied game design for 6 years, and now I am on the brink of finishing my PhD in game design and the use of artificial intelligence to generate game content. Many experiences through my education have shaped my interest in this research topic; from creating a small text adventure to spending long hours to build a game engine just to prototype a garbage truck simulator, to doing a small god game like *black and white* (although those villagers never followed me). You see, I had no doubt I wanted to do games, but I had no particular creative skill that I could hone to make something out of this, or so I thought. I was in awe playing these games for their fun and enjoyable moments, design, art, stories, and the experience they created. Yet, I saw myself as completely out of touch with these elements, and I think I can [nowadays] narrow it down to what I felt was a lack of creativity and creative thinking. How does creativity works? What elements constitute the creative process? How does it arise within us? This shaped, to a large extent, my interest in the exploration of artificial intelligence within creative domains and for creative tasks.

Creativity has always baffled me. Nowadays, I believe creativity is an outstanding ability that we [all] possess. Some believe that they are not creative creatures, but in reality, and being candid about it, we are all creative and creative actions are taken all the time. We are all confronted with challenges and situations that require us to be creative in how we approach them. Whether this means being creative for artistic purposes such as painting or developing games, questioning your field and writing a dissertation in political science, or creating variations on how to write your name, these all require creativity. Creativity varies on the task, process, outcome, and personal perception, which might be why people feel non-creative [1]. Nevertheless, creativity, like our other abilities, can be refined, developed, and fostered. The thesis that you are about to read challenges the misconception that there are non-creative people, yet it is not about creativity. This thesis explores a computational creativity system that collaborates with humans in the exciting and creative area of game design to enhance, augment, and support these human capabilities. However, in order to do so, this system needs to show creative output and, to some extent, recognize the human creative process. I present a computational creativity system that I have used to study game design and, in that endeavor, analyze and research [computational] creativity, the very thing that baffles me.

This thesis and its implications

Throughout this thesis, I will explore Human-AI collaboration as an approach to co-create game content. As a result, it is explored how to approach game design facets, particularly level and narrative design, with AI algorithms in tandem with human designers. A *Computational Designer*. A core part of the research concerns how to establish an *effective* collaboration and collaborative environment. This is addressed by studying the *Computational Designer* in a collaborative system, the Evolutionary Dungeon Designer (EDD). Studying these collaborative systems made two elements clear and worth their exploration; the different properties that arise as we establish this collaboration and the need for user models adjusted to the design task, process, and creator [2]. The former is, to some extent, a consequence of these Procedural Content Generation systems as discussed in [3], and in general, the tradeoffs that exist among the different techniques (e.g., expressive generators, controllable output, explainable processes, or adaptive systems). These are exacerbated and complexified when moved toward collaborative tasks since the human's current work

and design need to be considered. One way to address these is to create more adaptive experiences through modeling designers and their design process (i.e., *Designer Modeling*).

Now, you may feel inclined to say that by studying this *Computational Designer*, one might replace human designers with the caveat that we need their design traces and examples. This is an important ethical point on these types of systems, particularly AI-assisted design tools. A parallel discussion goes on on Twitter every now and then. For instance, Open AI's DALL-E 2 and GPT-3, Google's Imagen, or Github's CoPilot are just some of the systems that are in the eye of the hurricane regarding the use of human creative output to create these models, and the consideration and impact to human creativity and creative works. While you do not necessarily need to engage or agree in these discussions, they are relevant, important, and necessary. There is, of course, a possibility that these algorithms, such as the *Computational Designer* studied here, end up working autonomously. Yet, in this thesis, I make the argument for AI-assisted game design and explore the advantages and benefits of having such a system in a human-AI collaborative scenario for both the human designer and the algorithms. The findings support and embrace collaboration, not because it couldn't be used autonomously since we have demonstrated that "high-performing" content can be generated, but because its use is useless without the human designer. This is especially true when we consider the generation of subjective content where metrics must be reductionist and lose nuance and the adequate assembly or orchestration of content. However, *automated game design* is an important research area where the aim is to study how multiple game design elements, facets, and aspects could be modeled, assessed, and generated; to explore computational creativity systems.¹

Moving beyond the autonomous use of these techniques, there are other ethical considerations and implications I have come across throughout my PhD. Some of these points are factual discussions over what we can already see and experience. Others are speculative, to reflect, be thought-provoking, and raise awareness and discussion [4, 5]. For instance, what are the implications of implementing these human-AI collaborative systems in the workplace? These systems can completely change the workplace, how we work, and the interactions we have. This is visible when using automated-decision making systems in workplaces highlighting large problems with

¹For the interested reader I refer to the awesome tutorial by Mike Cook: <https://www.youtube.com/watch?v=dZv-vRrnHDA>.

AI systems such as bias, privacy, transparency, and fairness [6]. The introduction of *AI as a colleague* in the workplace would have a higher impact. Society and human colleagues will need to understand how these systems affect their work and how to collaborate with them. The benefits, as discussed in this thesis, are plenty, but its development needs to be gradual and elaborated together with the target group [7, 8].

Another contentious point is that it could also redefine the “perfect” colleague. As it will be shown and discussed throughout the thesis, within games, AI can be used to search unknown spaces for relevant content, simulate gameplay, gather statistics, or produce content that is both on the designers’ style and aligned to players’ requirements. This could redefine the landscape of what is expected from designers, creating false expectations. However, in my view, AI-assisted game design is part of the future in game development, and many of these systems could be used to improve the processes now in place. Popular systems like DALL-E, Imagen, or GPT, while not made with collaboration in mind, could be used for new interactions. Nevertheless, we need to be careful on how we implement these systems, need to have a wider reach to society, and understand better the needs of the target group.

Moreover, among the several points discussed by Mike Cook, one of them is the impact Procedural Content Generation (PCG) algorithms have on the developers’ workload. Rather than reducing their workload (one of the core ideas with PCG), more could be expected from the developers as PCG takes care of other work [9]. However, PCG takes care of asymmetrical workload; reducing repetitive and tedious tasks might mean that more laborious tasks await designers. Granted that these might be more interesting, they still require larger labor and different intensity from developers. Rather than working on a “sinusoidal” shape, where low (e.g., repetitive and tedious tasks) and high (e.g., creative output) intensity tasks are alternated, designers might end up working on a high-intensity plateau.

On a tangential point, not much discussion has taken place regarding these repetitive tasks. On the one hand, we aim at reducing them, thus, reducing the human workload to focus on what matters to them. On the other hand, these tasks might be necessary for the creative process, which is still unknown to us [10]. They might not be necessary for all, but perhaps novice designers require them in order to try something different, work around them, or simply because their creative process is longer than

more experienced designers. One outcome could be that these end up being replaced with another repetitive task, such as adjusting these algorithms to get the expected output.² A similar idea is seen within EDD, where designers take their time exploring how the *computational designer* adapts to them to understand how and when to use its suggestions. Given EDD's nature and goal, a similar creativity support process might be in place.

Moreover, I find one more point relevant to discuss with human-AI collaborative tools and their implementation, namely the role humans and AI have. Throughout this thesis, the discussion mainly surrounds the role of the computer as a colleague, but what are the alternatives, and how could that affect the collaboration and the workplace? Partlan et al. did a participatory design project to investigate how mixed-initiative tools and Computer Support Tool could be implemented and be in place, consulting with expert Game AI designers [8]. In their work, there seems to be a representative duality under the same category, "Designer direct implementation as editors," where designers want full control but want a tool as a partner as well. This duality is explored in the thesis, where one of the aims is to establish a colleague relationship, exploring ways for designers to have control over the algorithm without hindering expressivity. However, *colleague* is just one of the many roles the computer might have. How would the interaction be if either the human or the computer is the *manager*, *student*, or *teacher*? As we assign different roles, we expect different actions from the system, are able to grant different responsibilities and allow certain interactions such as constraining the design. For instance, as a *manager*, the system could organize multiple designers, arrange the produced content, or require and ask for certain goals,³ which could then mean that the system has more agency or initiative. On the other hand, as a *student*, the system could take a background role, learning from the designer (e.g., a designer model) and using different strategies such as Machine Teaching or Active Learning [13]. However, I believe these roles will be more fluid and dynamic throughout the design process. Instead of remaining static, they would change according to the needs of the project and designers.

²Systems such as *Danesh* explore the use of assessment tools for designers to test different PCG algorithms [11]

³The work by M in *Baba is Y'all* has the AI request for specific goals that it is missing, which the human can, of course, ignore [12]

CONTENT

ABSTRACT	IX
LIST OF PUBLICATIONS	XI
ACKNOWLEDGEMENT	XV
PREFACE	XVII
I COMPREHENSIVE SUMMARY	1
INTRODUCTION	1
Problem Statement	4
Research Questions	7
Pronouns, Style, and Clarification	12
BACKGROUND	13
Procedural Content Generation	13
Search-based Approach	15
Quality Diversity	16
Human-AI Collaboration	18
Mixed-Initiative Paradigm	19
Mixed-Initiative Co-Creativity	20

Modeling players and designers	24
Designer Modeling	25
Computational Creativity	26
AI METHODS.	29
Evolutionary Computation.	29
Evolutionary Algorithm Components	30
Representation	30
Evaluation	31
Selection	31
Variation Operators	32
Replacement.	32
MAP-Elites	33
Machine Learning	35
EVOLUTIONARY DUNGEON DESIGNER	37
Level Design	38
Room Generation	40
Narrative Design	41
Automatic Objective Assessment	42
Quest Design	42
Narrative Structure Design.	43
Designer Interaction	44
RESEARCH METHODOLOGY.	46
Evolutionary Dungeon Designer.	48
Computational Designer	48
Temporal Expressive Range Analysis	49
Designer Personas	49
Methods	50
Methodology Discussion.	51
CONTRIBUTIONS	53
Research Question 1	53

Research Question 2	56
Research Question 3	57
Research Question 3.1	58
Research Question 3.2.	60
Research Question 4	60
Research Question 4.1	61
Research Question 4.2.	62
DISCUSSION AND CONCLUSIONS	64
Control, Expressiveness, and Adaptability	65
Adaptive Experiences	66
Towards Holistic PCG and MI-CC	67
Revisiting Human and AI Roles	68
To Learn or to Evolve? That is the Question.	69
FUTURE WORK	71
Human-AI Collaborative Properties	71
Designer Modeling and Adaptive Experiences	72
Explainable AI.	72
Holistic Procedural Content Generation.	73
Exploring Agency and Initiative in Mixed-Initiative	73
REFERENCES	73
II PAPERS	105
Paper I - Fostering Creativity in the Mixed-Initiative Evolutionary Dungeon Designer	107
Introduction.	109
Related work	110
The Main Principles of Mixed-Initiative.	110

Mixed-Initiative Tools in Game Design	111
Dungeon Design in Videogames	111
The Evolutionary Dungeon Designer	112
Improving the Mixed-Initiative Evolutionary Dungeon Designer	113
The Suggestions View	114
The Room View	116
User Study	118
Results and Discussion	119
Conclusions and Future Work	120
Paper II - Assessing Aesthetic Criteria in the Evolutionary Dungeon Designer	128
Introduction	131
Related Work	133
The Evolutionary Dungeon Designer	133
Assessing Aesthetic Criteria	133
Preserving Custom Aesthetic Structures	134
Evaluating Symmetry and Similarity	135
Symmetry evaluation	136
Similarity evaluation	136
Conclusions and Future Work	138
Paper III - Empowering Quality Diversity in Dungeon Design with Interactive Constrained MAP-Elites	143
Introduction	145
Background	146
Dungeons	146
Map-Elites for illuminating search spaces	147
Evolving Dungeons as a Whole, Room by Room	147
The mixed-initiative workflow in EDD	149
Interactive Constrained MAP-Elites	150
Illuminating Dungeon Populations with MAP-Elites	151
Dimensions	151
Continuous Evolution	152
Algorithm	153

Experiments	154
Results and Discussion	155
Conclusions and Future Work	158
Paper IV - Learning the Designer's Preferences to Drive Evolution	165
Introduction	167
Previous work	168
Mixed-Initiative Co-Creativity	168
Procedural Content Generation via Machine Learning	168
The Evolutionary Dungeon Designer	169
Designer Preference Model	170
Model Update and Usage	171
Dataset creation:	172
Training and usage:	172
Evaluation	174
Model performance, integration, and setup	174
User Study	175
Open Problems and Future Work	175
Dataset	176
Preference modality	177
Dynamic-Dynamic System vs. Dynamic-Static System	178
Future Work	179
Conclusion	180
Paper V - Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions	187
Introduction	189
Previous Work	190
Map-Elites for illuminating search spaces	190
Evaluation of Procedural Content Generators	191
Evolving Dungeons as a Whole, Room by Room	191
Interactive Constrained MAP-Elites	193
Illuminating Dungeon Populations with MAP-Elites	194
Dimensions	194

Continuous and Interactive Evolution	197
Algorithm	198
Performance Evaluation across Dimensions	200
Expressive Range Analysis	201
Alternative Scenarios	205
Fitness Evaluation.	205
Conclusions and Future Work	207
Paper VI - To Make Sense of Procedurally Generated Dungeons	215
Introduction.	217
Related Work.	217
The Evolutionary Dungeon Designer.	217
Procedural Generation of Game Narrative	218
Generating Objectives for Dungeons	219
Results and Discussion	221
Conclusions and Future Work	225
Paper VII - Questgram [Qg: Toward a Mixed-Initiative Quest Generation Tool	229
Introduction.	231
Related Work.	232
Story and Quest Generation	233
Mixed-Initiative Co-Creativity	234
Quest Generation	235
Quest Actions.	238
Quest Grammar	238
Workflow.	239
Evaluation	239
Expressive Range Analysis.	240
Experiment Discussion	240
User Study	241
Manual Quest Creation.	242
Automatic Quest Creation.	243
Mixed-Initiative Quest Creation	243

Automatic Suggestions	244
Quest Actions	244
Usability	245
Creativity	245
Overall Experience and Missing Features	245
User Study Discussion	245
Conclusions and Future Work	247
Paper VIII - Assessing the Effects of Interacting with MAP-Elites	255
Introduction	257
Background	258
Methods for Evaluating PCG	259
Variants of MAP-Elites	259
Approach	261
Experiment Setup	261
Metrics	264
Results and Analysis	264
Case 1 - Design Opens New Regions of the Generative Space	266
Case 2 - Subtle Changes in the Design Reflected in the Generation of MAP-Elites	266
Case 3 – Exploring Multiple Properties	267
Discussion	268
Conclusions and Future Work	269
Paper IX - Designer Modeling through Design Style Clustering	276
Introduction	279
Background	280
Designer-centric Perspective	281
The Designer Preference Model in EDD	282
Concepts and Definitions	283
Design Style	284
Room Style Clustering	284
Data Collection	285
Dataset pre-processing	287

Clustering and Analysis	287
Cluster Labelling	289
Designer Personas	292
Unique Trajectories	293
Archetypical Paths through Style Space	294
Architectural-focus	294
Goal-oriented	294
Split central-focus	295
Complex-balance.	295
Preliminary Evaluation	297
Discussion.	297
Conclusions.	298
Paper X - TropeTwist: Trope-based	
Narrative Structure Generation	306
Introduction.	309
Related Work.	310
Building narrative structures with tropes	312
TropeTwist.	313
Trope Patterns	313
Micro-Patterns	314
Meso-Patterns.	315
Auxiliary Patterns	317
Proof-of-Concept	318
Evolving Narratives with Graph Grammars	318
Experiments.	320
Discussion and Limitations	322
Conclusions and Future Work	323
Paper XI - Story Designer: Towards a Mixed-Initiative Tool to	
Create Narrative Structures	332
Introduction.	335
Related Work.	337
Story Designer.	337
TropeTwist.	338

Workflow	339
Evolving narrative structures with Graph Grammars	340
Behavior Dimensions for Graph Grammars	341
Experiment Setup	345
Metrics	345
Narrative Graphs for Experiments	346
Result and Analysis	347
Conclusions and Future Work	350

Paper XII - Towards AI as a Creative Colleague in Game Level

Design	360
Introduction	363
Related Work	364
AI Roles and Adaptability	365
Altering Human-AI collaboration dynamics	365
AI Version 1 (AIv1) - Low degree of agency	366
AI Version 2 (AIv2) - Medium degree of agency	366
AI Version 3 (AIv3) - High degree of agency	366
The Design of the AI	366
Experiment Setup	367
Results	369
Perceptions of the different AI versions	370
AI's behavior	371
The Creative Relationship	372
The Creative Process	372
Constraints and Design Goals	372
Discussion	373
Willingness to include the AI in the design process	373
Variants of the users	373
Frustrating factors and Constraints	374
The concept of a well performing, high agency co-creator	375
Conclusion	376
Appendix A	377
Interview procedure and questions	377
Instructions	377
Interview Questions	378

Acronyms

AI Artificial Intelligence.

CC Computational Creativity.

CE Cluster-Elites.

CMA-ES Covariance Matrix Adaptation Evolution Strategy.

CMA-ME Covariance Matrix Adaptation MAP-Elites.

CVT-MAP-Elites Centroidal Voronoi Tessellation-MAP-Elites.

DSRM Design Science Research Methodology.

EA Evolutionary Algorithm.

EC Evolutionary Computation.

EDD Evolutionary Dungeon Designer.

FI-2Pop Feasible-infeasible Two-Population.

GVG-AI General Video Game AI.

HCI Human-Computer Interaction.

IC MAP-Elites Interactive Constrained MAP-Elites.

MAP-Elites Multi-dimensional Archive of Phenotypic Elites.

ME-MAP-Elites Multi-Emitter MAP-Elites.

MI Mixed-Initiative.

MI-CC Mixed-Initiative Co-Creative.

ML Machine Learning.

NSLC Novelty Search Local Competition.

PCG Procedural Content Generation.

PD Participatory Design.

QD Quality Diversity.

RL Reinforcement Learning.

RtD Research through Design.

SSL Self-Supervised Learning.

VGDL Video Game Description Language.

Part I.

COMPREHENSIVE SUMMARY

INTRODUCTION

We all test the rules, and consider bending them; even a saint can appreciate science fiction. We add constraints [...] to see what happens then. We seek the imposed constraints [...] and try to overcome them by changing the rules. We follow up hunches [...], and - sometimes - break out of dead-ends. Some people even make a living out of pushing the existing rules to their limits, finding all the computational 'cans' that exist: creative tax-lawyers call them loopholes (and creative tax-legislators close them).

Margaret Boden

The Creative Mind: Myths and Mechanisms, pp. 58

Since the dawn of time, we humans have been searching [and in need] for tools to develop our ideas or execute mundane objectives. As time and technology advanced, more sophisticated types of assistance emerged to cope with humans' needs, such as vehicles to traverse longer paths or ways to facilitate writing. With the invention of hardware and software, its ubiquity, and the raise of Artificial Intelligence (AI), a new path for human assistance opened up. Tools that were used to facilitate our work or assist us into doing repetitive work, could now provide advance assistance with smarter tools that allows us to work more efficiently. However, tools that assist us in our tasks are not the only key factor; the collaboration between humans has remained virtually unchanged as an essential way to move forward and to develop new experiences. Not only to achieve greater objectives as a group but also to develop as individuals. While the current tools to support humans' work and creative output are valuable in many ways; this raises an essential question that holistically motivates this thesis:

How can we create tools that no longer behave just as aid to support our work but can collaborate with us, to some extent, in the same way as human collaboration functions?

This thesis focuses on how game design related tasks such as the creation of game facets or designers' design process could be explored through Human-AI collaborative tools. While Procedural Content Generation algorithms could be employed for automated game design [14–16], this thesis takes a different approach and explores game design; and the creative, organizational, and collaborative tasks that encompass that; through Human-AI collaboration. We focus on fostering, enhancing, and augmenting human abilities such as creativity. Multiple approaches, systems, and algorithms representing a computational designer to collaborate in the creation of content with a human designer are developed. These tasks could not only be assisted by AI, but rather AI could be a *colleague* in the design process. To tackle these shared tasks, a mutual feedback loop could be established, whereby AI and humans could inspire each other to explore unknown areas in the design landscape and reach better and more creative solutions.

Problem Statement

The proposed question is not new and has been approached by different disciplines, under the Mixed-Initiative (MI) paradigm. MI refers to the collaboration between *human* and *computer* where both have some proactive initiative to solve some task. MI can be seen as a multi-agent collaboration scenario, where the interaction should be flexible, allowing for a continuous negotiation of initiative and leverage on each other's strengths to solve a task [17]. *Initiative* was described by Novick and Sulton as a multi-factor model that combines: choosing the task, choosing the agent in control and how the interaction is established, and choosing the expected outcome from the collaboration [18].

Moreover, Horvitz discussed such a question in terms of Intelligent User Interfaces [19], describing mixed-initiative systems and interfaces as a more natural collaboration in a user interface that emerges from intertwining human control and manipulation, and automation [20]. Horvitz presented several principles of mixed-initiative interaction and its challenges, many of which still exist [21], mainly describing this interaction as conversation systems between AI and humans [22]. Moreover, Yannakakis et al. introduced the Mixed-Initiative Co-Creative (MI-CC) paradigm for the co-creation of creative content, where both AI and humans alternate in the initiative to co-design and solve tasks [23]. Their work describes key findings and discussions for how MI-CC does not only help human designers solve tasks, but also fosters their creativity through an interactive feedback loop and

lateral thinking [24–26].

Nevertheless, this collaborative approach raises an *initiative* challenge for either agent: Which agent should have the initiative at different stages of the development and over the goal? The question reflects the diffuseness of the challenge and situation, as many factors need to be considered before appropriately indicating this. At the very least, some could say that depending on the task to be performed and the expertise of both, either would clearly be the one taking the development initiative. Whereas others would position the human as the one always in control. Yet, even with a clear answer, what happens in creative tasks to the expressivity of one of the sides due to the other taking the initiative?

Furthermore, one context where the MI paradigm would be very beneficial is games. Games, either digital or tabletop, are created through a complex creative process that couple together many different creative facets in different ways. Games contain a large amount of creative content carefully combined and intertwined to craft specific experiences, with the addition of rules that dictate how a player is to interact with it. In contrast with other creative content, games are multifaceted, content-intensive, and should be interacted, experienced, and enjoyed by others, which also creates a complex subjective task [25]. Usually, games are developed by more than a person (although many exceptions exist [27–29]), reaching to hundreds and thousands of developers, with each developer specialized in different areas such as gameplay, AI, animation, concept art, etc. Each creates a specific part of the game and the content through collaboration and following a road map [30, Chapter 14]. However, no matter the team’s size and talent, the fact remains that developing games is a hard challenge [31]. As technology advances, the requirements increase substantially for any game facet, coupled with the users’ increase demand, the higher competitiveness in the market, and the launch of many more platforms [32].

Procedural Content Generation (PCG) is a field within computational intelligence in games, that focuses on the use of algorithms to create game content [33]. PCG algorithms have been used to aid in the creation of a plethora of games such as No Man Sky [34], Spelunky [35], or Minecraft [27], to the extent that PCG and AI have enabled experiences and interactions that were not possible before [36–38]. Moreover, as one of the properties of PCG is to increase replayability by creating an abundance of well-made content [3], games are not the only beneficiaries of PCG

methods. For instance, they have the opportunity to be used to increase the generality of Machine Learning (ML) approaches [39], or a step towards open-endedness Evolutionary Computation (EC) [40].

Moreover, in design and creative tasks such as games, the designer usually has intentions in what they are creating and goals that they want to achieve with their design. Thus, to enable deeper MI levels to co-create content, some control mechanisms with a varying degree of control over the algorithms might be necessary for the designer. Through this, the designer could direct or constraint the generated content by the computational designer and oversee that it is within their intentions and goals. In this case, each agent's control and expressive properties are at the expense of the other agents, as it constrains the space of possibilities [41]. This is especially relevant when the aim is a creative work such as games, where the creative expression needs to be fostered [26]. Yet, it becomes particularly challenging when using mixed-initiative methods, where smart approaches need to be in place for a natural conversation and successful collaboration. The more control is given, the more constraint it exists, but is this a problem? Is it inevitable? Boden explains it conspicuously "... We [humans] seek the imposed constraints [...], and try to overcome them by changing the rules. [10]". Constraints limit the space, and as a consequence, they are overcome by encountering creative solutions.

For this interaction to be complete, the human needs to understand the AI's behavior through interpretable and explainable models and systems, and the AI needs to recognize and interpret the intentions of the humans seamlessly as they create their content. The former is the focus of *Interpretable and Explainable AI*, which seeks to create or adapt models and systems for a better workflow between humans and AI, where humans could understand the AI's decision process to enable trust relationships and reach deeper interactions [42–44]. The latter would mean that the AI could adapt its behavior and functionality to the needs, expertise, and workflow of individual designers or a specific group of designers. To do so, the AI must analyze several design processes, such as the designer's preferences, styles, and goals, which holistically is called *Designer Modeling* [2, 45]. How to create these models and use them to develop adapted experiences is a complex challenge, and understanding the implications of its usability in the control-expressive properties, as well as other consequences, is not trivial.

To explore this, the main body of work presented in this dissertation is applied and evaluated through the Evolutionary Dungeon Designer (EDD), a Mixed-Initiative Co-Creative system, where designers can create levels and narrative elements, such as quests or objectives for adventure and dungeon crawler games such as *Zelda* [46] or *The Binding of Isaac* [47]. In EDD, the human designer can, on the one hand, quickly create interconnected rooms forming a dungeon to be experienced by players. Meanwhile, the computational designer collaborates by providing suggestions using different algorithms and following multiple heuristics. On the other hand, the human designer can compose quests using the level information and the overarching narrative structure.

Research Questions

As motivated thus far, this thesis focuses on exploring different approaches for procedurally generating content for games, specifically through the MI-CC paradigm, where a human designer collaborates with an underlying AI to create creative content. Exploring the role of *computers as colleagues* as defined by Lubart [48], this thesis delves into the use of MI-CC tools and the multiple properties that emerge from the dynamic interaction between AI and Humans. The aim is to understand how we can enable a rich, fruitful, and better feedback loop in these types of tools using and developing novel AI techniques in the field of Evolutionary Computation and Machine Learning to improve the interaction and create adapted experiences. The thesis also analyzes and studies the requirements, challenges, and benefits of enabling in-depth collaboration, tailored experiences, the properties that emerge (some seemingly competing properties), and their dynamics. Therefore, this thesis aims at addressing, discussing, and exploring the following research questions:

RQ1. How can we use and integrate multiple algorithms such as quality-diversity algorithms and grammars into a mixed-initiative approach to help designers produce high-quality content and foster their creativity while allowing them to control, to a certain extent, the generated content?

There are plenty of approaches and algorithms such as Evolutionary Computation, Machine Learning, or Wave Function Collapse that have been used to generate content as it will be discussed in the background chapter (sec. I). In this thesis, we are interested in exploring these algorithms and approaches, particularly quality-diversity algorithms, pattern-based

systems, grammar systems and their use as encoding representation in MI-CC systems. In these MI-CC systems, the aim is to have more controllable yet expressive generators, and at the same time explore how they can be used to foster the designers' creativity and establish better human-AI collaboration and interactions.

Quality Diversity (QD) algorithms are a relatively new family of algorithms, specifically aimed at tasks and environments that require the strengths of convergence and divergence search [49]. Leveraging on QD algorithms to search for a surfeit of heterogeneous content while not losing sight of the content's quality could enable MI-CC systems to explore a big area of the generative space producing more diverse and high-quality solutions. Through this, the system could propose a higher range of diverse solutions to the user, aiming at fostering the creativity of the human designer [24]. Thus, how to integrate QD algorithms in MI-CC systems that need to take into account the human work to provide valuable input is a promising open research area and one that this thesis explores. However, it is paramount to understand how to effectively use QD algorithms in these systems to fully leverage their expressive power while providing control to human designers.

RQ2. How can we use player and designer data to better understand their behaviors and procedures to enhance and adapt Mixed-Initiative Co-Creative systems?

Games and creative contexts are spaces where both players and designers can express themselves, producing data on how they both interact. Research areas such as Experience-driven PCG [50], player modeling [51, 52] or designer modeling [2], explore the use of such data to understand particular users [2, 53, 54] and to improve and enhance the experiences of players and designer. Especially focusing on enabling adaptive experiences [55] and more accurate heuristics [56–58]. However, how to use the data (and even what to collect) is still an open research area, especially when applied to adaptive experiences for MI-CC tools with only a few relevant examples [45, 59, 60]. Furthermore, the importance of enhancing the experience of MI-CC tools' users lies in the search for deeper understanding and collaboration between humans and AI, which could enable a better experience for both.

RQ3. How can we model different designers' procedures and use them as surrogate models to anticipate the designers' actions, produce content

that better fits their requirements, and enhance the dynamic workflow of mixed-initiative tools?

RQ3.1 What trade-offs arise from modeling and using designer's procedures to steer the generation of content towards personalized content?

RQ3.2 What constraints are created over the generative process when using designer models?

The advantage of having the human and AI collaborating is analogous to humans collaborating, each one with their own set of strengths and weaknesses to reach greater objectives and develop each other. However, mixed-initiative collaboration requires both human and AI to understand each other and the goals that the human aim to reach [18, 21]. This creates a particular problem where the AI needs to identify certain processes and characteristics of the human. When employing MI-CC to co-create games and creative artifacts; this translates to design processes, style, preferences, intentions, and goals. This thesis aims to explore how to model different designer procedures such as preference or style, using several Machine Learning methods, and how to best use these as surrogate models to produce better content and enhance designers' experience using MI-CC systems.

RQ2 and RQ3 drive the research on how to gather and use different types of data, i.e., player and designer data, and whereas designer modeling could be used in the MI-CC feedback loop to create adaptive experiences. Through RQ3.1 and RQ3.2, this thesis focuses on exploring the trade-offs of using designer modeling. Specifically, the interest lies in the challenges and benefits that designer modeling creates for the algorithms and designers, and the overall experience that the designer wants to create, i.e., the game.

Moreover, the constraints that emerge from using these models as surrogate models to steer the content generation are not trivial to address and are essential to study to understand and analyze their effect and extent. Using these models will inevitably create constraints over the generation process as we aim to adapt the experience to each designer or group of designers. Therefore, RQ3.2 specifically aims at understanding: what are these constraints? What is constrained? And whether these constraints are positive or negative?

RQ4. How can level design and narrative interact, act as constraints, be intertwined, and in general, have an active role affecting each other to

produce a holistic system?

RQ4.1 What are the factors to be considered when implementing such a paradigm and system in a mixed-initiative application, where a designer will be able to interact with the content?

RQ4.2 What are the effects of producing and using a holistic system for the creative process of a designer, and what challenges are imposed on computational creativity?

The intertwined, multi-faceted, and collaborative nature of games invites the exploration of how to generate different facets and how to intertwine them. Facets in a game are seldom produced in a vacuum, and the coherence of a game and cohesion of game content rely on these facets to be relevant and aligned with each other. The automatic generation of game content could follow a similar process, so the resulting content feels coherent, which has been categorized as Holistic PCG [61, 62]. Usually, systems that generate several facets focus on a hierarchical and step-by-step process where each facet is generated successively and, at times, not relying on the other generated facets but rather on an overarching design goal. The generation of game facets could then have some feedback loop, where content generated in one facet has an effect on another, and vice-versa, generating content in unison, i.e., orchestrating game content.

Moreover, the importance of space or game worlds and narrative has been pointed out by previous research in different disciplines. For instance, Aarseth links the space to the quest in games, both dependant on each other [63]. Ashmore and Nitsche present a similar point in relation to the games' interactivity as they discuss that a generated level without depth and context lacks interest for the final user [64], further discussed and related by Kybartas and Bidarra with a focus on story automation [65]. Similarly, Dehn [66] defines space (i.e., the world) as a post-hoc development and justification for authored events, while Lebowitz [67] argues for the opposite view, the story gives meaning to a created world. Looking at the narrative-space discussion from whichever angle and perspective, it is noticeable that one requires the other to develop fully and fruitfully. Therefore, the intertwined generation of both facets is a suitable first approach to holistic PCG, which this thesis explores.

Nevertheless, there are several challenges when posing content generation

as a multi-faceted and intertwined task, such as how to represent the content, what elements to use as constraints, or how to evaluate the generated content. Yet one of the objectives explored in this thesis is to add and combine this into an MI-CC system. This exacerbates the challenge, as human input and control become central, and we consider their input and initiative and the role of the AI in the system. Likewise, MI-CC challenges and considerations are extended, whereas the generated content must not only adapt and be relevant for the human (considering their input as well) but also maintain and adapt to other constraints from different facets.

Pronouns, Style, and Clarification

Throughout the thesis, the pronoun “we” will be used in favor of “I”, since the work and research achieved and presented in this thesis would not have been possible without my co-authors’ collaboration.

When referring to a player or designer, this thesis chooses the pronouns “they” and “their” to respect a gender-inclusive language. Moreover, throughout the thesis, it is referred to as user and designer alike, as a designer is the target user group within the possible user base of the systems and tools developed in this thesis. The player is referred to as the end-user: the user who could experience the creations in the Mixed-Initiative Co-Creative system.

When discussing the participants in a mixed-initiative system, i.e., AI and Human, this thesis uses the word “agent” when needed to refer to either, unless specifically discussing one in particular, as mixed-initiative systems have been described as multi-agent systems [17].

Finally, this thesis will refer to as “computational designer” to the overall AI system that interacts and collaborates with the human designer to create content through the MI-CC system.

BACKGROUND

This chapter offers an overview of the different fields surrounding the central subject of study in this thesis, i.e., the collaboration between AI and humans to co-create game content, and the related RQs. First, Procedural Content Generation is explored with multiple examples of the type of content that might be created. It is then presented the search-based approach, quality-diversity algorithms, and the Mixed-Initiative paradigm as they are the main approaches and paradigms used throughout the thesis. Then, player and designer modeling is presented to give an overview of the concepts and the differences between them, and examples of each computational model. Finally, creativity and computational creativity are explored by briefly analyzing the field's goals with the most relevant literature and presenting examples within the computational intelligence in games research area.

Procedural Content Generation

Game content is the main component of any game, as it is what players interact with to achieve the designers' developed experience. Game content refers to anything within the game, from the game's rules, a hero's backstory, or the levels to be traversed by players. However, game engines and Non-Player Characters (NPC) behaviors are not considered the same type of game content as the former is used to create the games themselves, and the latter refers to the AI behavior in-game (e.g., movement or combat). Furthermore, as higher possibilities for more complex games are provided by technology, game engines, and platforms, and developers and players set higher requirements, games have increasingly become content-intensive entertainment mediums.

To cope with this challenge and to relieve the burden and workload of game designers when creating all this content, several approaches have been proposed to create content under the field of Procedural Content Generation. PCG refers to the creation of content, mainly for games, using

algorithms, autonomously or with the assistance of users [33]. Content can be divided into game facets: audio, visuals, narrative, levels, rules, and gameplay [61], and have been categorized within the PCG field as *Game Bits*, *Game Space*, *Game Systems*, *Game Scenarios*, *Game Design*, and *Derived Content* [68].

There are plentiful of commercial games that utilize one way or another PCG such as *The Binding of Isaac* [47] or *Civilization* [69], to the point that some games rely critically on these algorithms, providing experiences otherwise not possible such as *Rogue* [37], *Dwarf Fortress* [70] or *AI Dungeon* [36]. However, there has been an increasing interest in PCG during the past decade in academia [71]. There exist multiple approaches addressing different challenges in the creation of content, resulting in algorithms that can autonomously create game rule's [72,73], narratives [64,74], levels [75–77], graphics [78,79], and audio [80,81].

Another approach is to focus in the generation of content in multiple facets aiming at creating intertwined content; known as Holistic PCG [61, 62]. By approaching content generation as a multi-faceted task, games and their content can be more coherent. There have been approaches focusing on generating complete games [72,82,83], but usually only some facets are targeted based on their criteria, requirements, and synergy. For instance, in [84] and [85], mechanics, graphics, and levels are generated together, although through different processes. Two facets that are commonly associated with each other are the narrative and level facet [64,86–88]. This is due to the fact that space requires context to make sense of it, and narrative requires space to develop. Nevertheless, these approaches usually follow a hierarchical procedure, where content in each facet is generated step-by-step such as in the work by Dormans [86], which generates the mission graph that guides the level generation.⁴ Yet, works like the one by Hoover et al. [89], Holtar et al. [90] or Karavolos et al. [91] present interesting results and examples of more intertwined content generation.

Furthermore, within the field of PCG, there exist [arguably] three main approaches to create content: constructive approach, generate-and-test approach, and search-based approach, each with their criteria [92]. Constructive approaches focus on generating content following a set of predefined rules that can create valid content without evaluating the quality of the content after generating, rather the content is evaluated as it is being

⁴This approach is also used in Dormans' games *Unexplored* and *Unexplored 2*.

constructed [93–95]. Conversely, generate-and-test approaches focus on creating content iteratively that instead of being continuously tested as the content is constructed, it is tested after generation to satisfy a set of constraints or objectives. When tested, the process might iterate on the design. In this approach, the designer’s focus is on creating the set of constraints to be satisfied [96,97]. Search-based approaches are a specialized case of the generate-and-test approach that aims at using some type of search algorithm, mainly Evolutionary Algorithms, to generate content by exploring the generative space and through this process, encounter interesting individuals with non-trivial characteristics [55,98].

Besides these three main approaches, there exist other ones to generate content. For instance, Constraint Solving algorithms such as Wave Function Collapse (WFC), do not directly map their procedures to the aforementioned processes [99,100]. Other examples are techniques within the PCG via ML approach [101] such as approaches to repair unplayable generated content [102] or generating content using learned probabilities from sample content [103]. However, this thesis focuses mainly on using a search-based approach to generate suitable content suggested to a designer in an interactive tool through QD algorithms [104]. Our approach relies on exploring the generative space informed by a designer’s design that helps focus the search in different areas of the space while still encountering diverse solutions for the designer.

Search-based Approach

The search-based approach is a specialization of the generate-and-test approach, where the aim is to use some search algorithm, being the most prominent, Evolutionary Algorithm. However, essentially any metaheuristic algorithm and from the stochastic search algorithm family could be used as well and fall under the umbrella of search-based approaches. The main distinction between the search-based approach and the generate-and-test approach is that search-based approaches evaluate the generated solution with a quality estimator, e.g., fitness function or novelty behavior, providing a continuous evaluation of the generated content. Such evaluation drives the next generation steps, as the estimation helps the search to find promising paths.

Search-based approach has been widely used in PCG and basically for the generation of all the types of game content such as levels [86], rules [73]

or weapons [105]. Moreover, the evaluation of the generated content is the most important part of search-based approaches, as well as the most challenging and complex. The used heuristics does not only need to be representative of the task at hand but also allows the expressive property of the search, as that is one of the main benefits of search-based approaches. Constraints to ensure quality [or playable] experiences are not enough, since that does not necessarily represent what a designer or player wants ⁵. However, evaluation functions come in all shapes and sizes, and they are all valid with their own set of ups and downs. For instance, they might come from game design concepts such as design patterns [106] or game level metrics [56, 57].

Quality Diversity

Quality Diversity (QD) algorithms are a family of algorithms under the approaches in Evolutionary Computation, that focuses on combining the benefits and strengths of both convergence search, i.e., focusing on optimization and objective, and divergence search, i.e., disregarding objectives and searching for diversity [49, 107]. Through this, QD algorithms seek to generate a collection of high-performing solutions that are as diverse as possible⁶. While convergence search refers mainly to the typical EC algorithms used for optimization, divergence search has increasingly being used to tackle many tasks that were previously dominated by convergence search. For instance, when the task or environment is deceptive, i.e., reaching the goal might be impossible or where plenty of local optima exist where a convergence search might get stuck. Lehman and Stanley proposed the Novelty Search algorithm, which introduces the idea of divergence search through ignoring objectives and searching for novel behaviors instead, with surprisingly good results [108, 109]. From that moment onward, several divergent search algorithms have been proposed, such as surprise search [105] or the Paired Open-Ended Trailblazer (POET) [110–112] to explore open-ended algorithms, as well as variations to novelty search such as constrained novelty search [113] or Novelty Search Local Competition (NSLC) [114].

NSLC is an example of a QD algorithm that leverage on the divergent search to explore the space for novel behavior among solutions and on

⁵One of the challenges of generating games [and game content], is that it requires them to be fun and interacted as discussed in the previous section.

⁶The following website serves as a database with research related to QD: <https://quality-diversity.github.io/> maintained by Antoine Cully

convergence search for preserving the high-performing individuals within the novel niches [114]. Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) is another algorithm in the QD family, and one that has gained considerable popularity in multiple areas such as games [115–117] and robotics [118, 119]. As the other QD algorithms, MAP-Elites explores the behavioral space for a collection of solutions that are both high-performing and diverse among each other, with the caveat that MAP-Elites discretizes the behavior space as a grid of cells informed by a set of feature dimensions that illuminate the behavior space. MAP-Elites’ goal is to fill each cell belonging to a set of discrete feature dimension values with a high-performing individual encountered in the search and retain it until a higher-performing individual with similar characteristics is encountered [120]. This characteristic allows the exploration of features orthogonal to the fitness function that allow the discovery of diverse behavioral repertoires [121–123].

One major challenge with MAP-Elites is the *curse of dimensionality*, since each new feature dimension used adds a new dimension in the search space. Thus, some MAP-Elites variation skip the grid architecture and focus on reducing the amount of feature dimensions or enabling the use of higher dimensions such as Centroidal Voronoi Tessellation-MAP-Elites [124] or Cluster-Elites [125]. Further, the Covariance Matrix Adaptation MAP-Elites algorithm combines the effective adaptive search of Covariance Matrix Adaptation Evolution Strategy with a map of elites, yielding large improvements for real-valued representations in terms of both objective value and number of elites discovered [126]. The work by Fontaine et al. was expanded into the Multi-Emitter MAP-Elites, improving the quality, diversity, and convergence speed of MAP-Elites in general [127]. Other work within MAP-Elites has focused on its robustness [128], multi-objective tasks optimization [129], or assessing its properties when coupled in interactive environments [130].

Moreover, within the field of games QD algorithms have started to be used extensively, especially MAP-Elites, both for gameplay and agent behaviors [131, 132], and the generation of content [104]. MAP-Elites has been used to create and find levels with just the right difficulty for a set of agents [133], to balance and create decks in hearthstone [117], or create levels for puzzle games through crowdsourcing [12]. Constrained MAP-Elites introduced by Khalifa et al. [134], combines MAP-Elites with the Feasible-infeasible Two-Population (FI-2Pop) algorithm [135], to generate bosses

for bullet hells games in Talakat. Since then, constrained MAP-Elites has been used in other projects and experiments to benefit from its strengths, such as to generate game levels based on mechanics as feature dimensions in Mario [115, 136], and was combined with interactive evolution resulting in the Interactive Constrained MAP-Elites [116, 137].

Thus far, the focus has been on discussing PCG and presenting algorithms that create content mostly autonomously. Automated game design is a complex task since it is required to create content (or full games) by itself with the help of heuristics, user models, and logic among the content created [16, 138–140]. However, another paradigm within PCG is the mixed-initiative paradigm, where AI can collaborate with a designer to co-design games. Through this, we could leverage the strengths of both to create content.

Human-AI Collaboration

An alternative path to work with AI would also be Human-AI collaboration instead of AI automation. On the one hand, humans have qualities that are paramount in the collaboration such as subjective and domain knowledge, expert intuition, and a holistic perspective regarding external aspects. On the other hand, AI have many qualities that are favorable for Human-AI collaboration. It can, 1) go through big amount of data and learn representations from that, 2) explore the possibility spaces in several dimensions, which might be too prohibiting for humans, and 3) have a holistic perspective on multidimensional aspects with specific domains.

In general, using AI algorithms to generate content come with a set of requirements and properties that we would like them to have [3]. For instance, the algorithm should be *expressive*: it should generate diverse content; *controllable*: it should be possible to steer the algorithm; or *adaptive*: it should adapt to other content. Expressiveness has been more widely research within PCG mainly as a means to evaluate generators [141–143]. Controllability, on the other hand, has been less explored but remains as a fundamental property for practitioners and designers [8, 144, 145]. Furthermore, as we reshape tasks and environments for Human-AI collaboration, these properties become even more relevant since they are coupled with human interaction. For instance, *adaptiveness* can be discussed by adapting to what the human creates, their criteria, or other generated content. Moving towards Human-AI collaboration raises up two other important properties as well, namely, *explainability* and *subjective evaluation*. *explainability* is

a long-term goal and research area in AI [43, 146], which is emphasized in these collaborative scenarios [42]. *Subjective evaluation* such as fun or interesting, is a hard task for optimization; thus, through Human-AI collaboration we can leverage human knowledge for assessment. These properties and their possible benefits and tradeoffs are explored in this thesis in multiple ways discussed in section I. For instance, designers are given means to control various aspects of the computational designer to steer the generation, and in turn, we explore how this affect the algorithm's expressiveness.

Moreover, the role that both human and AI play in this collaboration are as important as the properties that arise from it. Knowing, identifying, and setting the role for either agent sets the tone for the collaboration and what is expected from both. This was discussed by Guzdial et al., where designers perceived the collaboration differently depending on the assigned role for the AI, varying between: *friend*, *collaborator*, *student*, or *manager* [147]. Guzdial et al.'s work is based on the colleague role introduced by Lubart, where there is a partnership between computer and humans. Lubart discussed three other roles that the computer might have to promote creativity; *computer as nanny*: management of creative work; *computer as pen-pal*: communication service between collaborators; and *computer as coach*: Using creative enhancement techniques [48].

Mixed-Initiative Paradigm

Mixed-Initiative (MI) refers to the collaboration between Computer and Human to solve some task where both have a proactive initiative into solving the task regardless of the degree of such initiative [148]. Yet while this definition clearly separates MI approaches from others that “simply” assist humans in their tasks, it still remains a very disputed concept as: which agent initiates the “conversation”, what task to be solved, and what initiative to take in each step remain unknown. Novick and Sutton discuss MI by analyzing a set of MI systems, and conclude that the initiative in MI is a multi-factor model, described as: 1) *choice of task*: describing the task; 2) *choice of speaker*: describing which agent is in control and how the interaction works; 3) *choice of outcome*: describing what is the outcome of the interaction [18]. Moreover, Allen describes MI systems as multi-agent collaboration scenarios. These need to have a flexible interaction strategy, leveraging each agent's strengths to solve the tasks, and that involves a continuous negotiation between agents to determine roles, i.e., initiative;

thus, collaborating as a team [17]. The initiative will vary depending on which agent can solve a determined problem, providing solutions and taking the control while the other agents, e.g., a human or group of models, assist in the procedure [149]. Similarly, Horvitz discusses MI as a more natural collaboration between agents that explicitly integrate human control and manipulation, and [AI] automation strategies and their contributions to achieve some [shared] task [20, 21].

Mixed-Initiative Co-Creativity

Yannakakis et al. introduced the Mixed-Initiative Co-Creative paradigm for the co-creation of creative content such as games, and regarding PCG, where machine and humans alternate initiative to co-design content [23]. Their work and discussion on the capabilities of such interaction to foster creativity on both humans and machines is pivotal for understanding and develop MI-CC tools that can reduce the designer's workload, foster their creativity, and in general, improve the design and creative process [24, 26].

Germinate [150] is a MI-CC system to co-create rhetorical games using the constraint-based game generator Gemini [96] under-the-hood. In Germinate, the designer can, in iterations, specify a set of constraints and properties they want games to have and which the generator will consider. The designer is then presented a set of games that they can play and inspect, and which they can use to modify the set of constrained previously set, improving their understanding of their own intent. Germinate focuses on accessibility by leveraging on the concept of Casual Creators [151] within the MI-CC paradigm, allowing through this iterative process, the designer to focus in the constraint that reflects their intent rather than any knowledge within game technology.

Delarosa et al. presented an innovative MI-CC system, where the computational designer is represented as three different agents with different representations trained using Reinforcement Learning (RL), suggesting specific changes to the designer as they create Sokoban levels [152]. Their approach is the first implementation of the work by Khalifa et al. that introduced a new approach to create content: PCG via RL [153]. In PCG via RL, the level creation process is set up as an RL problem, i.e., a sequential task, where the agent can learn policies to maximize the quality of the final level. Khalifa et al. approach use three different representations, i.e., different types of agents, to create levels: *Narrow*: at each step the

agent is located randomly in the level and can perform an action in such place; *Turtle*: at each step the agent can move and change tiles in the way; and *Wide*: at each step the agent has control of location and placement of tiles. Likewise, Delarosa et al. work includes the same agents and have an identical premise, i.e., level generation as an RL problem, with the caveat that these agents must now learn and adapt to a designer's design. The designer is suggested levels based on their own by each of the agents, which the designer might pick or disregard and continue editing. Their work was evaluated through thirty-nine sessions and showed that, on average, the levels created using AI suggestions were more playable and complex.

The Sentient Sketchbook is a tool where designers can co-create low-resolution sketches of strategy levels while being presented augmented information about their creation and suggested variations using multiple heuristics and objectives [154]. In the Sentient Sketchbook, the designer focuses mainly on creating the sketch they envision, while the computational designer focuses on three main aspects. 1) Provide *suggestions* adapted to the designer's current design using constrained novelty search [113]. 2) Provide *augmented information* on how the level is formed such as resource safety or navmesh. And 3) provide *multiple levels of visualization* that transform the designer's sketch into usable levels. Further, the main feature of the tool and its most innovative one is the suggestions by means of an EA powered by three different search algorithms: objective-driven, objective-driven with diversity preservation, and novelty search [155]. The work by Liapis et al. is seminal to analyze and understand how MI-CC systems have evolved and the benefits that they have for designers and AI likewise.

Cicero is a special kind of MI-CC system, where the focus is on helping designers create complete games in the General Video Game AI (GVG-AI) framework⁷ [156] and Video Game Description Language (VGDL) [157], rather than individual game content [158]. In Cicero, the aim is to let the designer create the game they want while receiving suggestions on what content might be added next related to sprites, mechanics, interactions between entities, stats, or game's rules [159]. Technically, Cicero uses a recommender system (Pitako) that using the A-Priori algorithm, learned the multiple and common sequence of actions, sprites, and rules that compose all the database of games in the GVG-AI system. Thus, the suggestions that

⁷<http://www.gvgai.net/>

the designer receives are based on their creation and the statistics behind it in the system rather than exploring possible solutions as for instance, in the Sentient Sketchbook. Machado et al. evaluated Cicero in a user study with eighty-seven students demonstrating that it increased the users' levels of accuracy and computational affect when assisted, and supported one of the main benefits of MI systems, the decrease of participant's workload [160].

Tanagra presents a collaborative scenario where the designer can create platform levels together with an AI that focuses on menial tasks of the creation process, and which in any moment the designer can request to "fill the blank" [161]. Throughout the design process, the designer can place constraints with actual platforms. The AI using a reactive planner either creates a playable level considering the constraints or informs the designer that no level can be created satisfying the set of constraints. Through this, the design process shifted from focusing on the correct placement of platforms, respecting all the possible game rules, to focusing on providing subjective evaluation and exploring the generated content.

While Tanagra presents an approach where the computational designer is designated to "fill the blank" based on the designer's design, more autonomy and initiative can be given to the computational designer for creating content in a continuous design process with the same premise. Morai Maker is a MI-CC tool to co-create levels in the Mario AI framework [162] (a Super Mario Bros. [163] clone for AI research⁸) through turn-taking phases between designer and computational designer [164]. The designer is initially in command of creating the first sketch of the level. Then by passing the turn, the computational designer can add content to the level and when finished, passes the turn and so on and so forth, until the designer is satisfied with their creation. One of the main innovations of the work by Guzdial et al. is that the computational designer is trained through RL, learning as it takes each turn since the designer can delete unwanted content created by the computational designer. Through this, the computational designer continuously learns to adapt to the designer's requirements and goals with positive and negative reinforcement.

Moreover, Lucas and Martinho presented 3Buddy [165], a MI-CC system to create dungeons in the game Legend of Grimrock 2 [166], where the computational designer acts as a colleague working in lockstep. Like

⁸Ahmed Khalifa is the current mastermind behind the Mario AI Framework: <https://github.com/amidos2006/Mario-AI-Framework>

Morai Maker and Tanagra, and with the idea of a conversation between agents, the designer is suggested variations to their current design when requested, which they can use to replace their design, discard it, or use parts of it. The computational designer uses an EA generating individuals in three different pools: *convergence*: similarity between current design and generated individuals, *innovation*: dissimilarity between current design and generated individuals, and *guidelines*: following human-input constraints. The most interesting aspect of 3Buddy is that the designer can specify an area where they will work on and another where the computational designer should focus, thus working simultaneously on different areas of the dungeon.

Furthermore, Karth and Smith's approach uses a modified version of the WFC algorithm [99], which while not strictly a MI-CC system; their approach focuses on the designer providing positive or negative examples to the algorithm, for it to use it to generate variations following such rules. Their novel approach presents a different design process somewhat similar to Morai Maker. Designers show the algorithm what they like and dislike to drive the algorithm's output to their goal [100].

This thesis revolves around the Evolutionary Dungeon Designer (EDD), a MI-CC tool to co-create adventure and dungeon crawler games, particularly, developing their narrative [167, 168] and levels [116]. EDD uses the Interactive Constrained MAP-Elites QD algorithm to continuously suggest adaptive, diverse, and high-performing solutions [137]. As EDD is the main research tool developed and used in this thesis, a chapter is reserved for presenting the tool, all its features and algorithms, and discussing the main contributions around it.

However, while MI-CC systems bring many benefits to design tools such as reducing workload, fostering creativity, providing adaptive experiences, learning design concepts, making game design tools more accessible, or creating various experiences, they have not been adopted by the game industry yet [8]. This is because, firstly, MI-CC tools and common computer-aided design tools such as game engines (Unity, 2005; Unreal Engine 4, 2014), differ in their goals. In the former, the focus is on leveraging each agent strengths and where one's weakness, such as lack of knowledge in game design, can be supplied by the other agent. For instance, using game design patterns to help designers build levels [41, 169]; thus making these tools more accessible. In the latter, the focus is on providing a plethora

of interconnected tools and systems unified in a system that relies on the designer having the complete initiative and expert knowledge to connect the bits that form the design of the game. Secondly, to have a natural dialogue and collaboration between AI and designers as discussed by Horvitz [20], both need to understand each other design processes such as intentions and goals. Thirdly, to enable more autonomy in the interaction between human and machine, and give a varying degree of initiative to the machine to co-create the game content a game designer has as a goal, these tools are required to identify and use different designer's processes and design procedures. Therefore, the following section is devoted to discussing *designer modeling*, an approach to achieve the before-mentioned third point, through modeling certain designer's processes and use them to drive the generation of content.

Modeling players and designers

Player modeling relates to the study of players in-game to compose computational models on the player's characteristics that arise when interacting with games as cognitive, affect, and behavioral patterns [170, 171]. Through this, the aim is to understand the player's experience when interacting with a game. Player modeling usually relies on data-driven and ML approaches with user-generated gameplay data, and have been used with a vast amount of goals. For instance, for automating playtesting [52, 172], identifying player types (using Bartle's taxonomy [173]) based on their playstyle [53], to understand and model in-game player's motivations [54], or for market purposes, to understand how players play and are engage in free-to-play games [174–176].

Furthermore, the combination of Machine Learning (ML) with PCG has led to the rise of Procedural Content Generation via Machine Learning (PCGML), defined as the generation of game content by models that have been trained on existing game content [101]. PCGML has been used for autonomous content generation [177], content repair [178], mixed-initiative design [164], or content adaptation [179]. The use of user models is essential for the generation of adaptive and tailored content, and when discussed in the context of PCG, usually relates to experience-driven PCG [50].

Content adaptation can take place as players play or use the content online or offline, building models from collected data. For instance, Duque et al. adapt and adjust the difficulty of generated content as players play the game using bayesian optimization [179]. Summerville et al. model players

automatically and implicitly by learning from video traces; generating levels that correspond to the latent player models [180]. Player models could also be used to enhance and adapt design tools, specifically MI-CC tools [52, 181]. Further, training models on gameplay data from *Tom Clancy's The Division* have also been used to model, and therefore find predictors of player motivation [54], which renders a very valuable tool for understanding the psychological effects of gameplay. Former research followed a similar approach in *Tomb Raider Underworld*, training player models on high-level playing behavior data, identifying four types of players as behavior clusters, which provide relevant information for game testing and mechanic design [53]. Melhart et al. take these approaches one step further by modeling a user's *Theory of Mind* in a human-game agent scenario [182], finding that players' perception of an agent's frustration is more a cognitive process than an affective response.

Designer Modeling

Understanding player behavior and experience, as well as predicting the player's motivation and intention, is key for mixed-initiative creative tools while aiming to offer in real-time user-tailored procedurally generated content. Nevertheless, the main user of MI-CC tools are designers, and gameplay data is replaced by a compilation of designer-user actions and AI model reactions over time while both user and model are engaged in a mutually inspired creative process. A fluent MI-CC loop should provide good human understanding and interpretation of the system, as well as accurate user behavior modelling by the system, capable of projecting the user's subsequent design decisions [183]. In the same line, goal thirteen in the guidelines for Human-AI interaction [184] highlights the importance of learning from user behavior and personalize the user's experience by learning from their actions over time.

Shifting towards a designer-centric perspective means that besides focusing on player modeling, it is necessary to focus on modeling the designers. Liapis et al. [2, 45] introduced designer modeling for personalized experiences when using computer-aided design tools, with a focus on the integration of such in automatized and mixed-initiative content creation. The focus is on capturing the designer's style, preferences, goals, intentions, and iterative design process to create designer models. Through these models, designers and their design process could be understood in-depth, enabling adaptive experiences, further reducing their workload and foster-

ing their creativity. Most of the focus has been on player modeling when generating content [52, 181], but the nature of MI-CC systems and its *adaptiveness* goal require this designer-centric shift. Modeling preferences has been the focus on some recent work using as a proxy the designer's content and their choices [59, 60, 185], but other modes such as style, underlying goals, or design process are interesting avenues [186]. However, how to capture this content, how to operationalize it into working models, or where to apply them in the pipeline are open questions.

As part of this thesis work, two approaches to model different designer's processes have been proposed, the designer's preference model [185] (PAPER IV), and design style cluster together with designer personas [186] (PAPER IX). The work presented in PAPER IV introduced the Designer Preference Model, a data-driven solution that learns from user-generated data in the EDD. This preference model uses an Artificial Neural Network to model the designer's preferences based on the choices they make while using EDD, which is then used to drive the content generation. Moreover, The work presented in PAPER IX uses data from the design process of 180 sessions to analyze the room styles created along the process, yielding twelve clusters representing such styles. The design process was again analyzed in function of these formed clusters, where we encountered four archetypical paths, i.e., designer personas, that were most commonly taken by designers with the aim to be used to drive the generation of content towards more adapted content.

Computational Creativity

Creativity is “the ability to produce work that is both novel (i.e., original, unexpected) and appropriate (i.e., useful, adaptive concerning task constraints) [187]”. How creative processes occur, how an individual might come up with novel ideas, or how to assess creativity is very much an open research area [10, 188–190]. Moreover, Computational Creativity is a multidisciplinary field that studies computational systems that demonstrate human-like creative behaviors [191]. As a multidisciplinary field, CC is not only interested in the algorithms or the outcome; it also aims to study the creative process and psychological causes of creative behaviors. Thus, through CC, some core concepts and research areas in creativity can be addressed. For instance, in *the Creative Mind: Myths and Mechanism*, Boden studies and analyzes *Creativity* and *creative behaviors* with the use and help of AI through the lenses of Computational Creativity. Boden discusses

three forms of creativity: *combinatorial*: combining existing knowledge in unfamiliar ways to produce new artifacts; *exploratory*: exploring the conceptual space to encounter possible ideas; *transformational*: transforming the conceptual space, the imposed constraints, and the encountered ideas [10].

Within CC, games have been proposed as the optimal artifact to create to test the creative-like abilities of a CC system, since games are *content-intensive, multi-faceted content*, and should be *interacted with and experienced* [25]. As described above, game content relates to the main facets that represent any game: audio, visuals, narrative, levels, rules, and gameplay [61]. Thus, creating systems that develop, to some extent, games poses an interesting application and challenge for CC, which can address some of the core questions in CC. For instance, investigating the creative process not only to create one type of content but the arrangement of such in a harmonious way as a team of humans creatively does, or the assessment of such content.

Using the combinatorial creativity form from Boden, Guzdial and Riedl proposed conceptual expansion. Conceptual expansion is an approach that combines neural networks trained to recognize or generate specific content to produce a *combinet* that could be used to recognize or generate novel content, which lacks enough data to use it to train a new ml model [192]. Moreover, they applied their approach to the conceptual expansion of games, with the same idea of creating novel combinations of games from a set of models trained to produce content for specific games [82]. In the same line, Sarkar et al. proposed the use of variational autoencoders (VAE) to create new levels by training the VAE with game levels from Super Mario Bros. and Kid Icarus. Through this, the VAE learns a representation of both game levels, and using [193, 194].

Moreover, Mikkulainen discusses the use of Evolutionary Computation to achieve creative AI, which refers to the use of AI not only to create and perform creative tasks such as generating games, but also to encounter creative solutions to complex multidimensional problems. In his work, he reflects on the aims of the AI field and discusses the use of search-based approaches for exploring complex multidimensional spaces filled with “unknown unknowns” with exciting results [195]. This is further supported by the collection of Lehman et al. [196] that presents creative examples by a myriad of EC researchers. Likewise, Sarkar discusses

leveraging on creative AI techniques to approach game design, and with such demonstrated exploratory work on how it could be achieved, and the benefits from it [197]. Specifically, Sarkar discusses the co-design aspect that can be enabled through creative AI techniques, which is especially relevant for this thesis and the development of effective MI-CC systems.

AI METHODS

This section describes the approaches relevant to the work presented in this thesis. First, an introduction is given to the main area of the thesis, Evolutionary Computation (EC), followed by a brief introduction to MAP-Elites. Finally, Machine Learning (ML) is briefly introduced and discussed.

Evolutionary Computation

Evolutionary Computation (EC) is a subfield within AI inspired on Darwin's theory of evolution [198] and Darwinian principles of natural selection and evolution of population over generations to primarily solve/optimize a problem or task, with the premise of "survival of the fittest". EC is a family of population-based algorithms that focuses on searching a multidimensional space for solutions through executing an iterative refinement loop. The basic premise is that by having a set of individuals in an environment to be experienced or with tasks to be solved, arises competition that causes natural selection, which results in finding high-performing solutions. Evolutionary Algorithm (EA) is a subset of EC, which applies a set of evolutionary mechanisms in the refinement cycle: *selection*, *variation operators*, *evaluation*, and *replacement* [199].

A typical EA starts by *creating* a set of random solutions in a multidimensional space and *evaluates* them using some fitness function. Based on this measurement, solutions can be sorted, and better candidates can be *selected* to seed the next generation and *variation operators* such as recombination or mutation can be applied to them to create a new set of candidates, i.e., the offspring. These solutions are once again *evaluated*, and compete against the current population to *replace* it and become part of the next generation. This process is repeated until a solution of sufficient quality is found, which ends the execution. In such a loop, Eiben and Smith highlight two evolutionary mechanisms as fundamental for continuously



Figure 1: Example of direct encoding

producing and encountering high-performing individuals and creating diversity. The *selection* mechanism, which increases the pressure on selecting high-performing individuals for variation that ultimately results in increasing the quality of the population. The *variation operator*, which varies individuals to produce candidate solutions, creating the necessary diversity, and achieving novelty [199].

Moreover, under the subset of EA there exist four main variants: *Genetic Algorithms*, *Genetic Programming*, *Evolution Strategies*, and *Evolutionary Programming*. However, their distinction is mainly on the individuals' encoding, i.e., how each individual or solution is internally represented, which also limits what variant operators can be applied. For instance, Genetic Algorithms use genes encoded as finite strings, while Evolution Strategies' individuals are represented as real-valued vectors where approaches such as Covariance Matrix Adaptation could be applied [126]. Genetic algorithms are used for the work presented in this thesis. The content to be evolved (i.e., levels in a dungeon) is represented as a finite set of integers, and operators such as mutation and recombination through crossover can be applied.

Evolutionary Algorithm Components

Representation

Individuals (i.e., solutions) within a population have two representations: genotypes and phenotypes. *Genotype* is the individual's internal representation within the EA, while *Phenotype* is the "translation" of the genotype when acting in the environment. For instance, human genotypical representation is the DNA (genotype), while the body, brain, organs, etc. are the phenotypical representation. Such representation and translation are also called *encoding*, which can fluctuate from direct to indirect encoding.

Direct encoding refers that the genotype is mapped bit-by-bit to the phenotype. In contrast, indirect encoding refers to the opposite, the genotype is minimally encoded, and its representation does not match the phenotype. For instance, in the case of evolving tile-based rooms in a dungeon, a direct encoding could mean that the genotype is an array with integers, each denoting a space in the room and the corresponding tile in the phenotype (shown in figure 1). A lesser direct encoding could have the genotype be an array that groups together areas of the room and marks them as specific areas, or an even lesser let it just be a ruleset (such as in L-systems [75, 200]) to create the levels. Encoding and representation of individuals are one of the main challenges in EA since the encoding can drastically change the evolutionary mechanisms and how the content is generated and explored [201–203].

Evaluation

In an EA, it is usually used a fitness function to assess the population and solutions. This estimates the quality of the solutions by testing some metrics that estimate and rank these solutions. Fitness functions are usually context-dependent and help solve the tasks at hand by using some representative heuristic of the task. For instance, if evolving levels in a game, quality can be measured based on the tile distribution or the challenge vs. reward. However, objective-based functions might not be the best way of evaluating content. Environments and tasks could be deceptive with a space filled with local optima, limiting the search; or diversity among the solutions might want to be rewarded rather than just ranking. Stanley and Lehman discuss such challenges with objective-based functions and proposed using divergent searches for stepping stones to solutions, with the aim of open-endedness [204]. More work and discussion on this is presented in section I.

Selection

Selection is used to choose the parents of the next generation of candidate solutions; thus, it chooses which individuals variation operators will be applied to. Selection approaches are biased towards selecting high-quality individuals; after all, these are the ones with the best chances to generate better candidates. However, to counter greedy strategies and avoid getting stuck in local optima, lower-quality individuals still get a chance to be selected (with a similar idea to tabu-search).

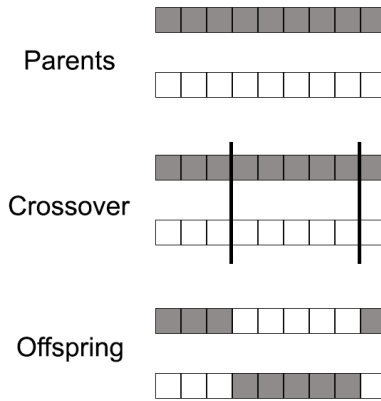


Figure 2: Example of the crossover variation operator

Variation Operators

Once individuals from a population are selected, a set of variation operators are applied to create candidates for the next generation. The operators might be *Mutation* or *recombination* as crossover. Mutation is applied to a single individual with the aims of varying some genes from the genotype to create variation and diversity. For instance, if mutation is not applied, the EA is limited to the genes encountered in the initial population; if a key gene was not produced, the global optima might never be found. Common mutation operators are to swap a gene for another in the genotype or randomly change a gene for a random value. Crossover requires at least two individuals that can, as the name indicates, cross their genes to form new candidates. Such is exemplified in figure 2. The result of applying these operators to the selected parents is a set of candidate solutions (offspring) that are evaluated and compete against the current population for a place in the next generation.

Replacement

Replacement strategies mainly focus on replacing lower-quality individuals from the population for better candidates generated through the variation operators, hence, “survival of the fittest”. However, as it will be explained in section I, not all EA work the same. Some have local competition with individuals with similar behavioral novelty [114]; other approaches have competition among phenotypical similars to preserve innovations encountered in the search space [205].

MAP-Elites

Quality Diversity algorithms are a relatively new family of algorithms that leverage the strengths of divergence and convergence search [49]. One algorithm from this family is MAP-Elites, which explores the space yielding a collection of high-performing and diverse solutions. The main characteristics of MAP-Elites are: 1) that it returns a collection of high-performing yet diverse solutions to address multiple wanted behavioral characteristics, for instance, robots that can move around and have a set of different behaviors and characteristics [118]. 2) Using behavioral features as dimensions and discretizing the space with cells helps the algorithm illuminate the search space and control niches of solutions. And 3) By using these features and cells, the algorithm is able to substantially explore the space finding in the way a greater amount of solutions than in other approaches. MAP-Elites has been extensively evaluated and resulted in a far superior approach to convergence search (i.e., objective-based) or divergence search (e.g., novelty search) alone, and other QD approaches such as NSLC [120].

MAP-Elites, while superior to other approaches, its iterative cycle is not that different as the one presented earlier in this chapter. The main changes that MAP-Elites introduce are: 1) instead of having one population, the EA uses cells; and 2) besides the fitness function to calculate the quality of individuals, it requires a set of behavioral feature dimensions to divide the space into cells and where individuals will be stored. In the vanilla version of MAP-Elites, each cell contains one individual, and as the search encounters new individuals within the same cell, the individual with higher-quality is kept, and the other is disregarded. This way, the algorithm is able to preserve only high-quality solutions while retaining diverse solutions.

Moreover, the algorithm is shown in Listing 1, which first initializes a collection of individuals evaluated with the fitness function and tested for their behavioral features to be placed in specific cells. Then random selection⁹ occurs on top of cells to pick individuals that then compete in a classical selection strategy (e.g., comparing fitness) to produce candidate solutions. Variation operators are applied to the selected parents, and the offspring are evaluated with the fitness function and tested for their behavioral features. Depending on the cell the offspring belong, they will

⁹Gravina et al. have experimented on using other selection approaches with beneficial and exciting results [206]

need to compete if occupied with the current occupant or if unoccupied, the offspring is placed in the cell, and the cycle restarts. With such a simple algorithm, MAP-Elites can encounter and return a collection of diverse and high-performing individuals. Firstly, this is due to the pressure in cells for high-performing individuals. Secondly, due to retaining individuals in these multidimensional cells, where they might be entirely different for another solution (e.g., have a complete different genotype) yet be as high-performing.

Algorithm 1 Pseudocode description of the MAP-Elites Algorithm. Taken from [120]

```

procedure MAP-ELITES ALGORITHM (SIMPLE, DEFAULT VERSION)
  ( $\mathcal{P} \leftarrow \emptyset, \mathcal{X} \leftarrow \emptyset$ )
  for iter = 1  $\rightarrow$  I do
    if iter < G then
       $\mathbf{x}' \leftarrow \text{random\_solution}()$ 
    else
       $\mathbf{x} \leftarrow \text{random\_selection}(\mathcal{X})$ 
       $\mathbf{x}' \leftarrow \text{random\_variation}(\mathbf{x})$ 
       $\mathbf{b}' \leftarrow \text{feature\_descriptor}(\mathbf{x}')$ 
       $p' \leftarrow \text{performance}(\mathbf{x}')$ 
      if  $\mathcal{P}(\mathbf{b}') = \emptyset$  or  $\mathcal{P}(\mathbf{b}') < p'$  then
         $\mathcal{P}(\mathbf{b}') \leftarrow p'$ 
         $\mathcal{X}(\mathbf{b}') \leftarrow \mathbf{x}'$ 

  return feature-performance map ( $\mathcal{P}$  and  $\mathcal{X}$ )

```

Furthermore, MAP-Elites have become popular and attractive due to all the abovementioned benefits and characteristics, which have not only spread its use in many fields and many experiments but also sparked many variations. The Constrained MAP-Elites by Khalifa et al. [134] is the one this thesis relies on and has expanded. Their approach added populations in each cell rather than individuals, preserving even more solutions, and combined MAP-Elites with FI-2Pop, which yielded two populations per cell, one driven by the fitness and the other driven by satisfying a set of constraints. Through this, they applied the same process as with the vanilla MAP-Elites but per population (i.e., feasible and infeasible), which resulted in useful and interesting results for the generation of bullet hells bosses.

Machine Learning

Machine Learning (ML) is a sub-field of AI that focuses on using learning algorithms that can learn from data and that are trained through some strategy such as supervised learning or reinforcement learning [207]. Formally (and generally), learning in ML was operationally defined by Mitchell [208] as: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .”

The task T in ML is not the learning *per se*, rather learning is the way to attain the ability to solve the tasks. ML helps us solve complex tasks that are deemed too complex to be solved by fixed programs such as the creation of games [101] or playing games [209, 210]. Tasks in ML might be a *classification* task: what category k an input belongs to [211]; a *regression* task: where it is asked to predict some numerical value based on some input; *synthesis* task: create new examples based on the training samples [212]; or *machine translation* task: translate an input from one language to another [213].

Moreover, performance P relates to how a learning model is assessed to check that it is learning from experience E to tackle task T . Depending on the type of task that the model must solve, the performance measure would vary, as it is dependant on it, similarly as to how fitness functions are dependant on the problem to be solved. The usual performance measures used are **accuracy** and **error rate**. For tasks such as image generation the **inception score** is normally used [214] or for machine translation the **BLEU** is used to measure the quality of the translated text [215]. The key aspect when evaluating performance in a task is that the learning model should be tested with data it has not used for learning, thus showing the ability of the model to solve “unknown” tasks [207].

Furthermore, experience E is related to the data and examples provided to the tool and what strategy is used to train a learning model. Learning strategies in ML can be categorized in two main learning strategies: *Supervised* and *Unsupervised* learning. However, Reinforcement Learning (RL) has gained tremendous interest from the research community, and is the current learning strategy that is used to solve many tasks due to the metaphor regarding how human’s learn and the fact that the learning happens by experiencing the environment rather than learning the data [216]. Self-Supervised Learning (SSL) has also been gaining popularity as an approach

to move away from traditional supervised training by not using human-annotated dataset to learn representations of data [217].

Within Procedural Content Generation and games, ML has been increasingly used, gaining popularity to generate different types of content. PCG via ML is a prospect area that encompasses all algorithms and approaches where the generated content is the output of models trained on existing content [101]. On the other hand, ML can take advantage of PCG approaches that continuously create content to increase generality, as discussed by Risi and Togelius [39]. Moreover, Liu et al. discuss the different deep learning approaches used thus far for PCG, the open areas for research, and more in detail on the benefit of these approaches both for the deep learning community and the PCG community [218].

EVOLUTIONARY DUNGEON DESIGNER

This section presents the Evolutionary Dungeon Designer (EDD) research tool, all its subsystems and features, and the multiple interactions between the human designer and the computational designer. First, it describes the tool's objectives, the iterations it has gone through, and game design patterns as a significant feature. Then, it presents and discusses the room and narrative generation process, the intertwined attempts, and multiple designer interactions. Finally, it describes the tool's workflow and presents some examples.

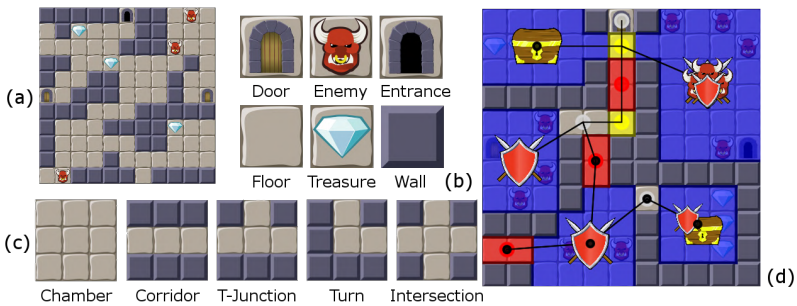


Figure 3: Main level design components in EDD. (a) Basic example room, (b) different placeable tiles (the boss tile looks the same as the enemy tile but 3 times bigger), (c) spatial-patterns and (d) meso-patterns

The EDD is an MI-CC system to create adventure and dungeon crawler content, similar to the ones found in Zelda [46] or the Binding of Isaac [47]. The main feature of EDD is the collaboration between human and computational designer to create levels and narrative. Within level design, the human designer focuses on editing the room to design their goals while in parallel, they are continuously offered a set of suggestions adapted to their

current design. In addition, the designer's design is continuously evaluated for design patterns, feasibility and playability constraints, and enhanced information of the rooms such as door safety or enemy-treasure balance. Within narrative design, the human designer can focus on the creation of quests and overarching narrative structures, while the computational designer collaborates by suggesting quest actions and auto-completing quests, and suggesting diverse narrative structures adapted from the designers structure and level design constraints. Likewise, the computational designer assesses the level design changes that might invalidate quests and change narrative structure constraint.

The first iteration of EDD was described and presented by Baldwin et al. [41, 219]. In their work, the aim was on creating the first steps towards a MI-CC system that allowed the designer to create fixed-sized individual rooms while receiving four generated rooms with diverse targets upon request. EDD was developed further with a revamped UI, allowing the creation of complete dungeons, and providing augmented information on the suggestions compared to the current design [26, 220]. The next version of EDD allows the designer to create dungeons in whatever layout preferred, and incorporates the Interactive Constrained MAP-Elites, providing a customizable grid of suggestions steered by the selected feature dimensions [116, 137]. Narrative aspects were included in the following iterations. Automatic objective assessment [221] and quest creation and generation [167, 222] were included in the next iteration. Dungeons were assessed for overarching objective placement, and designers could add different NPCs and quest items in their levels to couple quest actions to them, receiving suggestions from the computational designer. Finally, the last iteration included the creation of overarching narrative structures [168, 223], where designers could set characters, factions, conflicts, objectives, and main or side events, while receiving suggestions in the same line as when creating rooms using the IC MAP-Elites. All of the available views where the designer can create content and interact with different parts of the system are shown in figure 4. In addition, EDD now includes prototype implementations of different designer models: the designer preference model [224] and designer personas [186].

Level Design

Dungeons in EDD are a cyclical graph composed of interconnected rooms. Rooms are a rectangular $M \times N$ grid of tiles, which might be: *Floor*, *Wall*,

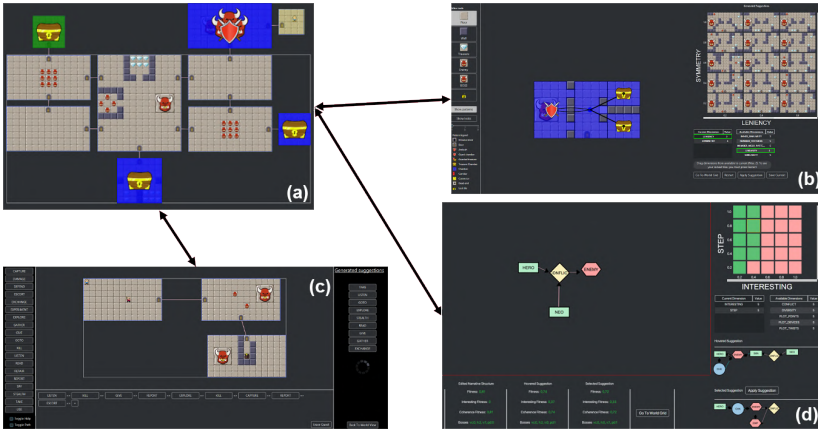


Figure 4: Workflow of the Evolutionary Dungeon Designer. (a) Shows the world view with a prototype dungeon (PAPERS I, II, III) and its automatically assessed objectives (PAPER VI). (b) Shows the room view, the main interactive view of EDD. In this view the designer can edit their room while the computational designer is offering suggestions for their design (PAPERS I, II, III, V). (c) Shows the QuestGram view, where designers can use their current dungeon to add different quest actions linked to the room elements (PAPER VII). Finally, (d) shows the Story Designer view, where designers can create the overarching narrative structure for their game (PAPERS X, XI).

Treasure, Enemy, Enemy Boss, and Door (all shown in figure 3.b). *Wall* tiles are obstacles that cannot be traversed by the player, while all the other are considered passable and could be “interacted” by the player. *Enemy boss* tiles are a special type of tile that occupies a 3×3 area, as its challenge might be comparable to nine enemies, and only two at a time can co-exist in a room. Doors cannot be placed at will by the designer but might be added when connecting rooms in the world view.

An essential part of EDD is its use of level design patterns to hierarchically divide a room into micro-patterns (inventorial and spatial patterns) and their combination into meso-patterns. The level design patterns in EDD are based on [106, 169, 225]. While EDD is an MI-CC tool to create dungeons for adventure games, we can leverage design patterns to create a generic and domain-independent tool. Through this, we do not need strict definitions or constraints based on specific tiles or functionalities. Rather we rely on patterns that can be made into specific content by the designer if needed.

The design patterns are used in two ways. First, they are used to evaluate the designer’s design and show the designer how the system categorizes different parts of the rooms. Second, and more important, design patterns are used to estimate the quality of a generated room. By extracting the

patterns of the designer's design and evaluating the generated rooms based on that, design patterns can be used as goals to be achieved by the generated content. In EDD, we have defined *micro-patterns* as the minimum bits that compose the room, and are divided into *inventorial* and *spatial* patterns. *Inventorial patterns* are individual passable tiles, and *spatial patterns*, are a combination of passable tiles and walls resulting in chambers, corridors, and different intersections.

Meso-patterns are the next level in the design pattern hierarchy. They are the composition of micro- and meso-patterns. The meso-patterns that are currently implemented are (shown in figure 3.d): *dead-end*, *ambush*, *guard chamber*, *treasure chamber*, *guarded treasure*.

Room Generation

The main feature of EDD is to suggest variations of the designer's work that are adapted to their design, interesting, and can foster the designer's creativity. Through this, we seek that EDD ends representing the role of a colleague, as discussed by Lubart [48]. The suggestions use the designer's current room configuration as target ratios (e.g., the number of corridors, the number of inventorial patterns, etc.) to provide adaptive suggestions. However, due to the algorithm's nature, the designer is also suggested content that respects these ratios but might use them differently with a different goal.

For instance, if the designer is creating a room with many corridors, such as a labyrinth, they will be provided with suggestions with a similar distribution of corridors, but utilizing the rest of space in different ways, as shown in figure 5.

EDD uses the IC MAP-Elites (explained in detail in PAPERS III, V) to generate and suggest rooms to the designer. Its main features are the use of divergent and convergent searches, the search space division into cells, the use of behavior feature dimensions, the constrained population per cell, and the designer's ability to interact with it. Through this process, EDD can provide a grid of evolved high-performing suggestions, adapted to the designer's current design, while representing a diverse set of solutions. For instance, in figure 5, it is shown multiple evolutionary runs when using the same design and the set of generated suggestions. It can be observed that the designer is provided with a set of suggestions that retained their expected ratios and design, but diverse enough that the designer can browse

many different variations. It can also be observed the effect of the different feature dimensions, as some of them do not match the expected ratios adequately. Thus, producing bigger variations in the solutions to explore the search space. In figure 6, it is presented an overview of the IC MAP-Elites steps applied to the level design facet and the multiple areas designers can interact.

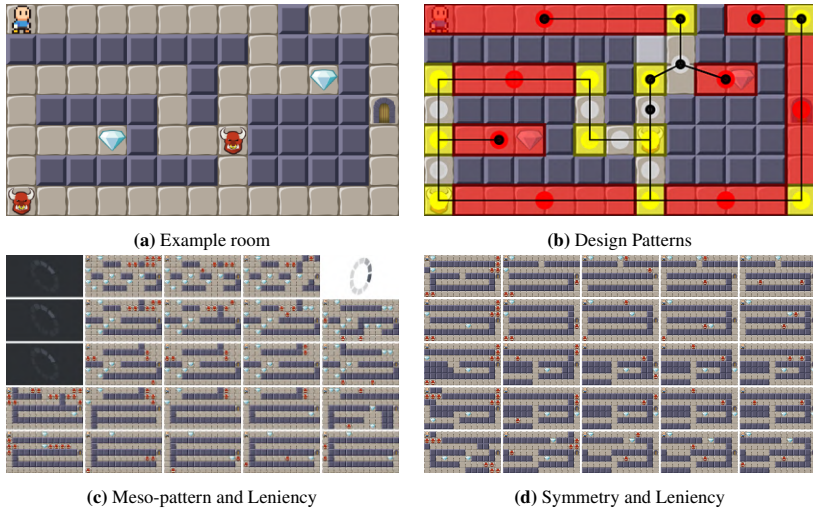


Figure 5: Example of a possible room created by a designer (a), the design patterns identified by the system (b), and two suggestion grids presented to the designer (c and d). The rooms in both suggestion grids, were generated using IC MAP-Elites using respectively, #meso-patterns and leniency (c), and symmetry and leniency (d) as dimensions.

Narrative Design

Within EDD, narrative is explored in three individual components: main and secondary objectives based on dungeon’s topology and level design patterns (PAPER VI), quests based on the individual tiles (PAPER VII), and overarching narrative structure (PAPERS X, XI). These components part from the same idea, using and leveraging patterns and structures to formalize the generation and propose content to the designer. It is important to also highlight that in EDD, in its current state, these narrative components are subordinate to the level design facet. This means that changes in the rooms and dungeon have effects in the narrative, which the computational designer assess, informs the user of the needed changes, and aids on these changes.

Automatic Objective Assessment

Leveraging on the dungeon's structure and the patterns that arise from the designers' design for each level, we calculate possible objectives in the dungeon. The main idea is to utilize the dungeon as much as possible, which means that dead-end rooms without bosses would yield side objectives for players to encourage exploration through side objectives. Likewise, depending on the content within rooms, a room would be selected as the main objective for players, prioritizing boss rooms and rooms farthest from the start position.

The process is simple and asynchronous from the design tasks, but with significant design impact. It gives insight to designers on what type of objectives are assessed based on the rooms they are designing, and the dungeon's topology. However, these objectives just show the assessment done in EDD, but have no functional use. This means that designers could take this objective assessment, agree with it, change the design to achieve other objectives for players, or disregard the assessment.

Quest Design

Narrative was further explored by implementing a quest editor for designers to compose quests within EDD, called QuestGram (in PAPER VII). QuestGram provides a new view in EDD for designers to concatenate abstract quest actions connected to individual tiles among all the designed rooms to design quests (a long quest sequence with implicit subquests, which was reformulated as explicit subquests in [222]). The system also makes use of two new tiles, quest item and quest NPC that certain quest actions can be assigned to. For instance, "Listen" can only be associated to quest NPCs, while "Damage" can be assigned to several tiles.

QuestGram is a mixed-initiative system within EDD that complements EDD's original purpose as a mixed-initiative system for level design. The computational designer uses grammars (specifically, L-System) to propose suggestions to the designer based on the quest they have created so far. The grammar, quest actions that are provided to the designer, and the type of NPC are adapted from the work by Doran and Parberry [226], which analyzed quests in four different MMOs, and extracted their common structure. As the designer adds quest actions and connects them to different tiles, the computational designer continuously generate quests using the grammar, and the ones that match the current quest are used to propose the

next possible quest action for the designer. The designer can then either use or disregard this suggestions. They can also request a quest action from the system ad hoc at any position of the quest by simply clicking on it.

QuestGram is coupled to the level design facet since it requires rooms to be designed beforehand for quest actions to be connected to some tangible element. This means that editing the rooms by adding or removing elements affect what quest actions can be added, and can also invalidate parts of the quest in case of tile, doors, or room removal. Once the designer is back at the quest view, they are showed the affected quest actions and can either decide to manually replace, remove, or use the suggestion to readjust their quest.

Narrative Structure Design

When creating objectives, stories, and narrative in general, we can consider their design in a different abstraction level. Rather than defining specific objectives, creating quests bit by bit, or coupling these elements to actual in-game elements, we can define narrative structurally, which can describe an experience or story with an abstract representation as argued by Barthes [227] and Szilas [228]. We can create structures using a set of generic building blocks that when connected together define narrative structures. This, in turn, define generic aspects of a story, leading to the identification of events, roles, and narrative elements, without the need of defining the plot or how they will take place.

In PAPER X, we introduced *TropeTwist* as a step towards defining and identifying narrative structures in games. TropeTwist uses tropes extracted from TVTropes [229] as building blocks and patterns, which can compose together structures represented as graphs in TropeTwist. To evaluate and analyze these structures, we defined a set of patterns that can arise from the interconnection of tropes: *micro-patterns*, fundamental narrative units in the system such as conflicts, characters, or plot devices; *meso-patterns*, emerging from the combination of micro-patterns and other meso-patterns, they denote spatial, semantic, and usability relationships within the narrative graph; and *auxiliary patterns*, used to identify structural gaps in the graph. Patterns are evaluated based on their quality, and in turn, these are used to evaluate the structure as a whole syntactically (i.e., coherence and consistency) and semantically.

TropeTwist was implemented within EDD as a parallel mixed-initiative

system called Story Designer (PAPER XI). Story Designer lets designers create narrative structures in the form of narrative graphs as seen in fig. 4.d, using TropeTwist’s building blocks while receiving suggestions adapted to their structures. Suggestions are generated using IC MAP-Elites adapted to generate narrative structures by evolving graph grammars (i.e., adding and removing rules, and changing both their left- and right-hand side part of rules). Thus, the designer can freely create their narrative graph and the computational designer proposes a suggestion grid adapted to their graph, akin to the level design facet.

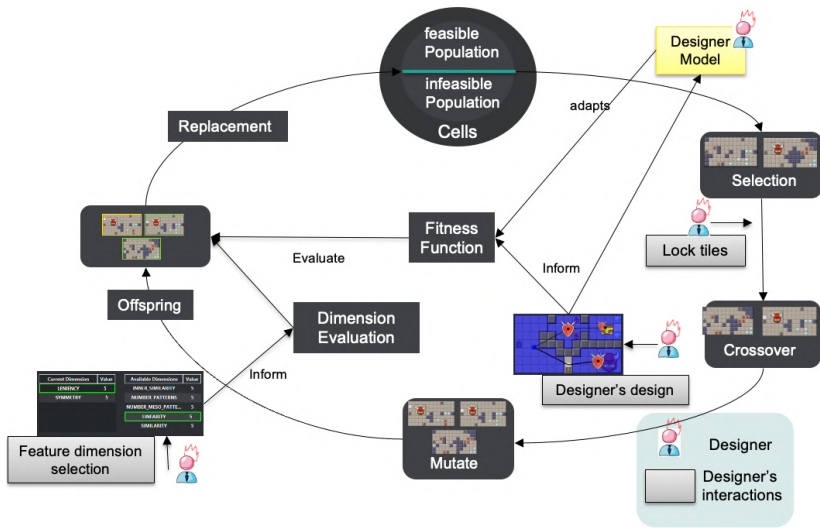


Figure 6: IC MAP-Elites evolutionary loop showing the possible designer interactions and what part of the loop these interaction affect

Designer Interaction

A significant feature in EDD, is the control and interaction designers have with non-intuitive components of the algorithms. In EDD, the designer always have influence on what the computational designer will create either directly or indirectly. Within the level design facet, the designer can interact with the IC MAP-Elites in four different ways: 1) *Locking tiles*: A special tile that locks tiles in the room to be unchangeable (PAPER II). 2) *Feature dimensions*: The designer can change the feature dimensions of the IC MAP-Elites at any time, reshaping the search space (PAPERS III, v). 3) *Current design*: The designer is indirectly in constant interaction with the IC MAP-Elites by simply designing their room, which adapts the

evaluation of generated solutions (PAPER I). 4) *Designer Model*: As the designer creates their content and uses the suggestions, the system is fed with their interactions, trying to form a model of their preferences (PAPER IV) and a style and goal model based on their associated designer personas (PAPER IX).

Within the narrative facet, the designer can interact with it by controlling or influencing the respective algorithm such as the IC MAP-Elites when designing narrative structures or quests, and with the facet itself based on their level design. Concretely, the designer can interact with the algorithms within the facet in two ways: 1) *Developing Quest*: As the designer adds or remove quest actions, the computational designer provides “next quest action” suggestions based on the quest up to the selected quest action by the designer (PAPER VII). 2) *Feature dimensions and narrative structure*: Similar to the level design facet, the designer can interact and change the feature dimensions of the IC MAP-Elites when creating narrative structures and at the same time, IC MAP-Elites constantly adapts to the designer’s narrative structure, leveraging the design as target metric (PAPER XI). Further, the designer’s level design conditions what quest actions will appear and the validity of the quests, what objectives are assessed within the dungeon, and constraints the search space of the IC MAP-Elites when generating narrative structures.

RESEARCH METHODOLOGY

This section describes the overarching research methodology, the methods used and how they were applied, a methodological reflection, and a discussion of other methodologies that were considered, and which are as valid for this thesis.

To explore game design as a Human-AI collaborative task, how game content might be generated and adapted, and how to better integrate AI in these systems and processes, this thesis mainly follows a Design Science Research Methodology (DSRM) [230]. DSRM aims at producing innovative artifacts to address a set of problems, challenges, and research questions within an area of concern. This thesis investigates this through a mix of qualitative and quantitative methods. Qualitative methods such as user studies, interviews, and controlled experiments, to collect data from usability, experience, and clarity of the interaction. Quantitative methods such as simulation and experimentation, where the goal is to evaluate specific properties, trade-offs, and the scope of the artifacts.

Moreover, artifacts created for the purpose of creating technology-innovation, are categorized according to DSRM in the following four types: *constructs*, *models*, *methods*, and *instantiations*. *Constructs* are the symbols and composed language to represent and define the problem and possible solutions. *Models* are the representations of the problems and solutions, which uses the previously defined constructs. *Methods* are the processes that are followed to solve the represented problem. Finally, *instantiations* are the final systems used to demonstrate that the other categories can be used as solutions to address the defined problems [230]. Through the thesis, four artifacts have been developed.

Evolutionary Dungeon Designer

the Evolutionary Dungeon Designer, defined as **instantiation**, is the main research tool in this thesis where we can investigate game design and multiple content creation aspects as well as the interaction between human designer and computational designer. EDD has enough flexibility to explore this interaction and to use it as a research tool to investigate the performance, behavior, and usability of multiple AI techniques and models. Thus, by developing and evaluating EDD, it is possible to support the development and creative expression of humans, and to test systems that aim at improving and exploring the co-creative, co-creation, and co-design capabilities of MI-CC.

We have developed four systems that build on top of EDD to explore this interaction. EDD main content generation area is within level design, which is explored and developed in PAPERS I-III, V, VI. Within EDD, we have also explored the generation of narrative, and its connection and usability alongside the level design facet. Narrative is explored and implemented in QuestGram (PAPER VII), TropeTwist (PAPER X), and Story Designer (PAPER XI).

Computational Designer

The computational designer (composed of multiple AI techniques) related to **model, method, and instantiation**, where each underlying AI technique is implemented, used, and evaluated within EDD. Through this, it is explored how different approaches affect the human-AI interaction, while aiming towards fostering creativity, reducing workload, and creating a much tighter Human-AI dynamic. These AI techniques range from addressing specific challenges such as explicit controllability by human designers and their aesthetic consideration (PAPER II), to the use of quality-diversity algorithms to generate content in different facets (PAPERS III, V, X, XI), to the use of different type of grammars such as L-systems (PAPER VIII) and graph grammars (PAPER X).

Furthermore, it is relevant to explore not only what algorithms and AI techniques are used for the Computational Designer, but also its role in the interaction. Thus, the role of the computational designer and its agency was preliminary explored in PAPER XII. We analyzed and evaluated through a controlled experiment in the form of user study, the interaction and user experience of the human designer when the computational designer gains

more agency in the final design.

Temporal Expressive Range Analysis

Expressive Range Analysis (ERA) is a popular and relevant method to evaluate content generators, since it allows the inspection of the diversity and expressiveness of a content generator with regards to feature dimensions and use as a comparison with other algorithms [3, 142, 231]. In PAPER VIII, we extended ERA and presented Temporal Expressive Range Analysis (TERA) as a method to better use ERAs in interactive PCG and MI-CC systems. TERA relates in DSRM as a **method**, and is proposed as a evaluation method to analyze and assess these interactive systems. TERAs allow the inspection and analysis of changes in the ERA over a defined period such as generations, design editions, or time; and how the algorithms react and adapt to constant changes in the input. We introduced an aggregated and non-aggregated version. Non-aggregated TERAs show the delta maps of the generator, meaning where the generator has focused and the space covered for a specific period part. Aggregated TERAs show the density of generated content over all the defined period and the coverage up to the specific period. TERAs were used in PAPERS VIII AND XI to assess the content generated with the IC MAP-Elites for game levels and narrative structures.

Designer Personas

In this thesis, we argue about the importance of *Designer Modeling* in MI-CC systems as a step towards adaptability, tailor-made experiences, and better interactions. We approached and investigated this in PAPERS IV AND IX. Nevertheless, in PAPER IX we presented and proposed an approach, related in DSRM as a **method**, for a data-driven modeling of the design process, style, and goals. This method allows for the identification of *Designer Personas* as archetypical paths through a clustered design style space in the respective design space. *Designer Personas* take the task of recognizing style and adapting content towards that to a higher abstraction layer, where both the final design and individual design edits are less relevant, and the process and trajectory through a clustered style space based on designer's data is highlighted. For instance, rooms for a game like the binding of Isaac [232] could be classified based on multiple characteristics such as the room's objectives regarding enemies and treasures, access to different areas, or hidden encounter and treasures. Moreover, different designers

could reach the same room style through different paths, where the focus along the creation could vary. Some designers would focus on the room's architecture before anything else, whereas others would focus first on the objectives a player must achieve.

Methods

All the publications included in this theses have followed some of the possible design evaluation methods introduced by Hevner et al., which are categorized into observational, analytical, experimental, testing, and descriptive [230]. Thus far, the focus has been on doing analytical, experimental, and descriptive methods, with a mix of quantitative and qualitative data collection.

The systems were evaluated with *analytical* and *experimental* methods. Analytically through a dynamic analysis to evaluate qualities such as performance and expressiveness. Experimentally through simulation with artificial and meaningful data to, for instance, analyze the exploration capabilities, reactivity, interaction, and adaptability of the algorithms. The *dynamic analysis* focused on performing expressive range analysis and temporal expressive range analysis (PAPERS III, V, VIII, XI). The *simulation* has focused on evaluating how the tool responds to different inputs and the robustness of algorithms with noisy and unpredictable data such as in PAPERS III, VII, OR XI.

Moreover, *controlled experiments* in the form of user studies have also been conducted to collect qualitative data on how the focus group (i.e., game designers) experience and use the tool. Through this, informed decisions can be taken on focus areas that might be important and relevant to explore as shown in the results of PAPERS I, IV, VII, IX, XII.

Finally, Hevner et al. [230] recommended the use of *descriptive* methods only when innovative artifacts cannot be evaluated using any of the other evaluation methods. Therefore, having in mind that this is still an unexplored research area, descriptive methods such as *Informed Argument* are required to build support for an artifact's necessity, challenges, and applications. For instance, the method and model developed in PAPER IX, used data from user studies that resulted in the clustering of the design space into design style clusters. We further used this to explore and identify Designer Personas.

Methodology Discussion

This thesis embraces DSRM as the principal methodology employed to explore, analyze, and address the different artifacts and RQs. However, it is essential to have a methodology discussion for those methodologies that could have been as valid in the frame of this thesis, with their own set of tradeoffs.

The field of Human-Computer Interaction (HCI) has studied for a long time the ways interactions can be created, approached, and established to foster, encourage, and address different functionalities and capabilities of humans. Likewise, the interaction between humans and AI has been the focus of much of the recent work by the community, which is now developing further than the computer science and information system frontiers towards intertwining with many other science fields [233].

However, these interactions, especially when the human and the AI do not necessarily need to have asymmetric functionalities (i.e., having an intelligent system that merely assist) and that both can participate in reciprocal stimuli to reach a common goal, still have uncountable challenges and questions to research and explore. This is thus, one of the motivations behind this thesis, which in part calls for the use of a more exploratory and holistic methodology that can then observe, analyze, model, and address the unknown space of problems and solutions.

An exploratory methodology that could have been used is Research through Design (RtD) [234] since, after all, the iterative design of multiple artifacts and the investigation of how this opens new spaces in the research is important to understand those unknown spaces. RtD is a methodology from the field of HCI, which aims at tackling “Wicked Problems” meaning unexplored, unclear, and complex scenarios and situations that are not easily simplified. *Design Thinking* is the design process used to describe RtD, consisting of *grounding, ideation, and iteration*. Following the RtD methodology, one addresses these problems through a holistic approach with an iterative process that reframes the problem through designing artifacts and prototypes as a means of knowledge. The knowledge generated from these prototypes reframes the space of problems and solutions, which in turn provides an important opportunity for researchers on focusing in the future state and on defining a preferred state to be achieved [235].

Moreover, Participatory Design (PD) matches as well the requirements of this thesis, especially understanding that the research while focusing on

the development of technological innovations and artifacts, has a human-centered perspective [236]. PD is a methodology that involves the target user of a specific type of design in the design and research process. Through this, the necessities of the target user are gathered, and their visions, expertise, and understanding are incorporated and included in the proposed solutions. Usually, the researcher wants to explore certain aspects of an unknown space, although it is not necessary to be unknown, together with the target users. These users understand and manage different problems and solutions in the design space, thus by co-designing the solution, they are involved and are part of the solution rather than this being delivered to them [236].

These methodologies could allow the work to be, in principle, more exploratory and letting the users actually participate in the design of the human-AI interaction loop. However, through the methods within DSRM and the research plan, this thesis has explored to a certain extent the incorporation of important aspects of these methodologies. While the main focus is on EDD to explore this interaction; throughout this thesis, multiple prototypes and artifacts addressing the underlying AI have been developed to explore different design capabilities and interactions that have helped reframed problems and space of solutions, akin to RtD.

Furthermore, we have managed to evaluate qualitative aspects of the tools and the interaction designers have with it through user studies. Thus, from our focus groups (i.e., game designers), there are important points to approach and address that motivates the next steps in the thesis. For instance, in the user study done in PAPER I, it was a recurrent topic that designers wanted more control over the tool and algorithms. Designers did not see a relation between their work and what the tool offered (i.e., preserving their design, intentions, and ideas), while of course, still expecting interesting solutions from the algorithm. This topic, supported by previous research, have chained a set of improvements and implementations in order to balance the requirements from designers. For instance, in PAPERS IV, IX, and in general, the steps towards capturing and developing designer models to be used as a complement evaluation in the interaction between the designer and the artificial design, have been motivated by the struggle and balance between controllability and expressivity.

CONTRIBUTIONS

This section summarizes the research contributions of the publications that this thesis compiles and unifies. First, contributions and publications are linked to different RQs. Then, each RQ is presented and discussed from the perspective of the different included papers' contributions.

Table 1: Relationship between the different research questions and the publications

RQ	Papers
RQ I	I, II, III, V, VII, VIII, X, XI, XII
RQ II	I, IV, IX, XII
RQ III	IV, VIII, IX
RQ III.I	IV, VIII IX
RQ III.II	IV
RQ IV	VI, VII, X, XI
RQ IV.I	VII, X
RQ IV.II	VII, X, XI

RQ1: How can we use and integrate multiple algorithms such as quality-diversity algorithms and grammars into a mixed-initiative approach to help designers produce high-quality content and foster their creativity while allowing them to control, to a certain extent, the generated content?

QD algorithms have been recently introduced as a family of algorithms that leverage both convergent and divergent searches' strengths. Specifically using strategies that help the search explore a greater area of the space while retaining high-performing individuals. However, how to handle these algorithms together with a human user giving inputs, changing conditions, and with certain goals in mind is non-trivial. Moreover, using these [and

other] algorithms in collaboration with human users, providing control over the algorithm’s output, is an open research area. This is mainly due to the many non-intuitive aspects of these algorithms, such as the variation operators or the genotype-to-phenotype conversion in Evolutionary Algorithms. Likewise, this is due to the use of design tools such as EDD by inexperienced human users or non-programmers, and the focus of these tools, where the human user should design their objective rather than focusing on the algorithms.

Therefore, we have focused on giving designers control over multiple non-intuitive aspects of the EA, specifically, the Interactive Constrained MAP-Elites (IC MAP-Elites). In PAPER I and based on related work [41], the goal was to understand and analyze what are the challenges game designers encounter with MI systems (specifically, with EDD). This drove the study into how to give control to users while preserving the algorithm’s expressiveness and the use of QD algorithms as an alternative. In PAPER II, it was investigated how to give the designer explicit control over non-intuitive strategies and parameters in an intuitive way. Specifically, what “genes” could be selected and which not for crossover and mutation within the EA, and as a consequence, preserve the designer’s intentions with their design.

Built on top of the Constrained MAP-Elites by Khalifa et al. [134], in PAPER III and PAPER V, we introduced the Interactive Constrained MAP-Elites, the first use of MAP-Elites in a mixed-initiative setup. Through IC MAP-Elites we added: *interaction* for the designer with MAP-Elites, *continuous adaptation* of the generative space of the algorithm to the ever-changing designer’s design, and a set of dungeon-like related features. This resulted in increasing diversity in the exploration of the search space while retaining high-performing solutions. Furthermore, it was established control mechanisms of non-intuitive aspects of the EA for the designer through controlling the feature dimensions that discretize the search space. The interaction and collaborative benefits for both humans and MAP-Elites were further explored in PAPER VIII. Our experiments support that the human designer’s interaction can aid the algorithm to find new areas and constant generation of novel individuals (i.e., increase and aid its expressiveness), and at the same time, the human designer would receive more adaptive and tailor solutions by indirectly guiding and controlling the search.

While the aforementioned papers focused mainly on level design aspects, PAPERS IX, X, XI focus on the generation of narrative elements such

as quests (PAPER VII) and narrative structures (PAPERS X, XI), and their interconnection with level design. In PAPER VII, we explored the use of quest patterns extracted by Doran and Parberry [226], in QuestGram, an MI-CC tool implemented in EDD. In fast iterative steps, designers could create levels and use QuestGram to create quests for the system using abstract quest actions. The system used entirely grammars to produce suggestions to designers, which were based on what the designer was creating. Designers could then control the computational designer output by requesting specific elements and could further use this when quests became invalid due to changes in level design; then, designers could simply resolve with the suggested action. In PAPERS X, XI, the generation and co-creation of narrative structures are explored. Our objective was to explore narrative at a higher abstraction level; whereas quests provide tangible objectives, narrative structures provide structural information on the game, such as overarching objectives and conflicts, roles, factions, and relevant plot events. Narrative structures are encoded as graphs, and the computational designer uses the IC MAP-Elites to search graphs generated using graph grammars. Similar to PAPERS III, V, VIII, the designer can guide and control the algorithm with their current graph while also constraining the search with level design elements.

Finally, in PAPER XII we explored the impact of AI agency in MI-CC systems. Human Designers and Computational Designers are progressively losing and gaining agency, respectively, over the final design. Our preliminary results indicate that losing control over the AI, which as an indirect consequence, takes the initiative over the design's goal, and that this is not aligned with their view, frustrated most designers. Our goal was to investigate if, by giving more control to the computational designer and constraining the design space, the human designers' creativity could be fostered to overcome these constraints [10, 237, 238]. Our approach then becomes a naive baseline to use as a comparison when further investigating how to vary human-AI collaborative capabilities to achieve goals and tasks, and foster human abilities.

RQ2: How can we use player and designer data to better understand their behaviors and procedures to enhance and adapt Mixed-Initiative Co-Creative systems?

Mixed-Initiative Co-Creative tools such as EDD, can benefit greatly from player and designer data. However, how to collect and use this is not straightforward, especially when the focus is not only to analyze and understand the user's behavior but also to actively use the data to enhance and adapt their experiences. For instance, player data such as where they are observing [239] or their experience [50] can be used to model how the end-user might perceive certain content. Designer data can be used to understand design processes and enhance the designer's experience by creating designer-tailored content and by modeling common designer practices and processes [2].

Therefore, we have delved into collecting, analyzing, and using player and designer data, reported in PAPER I, IV, IX, XII. Thus far, we have conducted four user studies with this in mind: The first, reported in PAPER I, where we collected qualitative data from experienced game designers on the interaction with EDD as a game design tool. The second user study reported in PAPER IV was conducted with beginner game designers, i.e., first-year game design students, and consisted of a mix of quantitative data, i.e., actions within the tool, and qualitative data, i.e., comments on the experience and usability of the tool. The third user-study is reported in PAPER IX, which consisted primarily of increasing the scope of the study in PAPER IV with data from a more diverse and wider group. Finally, the fourth user study is reported in PAPER XII, where we evaluated how designers and their final design is influenced by an AI with increasing agency over the final design.

With the data collected and the techniques employed, it was possible to analyze certain designers' actions and characteristics. The data collected corresponding to the designer's actions in PAPER IV and PAPER IX, not only allowed us to create models representing certain designers' procedures but also highlighted interesting design processes. Further, the data collected in PAPER XII gave us an insight into the design process of- and highlighted constraints in the design space for both humans and AI. The results help us understand further problems with establishing deeper collaborations with AIs in colleague roles, which is the overarching goal within MI-CC.

RQ3: How can we model different designers' procedures and use them as surrogate models to anticipate the designers' actions, produce content that better fits their requirements, and enhance the dynamic workflow of mixed-initiative tools?

There is a need for the AI to recognize design and creative procedures to have an aligned collaboration with the user. This is in order to create adaptive experiences and to fruitfully make these experiences enable an in-depth loop between humans and AI. Therefore, collecting designers' data as they worked in the tool was paramount, as described in the contributions for RQ2. However, making use of this data into a functional model of the designer is not trivial, as well as what to do with such models. PAPER IV and PAPER IX explore such a paradigm, where we proposed multiple approaches to model different but related processes. Likewise, in PAPER VIII, we analyze the effects for both designer and algorithm when the designer interacts with MAP-Elites.

The approach presented in PAPER IV focused on creating a preference model of the designer. This was then used to steer the generation of suggestions into more meaningful, interesting, and preferred suggestions. It leveraged in the IC MAP-Elites implicit relation between cells along the behavior dimensions and the suggestion grid's visualization. Through this, it estimated and collected the designer's preferences based on the current set of suggestions. As the designer chose suggestions in the grid, an ad-hoc preference matrix was placed, estimating each suggestion's preference in the grid. The estimated preference was used to compose a training set to subsequently train-and-test a neural network representing the designer's preference. The network was then used in the fitness evaluation of each new individual in the EA. The designer was then proposed a new set of suggestions that fitted their preferences, adapting seamlessly to the designer without interrupting their design process.

Moreover, the results from PAPER IV, drove the approach presented in PAPER IX, which focused on creating a general offline model of design style, specifically, when creating dungeons. To create such a model, we conducted two user studies with a diverse group of participants, i.e., game design students, game industry practitioners, and AI in games researchers. From these studies, it was used the design process of each of the created rooms (180 unique rooms), i.e., from an empty room to its final version. By clustering this design process, we were able to identify twelve

representative clusters. Further, by analyzing the same design processes, but in relation to the clusters rather than individual changes, we identified four *designer personas*. These designer personas are archetypical paths that most designers followed during the design process.

Both publications present examples of how multiple design processes can be modeled as a designer model, their usability, and their impact in the generation process. The preference model in PAPER IV was presented, implemented, and tested. While the designer personas and design style clusters in PAPER IX were discussed from a wider perspective on how they could be used. Furthermore, besides aiming at modeling different designer's procedures, the main difference is how they are created: the preference model is an online-personal model that uses data from single designers. In contrast, the design style clusters and designer personas are offline-group models created on data from a diverse and wider group. Thus, we explored multiple paths to capture design processes and use them to enhance design tools through both.

On the other hand, in PAPER VIII we did not model design processes, rather we assessed how the designer could steer and influence the computational designer just by designing their levels. In EDD we use the designers design steps implicitly in the search algorithm and adapt the fitness functions towards that. This rather simplistic way of modeling their design process is enough to steer IC MAP-Elites towards areas of interest, and adapt the content towards the designers.

RQ3.1: What trade-offs arise from modeling and using designer's procedures to steer the generation of content towards personalized content?

In PAPER VIII, we introduced TERAs as a way to study how design steps affect the search space throughout a time period. With TERAs, we could identify how the designer's design affected the search space and changed the algorithm's focus. Our results showed that IC MAP-Elites is able to adapt and have stable generation of content as the designer's design moves through the design space. However, this adaptability, while wanted, makes the algorithm focus more on particular regions of the search space, exploiting those for the designer to receive more relevant content, and leaving some others unexplored. At the same time, IC MAP-Elites also benefits by adapting to the designer, as unexplored regions that could be deemed mutually exclusive, are fully explored once the designer is able to

provide examples there.

Moreover, PAPER IV and PAPER IX present novel approaches to model specific designer procedures to create content and collaborate with designers in a more adaptive and meaningful way. However, the application of these models into the design process and to drive the computational designer's collaboration arises multiple challenges and benefits. Such trade-offs were explored and discussed in PAPER IV where some of these trade-offs were posted as three open areas for active research: 1) *Dataset Creation*: the challenge on acquiring data, 2) *Preference Modality*: the challenge on using representative data, and 3) *System's Training-and-Usage*: the challenge to train and use ML models dynamically or statically.

As designers use and interact with design tools, it must be decided what type of data should be used that better represent the procedure or process to be captured. Individual data could allow for a more adaptive experience, but collecting data from a single designer in a single session, might not be enough to accurately train such a model as in PAPER IV. In contrast, using collective data might decrease that tailored experience, but it could point out towards frequent processes that are simultaneously followed by several designers as in PAPER IX.

Nevertheless, despite the use of collective or individual data to create these models, the challenge remains on what data captures the different processes faithfully. Seemingly representative data could be dependant on other attributes, which might be or not counterproductive to collect or analyze. In PAPER IV, the data used was based on the suggestion grid, which implicitly used the cell relation of the behavior dimensions. For example, this meant that selecting a very symmetric room as the preferred one automatically meant that the less preferred suggestion was an asymmetric room. For other processes, it might be simpler to match data with the process. For instance, the design style clusters presented in PAPER IX were formed using individual rooms' design process. However, each designer's design process is different and still presents many unknowns; thus, the challenge of using representative data prevails.

Likewise, *concept drift*: the constant change in the training set for an ML model is a major challenge in design tools, as when designers use the tool, they have an ever-changing design process that varies greatly. This was highlighted by testers in PAPER IV, as the computational designer's suggestions were not aligned with the current design, mainly because how

and when the model was trained. The model was trained every time the designer chose a suggestion, and these events could be very far away from each other. For instance, the designer starts with some goal when creating their room, and when choosing a suggested room, they expect this to help them reach their goal. However, their goals are by no means needed to be taken with them to the next room. Such a challenge partly motivated the research in PAPER IX. Where rather than having a model that tries to update as the designer traverses through the generative space, the space is already clustered, and models do not update with the designer. Through this approach, we aimed at clustering the designer's design in an already clustered design style space. Through this, they could be provided with adaptive experiences as the computational designer could make informed decisions based on where the designer's design is and where is headed.

RQ3.2: What constraints are created over the generative process when using designer models?

Regardless of these trade-offs, using designer modeling impose implicit constraints in the generative system similar to any other system that adapts its functionality to satisfy a set of constraints. These constraints act as a set of guidelines to help the generative process select more appropriated suggestions such as in PAPER IV, or to indicate possible steps the designer might take as in PAPER IX. However, having these constraints also limits, to some extent, the generative space and expressive range of the AI. In PAPER IV, this was explored by using the preference model as part of the weighted sum of the fitness function to infer if a generated room might be preferred. Through this, the EA evaluated the generated content objectively through the fitness function, and subjectively through the preference model. If using an accurate model, the designer could receive suggestions aligned with their preference. This could mean that the generation would focus on some specific area of the generative space with the possibility of limiting both the creativity of the computational designer and fostering the designer's creativity.

RQ4: How can level design and narrative interact, act as constraints, be intertwined, and in general, have an active role affecting each other to produce a holistic system?

Level design and narrative are two facets that are continuously linked [63–65]. Not only because of their importance for players but due to the role

that they have and play in each other's output and human perception. The designer's creations as constraints to generative models and algorithms within their respective facets are explored in RQ1 and the designer models' constraints in the generation in RQ3. In RQ4, we are interested in the role that other content representations and facets have as constraints for the generation of "unrelated" content, e.g., the level design in narrative design. We investigated this by developing parallel systems such as EDD (PAPERS I, II, III), QuestGram (PAPER VII) or TropeTwist (PAPER X), that then could be intertwined (PAPER XI).

Furthermore, within Holistic PCG, we explored how and what level design aspects, either indirectly or directly, and implicitly or explicitly, could have a role in narrative. In PAPER VI, we considered the existing patterns in levels and the dungeon's structure to automatically assess the main and side objectives in the game. This means that designers could control [indirectly] the objectives with their level design. In PAPER VII and further explored in [222], designers had direct control over objectives and quests that would exist within EDD using abstract quest actions and dependant on the level design elements. Designers could add NPCs, quest items, and the rest of the elements, which could be used to create quests. Any addition or removal when designing the dungeon would alter the possible quests. PAPER X and PAPER XI explored the creation of narrative in a higher abstraction layer. Instead of defining plots, quests, or stories, designers could shape and design narrative structures. Similar to QuestGram, level design elements could be used to constraint these structures and their generation as shown in PAPER XI.

RQ4.1: What are the factors to be considered when implementing such a paradigm and system in a mixed-initiative application, where a designer will be able to interact with the content?

PAPER VII and PAPER XI present approaches where narrative and level design is implemented, exemplified and tested in a mixed-initiative system. These systems, QuestGram and Story Designer, respectively, are preliminary steps towards *holistic mixed-initiative systems*, where designers can interact with EDD and design parts of both facets. However, as the designer creates these facets, it becomes important what content is used, how it is represented, and how changing it could affect the other facet.

In PAPER VII, we explored the iterative loop of designing both facets,

where altering the level design had a direct effect on the created quest. How changes in the level design affect the quests might not be straightforward, especially when constantly iterating between both. Thus, we emphasized the communication of how these changes affected the quest, either because elements were removed or paths were blocked, which in turn showed to the designer what quest actions were affected. The designer could then manually change them or use one of the proposed suggestions to fix the quest for them. In PAPER XI, Story Designer was proposed as an MI-CC tool to build narrative structures using TropeTwist and EDD as foundations. Through four controlled studies using simulated data, we explored how level design data could be used to constrain the generated suggestions. Level design data was used to limit the quantity and roles to appear in the final generated and suggested narratives (i.e., how many villains, heroes, and plot devices). Thus, communication about how elements are interpreted is necessary, as well as investigating how the system is constrained and how to show these constrained spaces to designers. In Story Designer, it was chosen to constraint the algorithm, but not the manually edited structure. This is due to the ambiguous nature of TropeTwist, and how quickly changes to the structure reformulate these roles and patterns, which is visible in PAPER XI experiment 4.1-4.5. Thus, the computational designer is constrained, but not the human designer. However, the computational designer should adapt to the human structure, which could arise interesting uses of narrative meso-patterns to overcome the imposed constraints.

RQ4.2: What are the effects of producing and using a holistic system for the creative process of a designer, and what challenges are imposed on computational creativity?

There are many factors and aspects to consider when discussing holistic PCG systems, their implementation in MI-CC systems, and how to best do this to assess the human-AI collaboration and interaction. However, due to the different interactions designers and computational designers have with the content, and how the content is intertwined, it is relevant to explore the effects and challenges for them. When designers used QuestGram (PAPER VII), they reported increased creativity when using the mixed-initiative, especially when quests became large due to, for instance, containing too many level design elements as designers got stuck (akin to writers' block) and the system showed alternatives. This points towards that in the early stages of the design, designers might not be overwhelmed with creating

content in different facets, but as content increases, mixed-initiative systems slowly become more relevant. This is understandable, as game facets are usually not designed and developed by a single designer but by a team collaborating, which is the aim of MI-CC. Given the system's simplicity and the use of patterns, the computational designer in QuestGram is able to cope with these requirements. However, in Story Designer (PAPER XI), rules are less clear, and we use IC MAP-Elites to search the possibility space. We assessed, through simulations, how constraints affect the search space, which showed similar, stable, and consistent results regardless of constraints. This points that the task itself is hard to explore, but as shown in PAPER XI experiment 4.1-4.5, and in PAPER VIII, IC MAP-Elites benefit from the human interaction; thus, it is a promising area to keep exploring. We, however, hypothesize that Story Designer will create a similar situation for designers as in QuestGram; the longer the design session, the more relevant the mixed-initiative and the AI role will have in the design process. This phenomenon could then be intrinsically attached to the holistic nature of these tools, which should be considered further.

DISCUSSION AND CONCLUSIONS

This thesis explored game design and game content creation through human-AI collaboration. We used EDD and its extended inner systems (QuestGram, TropeTwist, and Story Designer) to analyze and study MI collaboration. The focus has been on developing techniques and algorithms to investigate this interaction to highlight and argue for the benefits that can be achieved. Specifically, mutual inspiration to explore unknown design areas, foster the designer’s creativity, and establish adaptive experiences.

Control, Expressiveness, and Adaptability

The interaction between designers and AI arises multiple dynamic properties such as initiative, control, and expressivity. *Initiative* relates to how either agent engages in the tasks and to what extent. *Control* relates to the control mechanisms enabled for either agent to direct or constrain the output of other agents based on some criteria. *Expressivity* relates to the diversity of solutions that either agent can create. A strong candidate to cope with both the control and expressivity dynamic properties that arise in the interaction are QD algorithms. Thus, in PAPER III and PAPER V, it was introduced and implemented the Interactive Constrained MAP-Elites, a variation of the MAP-Elites algorithm. Among its features, it enabled the designer to select feature dimensions (a key component of MAP-Elites, explained in section I). The results from PAPER III pointed towards enabling enough control since selecting feature dimensions and their granularity changes the search landscape and the retained solutions. However, this changes the features a designer might be interested in but does not limit the algorithm’s expressivity to create diverse solutions. Further, the fitness function of IC MAP-Elites continuously adapts to the designer’s design, acting as an indirect control mechanism.

The results from PAPER III and PAPER V are further supported by the results in PAPER VIII, where the behavior and generative space of IC MAP-

Elites were analyzed by simulating design sessions. Given that one of the designer's interactions is the automated adaptation of the fitness based on the current design, we examined and evaluated how the search varied and adapted to a design session. The results showed that the algorithm adapts adequately, and the designer has, to a large extent, an impact on the generative space with their design. More exciting is that IC MAP-Elites is able to explore new areas of the space by adapting to the designer's design.

In short, controllability, in many cases, is a competing property with expressivity. Especially since the control and constraints imposed could limit the expressive range for any of the constrained agents. However, in this thesis, it is shown that QD algorithms; specifically, IC MAP-Elites, can cope with this, showing robustness, adaptability, and stability when interacting with it. Designers are provided with meaningful control, yet IC MAP-Elites adapted and kept exploring vast amounts of the generative space encountering high-performing solutions. Nevertheless, PAPER XII showed the importance of agency regarding control, adaptation, and initiative for either agent. Our preliminary results reported high frustration levels as human designers got too constrained, losing agency and ultimately control over the final design and objective. Constraints limit the space, and as a consequence, they need to be overcome by discovering creative solutions. However, the system needs to provide enough space and creative freedom for designers to encounter these creative solutions.

Adaptive Experiences

The work in PAPERS I, II, III, V, VII, XI, XII and the interest in seeking alternative approaches to foster creativity, create adaptive experiences, and enable more autonomy and initiative for the AI directed the research towards designer modeling. PAPER IV and PAPER IX presented designer modeling examples by modeling different design procedures. These could be used as surrogate models to enhance the understanding of design processes and the usability of design tools, such as EDD. PAPER IV presented a clear artifact design used to steer the generation of new suggestions based on the in situ created preference model. This work demonstrated the benefits that come with integrating these models in the MI loop, such as the possibility of seamlessly creating preferred content. However, it also demonstrated the challenges of selecting and collecting representative data or training-and-using models as designers develop.

Furthermore, PAPER IX presented the development of a novel model

to analyze the designer's design process, which could inform generative processes on the designer's style, goal, and intentions. The analysis of the resulting clusters based on each designer's design process resulted in the designer personas. These designer personas were presented as archetypical paths taken by designers through the clustered style space. Both models allow for the analysis of design and creative processes from a higher abstraction level rather than specific steps akin to procedural personas or game design patterns.

These approaches toward designer modeling have shown the capabilities of modeling several procedures and how they could be used. They also show that design processes can be analyzed more abstractly, yielding interesting similarities among seamlessly different designers or design processes. Designer modeling has the possibility to create adaptive experiences for an individual or group of designers and could enable more autonomy and initiative for the AI. However, whereas this and its usability as surrogate models to enhance the collaboration, interaction, and generation produce actual benefits to the dynamic workflow of MI-CC tools remain open for exploration as a promising area.

Towards Holistic PCG and MI-CC

In PAPERS VII, XI, we experimented with MI-CC narrative generation where adaptation and control were not only based on what the designer directly created in the narrative facet but also elements from the level design facet. QuestGram limited designers with the level design content, and the computational designer could be used repeatedly to address level design changes. On the other hand, Story Designer limited the computational designer with level design constraints, but the human designer could create freely. This could allow the human designer to be more exploratory and possibly change constraints back to the level design while the computational designer tries to adapt and overcome the constraints. However, this is left to future work, where these systems should be fully integrated and evaluated with user studies to understand the space of possibilities.

Furthermore, our approaches in connection with related work show the possible intertwining capability of narrative and level design. For instance, for a designer, as discussed in PAPER VII and in [222], it feels natural to connect these elements in relation to objectives since these are already there, to some extent, when designing the levels but not formalized. The relevance of MI-CC tools is highlighted when there is a large search space,

including many level design elements and other elements within the facet.

Revisiting Human and AI Roles

Lubart discusses four different roles a computer might take to promote creativity; *Nanny, pen-pal, coach, and colleague* [48], further discussed by Guzdial et al. based on how designers perceived the AI collaborator [147]. These roles and how designers perceive them are essential to be explored to be understood properly. The role given to the AI could condition the experience and what is expected from each agent. In PAPER XII, we preliminarily explored the effects of the computational designer having more agency and, thus, having more decisions as a creative colleague could have. Our results show that more work is required, such as aligning and recognizing the goals, objectives, preferences, and intentions of human designers to be able to explore deeper relationships and collaborations.

Establishing different roles such as colleague and collaborator might require some user model within the system, such as designer models [2]. Preference models [59, 185] have been built based on designers' choices and used as surrogate models to evaluate further generated content. Similarly, using the designers' creation, the designers' processes and styles could be modeled to inform other systems and adapt the generated content [45, 60, 186]. These models were explored in PAPERS IV, IX, where preference models and style and process models were built, respectively. The aim is to use these models as a complement in the search to tailor the suggested content. This would then direct the search towards areas that are not captured by the objective function and generate more adaptive content.

Nevertheless, Human-AI collaboration is a multi-factor system, and its properties and parameters require that we consider them holistically. Roles, models, interactions, system goals, and purpose are some of the factors to consider [240]. Systems like the one studied in this thesis (i.e., EDD) are creative support tools (CST) that are meant to be used to elaborate some output. On the other hand, CSTs could also be employed and used as autotelic systems. Their purpose and meaning are within the tool and the activity itself rather than the quality of the final product. *Casual Creators* are such a system, where the goal is no other than exploring creativity through creation and short grokloops [151]. Recently, Kreminski et al. proposed *reflective creators*, a particular form of autotelic CSTs, that are meant for users to reflect on their work and creative process [241]. Recognizing what type of CST is to be developed and tested would then specify the

possibilities within the system and the research questions to be addressed.

To Learn or to Evolve? That is the Question

Machine Learning have shown very good and exciting results in many areas [101, 242], and different type of tasks, especially and recently using generative models for language [243], images [244, 245], or games [218]. ML has gained traction and momentum to generate content, mainly due to advances in Deep Learning [246]. Within PCG, this has been categorized as PCGML [101]. Techniques to generate game content using ML have been fruitful as well. Several explored approaches could have been used and are comparable to what is proposed in this thesis with evolutionary algorithms. For instance, Sarkar et al. worked on a series of approaches that combine game content from different games to produce playable levels [76, 197, 247]. Likewise, the works by Snodgrass et al. [95] or Guzdial et al. [82] are great examples of just some approaches to generating content.

Nevertheless, search-based approaches and evolutionary algorithms are still predominant within PCG [71]. While ML approaches show promising results, they rely on [substantial] data to train these models to generate content. This is, firstly, not a necessity for EAs, and secondly, a big bottleneck in PCG as datasets are not widely available. The latter is particularly true when using custom-made systems such as EDD. In PAPER IV, this was part of the issue for creating accurate preference models, among other challenges such as the quick-and-dynamic interactions (i.e., short grokllops [151]). There are, of course, approaches to subdue these problems, such as bootstrapping [212], and this varies depending if the model is to be trained offline or online. Another relevant point is the out-of-distribution exploration problem that ML models encounter. If a model is trained on a certain dataset, the model is constrained to that distribution. This does not mean that generative models couldn't create novel variations, as shown by Sarkar et al. [194] with variational autoencoders (VAE) or Generative Adversarial Networks [207]. However, the search space is considerably reduced. It is, for all purposes, constrained to the distribution; thus, out-of-distribution generation is a big challenge for ML and generative models.

These two challenges are the main technical reasons we focus solely on using evolutionary algorithms, specifically MAP-Elites. EAs can escape the need for data and can explore the search space and generate content out-of-distribution, or at least have a better chance for it. This is particularly

true when using QD-algorithms, given their divergent and convergent capabilities [49]. The complexity then resides in the formulation of an expressive and good enough objective function. In this thesis, we approached this by using Interactive Evolution [248], where we can leverage the human design and extract from it qualities that the human might want from the algorithm. In PAPER VIII, we explore the adaptability and stability aspects of IC MAP-Elites, and how interaction affects both the algorithm and the designer positively. The designer is able to steer evolution and the search for relevant content, which could be a more intuitive way to explore the space than it would if using ML.

Many approaches blend EA and ML as well, which could have been explored, such as [249]. This has not been fully explored and could be a better approach to generating content. Similar to how Mixed-Initiative systems leverage the strengths of humans and AI, ML and EA could leverage their strengths together. Learning from data, especially the one produced by the target group, is relevant and crucial to making systems create more human-like content. Yet, it is as important for the algorithm to be able to explore out-of-distribution spaces, which ML is notoriously bad at doing. Thus, EA could explore with relative confidence these spaces. On the other hand, and separated from PCGML; ML could be used as it is used in this thesis (PAPERS IV, IX), for creating user or designer models that are used as surrogate models to evaluate content [50, 181]. These models could then be used as objective functions or to complement it, steering the search.

FUTURE WORK

We can only see a short distance ahead, but we can see plenty there that needs to be done.

Alan M. Turing, Computing Machinery and Intelligence

The work in this thesis had an ambitious overarching objective, which was partly explored, but that revealed many more exciting areas and paths to continue exploring.

Human-AI Collaborative Properties

This thesis has emphasized controllability as a core property in MI-CC systems that needs to be taken into consideration. We have argued that control can come at the expense of expressivity but with the aim of adaptability towards the designer. Using IC MAP-Elites, we have shown that not only expressivity does not need to be hindered by this, but that control can be a win-win situation for both the designer and the algorithm. However, these properties: *controllability*, *expressivity*, or *adaptability*, miss much of their nuance and their usability when taken as overarching properties. For instance, how can we calculate *controllability* or *adaptability*, and what would that mean? What would it mean to be 30% percent adapted? These questions highlight the problem that these properties encompass too much meaning and require to be broken down and explored deeper. For instance, *controllability* could be explored on how control varies when this is direct or indirect; or when applied to the process, the product, or the other agents; or how it affects and changes the role of each agent. Exploring how these properties can be broken down and how to analyze systems with them to move towards deeper and more developed interactions and collaborations is an exciting open research area.

Designer Modeling and Adaptive Experiences

Two approaches have been proposed to model different designer procedures as designer modeling, but more work needs to be in place to operationalize the findings and models. For instance, using and transforming the designer personas and the design style clustering presented in PAPER IX into an applicable model to study designers and their design and creative process. Moreover, how to use these models to steer the generation of content and create adaptive experiences remains open, as the interaction with designers is dynamic and heterogeneous.

Another interesting future work would be to use these adaptive models to identify and understand the designers' intentions. By knowing the designer's intentions and their current design process, one could adapt and tailor the system towards their needs. For instance, it would be important to find the critical points in the design process where AI could need to change role, adapt the generated content, or take more control, agency, or initiative.

Explainable AI

To establish an in-depth relationship between human and AI, trustworthiness is also required from the AI in order to give more autonomy, responsibility, and initiative to the AI in creative tasks. Explainable AI is a research field that aims at increasing the transparency of AI systems, making AI systems more accessible and more understandable [43, 44].

Zhu et al. [42] proposed the field of eXplainable AI for Designers (XAID) as a human-centered perspective on MI-CC tools. This work discusses three principles of mixed-initiative, *explainability*, *initiative*, and *domain overlap*, where the latter focuses on the study of the overlapping creative tasks between game designers and black-box PCG systems in mixed-initiative contexts. Xie et al. present an example of this [250], where they explored visualization techniques through an interactive level designer tool called *QUBE* to explain and introduce machine learning principles to game designers.

Nevertheless, given the nature of human-AI tools, such as EDD, and the relationship that humans and AI have in these settings, *explainability* could be further explored and intrinsic to the collaboration. For instance, we observed that when using EDD, designers tested the suggestions, how they adapted to their different content, and what are its limitations by just designing their content and focusing on the suggestion grid. Thus,

as designers explore and try different alternatives with these algorithms and models, there comes the possibility of better understanding and interpretation, which is an exciting path to explore.

Holistic Procedural Content Generation

In this thesis, we scratched the surface of holistic PCG and its implementation in MI-CC systems. Whole systems, where facets are not only intertwined, but designers can start the design process from any of them, is a very interesting path. If designers were able to start by designing quests or objectives, and the system could adapt to these and generate levels or structures to support these or vice versa, MI-CC tools would be more aligned with how design processes start. These do not often start from the same starting point but tend to vary, and allowing that variance might be necessary to have different and more fruitful creative experiences. The system would then be in accordance with both Dehn's definition, that space (world) is developed as post hoc justification for events by authors [66], and with Lebowitz, that the story gives meaning to a created world [67].

Exploring Agency and Initiative in Mixed-Initiative

As paths towards adaptive experiences that recognize several designer procedures are explored, it is expected to establish the foundations of a deeper relationship between humans and AI. Through this thesis, control mechanisms were given to designers while not reducing the expressiveness of the AI. Further, examining and modeling different designer procedures, such as their design process, style, preferences, and goals as they create content, have enabled this thesis to explore important steps towards a mature relationship. It is hypothesized that this is needed to create more autonomy and initiative for the AI and to establish a trustworthy relationship between both agents, yet, this might not be enough. This thesis barely scratched the surface of the relationship, the possibilities that emerge with them, and the AI's initiative and role. In PAPER XII, we took the first steps toward altering the role and agency of both humans and AI, and had interesting results that still require much exploration. Therefore, another future direction is to explore varying autonomy, agency, and initiative for both agents and how that affects the interaction and design and creative process of the co-design and MI paradigm.

References

- [1] J. C. Kaufman and R. A. Beghetto, “Beyond Big and Little: The Four C Model of Creativity,” *Review of General Psychology*, vol. 13, pp. 1–12, Mar. 2009. XVIII
- [2] A. Liapis, G. N. Yannakakis, and J. Togelius, “Designer modeling for personalized game content creation tools,” in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19, pp. 11–16, AI Access Foundation, 2013. XVIII, 6, 8, 25, 56, 68
- [3] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016. XVIII, 5, 18, 49
- [4] C. Fiesler, “Innovating Like an Optimist, Preparing Like a Pessimist: Ethical Speculation and the Legal Imagination,” *Colorado Technology Law Journal*, vol. 19, no. 1, 2021. XIX
- [5] S. Klassen and C. Fiesler, ““Run Wild a Little With Your Imagination”: Ethical Speculation in Computing Education with Black Mirror,” in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1, SIGCSE 2022*, (New York, NY, USA), pp. 836–842, Association for Computing Machinery, 2022. event-place: Providence, RI, USA. XIX
- [6] B. Lepri, N. Oliver, and A. Pentland, “Ethical machines: The human-centric use of artificial intelligence,” *iScience*, vol. 24, p. 102249, Mar. 2021. XX
- [7] G. Lai, W. Latham, and F. F. Leymarie, “Towards Friendly Mixed Initiative Procedural Content Generation: Three Pillars of Industry,” in *International Conference on the Foundations of Digital Games, FDG*

- '20, (New York, NY, USA), Association for Computing Machinery, 2020. event-place: Bugibba, Malta. XX
- [8] N. Partlan, E. Kleinman, J. Howe, S. Ahmad, S. Marsella, and M. Seif El-Nasr, "Design-Driven Requirements for Computationally Co-Creative Game AI Design Tools," in *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, FDG'21, (New York, NY, USA), Association for Computing Machinery, 2021. event-place: Montreal, QC, Canada. XX, XXI, 18, 23
- [9] M. Cook, "The Social Responsibility of Game AI," in *2021 IEEE Conference on Games (CoG)*, (Copenhagen, Denmark), pp. 1–8, IEEE, Aug. 2021. XX
- [10] M. Boden, *The Creative Mind: Myths and Mechanisms*. Routledge, 2004. XX, 6, 26, 27, 55
- [11] M. Cook, J. Gow, G. Smith, and S. Colton, "Danesh: Interactive Tools For Understanding Procedural Content Generators," *IEEE Transactions on Games*, 2021. XXI
- [12] M. Charity, I. Dave, A. Khalifa, and J. Togelius, "Baba is Y'all 2.0: Design and Investigation of a Collaborative Mixed-Initiative System," 2022. XXI, 17
- [13] A. Tegen, P. Davidsson, and J. A. Persson, "A Taxonomy of Interactive Online Machine Learning Strategies," in *Machine Learning and Knowledge Discovery in Databases* (F. Hutter, K. Kersting, J. Lijffijt, and I. Valera, eds.), (Cham), pp. 137–153, Springer International Publishing, 2021. XXI
- [14] M. J. Nelson and M. Mateas, "Towards Automated Game Design," in *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing* (R. Basili and M. T. Paziienza, eds.), (Berlin, Heidelberg), pp. 626–637, Springer Berlin Heidelberg, 2007. 4
- [15] M. Cook, "Software Engineering For Automated Game Design," in *2020 IEEE Conference on Games (CoG)*, pp. 487–494, 2020. 4
- [16] M. Cook, "Getting Started in Automated Game Design," 2020. Publication Title: Tutorial at the 2020 IEEE Conference on Games (COG). 4, 18

- [17] J. F. Allen, “Mixed-Initiative Interaction,” in *Intelligent Systems, Trends and Controversies section*, pp. 14–16, 1999. 4, 12, 20
- [18] D. Novick and S. Sutton, “What is Mixed-Initiative Interaction?,” in *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, Jan. 1997. 4, 9, 19
- [19] L. Birnbaum, E. Horvitz, D. Kurlander, H. Lieberman, J. Marks, and S. Roth, “Compelling Intelligent User Interfaces—How Much AI?,” in *Proceedings of the 2nd International Conference on Intelligent User Interfaces, IUI '97*, (New York, NY, USA), pp. 173–175, Association for Computing Machinery, 1997. event-place: Orlando, Florida, USA. 4
- [20] E. Horvitz, “Uncertainty, Action, and Interaction: In Pursuit of Mixed-Initiative Computing,” in *Intelligent Systems, Trends and Controversies section*, pp. 17–20, 1999. 4, 20, 24
- [21] E. Horvitz, “Principles of Mixed-Initiative User Interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '99*, (New York, NY, USA), pp. 159–166, Association for Computing Machinery, 1999. event-place: Pittsburgh, Pennsylvania, USA. 4, 9, 20
- [22] E. Horvitz and T. Paek, “A Computational Architecture for Conversation,” in *Proceedings of the Seventh International Conference on User Modeling, Banff, Canada, New York: Springer Wien*, pp. 201–210, June 1999. Edition: Proceedings of the Seventh International Conference on User Modeling, Banff, Canada, New York: Springer Wien. 4
- [23] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-Initiative Co-Creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 4, 20
- [24] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, “Can Computers Foster Human Users’ Creativity? Theory and Praxis of Mixed-Initiative Co-Creativity,” *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016. 5, 8, 20
- [25] A. Liapis, G. N. Yannakakis, and J. Togelius, “Computational Game Creativity,” in *Proceedings of the fifth International Conference on*

Computational Creativity (ICCC 2014) (H. Toivonen, S. Colton, M. Cook, and D. Ventura, eds.), 2014. 5, 27

- [26] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering Creativity in the Mixed-initiative Evolutionary Dungeon Designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 5, 6, 20, 38
- [27] Mojang Studios, “*Minecraft*.” Game [PC], May 2009. Microsoft Studios, Washington, US. Last played October 2019. 5
- [28] T. Fox, “*Undertale*.” Game [PC], 2015. Fox Toby, Boston, US, Last played December 2018. 5
- [29] E. Barone and ConcernedApe, “*Stardew Valley*.” Game [PC], 2016. Chucklefish, London, UK. Last played January 2018. 5
- [30] T. Fullerton, C. Swain, and S. Hoffman, *Game Design Workshop: Designing, Prototyping, & Playtesting Games*. Gama Network, Taylor & Francis, 2004. 5
- [31] J. Blow, “Game Development: Harder Than You Think,” *Queue*, vol. 1, pp. 28–37, Feb. 2004. 5
- [32] M. Washburn Jr., P. Sathiyarayanan, M. Nagappan, T. Zimmermann, and C. Bird, ““What Went Right and What Went Wrong”: An Analysis of 155 Postmortems from Game Development,” in *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016 SEIP Track)*, May 2016. Edition: Proceedings of the 38th International Conference on Software Engineering (ICSE 2016 SEIP Track). 5
- [33] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018. 5, 14
- [34] Hello Games, “*No Man’s Sky*.” Game [PC], August 2016. Sony Interactive Entertainment, Tokyo, Japan. Last played March 2017. 5
- [35] D. Yu and Mossmouth, LLC, “*Spelunky*.” Game [PC], December 2008. Mossmouth, California, US. Last played December 2019. 5
- [36] N. Walton, “*AI Dungeon 2*.” Game [PC], 2019. Latitude Inc. Last played April 2020. 5, 14

- [37] M. Toy, “*Rogue*.” Game [PC], 1980. Epyx, San Francisco, US. Last played November 2017. 5, 14
- [38] D. Braben and I. Bell, “*Elite*.” Game [BBC Micro and Acorn Electron], 1984. Acornsoft, Cambridge, UK. Last played February 2012. 5
- [39] S. Risi and J. Togelius, “Increasing generality in machine learning through procedural content generation,” *Nature Machine Intelligence*, vol. 2, pp. 428–436, Aug. 2020. 6, 36
- [40] J. Clune, “AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence,” 2019. _eprint: 1905.10985. 6
- [41] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Mixed-initiative procedural generation of dungeons using game design patterns,” in *Proc. IEEE Conf. Computational Intelligence and Games (CIG)*, 2017. 6, 23, 38, 54
- [42] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable AI for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, Aug. 2018. ISSN: 2325-4270. 6, 19, 72
- [43] F. Doshi-Velez and B. Kim, “Considerations for Evaluation and Generalization in Interpretable Machine Learning,” in *Explainable and Interpretable Models in Computer Vision and Machine Learning* (H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven, eds.), pp. 3–17, Cham: Springer International Publishing, 2018. 6, 19, 72
- [44] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, vol. 6, pp. 52138–52160, 2018. Publisher: IEEE. 6, 72
- [45] A. Liapis, G. N. Yannakakis, and J. Togelius, “Designer modeling for Sentient Sketchbook,” in *Proc. IEEE Conf. Computational Intelligence and Games*, pp. 1–8, Aug. 2014. ISSN: 2325-4289. 6, 8, 25, 68

- [46] Nintendo R&D4, “*The Legend of Zelda.*” Game [NES], 1986. Nintendo, Kyoto, Japan. Last played December 2017. 7, 37
- [47] E. McMillen and F. Himsl, “*The Binding of Isaac.*” Game [PC], 2011. McMillen, Edmund and Himsl, Florian, California, US. Last played March 2020. 7, 14, 37
- [48] T. Lubart, “How can computers be partners in the creative process: Classification and commentary on the Special Issue,” *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 365 – 369, 2005. 7, 19, 40, 68
- [49] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality Diversity: A New Frontier for Evolutionary Computation,” *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016. 8, 16, 33, 70
- [50] G. N. Yannakakis and J. Togelius, “Experience-Driven Procedural Content Generation,” *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011. 8, 24, 56, 70
- [51] C. Pedersen, J. Togelius, and G. N. Yannakakis, “Modeling Player Experience for Content Creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 54–67, 2010. 8
- [52] C. Holmgård, M. C. Green, A. Liapis, and J. Togelius, “Automated Playtesting With Procedural Personas Through MCTS With Evolved Heuristics,” *IEEE Transactions on Games*, vol. 11, pp. 352–362, Dec. 2019. 8, 24, 25, 26
- [53] A. Drachen, A. Canossa, and G. N. Yannakakis, “Player modeling using self-organization in Tomb Raider: Underworld,” in *2009 IEEE Symposium on Computational Intelligence and Games*, pp. 1–8, Sept. 2009. ISSN: 2325-4289. 8, 24, 25
- [54] D. Melhart, A. Azadvar, A. Canossa, A. Liapis, and G. N. Yannakakis, “Your Gameplay Says It All: Modelling Motivation in Tom Clancy’s The Division,” in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, Aug. 2019. ISSN: 2325-4270. 8, 24, 25
- [55] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the Galactic Arms Race video game,” in *2009 IEEE Symposium on Computational Intelligence and Games*, Sept. 2009. 8, 15

- [56] J. R. H. Mariño, W. M. P. Reis, and L. H. S. Lelis, “An Empirical Evaluation of Evaluation Metrics of Procedurally Generated Mario Levels,” in *AIIDE*, 2015. 8, 16
- [57] A. Canossa and G. Smith, “Towards a procedural evaluation technique: Metrics for level design,” in *The 10th International Conference on the Foundations of Digital Games*, p. 8, sn, 2015. 8, 16
- [58] A. Summerville, J. R. H. Mariño, S. Snodgrass, S. Ontañón, and L. H. S. Lelis, “Understanding Mario: An Evaluation of Design Metrics for Platformers,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG ’17*, (New York, NY, USA), Association for Computing Machinery, 2017. event-place: Hyannis, Massachusetts. 8
- [59] A. Liapis, G. N. Yannakakis, and J. Togelius, “Adapting Models of Visual Aesthetics for Personalized Content Creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012. 8, 26, 68
- [60] E. Halina and M. Guzdial, “Threshold Designer Adaptation: Improved Adaptation for Designers in Co-creative Systems,” in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22), Special Track on AI, the Arts and Creativity*, 2022. 8, 26, 68
- [61] A. Liapis, G. N. Yannakaki, M. J. Nelson, M. Preuss, and R. Bidarra, “Orchestrating Game Generation,” *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019. 10, 14, 27
- [62] C. Salge, M. C. Green, R. Canaan, and J. Togelius, “Generative Design in Minecraft (GDMC): Settlement Generation Competition,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, (New York, NY, USA), Association for Computing Machinery, 2018. event-place: Malmö, Sweden. 10, 14
- [63] E. Aarseth, “From Hunt the Wumpus to EverQuest: Introduction to Quest Theory,” in *Entertainment Computing - ICEC 2005* (F. Kishino, Y. Kitamura, H. Kato, and N. Nagata, eds.), vol. 3711, (Berlin, Heidelberg), pp. 496–506, Springer Berlin Heidelberg, 2005. Series Title: Lecture Notes in Computer Science. 10, 60

- [64] C. Ashmore and M. Nitsche, “The Quest in a Generated World,” in *Proceedings of the 2007 DiGRA International Conference: Situated Play*, The University of Tokyo, Sept. 2007. 10, 14, 60
- [65] Q. Kybartas and R. Bidarra, “A Survey on Story Generation Techniques for Authoring Computational Narratives,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, pp. 239–253, Sept. 2017. 10, 60
- [66] N. Dehn, “Story Generation after TALE-SPIN,” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI’81*, (San Francisco, CA, USA), pp. 16–18, Morgan Kaufmann Publishers Inc., 1981. event-place: Vancouver, BC, Canada. 10, 73
- [67] M. Lebowitz, “Creating a Story-Telling Universe,” in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 1, IJCAI’83*, (San Francisco, CA, USA), pp. 63–65, Morgan Kaufmann Publishers Inc., 1983. event-place: Karlsruhe, West Germany. 10, 73
- [68] M. Hendriks, S. Meijer, J. Van Der Velden, and A. Iosup, “Procedural Content Generation for Games: A Survey,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, Feb. 2013. Place: New York, NY, USA Publisher: Association for Computing Machinery. 14
- [69] Firaxis Games and S. Meier, “*Civilization IV*.” Game [PC], October 2005. 2K Games, California, US. Last played November 2009. 14
- [70] T. Adams and Z. Adams, “*Dwarf Fortress*.” Game [PC], August 2006. Last played December 2013. 14
- [71] A. Liapis, “10 Years of the PCG workshop: Past and Future Trends,” in *Proceedings of the 11th Workshop on Procedural Content Generation, FDG*, 2020. 14, 69
- [72] C. Browne and F. Maire, “Evolutionary Game Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 1–16, 2010. 14
- [73] J. Font, T. Mahlmann, D. Manrique, and J. Togelius, “Towards the Automatic Generation of Card Games through Grammar-Guided Genetic Programming,” in *Proceedings of the 8th International*

Conference on the Foundations of Digital Games, FDG '13, pp. 360–363, Jan. 2013. 14, 15

- [74] P. Ammanabrolu, W. Broniec, A. Mueller, J. Paul, and M. Riedl, “Toward Automated Quest Generation in Text-Adventure Games,” in *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*, (Tokyo, Japan), pp. 1–12, Association for Computational Linguistics, 2019. 14
- [75] N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, and M. O’Neill, “Evolving levels for Super Mario Bros using grammatical evolution,” in *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2012*, (Granada), pp. 304–311, IEEE, 2012. 14, 31
- [76] A. Sarkar and S. Cooper, “Sequential Segment-based Level Generation and Blending using Variational Autoencoders,” in *Proceedings of the 11th Workshop on Procedural Content Generation in Games*, 2020. 14, 69
- [77] M. C. Green, L. Mugrai, A. Khalifa, and J. Togelius, “Mario Level Generation From Mechanics Using Scene Stitching,” 2020. _eprint: 2002.02992. 14
- [78] L. Horsley and D. Perez-Liebana, “Building an automatic sprite generator with deep convolutional generative adversarial networks,” in *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 134–141, 2017. 14
- [79] J. Pagnutti, K. Compton, and J. Whitehead, “Do You Like This Art I Made You: Introducing Techne, A Creative Artbot Commune,” in *Proceedings of the 7th Workshop on Procedural Content Generation, FDG*, 2016. 14
- [80] M. Scirea, J. Togelius, P. Eklund, and S. Risi, “MetaCompose: A Compositional Evolutionary Music Composer,” in *Evolutionary and Biologically Inspired Music, Sound, Art and Design* (C. Johnson, V. Ciesielski, J. Correia, and P. Machado, eds.), (Cham), pp. 202–217, Springer International Publishing, 2016. 14
- [81] A. K. Hoover, *Functional Scaffolding for Musical Composition: A New Approach in Computer-Assisted Music Composition*. PhD The-

- sis, University of Central Florida, 2014. Published: PhD Dissertation, University of Central Florida. 14
- [82] M. Guzdial and M. Riedl, “Conceptual Game Expansion,” 2020. [_eprint: 2002.09636](#). 14, 27, 69
- [83] M. Cook, S. Colton, and J. Gow, “The ANGELINA Videogame Design System, Part I,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, pp. 192–203, June 2017. 14
- [84] M. Cook and S. Colton, “A Rogue Dream: Automatically Generating Meaningful Content For Games,” in *Joint Proceedings of the AIIDE 2014 Workshops co-located with 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2014)*, 2014. 14
- [85] M. Treanor, B. Blackford, M. Mateas, and I. Bogost, “Game-O-Matic: Generating Videogames that Represent Ideas,” in *Proceedings of the Procedural Content Generation Workshop at the Foundations of Digital Games Conference, 2012*. 14
- [86] J. Dormans and S. Bakkes, “Generating Missions and Spaces for Adaptable Play Experiences,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 216–228, Sept. 2011. 14, 15
- [87] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, “Toward supporting stories with procedurally generated game worlds,” in *2011 IEEE Conference on Computational Intelligence and Games (CIG’11)*, (Seoul, Korea (South)), pp. 297–304, IEEE, Aug. 2011. 14
- [88] A. M. Abuzurairq, A. Ferguson, and P. Pasquier, “Taksim: A Constrained Graph Partitioning Framework for Procedural Content Generation,” in *2019 IEEE Conference on Games (CoG)*, (London, United Kingdom), pp. 1–8, IEEE, Aug. 2019. 14
- [89] A. Hoover, W. Cachia, A. Liapis, and G. Yannakakis, “AudioInSpace: Exploring the Creative Fusion of Generative Audio, Visuals and Gameplay,” in *Proceedings of the EvoMusArt conference*, vol. 9027, pp. 101–112, Springer, Apr. 2015. 14
- [90] N. I. Holtar, M. J. Nelson, and J. Togelius, “Audioverdrive: Exploring bidirectional communication between music and gameplay,”

- in *Proceedings of the International Computer Music Conference*, pp. 124–131, 2013. 14
- [91] D. Karavolos, A. Liapis, and G. N. Yannakakis, “A multi-faceted surrogate model for search-based procedural content generation,” *IEEE Transactions on Games*, 2019. Publisher: IEEE. 14
- [92] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-Based Procedural Content Generation: A Taxonomy and Survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 172–186, Sept. 2011. 14
- [93] N. Shaker, A. Liapis, J. Togelius, R. Lopes, and R. Bidarra, “Constructive generation methods for dungeons and levels,” in *Procedural Content Generation in Games*, pp. 31–55, Cham: Springer International Publishing, 2016. 15
- [94] M. C. Green, A. Khalifa, A. Alsoughayer, D. Surana, A. Liapis, and J. Togelius, “Two-Step Constructive Approaches for Dungeon Generation,” in *Proceedings of the 14th International Conference on the Foundations of Digital Games, FDG '19*, (New York, NY, USA), Association for Computing Machinery, 2019. event-place: San Luis Obispo, California, USA. 15
- [95] S. Snodgrass, “Levels from Sketches with Example-Driven Binary Space Partition,” in *Proceedings of the Fifteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2019, October 8-12, 2019, Atlanta, Georgia, USA* (G. Smith and L. Lelis, eds.), pp. 73–79, AAAI Press, 2019. 15, 69
- [96] A. Summerville, C. Martens, B. Samuel, J. Osborn, N. Wardrip-Fruin, and M. Mateas, “Gemini: Bidirectional Generation and Analysis of Games via ASP,” in *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2018. 15, 20
- [97] V. Volz, J. Schrum, J. Liu, S. M. Lucas, A. Smith, and S. Risi, “Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network,” in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '18*, (New York, NY, USA), pp. 221–228, Association for Computing Machinery, 2018. event-place: Kyoto, Japan. 15

- [98] J. M. Font, R. Izquierdo, D. Manrique, and J. Togelius, “Constrained level generation through grammar-based evolutionary algorithms,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9597, pp. 558–573, 2016. ISBN: 9783319312033. 15
- [99] I. Karth and A. M. Smith, “WaveFunctionCollapse is Constraint Solving in the Wild,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG ’17*, (New York, NY, USA), Association for Computing Machinery, 2017. event-place: Hyannis, Massachusetts. 15, 23
- [100] I. Karth and A. M. Smith, “Addressing the Fundamental Tension of PCGML with Discriminative Learning,” in *Proceedings of the 14th International Conference on the Foundations of Digital Games, FDG ’19*, (New York, NY, USA), Association for Computing Machinery, 2019. event-place: San Luis Obispo, California, USA. 15, 23
- [101] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, “Procedural Content Generation via Machine Learning (PCGML),” *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018. Publisher: IEEE. 15, 24, 35, 36, 69
- [102] H. Zhang, M. C. Fontaine, A. K. Hoover, J. Togelius, B. Dilkina, and S. Nikolaidis, “Video Game Level Repair via Mixed Integer Linear Programming,” in *Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE.*, 2020. 15
- [103] S. Dahlskog and J. Togelius, “A multi-level level generator,” in *2014 IEEE Conference on Computational Intelligence and Games*, Aug. 2014. 15
- [104] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, “Procedural Content Generation through Quality Diversity,” in *2019 IEEE Conference on Games (CoG)*, 2019. 15, 17
- [105] D. Gravina, A. Liapis, and G. Yannakakis, “Surprise Search: Beyond Objectives and Novelty,” in *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO ’16*, (New

York, NY, USA), pp. 677–684, ACM, 2016. event-place: Denver, Colorado, USA. 16

- [106] S. Dahlskog, J. Togelius, and S. Björk, “Patterns, Dungeons and Generators,” in *Proceedings of the 10th International Conference on the Foundations of Digital Games, FDG 2015, Pacific Grove, CA, USA, June 22-25, 2015* (J. P. Zagal, E. MacCallum-Stewart, and J. Togelius, eds.), Society for the Advancement of the Science of Digital Games, 2015. 16, 39
- [107] A. Gaier, A. Asteroth, and J.-B. Mouret, “Are Quality Diversity Algorithms Better at Generating Stepping Stones than Objective-Based Search?,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO ’19*, (New York, NY, USA), pp. 115–116, Association for Computing Machinery, 2019. event-place: Prague, Czech Republic. 16
- [108] J. Lehman and K. O. Stanley, “Abandoning Objectives: Evolution Through the Search for Novelty Alone,” *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011. 16
- [109] J. Lehman and K. O. Stanley, “Revising the Evolutionary Computation Abstraction: Minimal Criteria Novelty Search,” in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO ’10*, (New York, NY, USA), pp. 103–110, Association for Computing Machinery, 2010. event-place: Portland, Oregon, USA. 16
- [110] R. Wang, J. Lehman, J. Clune, and K. O. Stanley, “POET: Open-Ended Coevolution of Environments and Their Optimized Solutions,” in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’19*, (New York, NY, USA), pp. 142–151, Association for Computing Machinery, 2019. event-place: Prague, Czech Republic. 16
- [111] R. Wang, J. Lehman, A. Rawal, J. Zhi, Y. Li, J. Clune, and K. Stanley, “Enhanced POET: Open-ended Reinforcement Learning through Unbounded Invention of Learning Challenges and their Solutions,” in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 9940–9951, PMLR, July 2020. 16

- [112] A. Dharna, J. Togelius, and L. B. Soros, “Co-Generation of Game Levels and Game-Playing Agents,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, pp. 203–209, Oct. 2020. 16
- [113] A. Liapis, G. N. Yannakakis, and J. Togelius, “Constrained Novelty Search: A Study on Game Content Generation,” *Evol. Comput.*, vol. 23, pp. 101–129, Mar. 2015. Place: Cambridge, MA, USA Publisher: MIT Press. 16, 21
- [114] J. Lehman and K. O. Stanley, “Evolving a Diversity of Virtual Creatures through Novelty Search and Local Competition,” in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO ’11, (New York, NY, USA), pp. 211–218, Association for Computing Machinery, 2011. event-place: Dublin, Ireland. 16, 17, 32
- [115] M. Charity, M. C. Green, A. Khalifa, and J. Togelius, “Mech-Elites: Illuminating the Mechanic Space of GVG-AI,” in *International Conference on the Foundations of Digital Games*, FDG ’20, (New York, NY, USA), Association for Computing Machinery, 2020. event-place: Bugibba, Malta. 17, 18
- [116] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering Quality Diversity in Dungeon Design with Interactive Constrained MAP-Elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 17, 18, 23, 38
- [117] M. C. Fontaine, S. Lee, L. B. Soros, F. De Mesentier Silva, J. Togelius, and A. K. Hoover, “Mapping hearthstone deck spaces through MAP-elites with sliding boundaries,” *Proceedings of the Genetic and Evolutionary Computation Conference*, July 2019. ISBN: 9781450361118 Publisher: ACM. 17
- [118] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, “Robots that can adapt like animals,” *Nature*, vol. 521, pp. 503–507, May 2015. 17, 33
- [119] B. Tjanaka, M. C. Fontaine, J. Togelius, and S. Nikolaidis, “Approximating Gradients for Differentiable Quality Diversity in Reinforcement Learning,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’22, (New York, NY, USA),

- pp. 1102–1111, Association for Computing Machinery, 2022. event-place: Boston, Massachusetts. 17
- [120] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015. 17, 33, 34
- [121] A. Cully, “Autonomous Skill Discovery with Quality-Diversity and Unsupervised Descriptors,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’19, (New York, NY, USA), pp. 81–89, Association for Computing Machinery, 2019. event-place: Prague, Czech Republic. 17
- [122] N. Justesen, M. González-Duque, D. Cabarcas, J.-B. Mouret, and S. Risi, “Learning a Behavioral Repertoire from Demonstrations,” in *2020 IEEE Conference on Games (CoG)*, pp. 383–390, 2020. 17
- [123] L. Grillotti and A. Cully, “Discovering Unsupervised Behaviours from Full State Trajectories,” in *ICLR Workshop on Agent Learning in Open-Endedness*, 2022. 17
- [124] V. Vassiliades, K. Chatzilygeroudis, and J.-B. Mouret, “Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 623–630, 2018. Publisher: IEEE. 17
- [125] V. Vassiliades, K. Chatzilygeroudis, and J.-B. Mouret, “A Comparison of Illumination Algorithms in Unbounded Spaces,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’17, (New York, NY, USA), pp. 1578–1581, ACM, 2017. event-place: Berlin, Germany. 17
- [126] M. C. Fontaine, J. Togelius, S. Nikolaidis, and A. K. Hoover, “Covariance Matrix Adaptation for the Rapid Illumination of Behavior Space,” 2019. _eprint: 1912.02400. 17, 30
- [127] A. Cully, “Multi-Emitter MAP-Elites: Improving Quality, Diversity and Data Efficiency with Heterogeneous Sets of Emitters,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’21, (New York, NY, USA), pp. 84–92, Association for Computing Machinery, 2021. event-place: Lille, France. 17

- [128] N. Justesen, S. Risi, and J.-B. Mouret, “MAP-Elites for Noisy Domains by Adaptive Sampling,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’19, (New York, NY, USA), pp. 121–122, Association for Computing Machinery, 2019. event-place: Prague, Czech Republic. 17
- [129] T. Pierrot, G. Richard, K. Beguir, and A. Cully, “Multi-Objective Quality Diversity Optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’22, (New York, NY, USA), pp. 139–147, Association for Computing Machinery, 2022. event-place: Boston, Massachusetts. 17
- [130] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, “Assessing the Effects of Interacting with MAP-Elites,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, pp. 124–131, AAAI, Oct. 2021. Section: Poster Papers. 17
- [131] D. Perez-Liebana, C. Guerrero-Romero, A. Dockhorn, L. Xu, J. Hurtado, and D. Jeurissen, “Generating Diverse and Competitive Play-Styles for Strategy Games,” in *2021 IEEE Conference on Games (CoG)*, pp. 1–8, IEEE Press, 2021. event-place: Copenhagen, Denmark. 17
- [132] Y. Zhang, M. C. Fontaine, A. K. Hoover, and S. Nikolaidis, “Deep Surrogate Assisted MAP-Elites for Automated Hearthstone Deck-building,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’22, (New York, NY, USA), pp. 158–167, Association for Computing Machinery, 2022. event-place: Boston, Massachusetts. 17
- [133] M. Gonzalez-Duque, R. Berg Palm, D. Ha, and S. Risi, “Finding Game Levels with the Right Difficulty in a Few Trials through Intelligent Trial-and-Error,” in *Proceedings of the IEEE Conference on Games (CoG)*, 2020. 17
- [134] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, ACM, 2018. 17, 34, 54

- [135] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a Feasible–Infeasible Two-Population (FI-2Pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008. Publisher: Elsevier. 17
- [136] A. Khalifa, M. C. Green, G. Barros, and J. Togelius, “Intentional Computational Level Design,” in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO ’19*, (New York, NY, USA), pp. 796–803, ACM, 2019. event-place: Prague, Czech Republic. 18
- [137] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, “Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions,” *IEEE Transactions on Games*, pp. 1–1, 2020. 18, 23, 38
- [138] J. Togelius and J. Schmidhuber, “An experiment in automatic game design,” in *2008 IEEE Symposium On Computational Intelligence and Games*, pp. 111–118, 2008. 18
- [139] G. A. B. Barros, M. C. Green, A. Liapis, and J. Togelius, “Who Killed Albert Einstein? From Open Data to Murder Mystery Games,” *IEEE Transactions on Games*, vol. 11, no. 1, pp. 79–89, 2019. 18
- [140] M. Cook, S. Colton, and J. Gow, “The ANGELINA Videogame Design System, Part II,” *IEEE Transactions on Computational Intelligence and AI in Games*, Nov. 2016. 18
- [141] G. M. Smith, *Expressive Design Tools: Procedural Content Generation for Game Designers*. PhD Thesis, University of California Santa Cruz, 2012. Published: PhD Dissertation, University of California Santa Cruz. 18
- [142] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, “A Comparative Evaluation of Procedural Level Generators in the Mario AI Framework,” in *Proceedings of the 9th International Conference on the Foundations of Digital Games, FDG ’14*, 2014. 18, 49
- [143] M. Kreminski, I. Karth, M. Mateas, and N. Wardrip-Fruin, “Evaluating Mixed-Initiative Creative Interfaces via Expressive Range Coverage Analysis,” in *Joint Proceedings of the IUI 2022 Workshops*:

- APEX-UI, HAI-GEN, HEALTHI, HUMANIZE, TExSS, SOCIALIZE* co-located with the ACM International Conference on Intelligent User Interfaces (IUI 2022), Virtual Event, Helsinki, Finland, March 21-22, 2022 (A. Smith-Renner and O. Amir, eds.), vol. 3124 of *CEUR Workshop Proceedings*, pp. 34–45, CEUR-WS.org, 2022. 18
- [144] S. Earle, M. Edwards, A. Khalifa, P. Bontrager, and J. Togelius, “Learning Controllable Content Generators,” in *2021 IEEE Conference on Games (CoG)*, pp. 1–9, 2021. 18
- [145] A. Madkour, S. Marsella, and C. Hartevelde, “Towards Non-Technical Designer Control Over PCG Systems: Investigating An Example-Based Mechanism For Controlling Graph Grammars,” in *Proceedings of the 17th International Conference on the Foundations of Digital Games*, 2022. 18
- [146] L. Holmberg, S. Generalao, and A. Hermansson, “The Role of Explanations in Human-Machine Learning,” in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1006–1013, 2021. 19
- [147] M. Guzdial, N. Liao, J. Chen, S.-Y. Chen, S. Shah, V. Shah, J. Reno, G. Smith, and M. O. Riedl, “Friend, Collaborator, Student, Manager: How Design of an AI-Driven Game Level Editor Affects Creators,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, (New York, NY, USA), Association for Computing Machinery, 2019. event-place: Glasgow, Scotland Uk. 19, 68
- [148] A. Liapis, *Searching for Sentient Design tools for Game Development*. PhD Thesis, IT University of Copenhagen, 2014. Published: PhD Dissertation, IT University of Copenhagen. 19
- [149] G. Ferguson and J. Allen, “Mixed-Initiative Systems for Collaborative Problem Solving,” *AI Magazine*, vol. 28, p. 23, June 2007. 20
- [150] M. Kreminski, M. Dickinson, J. C. Osborn, A. Summerville, M. Mateas, and N. Wardrip-Fruin, “Germinate: A Mixed-Initiative Casual Creator for Rhetorical Games,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020. 20

- [151] K. Compton and M. Mateas, “Casual Creators,” in *Proceedings of the Sixth International Conference on Computational Creativity, Park City, Utah, USA, June 29 - July 2, 2015* (H. Toivonen, S. Colton, M. Cook, and D. Ventura, eds.), pp. 228–235, computationalcreativity.net, 2015. 20, 68, 69
- [152] O. Delarosa, H. Dong, M. Ruan, A. Khalifa, and J. Togelius, “Mixed-Initiative Level Design with RL Brush,” 2020. _eprint: 2008.02778. 20
- [153] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, “PCGRL: Procedural Content Generation via Reinforcement Learning,” in *Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE’20*, pp. 95–101, AAAI Press, 2020. 20
- [154] A. Liapis, G. N. Yannakakis, and J. Togelius, “Sentient sketchbook: Computer-aided game level authoring,” in *Proceedings of ACM Conference on Foundations of Digital Games, 2013. In Print*, 2013. 21
- [155] M. Preuss, A. Liapis, and J. Togelius, “Searching for Good and Diverse Game Levels,” in *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, 2014. 21
- [156] D. Perez-Liebana, S. M. Lucas, R. D. Gaina, J. Togelius, A. Khalifa, and J. Liu, *General Video Game Artificial Intelligence*. Morgan & Claypool Publishers, 2019. Publication Title: Synthesis Lectures on Games and Computational Intelligence. 21
- [157] T. Schaul, “A video game description language for model-based or interactive learning,” in *2013 IEEE Conference on Computational Intelligence in Games (CIG)*, pp. 1–8, 2013. 21
- [158] T. Machado, A. Nealen, and J. Togelius, “Cicero: Computationally intelligent collaborative environment for game and level design,” in *3rd workshop on Computational Creativity and Games (CCGW) at the 8th International Conference on Computational Creativity (ICCC’17)*, 2017. 21
- [159] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, “Pitako-recommending game design elements in cicero,” in *2019 IEEE Conference on Games (CoG)*, IEEE, 2019. 21

- [160] T. Machado, D. Gopstein, A. Wang, O. Nov, A. Nealen, and J. Togelius, “Evaluation of a Recommender System for Assisting Novice Game Designers,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 15, pp. 167–173, 2019. 22
- [161] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 22
- [162] S. Karakovskiy and J. Togelius, “The Mario AI Benchmark and Competitions,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 55–67, 2012. 22
- [163] Nintendo R&D1 and Intelligent Systems, “*Super Mario Bros.*” Game [SNES], September 1985. Nintendo, Kyoto, Japan. Last played August 2020. 22
- [164] M. Guzdial, N. Liao, and M. Riedl, “Co-Creative Level Design via Machine Learning,” in *Joint Proceedings of the AIIDE 2018 Workshops co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2018), Edmonton, Canada, November 13-14, 2018.*, 2018. 22, 24
- [165] P. Lucas and C. Martinho, “Stay Awhile and Listen to 3Buddy, a Co-creative Level Design Support Tool,” in *Proceedings of the Eighth International Conference on Computational Creativity, Atlanta, Georgia, USA, June 19-23, 2017.* (A. K. Goel, A. Jordanous, and A. Pease, eds.), pp. 205–212, 2017. 22
- [166] Almost Human, “*Civilization IV.*” Game [PC], October 2014. Almost Human, Espoo, Finland. 22
- [167] A. Alvarez, E. Grevillius, E. Olsson, and J. Font, “Questgram [Qg]: Toward a Mixed-Initiative Quest Generation Tool,” in *Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG’21, (New York, NY, USA), Association for Computing Machinery, 2021.* 23, 38
- [168] A. Alvarez, J. Font, and J. Togelius, “Story Designer: Towards a Mixed-Initiative Tool to Create Narrative Structures,” in *Proceedings*

- [169] S. Dahlskog and J. Togelius, “Procedural Content Generation Using Patterns as Objectives,” in *Applications of Evolutionary Computation* (A. I. Esparcia-Alcázar and A. M. Mora, eds.), (Berlin, Heidelberg), pp. 325–336, Springer Berlin Heidelberg, 2014. 23, 39
- [170] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. Andre, “Player Modeling,” 2013. Publication Title: Dagstuhl Follow-Ups Volume: 6. 24
- [171] R. Thawonmas, J. Togelius, and G. N. Yannakakis, “Artificial General Intelligence in Games: Where Play Meets Design and User Experience,” *NII Shonan Meeting No. 130*, 2019. 24
- [172] S. F. Gudmundsson, P. Eisen, E. Poromaa, A. Nodet, S. Purmonen, B. Kozakowski, R. Meurling, and L. Cao, “Human-Like Playtesting with Deep Learning,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, 2018. 24
- [173] R. Bartle, “Hearts, clubs, diamonds, spades: Players who suit MUDs,” *Journal of MUD Research*, 1996. 24
- [174] A. Saas, A. Guitart, and Periañez, “Discovering playing patterns: Time series clustering of free-to-play game data,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, 2016. 24
- [175] A. F. d. Río, A. Guitart, and Periañez, “A Time Series Approach To Player Churn and Conversion in Videogames,” 2020. _eprint: 2003.10287. 24
- [176] A. Guitart, P. P. Chen, P. Bertens, and Periañez, “Forecasting Player Behavioral Data and Simulating In-Game Events,” in *Advances in Information and Communication Networks* (K. Arai, S. Kapoor, and R. Bhatia, eds.), (Cham), pp. 274–293, Springer International Publishing, 2019. 24
- [177] A. Sarkar and S. Cooper, “Towards Game Design via Creative Machine Learning (GDCML),” in *Proceedings of the IEEE Conference on Games (CoG)*, 2020. 24

- [178] M. Siper, A. Khalifa, and J. Togelius, “Path of Destruction: Learning an Iterative Level Generator Using a Small Dataset,” 2022. 24
- [179] M. Gonzalez-Duque, R. B. Palm, and S. Risi, “Fast Game Content Adaptation Through Bayesian-based Player Modelling,” in *2021 IEEE Conference on Games (CoG)*, pp. 01–08, 2021. 24
- [180] A. Summerville, M. Guzdial, M. Mateas, and M. Riedl, “Learning Player Tailored Content From Observation: Platformer Level Generation from Video Traces using LSTMs,” *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 12, no. 2, pp. 107–113, 2016. 25
- [181] P. Migkotzidis and A. Liapis, “SuSketch: Surrogate Models of Gameplay as a Design Assistant,” *IEEE Transactions on Games*, pp. 1–1, 2021. 25, 26, 70
- [182] D. Melhart, G. N. Yannakakis, and A. Liapis, “I Feel I Feel You: A Theory of Mind Experiment in Games,” *KI - Künstliche Intelligenz*, vol. 34, pp. 45–55, Mar. 2020. 25
- [183] K. Compton, *Casual Creators: Defining a Genre of Autotelic Creativity Support Systems*. PhD Thesis, University of California Santa Cruz, 2019. Published: PhD Dissertation, University of California Santa Cruz. 25
- [184] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, and others, “Guidelines for human-ai interaction,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019. 25
- [185] A. Alvarez and J. Font, “Learning the Designer’s Preferences to Drive Evolution,” in *Applications of Evolutionary Computation* (P. A. Castillo, J. L. Jiménez Laredo, and F. Fernández de Vega, eds.), vol. 12104 of *Lecture Notes in Computer Science*, (Cham), pp. 431–445, Springer International Publishing, 2020. 26, 68
- [186] A. Alvarez, J. Font, and J. Togelius, “Designer Modeling through Design Style Clustering,” *IEEE Transactions on Games*, pp. 1–1, 2022. 26, 38, 68

- [187] R. Sternberg and T. I. Lubart, “The concept of creativity: prospects and paradigms,” in *Handbook of Creativity*, Cambridge University Press, 1999. Section: 1. 26
- [188] R. Sternberg and C. U. Press, *Handbook of Creativity*. Cambridge University Press, 1999. 26
- [189] R. J. Sternberg, “Creativity or creativities?,” *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 370 – 382, 2005. 26
- [190] M. Csikszentmihalyi, *Creativity: Flow and the psychology of discovery and invention*. Creativity: Flow and the psychology of discovery and invention., New York, NY, US: HarperCollins Publishers, 1997. 26
- [191] S. Colton and G. A. Wiggins, “Computational Creativity: The Final Frontier?,” in *Proceedings of the 20th European Conference on Artificial Intelligence, ECAI’12*, (NLD), pp. 21–26, IOS Press, 2012. event-place: Montpellier, France. 26
- [192] M. Guzdial and M. O. Riedl, “Combinets: Creativity via Recombination of Neural Networks,” in *Proceedings of the Ninth International Conference on Computational Creativity*, 2018. 27
- [193] A. Sarkar, Z. Yang, and S. Cooper, “Controllable Level Blending between Games using Variational Autoencoders,” in *Proceedings of the EXAG Workshop at AIIDE*, 2019. 27
- [194] A. Sarkar and S. Cooper, “Generating and Blending Game Levels via Quality-Diversity in the Latent Space of a Variational Autoencoder,” in *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, FDG’21, (New York, NY, USA), Association for Computing Machinery, 2021. event-place: Montreal, QC, Canada. 27, 69
- [195] R. Miikkulainen, “Creative AI Through Evolutionary Computation: Principles and Examples,” *arXiv:2008.04212 [cs]*, Feb. 2021. arXiv: 2008.04212. 27
- [196] J. Lehman, J. Clune, D. Misevic, C. Adami, L. Altenberg, J. Beaulieu, P. J. Bentley, S. Bernard, G. Beslon, D. M. Bryson, N. Cheney, P. Chrabaszcz, A. Cully, S. Doncieux, F. C. Dyer, K. O. Ellefsen, R. Feldt, S. Fischer, S. Forrest, A. Ffenooy, C. Gagñe, L. Le Goff, L. M.

- Grabowski, B. Hodjat, F. Hutter, L. Keller, C. Knibbe, P. Krcah, R. E. Lenski, H. Lipson, R. MacCurdy, C. Maestre, R. Miikkulainen, S. Mitri, D. E. Moriarty, J.-B. Mouret, A. Nguyen, C. Ofria, M. Parizeau, D. Parsons, R. T. Pennock, W. F. Punch, T. S. Ray, M. Schoenauer, E. Schulte, K. Sims, K. O. Stanley, F. Taddei, D. Tarapore, S. Thibault, R. Watson, W. Weimer, and J. Yosinski, “The Surprising Creativity of Digital Evolution: A Collection of Anecdotes from the Evolutionary Computation and Artificial Life Research Communities,” *Artificial Life*, vol. 26, pp. 274–306, May 2020. 27
- [197] A. Sarkar, “Game Design using Creative AI,” in *NeurIPS 2019 Workshop on Machine Learning for Creativity and Design*, 2019. 28, 69
- [198] C. Darwin, *On the Origin of Species by Means of Natural Selection*. London: Murray, 1859. 29
- [199] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Natural Computing Series, Springer Berlin Heidelberg, 2007. 29, 30
- [200] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Berlin, Heidelberg: Springer-Verlag, 1996. 31
- [201] D. Ashlock, S. Risi, and J. Togelius, “Representations for Search-based Methods,” in *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, Springer, 2016. Section: 9. 31
- [202] J. Clune, K. O. Stanley, R. T. Pennock, and C. Ofria, “On the Performance of Indirect Encoding Across the Continuum of Regularity,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 3, pp. 346–367, 2011. 31
- [203] K. O. Stanley, “Compositional Pattern Producing Networks: A Novel Abstraction of Development,” *Genetic Programming and Evolvable Machines*, vol. 8, pp. 131–162, June 2007. Place: USA Publisher: Kluwer Academic Publishers. 31
- [204] K. O. Stanley and J. Lehman, *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer Publishing Company, Incorporated, 2015. 31

- [205] K. O. Stanley and R. Miikkulainen, “Evolving Neural Networks through Augmenting Topologies,” *Evol. Comput.*, vol. 10, pp. 99–127, June 2002. Place: Cambridge, MA, USA Publisher: MIT Press. 32
- [206] D. Gravina, A. Liapis, and G. N. Yannakakis, “Blending Notions of Diversity for MAP-elites,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’19, (New York, NY, USA), pp. 117–118, ACM, 2019. event-place: Prague, Czech Republic. 33
- [207] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. 35, 69
- [208] T. M. Mitchell, *Machine Learning*. USA: McGraw-Hill, Inc., 1 ed., 1997. 35
- [209] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015. 35
- [210] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, “Deep Learning for Video Game Playing,” *IEEE Transactions on Games*, vol. 12, no. 1, pp. 1–20, 2020. 35
- [211] T. Clanuwat, A. Lamb, and A. Kitamoto, “KuroNet: Pre-Modern Japanese Kuzushiji Character Recognition with Deep Learning,” in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 607–614, Sept. 2019. 35
- [212] R. R. Torrado, A. Khalifa, M. C. Green, N. Justesen, S. Risi, and J. Togelius, “Bootstrapping Conditional GANs for Video Game Level Generation,” in *Proceedings of the IEEE Conference on Games (CoG)*, 2020. 35, 69
- [213] M. Hartmann, Y. Kementchedjheva, and A. Søgaard, “Comparing Unsupervised Word Translation Methods Step by Step,” in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 6033–6043, Curran Associates, Inc., 2019. 35

- [214] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, “Improved Techniques for Training GANs,” in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), pp. 2234–2242, Curran Associates, Inc., 2016. 35
- [215] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL ’02, (USA), pp. 311–318, Association for Computational Linguistics, 2002. event-place: Philadelphia, Pennsylvania. 35
- [216] A. Juliani, A. Khalifa, V. P. Berges, J. Harper, E. Teng, H. Henry, A. Crespi, J. Togelius, and D. Lange, “Obstacle tower: A generalization challenge in vision, control, and planning,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019* (S. Kraus, ed.), IJCAI International Joint Conference on Artificial Intelligence, pp. 2684–2691, International Joint Conferences on Artificial Intelligence, 2019. 35
- [217] C. Doersch and A. Zisserman, “Multi-task Self-Supervised Visual Learning,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2070–2079, IEEE Computer Society, 2017. 36
- [218] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, “Deep learning for procedural content generation,” *Neural Computing and Applications*, vol. 33, pp. 19–37, Jan. 2021. 36, 69
- [219] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards Pattern-based Mixed-initiative Dungeon Generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG ’17*, (New York, NY, USA), ACM, 2017. 38
- [220] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing Aesthetic Criteria in the Evolutionary Dungeon Designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, 2018. 38
- [221] A. Flodhag, S. Tolinsson, A. Alvarez, and J. Font, “To make sense of dungeons: Developing procedural narrative for EDD with the help of macro patterns,” in *Extended Abstracts of the Annual Symposium*

- on Computer-Human Interaction in Play Companion Extended Abstracts*, CHI PLAY '20 Extended Abstracts, (New York, NY, USA), ACM, 2020. 38
- [222] J. Larsson and A. Hammer, *QuestStories - Quest and Narrative, The Role of the NPC*. B.S. Thesis, Malmö universitet, 2021. 38, 42, 61, 67
- [223] A. Alvarez and J. Font, “TropeTwist: Trope-based Narrative Structure Generation,” in *Proceedings of the 13th Workshop on Procedural Content Generation, FDG, 2022*. 38
- [224] A. Alvarez, *Exploring the Dynamic Properties of Interaction in Mixed-Initiative Procedural Content Generation*. PhD Thesis, Malmö universitet, 2020. ISBN: 978-91-7877-139-4 Series: Studies in Computer Science. 38
- [225] S. Bjork and J. Holopainen, *Patterns in Game Design (Game Development Series)*. USA: Charles River Media, Inc., 2004. 39
- [226] J. Doran and I. Parberry, “A Prototype Quest Generator Based on a Structural Analysis of Quests from Four MMORPGs,” in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, PCGames '11*, (New York, NY, USA), ACM, 2011. event-place: Bordeaux, France. 42, 55
- [227] R. Barthes, “Introduction à l’analyse structurale des récits,” *Communications*, pp. 1–27, 1966. 43
- [228] N. Szilas, “IDtension : a narrative engine for Interactive Drama,” in *1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, 2003*, p. 12, 2003. 43
- [229] C. Richmond and D. Schoentrup, “TV Tropes,” 2004. 43
- [230] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Q.*, vol. 28, p. 75–105, Mar. 2004. 47, 50
- [231] M. Cook, J. Gow, and S. Colton, “Danesh: Helping Bridge The Gap Between Procedural Generators And Their Output,” in *Proceedings of the 7th Workshop on Procedural Content Generation, FDG, 2016*. event-place: Dundee, United Kingdom. 49

- [232] E. McMillen and F. Himsl, “The Binding of Isaac,” 2011. Game [PC]. McMillen, Edmund and Himsl, Florian, California, US. Last played February 2022. 49
- [233] I. Rahwan, M. Cebrian, N. Obradovich, J. Bongard, J. F. Bonnefon, C. Breazeal, J. W. Crandall, N. A. Christakis, I. D. Couzin, M. O. Jackson, N. R. Jennings, E. Kamar, I. M. Kloumann, H. Larochelle, D. Lazer, R. McElreath, A. Mislove, D. C. Parkes, A. Pentland, M. E. Roberts, A. Shariff, J. B. Tenenbaum, and M. Wellman, “Machine behaviour,” *Nature*, vol. 568, no. 7753, pp. 477–486, 2019. Publisher: Springer US. 51
- [234] J. Zimmerman, J. Forlizzi, and S. Evenson, “Research through design as a method for interaction design research in hci,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, (New York, NY, USA), p. 493–502, Association for Computing Machinery, 2007. 51
- [235] J. Zimmerman, E. Stolterman, and J. Forlizzi, “An analysis and critique of research through design: Towards a formalization of a research approach,” in *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, DIS '10, (New York, NY, USA), p. 310–319, Association for Computing Machinery, 2010. 51
- [236] C. Spinuzzi, “The methodology of participatory design,” *Technical Communication*, vol. 52, pp. 163–174, May 2005. Article. 52
- [237] D. Bhaumik, A. Khalifa, and J. Togelius, “Lode Encoder: AI-constrained co-creativity,” in *2021 IEEE Conference on Games (CoG)*, pp. 01–08, IEEE, Aug. 2021. 55
- [238] O. A. Acar, M. Tarakci, and D. v. Knippenberg, “Creativity and Innovation Under Constraints: A Cross-Disciplinary Integrative Review,” *Journal of Management*, vol. 45, no. 1, pp. 96–121, 2019. _eprint: <https://doi.org/10.1177/0149206318805832>. 55
- [239] K. Makantasis, A. Liapis, and G. N. Yannakakis, “From Pixels to Affect: A Study on Games and Player Experience,” in *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 1–7, 2019. 56
- [240] J. J. Y. Chung, S. He, and E. Adar, “The Intersection of Users, Roles, Interactions, and Technologies in Creativity Support Tools,”

in *Designing Interactive Systems Conference 2021*, (Virtual Event USA), pp. 1817–1833, ACM, June 2021. 68

- [241] M. Kreminski and M. Mateas, “Reflective Creators,” in *Proceedings of the International Conference on Interactive Digital Storytelling*, p. 10, Springer International Publishing, 2021. 68
- [242] A. Canziani, A. Paszke, and E. Cukurciello, “An Analysis of Deep Neural Network Models for Practical Applications,” 2016. arXiv: 1605.07678. 69
- [243] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language Models are Few-Shot Learners,” in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 1877–1901, Curran Associates, Inc., 2020. 69
- [244] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical Text-Conditional Image Generation with CLIP Latents,” Apr. 2022. _eprint: 2204.06125. 69
- [245] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding,” 2022. 69
- [246] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015. 69
- [247] A. Sarkar and S. Cooper, “Dungeon and Platformer Level Blending and Generation using Conditional VAEs,” in *Proceedings of the IEEE Conference on Games (CoG)*, 2021. 69
- [248] H. Takagi, “Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation,” *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001. 70
- [249] J. Schrum, J. Gutierrez, V. Volz, J. Liu, S. Lucas, and S. Risi, “Interactive Evolution and Exploration within Latent Level-Design

Space of Generative Adversarial Networks,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO '20*, (New York, NY, USA), pp. 148–156, Association for Computing Machinery, 2020. event-place: Cancún, Mexico. 70

- [250] J. Xie, C. M. Myers, and J. Zhu, “Interactive Visualizer to Facilitate Game Designers in Understanding Machine Learning,” in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–6, 2019. 72

Part II.

PAPERS

PAPER I - FOSTERING CREATIVITY IN THE MIXED-INITIATIVE EVOLUTIONARY DUNGEON DESIGNER

Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, Chelsi Nolasco, and Axel Österman

ABSTRACT

Mixed-initiative systems highlight the collaboration between humans and computers in fostering the generation of more interesting content in game design. In light of the ever-increasing cost of game development, providing mixed-initiative tools can not only significantly reduce the cost but also encourage more creativity amongst game designers. The Evolutionary Dungeon Designer (EDD) is a mixed-initiative tool with a focus on using evolutionary computation to procedurally generate content that adhere to game design patterns. As part of an ongoing project, feedback from a user study on EDD's capabilities as a mixed-initiative design tool pointed out the need for improvement on the tool's functionalities.

In this paper we present a review of the principles of the mixed-initiative model, as well as the existing approaches that implement it. The outcome of this analysis allows us to address the appointed needs for improvement by shaping a new version of EDD that we describe here. Finally, we also present the results from a user study carried out with professional game developers, in order to assess EDD's new functionalities. Results show an overall positive evaluation of the tool's intuitiveness and capabilities for empowering game developers' creative skills during the design process of dungeons for adventure games. They also allow us to identify upcoming challenges pattern-based mixed-initiative tools could benefit from.

PUBLISHED IN

Proceedings of the 13th International Conference on the Foundations of Digital Games, ACM, 2018

FOSTERING CREATIVITY IN THE MIXED-INITIATIVE EVOLUTIONARY DUNGEON DESIGNER

Introduction

Mixed-initiative systems in game content creation [1] refer to the combination of functions produced by procedural content generation (PCG) algorithms and human designer intentions.

In today's design paradigm, it is a common approach to have humans and machines collaborate to maximize creativity during the design process and thus software have become a backbone tool for a designer to create artifacts within the areas of architecture, consumer product and interior design. As a result, computer-aided design (CAD) has been an important facet for design practices [2]. It could be argued that game development is a fast growing application area for this facet.

Games are part of an evolving medium of creative expression, but limitations still exist in regards to its design tools' accessibility due to the fast paced life cycle and expensive nature of game development. The rising cost in game development due to games' technological evolution has resulted in a push towards automatically generated content [3–5]. Cost drivers may include multiple factors, but in the context of work processes involving human designers and artists, they are commonly identified as a huge contributor, since they are expensive. Games' complexity in design, requires the involvement of tens to hundreds of staff across a development period that may span for years. This can negatively affect a company's profitability and the development team's innovative and creative vision.

PCG approaches and functionalities are used to reduce the workload of developers, and to promote cooperation between humans and machines by providing more diverse game content that could increase quality and re-playability [3, 6]. Various development tools and level editors can be used by human designers at their disposal, making them the sole driver of the creative process.

PCG, however, may limit the human designer's intentions by strictly following its own algorithms, disregarding the designer's desired parameters before generation [2]. Rather than simply being limited tools of support for the other, mixed-initiative systems can foster co-creativity in game design by combining the best of these two perspectives. Not only would it improve a development team's overall productivity, it can also guide and improve the creativity of smaller indie teams and individuals in developing more interesting and content-rich games with less worries about development costs [6, 7].

This paper is organized as follows: Section Related work presents the related work in Mixed-Initiative design and the previous results of Evolutionary Dungeon Designer, both of them as motivation for the current work. Section Improving the Mixed-Initiative Evolutionary Dungeon Designer describes the contributions of this paper together with a description of the last release of the tool. Section User Study presents the results from the user study conducted with game developers, followed by the conclusions and future work discussed in Section Results and Discussion.

Related work

The Main Principles of Mixed-Initiative

There are two different types of mixed-initiative [1]. The first type relates to the human creator having an idea and the computer being the mean of expression through aiding the human in their creative task (e.g. a text editor or Photoshop). The second type is described as the computer autonomously generating content and being evaluated, changed and edited by a human designer. This division is also described by [3] where they present a scale with the two extremes on opposite ends: purely human design on one and purely computational design on the other. In between these extremes are varying forms applied to mixed-initiative content generation tools used for game design.

Artificial intelligence (AI) techniques have become more common and essential on aiding designers while they develop games [8]. This motivates the use of mixed-initiative systems, which promotes the co-creativity between human designers and machines providing more interesting and exciting creations [2]. However, there are still some problems when generating content, thus advocating developers still doing manual designs from scratch. Some drawbacks of completely relying on PCG is the low reliability, believability, and high predictability of the game content - all which guarantees difficulties in evaluating the generated content like, for instance, a dungeon level's quality [9]. Therefore, by following the principles of

mixed-initiative through combining the content generation with the guidance and input from a human designer, you provide aspects from both parties, hopefully limiting the weaknesses from either.

Mixed-Initiative Tools in Game Design

Recent research in the field has presented different approaches to mixed-initiative authoring tool. These are *Tanagra*, *CICERO* and the *Sentient Sketchbook*. While all these provide mixed-initiative interfaces to the designer, they also share the limitation of addressing a specific game type.

Sentient Sketchbook aims at generating maps for strategy games such as *Starcraft* [10]. Users can sketch a low-resolution map that seeds an underlying evolutionary algorithm that provides suggestions. Low-resolution sketches reduce the creative strain on the user during the design process, but also makes it easier for the program to detect patterns in the map. Once the user deems the generated and edited low-resolution map good enough, the program can then generate the higher resolution map while still maintaining the patterns that were detected in the sketch.

Tanagra is used to develop 2D platformer levels [11], while still checking whether the generated content is playable or not. *Tanagra* offers users an empty grid where they can place different tiles such as floor, enemies, and coins. Mixed-initiative is implemented so that users can select tiles and objects they want to keep in their designs, while *Tanagra* generates new content around them.

CICERO focuses on the generation of dungeons for adventure games [7]. *CICERO* offers users the possibility to define the behavior of the game components they include in their designs, such as power-ups, win and lose conditions, and collision-triggered behavior. From these definitions *CICERO* recommends different game mechanics that would suit the game, such as the optimal weapon types to include in the game. By manually editing game content in the dungeon and having *CICERO* run the game and test different element combinations, users can understand how different layouts affect the generated gameplay.

Dungeon Design in Videogames

The dungeon is a popular level design archetype found in several popular game genres [12, 13]. Dungeons are also popular in PCG research, where different approaches have been presented for generating dungeon levels [1, 14–17]. These works emphasize the importance of considering goals, missions, the narrative or themes, visual style, and gameplay rules when designing levels, therefore they should be taken into consideration when developing a mixed-initiative tool for



Figure 1: The start screen lets users choose the dungeon dimensions.

content generation [9]. These factors are mostly decided by the human designer, thus a designer has to be integrated into the dungeon generation process.

Another key aspect to dungeon generation is player progression [18]. Designers ensure that the player's experience throughout a level will be coherent and effective, which will be affected by the content they create. This includes reward and challenge balancing among the rooms in a dungeon.

The Evolutionary Dungeon Designer

Previous research presented the *Evolutionary Dungeon Designer* (EDD) [19, 20] as a mixed-initiative authoring tool for designing dungeon rooms for adventure games. EDD automatically generates and suggests rooms to the user while the user is manually designing one of them. The user either form the room from scratch or from a previously generated suggestion. This is done by means of a FI-2Pop GA [21], where game design patterns are used both as input parameters and as objectives. These patterns involve micro-patterns (ENEMY, TREASURE, CHAMBER, CORRIDOR, CONNECTOR, ENTRANCE, and DOOR) as well as meso-patterns (AMBUSH, GUARD CHAMBER, TREASURE CHAMBER, and GUARDED TREASURE). EDD also ensures that all generated rooms are playable.

Initial experiments on EDD [19] validated its PCG system in terms of fitness optimization, pattern detection, and solution diversity, providing a sufficient level of control to the designer. The following iteration [20] explored the capabilities of EDD as a mixed-initiative level generator as a means of facilitating collaboration between human designers and PCG algorithms. Among its key features, the participants of a user study highlighted EDD as a useful framework for working

with game design patterns in the context of search-based problems. The suggestions were considered a good source of inspiration as well as time saving. This user study also shaped the roadmap for future improvements on EDD. This included extending EDD from room generation to complete dungeon generation, and preserving the users' designs to a higher degree in relation to both design patterns and room aesthetics.

This version of EDD extends previous work based on the aforementioned user study by implementing the following key improvements:

- The designer is now able to construct, develop, and edit a grid-based dungeon of different dimensions and inter-connected rooms, in contrast to a single room, which in turn, helps the designer on having context over their work on individual rooms and giving them more freedom on producing variations.
- The designer receives extended information about the consequences of their changes in individual rooms, and the differences between the current edited room and the proposed suggestions by the EA.
- The UI has been renovated to account for the newly added features by means of different views and options, as well as, a better structure and distribution of the different elements in the generator.
- Navigation tools have been added within a view and between views, which provides an overview of the dungeon, along with a better context of the edited room.
- The EA has been updated to assess and preserve the aesthetic criteria of the designer by means of a new capability of locking sections of an edited room for preserving custom aesthetic structures, and by extending the evaluation function through the measurement of symmetry and similarity in the provided suggestions, both which are further explained in [22].

Improving the Mixed-Initiative Evolutionary Dungeon Designer

fig. 1 shows the start screen in EDD, which starts a new workflow by prompting users to choose the maximum number of rooms in the dungeon to be developed. The dimensions range from 2x2 rooms up to 7x7 rooms in a square dungeon grid (also referred as world grid). From this point, the workflow offers users three different views: 1) a world view for dealing with aspects regarding the dungeon as a whole; 2) a room view which places the focus in a particular room in the dungeon; and 3) the suggestions view, which produces six different suggestions with diverging room configurations (e.g. more corridors or more chambers) for the

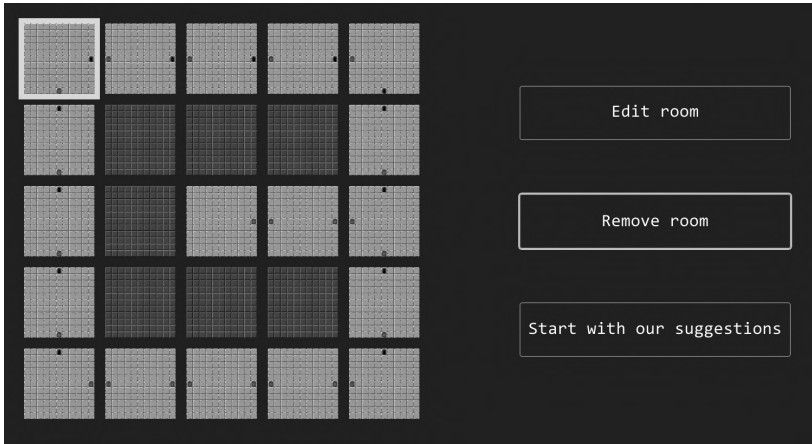


Figure 2: Sample world view showing a 5x5 dungeon with 7 disabled rooms. User actions are displayed in the rightmost buttons.

user to choose from. The user can freely alternate between views during the design process. The current dungeon layout can be saved at any moment from either of the views.

The world view (fig. 2) opens up right after the start screen, displaying a grid of the selected size composed by a fully connected set of empty rooms (all rooms are connected to their neighbors). The users can load a previously saved dungeon design, skipping the start screen and resume work from the state in which the dungeon design was saved.

From the world view users can then click and select any room to:

- disable or enable the room. Disabling makes the room inaccessible, removing all doors from the adjacent rooms. This can be undone by clicking *enable*. Single rooms that become isolated after all their neighbors have been disabled, are automatically disabled as well. figs. 3a and 3b show two examples of dungeons with several disabled rooms,
- get procedurally generated suggestions for that room in the suggestions view,
- load the room in the room view for manual editing.

The Suggestions View

By selecting “Start with our suggestions” in the world view (fig. 2) six uniquely generated rooms are presented to the user in a separate window: the suggestions

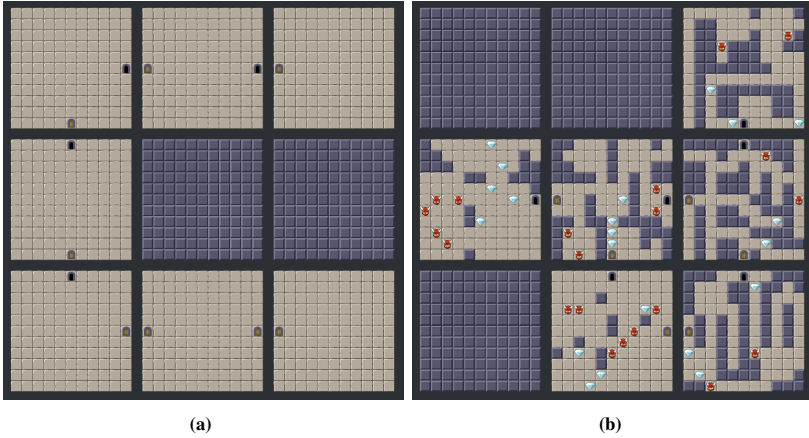


Figure 3: (a) 3x3 dungeon with 2 disabled rooms and 7 empty rooms, and (b) 3x3 dungeon completed dungeon with 3 disabled rooms.

view (fig. 4). When clicking any of the suggestions, it will replace the previously selected room in the dungeon.

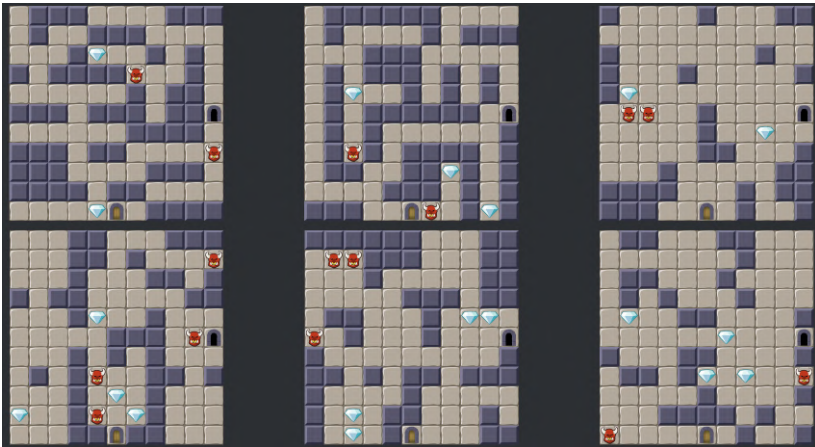


Figure 4: Six procedurally generated rooms presented to the user in the suggestions view.

A similar functionality was present in the former version of the tool, presented only once as the start screen. Now users can freely alternate between the world and the suggestions views, getting as many suggestions as they need, deciding whether to start creating every room from a clean state or to get inspiration from one of the generated rooms.

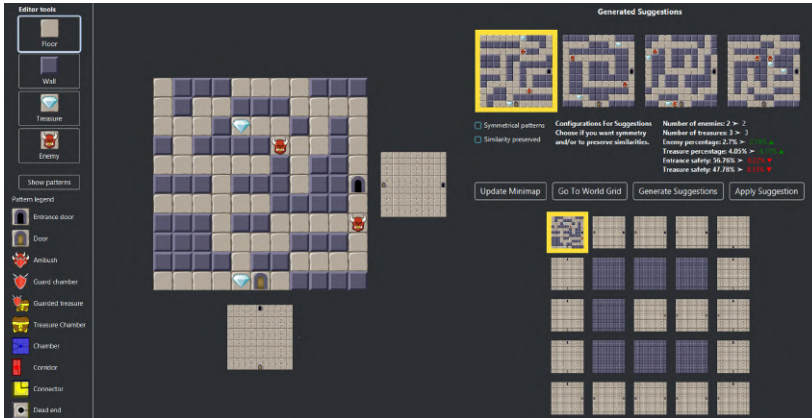


Figure 5: The room view while editing the top-left corner in a 5x5 dungeon with seven disabled rooms.

Suggestions preserve the door layout from the room that was selected in the world view. The suggestions shown in fig. 4 have been created for the room selected in fig. 2, placed in the top-left corner, and containing only two doors connecting them to their east and south neighbor.

The Room View

Users edit single rooms in the room view (fig. 5), regardless of whether these are new empty rooms, procedurally generated suggestions, or previously edited rooms. The room view is an improved and extended version of the main screen in the last version of EDD [20]. All functionalities from that version are still present: manually editing the room by changing tiles (floor, wall, enemy, and treasure), displaying an overlay view of the existing design patterns, and procedurally generating suggestions based on the current edited room's configuration.

Navigation is one of the crucial added features, allowing users to move around the dungeon without going back to the world view. Two other options offer navigation through the dungeon: the navigational buttons and the minimap. The navigational buttons are displayed next to each of the edited rooms' borders that contain a door. Provided that the room being edited in fig. 5 is located in the top-left corner, two navigational buttons are displayed right and below the room, respectively. Clicking a navigational button transports the user to that room, replacing the currently edited room with the targeted neighbor. Instead of using arrows or any other fixed picture, these buttons preview the neighboring rooms as a hint for generated users to help them

design the room currently being edited. The navigational buttons are automatically refreshed to reflect up-to-date changes performed to the neighboring rooms.

The minimap displays a scaled-down overall picture of the whole dungeon, highlighting the currently edited room with a yellow border. Users can navigate to any room, which is not disabled and is displayed on the minimap by clicking on it, replacing the current room. The buttons above the minimap allow users to go *Back To World Grid*, to *Update Minimap*, as well as request and select procedurally generated suggestions. The whole minimap is updated whenever the user navigates to a different room, but if the user wants to see the last changes applied to currently edited room reflected on the minimap, a manual refresh has to be done. This is done to reduce the workload derived from re-rendering the minimap automatically after every manual edition.

The generated suggestions work similarly to the previous version of EDD: four unique maps are generated by the underlying evolutionary algorithm in four subsequent evolutions, seeding the four initial populations with different sets of features extracted from the edited room. Each suggestion is evolved by means of a different fitness function, therefore addressing different goals to maximize diversity in the provided suggestions. Clicking on a suggestion highlights it, and clicking *Apply Suggestion* replaces the current room with the highlighted map. This differs from the previous version, in which maps were applied at the moment they were clicked on, occasionally causing work loss due to accidental replacements.

Additionally, highlighted suggestions display informative parameters below them. These describe meaningful features of the highlighted room that are relevant to both the human designer and the evolutionary algorithm's fitness calculation: number of enemies and treasures, enemy and treasure rate (in relation to floor tiles), and entrance and treasure safety (see [19] for a detailed description). These parameters are displayed as a comparison between their values in the edited room and in the highlighted suggestion, showing how they would change if the suggestion is applied.

Two checkboxes below the suggestions now offer users the possibility to ask specifically for the provided suggestions to address symmetry and similarity aesthetic features, respectively. By ticking the symmetry checkbox, two of the suggestions will be generated by the evolutionary algorithm using a symmetry fitness function, which enable the generation of symmetric rooms, (either vertically, horizontally, or diagonally). Analogously, ticking the similarity checkbox, the other two suggestions are generated with a similarity fitness function, which promotes the generation of room aesthetically similar (but never equal) to the currently edited

map. These fitness functions are presented in [22]. When both checkboxes are unchecked, the fitness functions described in [20] are used.

User Study

A user study was conducted in order to assess the impact on the design process caused by the improvements made to EDD. Five game developers participated in the study, which had the following structure:

- *Introduction to the purpose of the study.* Participants were asked whether they were familiar with the previous version of EDD.
- *Demonstration of the tool,* showcasing its workflow and features with a short example performed over a 3x3 dungeon.
- *Designing a dungeon.* After the demonstration the users were tasked to design a 3x3 dungeon within approximately 10 minutes, saving the work after that for a later analysis and discussion conducted in a structured interview with the participant after this phase. Two observers took notes of what the participant was doing, providing additional data for the later analysis.
- *Questionnaire.* The users were asked a few questions regarding their background in game design as well as dungeon-based games. They were also asked whether they had any previous experience with mixed-initiative tools.
- *Interview.* A semi-structured interview was conducted to provide data for an analysis and discussion about the tool, and its improvements. Audio was recorded for a later analysis.

As a result of the questionnaire, the following information was gathered from the participants:

- User 1 has been working for more than ten years in the game industry as a data scientist and user experience researcher. The user holds prior experience with RPGs and dungeon crawlers, and is familiar with the terms of mixed-initiative tools and has used The Sentient Sketchbook in the past. This user is the only one who participated in the former user study of EDD.
- User 2 has been working for six months as a project coordinator of eSports events and has long experience of playing dungeon crawlers and RPGs. This user is not familiar with mixed-initiative concepts and has never used a mixed-initiative authoring tool before.
- User 3 has been working for six years in the game industry as a user experience researcher and a biometrics expert. The user has prior experience

with dungeon style games, but has limited knowledge about mixed-initiative tools.

- User 4 has been working for nine years as a senior user experience researcher and has long experience of playing dungeon crawlers and RPGs. This user is not familiar with mixed-initiative concepts and has never used a mixed-initiative authoring tool before.
- User 5 has been working for three weeks as a game user researcher. This user has no experience with dungeon crawlers and dungeon-based RPGs, and is not familiar with mixed-initiative tools.

Results and Discussion

As with any raw qualitative data, the data collected from the user study has to go through a condensation process in order to isolate the most relevant information that will answer the research questions. In a blueprint provided by [23], the qualitative data obtained has undergone four out of five stages:

- *Material sourcing*: an audio recording the user study, interview materials, and the authors' own observations.
- *Transcription*: combining and writing down the observations and questionnaire answers for each participant in the user study.
- *Unitization*: dividing the data according to the mixed-initiative features of EDD.
- *Categorization*: dividing the data according to categories relevant to the research questions while taking into consideration the principles of mixed-initiative.

All participants in the user study perceived EDD as overall good and intuitive. table 1 shows their general consensus of EDD's usability and capability to foster creativity in dungeon design. table 2 lists the participants' most requested missing features.

The main goal of mixed-initiative interaction pertains to the flexibility of roles between the human and computer as a team and simplifying the general experience [24], and this was somewhat achieved by EDD. This could be proven by how features such as suggestions and the implementation of a whole dungeon with navigation have definitely supported the users when making decisions throughout the design process. As a result the experience was overall simple and intuitive. It could not be said, however, that the set goal has been fully achieved; a fully successful mixed-initiative system emphasizes interchangeable roles of the human

and computer while maintaining the balance between them. The participants in the user study did not feel restricted, but they still desired more control in EDD's assistance in the design process, as well as different suggestions that the designer cannot come up with themselves.

[25] provides a list of principles for mixed-initiative user interfaces which would enhance human-computer interaction. EDD has achieved four out of twelve in Horvitz's list of critical factors that would make up a fully successful mixed-initiative system:

- *Developing significant value-added automation*: providing an automated solution that cannot be achieved with direct manipulation. EDD provides a framework for the generation of complex dungeons of different sizes, together with suggestions of similar dungeon rooms and information parameters for these suggestions.
- *Considering uncertainty about a user's goals*: taking advantage of a user's uncertainty in their intentions. EDD provides the choice to initialize rooms in a dungeon with either an empty slate or from any of the generated suggestions.
- *Inferring ideal action in light of costs, benefits, and uncertainties*: considering the value of an automated service in regards to the usually expected value of taking actions. EDD's main motivation is to significantly reduce the cost of game design while maintaining and improving creativity, which has at least partially been fulfilled.
- *Employing dialogue to resolve key uncertainties*: establishing an efficient dialog between the human and computer when uncertainty arises while considering the costs of potentially disrupting the user. EDD extracts and displays relevant features in the edited and suggested rooms for the users to guide their decisions. The minimap also fulfill parts of this role.

There are other principles which are relevant to EDD which fall in line with the participants' feedback. For example, some principles such as the ability to continuously learn from the user's input and to preserve memory of their decisions and actions may pertain to the desired features of having more control in the generation maps and receiving more assistance in preserving their own manual designs for different purposes.

Conclusions and Future Work

The contributions presented in this work explore how the user interface and the mixed-initiative aspects in the Evolutionary Dungeon Designer have been improved,

as well as how they should be improved on further in order to increase creativity during the dungeon design process.

The addition of the world grid provides the adoption of a new workflow, which offers users the possibility to start designing either from empty rooms or PCG suggestions. Various changes in the user interface were made to accommodate the increased dungeon size. With a larger dungeon, navigation has proved to be a key functionality, as well as giving an overview of the adjacent rooms. Now users can get a better understanding of the context of the room currently being edited. In conjunction with the navigation and larger dungeons, a minimap was also added to further enhance the experience when designing a larger dungeon. Alongside these changes, aesthetic goals have been included in the generative process. Visual cues for room descriptors were added, so that the user can make a more informed decision when selecting suggested maps.

Compared to its previous version, EDD further empowers the mixed-initiative design process by providing more context, feedback, flexibility, and to some extent, the ability to address aesthetic features in the procedural suggestions. Creativity can directly adhere to the amount of interesting possibilities a designer can employ, which is relevant to providing rich contexts to dungeon designs. EDD offers a mixed-initiative experience that provides adequate flexibility for the designer's intentions as the results from the user study have shown.

Overall, our user study successfully shows the strengths of mixed-initiative tools for designers but it also reveals various limitations, which should be considered by the community when creating a mixed-initiative tool.

To a certain extent, controllability is preferred than expressivity, as the users continuously try to impose their vision, which is a non-trivial task for automated systems to capture, thus, the users are more likely to sacrifice to a certain degree expressivity and exploration of the tool by gaining control over the generated content.

The capability of proposing useful and novel suggestions is fundamental to fostering creativity and impulses the generation of more interesting content. Moreover, explicit information about the designers' changes and choices is important as it helps them understand the effect of their decisions.

Finally, this work has identified features that should still be taken into consideration for future versions of the tool, which are shown in table 2.

ACKNOWLEDGMENTS

The Evolutionary Dungeon Designer is part of the project *The Evolutionary World Designer* which is supported by The Crafoord Foundation.

Table 1: General consensus on EDD's features

Description	Participants' Consensus
World Grid of the dungeon	Its purpose of establishing an illusion of a fully realized dungeon is somewhat achieved. However, limitations exist with how it defines feasibility, a dungeon's starting point, and the entrances, which disrupts the designers' decisions.
World View	The world view's usefulness for the most part could not be established, other than for the purpose of going to the suggestions view (which was already seldom during the user study) and having a closer look at the entire dungeon without any distractions. Some participants preferred features to be already in the room view's minimap, and some wanted to see more specific functionalities within the world view itself.
Enabling and disabling rooms	As the user study restricted participants to create 3x3 dungeons, this feature for the most part has been neglected. This is also in part because of its accessibility only being in the world view, which proved to be an inefficient view in general. However, its use for bigger dungeon sizes later on was appreciated, especially for more intricate design purposes.
Suggestions View	Similarly to enabling and disabling rooms, it was quite difficult to encourage the use of this functionality due to the world view's inefficient usability. However, this could also be due to the dungeon's small size, as some participants expressed high interest in using more suggestions with larger dungeon sizes.
Minimap navigation	The minimap proved to be a strong tool not only for navigation purposes, but also for supporting design decisions and choices. The directional buttons were rarely used, but their room previews were helpful in emphasizing the current room's connection to adjacent rooms without looking at the minimap. On the other hand, this lowered the usability of the world view.
Parameters	The parameters were, in general, lacking. They served to be important in decision-making when choosing a suggested map in room view, but there were still doubts on their accuracy and sufficiency when providing information about the generated suggestions.
Generated maps for suggestions in room view	Suggestions in the room view proved to be very helpful in supporting the whole design process as they primarily acted as inspirations for the users. The most prominent comment among the users is the preference of having more control on how suggestions should be generated depending on different types of parameters.
Design patterns	The patterns' visualization was, in general, lacking and not self-explanatory. Some participants have expressed interest in using patterns as a parameter in the generation of suggestions.
Dark theme	EDD's dark theme for the user interface received a positive response as it makes working with the program easier.

Table 2: Participants' most requested features

Feature	Description
Design patterns	Their visualization and accuracy should be improved. Other than acting as visual guide for map information, they should be used to help generate rooms as well. They should also be available for the entire dungeon.
Parameters	They need to have more information about the specific room, and have better visualization in order to make the designer trust their accuracy more. The parameters should also consider the entire dungeon as a whole in different terms such as difficulty and balance.
Generated suggestions	In general, the participants want more variety and control in the generation of suggestions using different types of parameters e.g. their degree of similarity and fitness functions.
Redefined feasibility	Eddy 3.0's definition of feasibility should be revised which considers the whole dungeon and its connected rooms.
World View	The World View should be revised and enhanced with more special features which would encourage users to visit it more.
World grid	The computation of the whole dungeon should be improved. It should have an option to define a starting point. Its definition of entrance doors should be improved, as well as the calculation of distances of tile types.
Version control	Some participants want to preview suggestions within the Room View to help their judgment and the ability to save suggestions for later use. They also want to revert to old designs in case they have second thoughts.
Templates	Some participants want the ability to save their own manual designs to be carried over to other grids.
Automated assistance	The participants in general welcome a bit more automated assistance when doing manual designs, which can reduce clicking around the program. It should also not be too invasive for the designer.

References

- [1] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016. 109, 110, 111
- [2] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-Initiative Co-Creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 109, 110
- [3] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Springer, 2018. <http://gameaibook.org>. 109, 110
- [4] J. M. Font, R. Izquierdo, D. Manrique, and J. Togelius, “Constrained Level Generation Through Grammar-Based Evolutionary Algorithms,” in *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part I* (G. Squillero and P. Burelli, eds.), pp. 558–573, Cham: Springer International Publishing, 2016. 109
- [5] L. Doherty, M. Whitney, J. Shakya, M. Jordanov, P. Lougheed, D. Brokenshire, S. Rao, S. Menon, and V. Kumar, “Mixed-initiative in computer games: Algorithmic content creation in open-ended worlds,” in *AAAI Fall Symposium on Mixed-Initiative Problem-Solving Assistants*, pp. 46–56, aaai.org, 2005. 109
- [6] A. Liapis, G. N. Yannakakis, and J. Togelius, “Generating Map Sketches for Strategy Games,” in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS, Springer, 2013. 109, 110
- [7] T. Machado, A. Nealen, and J. Togelius, “Seekwhence a retrospective analysis tool for general game design,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG ’17, (New York, NY, USA)*, pp. 4:1–4:6, ACM, 2017. 110, 111

- [8] G. N. Yannakakis and J. Togelius, "A panorama of artificial and computational intelligence in games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, pp. 317–335, Dec 2015. 110
- [9] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-Initiative Design of Game Levels: Integrating Mission and Space into Level Generation," in *Proceedings of the 10th International Conference on the Foundations of Digital Games*, 2015. 110, 112
- [10] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient Sketchbook: Computer-Aided Game Level Authoring," in *Proceedings of the 8th Conference on the Foundations of Digital Games*, 2013. 111
- [11] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 201–215, Sept 2011. 111
- [12] Nintendo Research & Development 4, "The Legend of Zelda," 1986. 111
- [13] Blizzard North, "Diablo," 1996. 111
- [14] L. Johnson, G. N. Yannakakis, and J. Togelius, "Cellular Automata for Real-time Generation of Infinite Cave Levels," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, (New York, NY, USA), ACM, 2010. 111
- [15] J. Dormans, "Adventures in Level Design: Generating Missions and Spaces for Action Adventure Games," in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, (New York, NY, USA), ACM, 2010. 111
- [16] R. van der Linden, R. Lopes, and R. Bidarra, "Designing procedurally generated levels," in *Proceedings of the 2nd AIIDE Workshop on Artificial Intelligence in the Game Design Process*, 2013. 111
- [17] D. Karavolos, A. Liapis, and G. N. Yannakakis, "Evolving Missions to Create Game Spaces," in *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 2016. 111
- [18] E. Butler, A. M. Smith, Y.-E. Liu, and Z. Popovic, "A mixed-initiative tool for designing level progressions in games," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pp. 377–386, ACM, 2013. 112
- [19] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Mixed-initiative procedural generation of dungeons using game design patterns," in *2017 IEEE*

Conference on Computational Intelligence and Games (CIG), pp. 25–32, Aug 2017. 112, 117

- [20] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG ’17, (New York, NY, USA), pp. 74:1–74:10, ACM, 2017. 112, 116, 118
- [21] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a Feasible–Infeasible Two-Population (FI-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310 – 327, 2008. 112
- [22] A. Alvarez, S. Dahlskog, J. M. Font, J. Holmberg, and J. S., “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, (New York, NY, USA), ACM, 2018. 113, 118
- [23] K. J. Srnka and S. T. Koeszegi, “From words to numbers: how to transform qualitative data into meaningful quantitative results,” *Schmalenbach Business Review*, vol. 59, no. 1, pp. 29–57, 2007. 119
- [24] J. Allen, C. I. Guinn, and E. Horvitz, “Mixed-initiative interaction,” *IEEE Intelligent Systems and their Applications*, vol. 14, no. 5, pp. 14–23, 1999. 119
- [25] E. Horvitz, “Principles of mixed-initiative user interfaces,” in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 159–166, ACM, 1999. 120

PAPER II - ASSESSING AESTHETIC CRITERIA IN THE EVOLUTIONARY DUNGEON DESIGNER

*Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg,
and Simon Johansson*

ABSTRACT

The Evolutionary Dungeon Designer (EDD) is a mixed-initiative tool for creating dungeons for adventure games. Results from a user study with game developers positively evaluated EDD as a suitable framework for collaboration between human designers and PCG suggestions, highlighting these as time-saving and inspiring for creating dungeons.

Previous work on EDD identified the need of assessing aesthetic criteria as a key area for improvement in its PCG Engine. By upgrading the individual encoding system and the fitness evaluation in EDD's evolutionary algorithm, we present three techniques to preserve and account the designer's aesthetic criteria during the dungeon generation process: the capability of locking sections for preserving custom aesthetic structures, as well as the measurement of symmetry and similarity in the provided suggestions.

PUBLISHED IN

Proceedings of the 13th International Conference on the Foundations of Digital Games, ACM, 2018

ASSESSING AESTHETIC CRITERIA IN THE EVOLUTIONARY DUNGEON DESIGNER

Introduction

Procedural content generation (PCG) has been widely used to generate content in games for different reasons, due to constraints in memory [1], create new experiences for the user [2], animations [3] or more recently, to create most of the assets [4]. Moreover, interest in PCG has increased as researchers have explored ways to automate, reduce cost and, produce novel and interesting content, for instance, weapons [5], levels [6], music and sound [7, 8], and even complete commercial games [9].

Search-based procedural content generation (SBPCG) is a popular PCG approach that uses evolutionary algorithms (EA) for guiding the content generation process by means of evaluation functions [10]. Mixed-initiative SBPCG involves human users in the evolutionary process so that promotes the co-creation of human and machine-made designs [11].

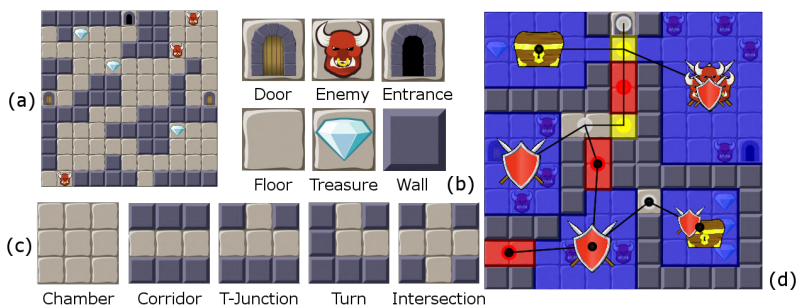


Figure 1: Current version of EDD and its different components. (a) Basic room, (b) different placeable tiles, (c) micro-patterns and (d) meso-patterns.

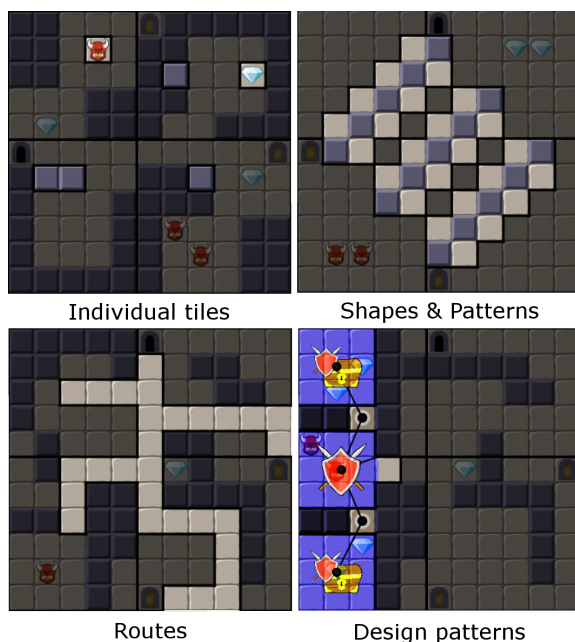


Figure 2: Different uses and possibilities that the designers can have for locking the tiles in the Room, in order to, preserve their manual changes and diverse objectives

As discussed by [12], it is important for a mixed-initiative SBPCG approach to evaluate the degree to which generated designs are aesthetically pleasing and interesting to the human designer. This is stressed by the designer’s will to imprint and preserve their custom designs on the generated content offered by the PCG system. It is a non-trivial task to know which parts the designer wants to preserve, as well as correctly balancing human and procedurally designed content in the generated solutions. This motivates the work presented here, in which we address the need for assessing aesthetic criteria by improving both: the solution encoding mechanism and the fitness evaluation function in EDD’s evolutionary algorithm.

This paper is organized as follows: Section Related Work presents previous and related works in mixed-initiative design. Section Assessing Aesthetic Criteria describes in detail the contributions of this paper and presents the results from the laboratory experiments used for validating them. Finally, Section Conclusions and Future Work summarizes and discusses these results, as well as sets future questions to be addressed by further research in the area of aesthetic criteria and EDD.

Related Work

Aesthetic criteria was specified by previous research as a key feature while evaluating content, as it leads to the generation of more customized content in the eyes of the human designer, whose aesthetic vision on the content is preserved [5, 13, 14].

Interactive evolutionary approaches incorporate human evaluation by allowing the user to select, either implicitly or explicitly, the parents of the next generation of procedurally generated individuals. In [15] system allows users to draw simple primitive shapes to seed an evolutionary algorithm and train a neural network with their aesthetic vision. In Galactic Arms Race [5] players preferences on the evolved weapons is implicitly deducted from the amount time they actively select those weapons during the gameplay.

[13], incorporated visual aesthetics as an evaluation of their generated spaceships by calculating different aesthetic concepts: symmetry along axes, weight distribution or design simplicity. Moreover, [16] generated levels for Mario using symmetry as objective function, which based on their user study, were as visually pleasing as the ones created by human designers and even more than other similar approaches.

The Evolutionary Dungeon Designer

EDD is a mixed-initiative authoring tool for generating dungeon rooms using a feasible-infeasible two population (fi-2pop) evolutionary approach, which is interactively evaluated and edited by a designer. The current version of EDD consists of six different building blocks that represent floors, walls, enemies, treasures, doors and entrances. This can be used by the user to brush paint and compose a NxM size room which, at its minimum, must hold one of each tile. Both the tiles and the finished room can be seen in Figure 1a) and b).

EDD takes the work presented in The Evolutionary World Designer [17] one step further, by procedurally generating rooms and their specific content. EDD's EA follows the approach of [13] using the evaluation of the user to change the internal evaluation and configuration of the system. Its fitness evaluation is driven by the use of game design micro- and meso- patterns, as shown in Figure 1 c) and d). A detailed description of EDD's pattern-based fitness, genetic algorithm and mixed-initiative approach can be found in [18] and [12].

Assessing Aesthetic Criteria

Our approach is divided in two; on one side, the algorithm implicitly has control over different aesthetic criteria using the edited room as a base to measure symmetry and similarity for the EA. On the other side, the designer was given control over

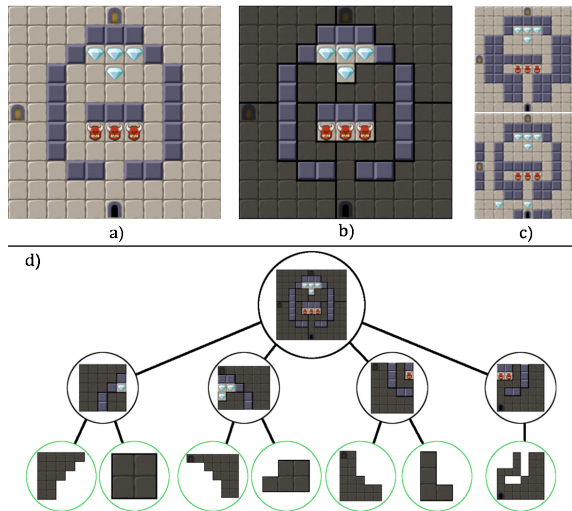


Figure 3: A sample edited room (a) with its division into zones (b) based on the tiles locked by the user. Suggestions preserve these locked tiles (c). The room and its zones are internally represented with a tree structure (d), where the leaf nodes (green) are the valid candidates to operate within an individual.

what they wanted to preserve by being able to select tiles in the room to be immutable (i.e. not changeable in following generations).

Preserving Custom Aesthetic Structures

To preserve the aesthetic criteria of a designer’s edited room, we give the users the ability to manually lock custom structures in it, preserving these in the upcoming suggestions. This is possible by incorporating a new brush which is used as a complementary modifier when editing the room. The designer can now lock any range of tiles, making it possible to preserve individual tiles, shapes, patterns, routes and even design patterns as shown in Figure 2.

The process to subdivide the room is straightforward; the designer is presented with the room to be edited, and by using the lock brush, the room seamlessly subdivides and creates zones, which classifies the room’s tiles into two sets: the immutable tiles (i.e. invalid or locked) and the mutable tiles (i.e. valid or unlocked).

An individual’s genotype is now changed from a direct encoding (each tile is a gene) to a semi-direct encoding using a tree structure, with the nodes of the tree as different zones of the room, constructed from the mutable and immutable tiles, and the leaf nodes, only containing sets of mutable tiles, as candidates to be used for crossing and mutation. Figure 3 shows the room, it’s division into zones and

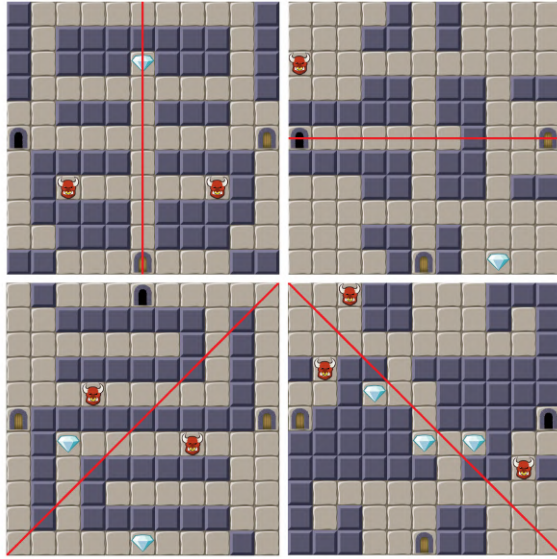


Figure 4: Different types of symmetry evaluated

the tree representation used by the EA.

The advantages of this representation are that it allows the EA to reduce the search space by only considering valid zones of the room, and improves the crossover operator by allowing the exchange of irregular shapes between individuals along different parts of the room.

In practice, this solution allows users to preserve any aesthetic change (either significant or detailed) that they want to keep in further generations, while still receiving novel suggestions created following the pattern-based fitness function. It also means that the construction of the dungeon can be performed differently: instead of manually editing a room first to later generate appealing solutions based on it, the user can now start from a suggestion, selecting parts of it that look promising that are kept through subsequent generations, until the user's needs and criteria are met.

Evaluating Symmetry and Similarity

While the pattern-based fitness function worked well for functionality purposes, it did not consider nor capture any aesthetic aspects into it. Therefore, in order to consider and preserve visual aesthetic criteria, we evaluate the rooms for their symmetry along the X and Y axes, backslash and front slash diagonal as

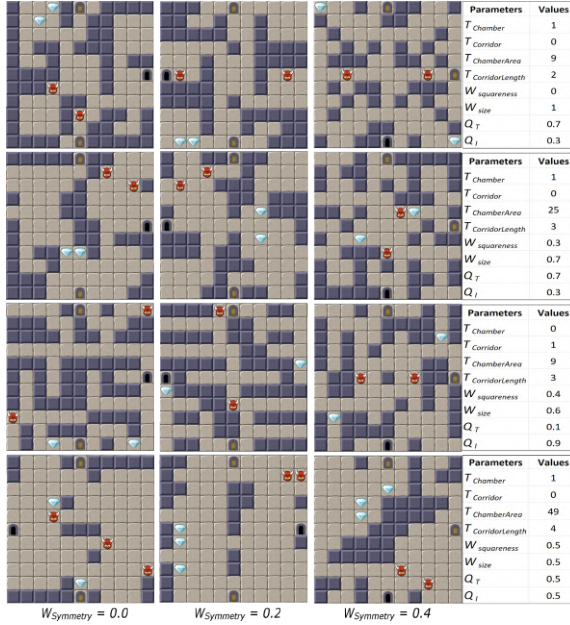


Figure 5: Each row shows three results ($W_{Symmetry} = 0, W_{Symmetry} = 0.2, W_{Symmetry} = 0.4$) produced under the settings displayed on the rightmost column. Metrics adapted from [18].

shown in Figure 4 and calculate the similarity that subsequent individuals had in comparison with the original edited room. For simplicity, we differentiate the room by impassable (i.e. walls) and passable (i.e. floor, treasure and enemy) tiles.

Symmetry evaluation

To calculate the symmetry of a room we evaluate the impassable tiles of one side against their corresponding tile on the other side for the X and Y axes and diagonals. The highest symmetric value is then used in equation 1 to calculate the fitness.

$$f_{symmetry} = \frac{highestSymmetricValue}{totalWalls} \quad (1)$$

Equation 1 allow us to calculate symmetry while also preventing the favoring of more walls. Once calculated, we weight the result into the individual's fitness, and as consequence it would favor more or less symmetric rooms and preserve the room's configuration as it can be seen in Figure 5.

Similarity evaluation

The similarity value between an edited room and successive evolved rooms is calculated by comparing every tile in the original with the corresponding tile in

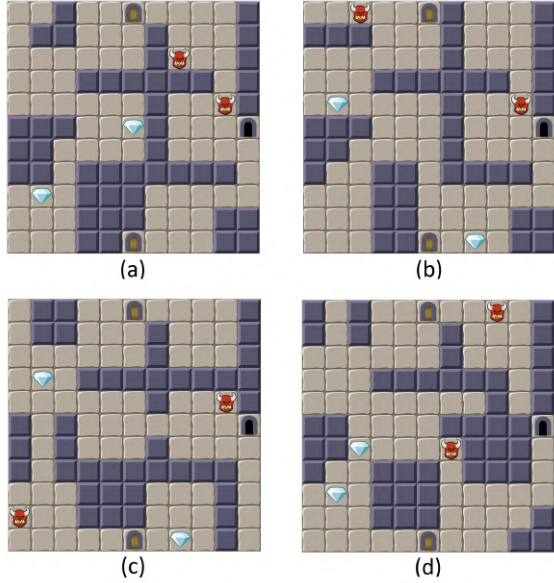


Figure 6: (a) Sample original room and the evolved solutions with different *idealSimilarity* values in order: (b) 0.95, (c) 0.90 and (d) 0.85.

subsequent individuals. Once the total amount of equal tiles is known, we calculate the similarity percentage based on the total amount of tiles, following equation 2.

$$similarityPercentage = \frac{totalTiles - notSimilarTiles}{totalTiles} \quad (2)$$

We introduced a second parameter called *idealSimilarity*, which represents how similar we want the individuals to be. Following equation 3 we measured the error between both similarities and used it as the similarity fitness.

$$f_{similarity} = 1 - |idealSimilarity - SimilarityPercentage| \quad (3)$$

The result of incorporating the similarity evaluation into the final fitness is shown in Figure 6 where is observable that depending on the *idealSimilarityPercentage* the original room goes from having a slight variation to start losing its resemblance.

Finally and expanding over the previous work on EDD [12], these calculations (i.e. $f_{symmetry}$ and $f_{similarity}$) are included into the existing fitness evaluation of an individual as shown in equation 4. $f_{inventorial}$ and $f_{spacial}$, evaluates the overall layout of the room, and the frequency and quality of the design patterns in the room, respectively. An in-depth explanation of both can be found in [12].

$$\begin{aligned}
f_{fitness}(r) = & \left(\frac{a}{10} f_{inventorial}(r) + \frac{b}{10} f_{spacial}(r) \right) \\
& + \frac{c}{10} f_{symmetry}(r) * f_{similarity}(r)
\end{aligned} \tag{4}$$

Conclusions and Future Work

In this paper, we have presented the advancements done on EDD in relation to the evolutionary system with different evaluations, encoding, genotype representation and strategies that aims on preserving and consider the designer's aesthetic criteria.

By introducing the capability of locking sections of a room, we changed the individual's encoding from direct to semi-direct, and in turn, offered new and easier possibilities to perform different operations to the individuals, as well as, allowing the designer to preserve individual tiles, shapes, routes and even design patterns.

Moreover, we successfully integrated and produced rooms evaluated on symmetry and similarity that held the overlying structure of the micro-patterns. The added evaluations establishes the path to preserve and consider more in-depth the designers criteria and produce personalized work that accurately transmit the ideas and intentions of the designer.

We aim to more thoroughly evaluate the system by incorporate the three techniques into a user study, similar to the one done by [12] to validate the tool's capacity on assessing the designer's criteria. It would be interesting to add more aesthetic concepts to evaluate the produced content, for instance, density, simplicity, sparseness and individuality.

The subdivision of the map could be extended to perform a parallel evolution on the custom aesthetic structures locked by the designers and propose interesting variations. Moreover, a zone analysis could be introduced to increase the dungeon's knowledge for the designer by suggesting changes to fulfill different player models, similar to Holmgård's approach [19], or paths and statistics. Finally, we would like to explore different types of representations towards more generative encodings to test, compare and measure the differences and advantages of the resulting maps.

We aim to further evaluate the system with different configurations and observe how the different fitness functions can interact and cooperate with each other to create more interesting content, as well as, joining both approaches for a case study, similar to the one done by Baldwin et al [12]. It would be interesting to continue using aesthetic concepts, for instance, density, simplicity, sparseness and individuality, to evaluate the content

Further use the division of the map by performing zone analysis, which could result on suggesting changes to the designers in order to fulfill different player models, similar to Holmgård's approach [19] or do a separated evolution on the manually locked tiles providing the designers with interesting shapes and patterns. Finally, we would like to go down the road towards more indirect encodings and test different approaches and, compare and measure the differences and advantages of the resulting maps.

ACKNOWLEDGMENTS

The project was supported by The Crafoord Foundation.

References

- [1] D. Braben and I. Bell, “Elite,” 1984. 131
- [2] M. Toy, G. Wichman, K. Arnold, and J. Lane, “Rogue,” 1980. 131
- [3] Maxis, “Spore,” 2008. 131
- [4] Hello Games, “No Man’s Sky,” 2016. 131
- [5] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Galactic Arms Race Video Game,” in *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG09)*, vol. 1, pp. 245–263, 2009. 131, 133
- [6] N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, and M. O’Neill, “Evolving levels for Super Mario Bros using grammatical evolution,” in *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2012*, (Granada), pp. 304–311, IEEE, 2012. 131
- [7] A. K. Hoover, P. A. Szerlip, and K. O. Stanley, “Interactively evolving harmonies through functional scaffolding,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO ’11*, no. Gecco, pp. 387–394, ACM, 2011. 131
- [8] M. Scirea, P. Eklund, J. Togelius, and S. Risi, “PRIMAL - IMPROV: Towards Co-Evolutionary Musical Improvisation,” in *Proceedings of the Computer Science & Electronic Engineering Conference (CEEC)*, pp. 172–177, IEEE, 2017. 131
- [9] C. Browne, “Yavalath,” 2007. 131
- [10] N. Shaker, J. Togelius, and M. J. Nelson, “The search-based approach,” in *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, ch. 2, pp. 17–31, Springer, 2016. 131
- [11] A. Liapis, G. N. Yannakakis, and J. Togelius, “Sentient sketchbook: Computer-aided game level authoring,” in *Proceedings of the 8th International Conference on the Foundations of Digital Games (FDG 2013)*, pp. 213–220, 2013. 131
- [12] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th*

- International Conference on the Foundations of Digital Games*, vol. 74, pp. 1–10, ACM, 2017. 132, 133, 137, 138
- [13] A. Liapis, G. N. Yannakakis, and J. Togelius, “Adapting models of visual aesthetics for personalized content creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012. 133
- [14] A. Machwe and I. C. Parmee, “Integrating aesthetic criteria with evolutionary processes in complex, free-form design – an initial investigation,” in *Proceedings of the IEEE International Conf. on Evolutionary Computation*, pp. 165–172, 2006. 133
- [15] J. Zhang, R. Taarnby, A. Liapis, and S. Risi, “DrawCompileEvolve: Sparking interactive evolutionary art with human creations,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9027, pp. 261–273, 2015. 133
- [16] J. R. H. Mariño and L. Lelis, “A computational model based on symmetry for generating visually pleasing maps of platform games,” in *AIIDE*, 2016. 133
- [17] J. M. Font, R. Izquierdo, D. Manrique, and J. Togelius, “Constrained level generation through grammar-based evolutionary algorithms,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9597, pp. 558–573, 2016. 133
- [18] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Mixed-initiative procedural generation of dungeons using game design patterns,” in *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2017*, pp. 25–32, 2017. 133, 136
- [19] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis, “Evolving Personas for Player Decision Modeling,” in *Proceedings of the Computational Intelligence and Games 2014 (CIG)*, 2014. 138, 139

PAPER III - EMPOWERING QUALITY DIVERSITY IN DUNGEON DESIGN WITH INTERACTIVE CONSTRAINED MAP-ELITES

Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius

ABSTRACT

We propose the use of quality-diversity algorithms for mixed-initiative game content generation. This idea is implemented as a new feature of the Evolutionary Dungeon Designer, a system for mixed-initiative design of the type of levels you typically find in computer role playing games. The feature uses the MAP-Elites algorithm, an illumination algorithm which divides the population into a number of cells depending on their values along several behavioral dimensions. Users can flexibly and dynamically choose relevant dimensions of variation, and incorporate suggestions produced by the algorithm in their map designs. At the same time, any modifications performed by the human feed back into MAP-Elites, and are used to generate further suggestions.

PUBLISHED IN

Proceedings of the 2019 IEEE Conference on Games (CoG), IEEE, 2019

EMPOWERING QUALITY DIVERSITY IN DUNGEON DESIGN WITH INTERACTIVE CONSTRAINED MAP-ELITES

Introduction

Procedural Content Generation (PCG) refers to the generation of game content with none or limited human input [1], where game content could be anything from game rules, quests, and stories, to levels, maps, items, and music. While PCG has been present in some games since trailblazing games like *Rogue* [2] and *Elite* [3], it has only been a popular academic research topic for a decade or so. Search-based PCG means using a global search algorithm such as an evolutionary algorithm to search content space [4].

Part of PCG's allure is the promise to produce game art and content faster and cheaper, as well as enabling innovative content creation processes such as player-adaptive games [5–7], data-driven content generation [8, 9], and mixed-initiative co-creativity [10]. Mixed-initiative co-creativity (MI-CC), a concept introduced by Yannakakis et al. [11], refers to a creation process through which a computer and a human user feed and inspire each other in the form of iterative reciprocal stimuli. Some examples of this are *Ropossum* [12], *Tanagra* [13], *CICERO* [14], and *Sentient Sketchbook* [15].

MI-CC aligns with the principles of lateral thinking and creative emotive reasoning: the processes of solving seemingly unsolvable problems or tackling non-trivial tasks through an indirect, non-linear, creative approach [16]. Even more, MI-CC provides insight and understanding on the affordances and constraints of the human process for creating and designing games [1].

A key mechanism in MI-CC approaches is to present suggestions to players, and these suggestions must have high quality but also be sufficiently diverse. So-called quality-diversity algorithms [17] are very well suited for this, as they find solutions

that have high quality according to some measure, but are also diverse according other measures. MAP-Elites [18] is a well-known algorithm of this type. Khalifa et al. [8] presented constrained MAP-Elites, a combination MAP-Elites with the feasible-infeasible concept from the FI2Pop genetic algorithm [19], and applied this to procedurally generating levels for bullet hell games.

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool for generating dungeons for adventure games using a FI2Pop evolutionary approach [20–23]. This paper presents the Interactive Constrained MAP-Elites, an implementation of MAP-Elites into EDD’s FI2Pop evolutionary algorithm, as well as introduces a continuous evolution process that takes advantage of MAP-Elites multidimensional discretization of the search space into cells. Results are analyzed and discussed regarding the improvements on quality diversity in the procedurally generated dungeons, as well as the effects of continuous evolution and dimension customization in a MI-CC approach.

Background

Dungeons

For more than 40 years, *dungeons* have frequently been the setting for digital games and provided players with entertainment and excitement in particularly computer role-playing games (CRPGs) and adventure games. It seems that dungeons, as game settings, are as popular as ever, and shows no signs of going away [24]. We can trace the first digital dungeons to the PLATO system back in 1975 [25,26] with games called “*pedit5*” [27] and “*moria*”. Even though the layout of the dungeon in “*pedit5*” was a fixed design, the game contained randomly generated encounters and rewards, making it a predecessor to the more commonly known *Rogue* [2] which provides the player with a new layout of the dungeon with every restart. However, prior to *Rogue*, the game *Beneath Apple Manor* [28] made for the Apple contained a level generator which gave the player the possibility to replay the game with a different layout when starting the game. This feature of dungeons as “randomized environments” is the key element in so called Dungeon Hack games [29].

With regards to CRPGs and adventure games, it should be noted that they share the mechanisms of adventure and exploration whereas combat is more common in CRPG [30]. Adventure games, on the other hand, more often contain puzzle solving as a mechanism. It is perhaps not that strange that dungeons are a popular setting for games in these genres, since they provide the following design elements: levels (several levels are needed with diverse layout and difficulty), collectibles (loot), boss fights, locked door and key (you need to find the key to open the

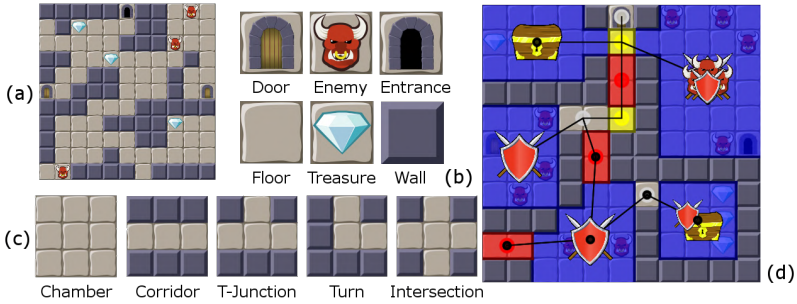


Figure 1: Main components in EDD. (a) Basic room, (b) different placeable tiles, (c) micro-patterns and (d) meso-patterns [21].

door), wildcard enemies (placement, type and strength), monster generators (new monsters are generated until this mechanism is destroyed), and finally, exits and warps (which acts as transitions to other parts indicating progress in the game) [30].

Map-Elites for illuminating search spaces

Quality-diversity algorithms are algorithms which search a space of solutions not just for the single best solution, but for a set of diverse solutions which are good. MAP-Elites maintains of map of good solutions [18] and is perhaps the most well-known quality-diversity algorithm. The map is divided into a number of cells according to one or more feature dimensions (commonly, two dimensions are used). In each cell, a single solution is kept. At every update, an offspring is generated based on one or more existing solutions. That offspring is then assigned to a cell based on its feature dimensions, which might or might not be the same as the cell(s) its parent(s) occupy. If the new offspring has a higher fitness than the existing solution in that cell, it replaces the previous item in the cell. This process results in a map of solutions where each cell contains the best found solution for those particular feature dimensions.

Evolving Dungeons as a Whole, Room by Room

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool that allows a human designer to create a 2D dungeon and its composing rooms (Figure 1.a), being the designer able to manually edit both the dungeon - by placing and removing rooms - and the rooms - by separately editing the tiles (Figure 1.b) that compose each room. EDD's underlying evolutionary algorithm provides procedurally generated suggestions, and is driven through the use of game design micro- and meso- patterns

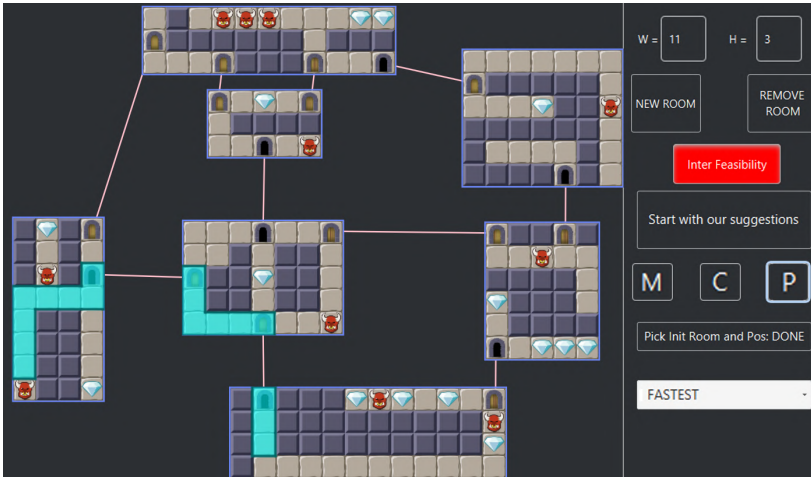


Figure 2: Screenshot of the dungeon editor screen in EDD, displaying a sample dungeon composed by seven rooms. The shortest path between two given tiles is highlighted in blue. The right pane contains all options for editing the dungeon. "M", "C", and "P" stand for "Move rooms", "Connect rooms", and "calculate Path".

(Figure 1.c and Figure 1.d). A detailed description of all EDD's features can be found in [20–23].

This section presents the latest version of EDD¹, which includes significant improvements based on the outcomes from the qualitative analysis discussed in [20]. The most significant upgrade is replacing the grid-based backbone that represented the dungeon by a more flexible graph-based representation. A dungeon is now a graph of interconnected rooms of any given size between 3×3 and 20×20 tiles. The smallest allowed dungeon is composed by two rooms connected once to each other. The designer can perform the following new actions:

- adding disconnected rooms to the dungeon. Rooms may also be removed at any time.
- connecting any pair of rooms by adding a new bi-directional connection to the graph. Rooms interconnect from and to passable border tiles (self-loops are not allowed). Both ends are marked with a door tile (Figure 1.b). A single border tile can only hold one connection, implying that a room can have as many connections as passable border tiles. Connections and rooms can be removed at any time, and their associated doors removed with them.

¹available for download at <https://drive.google.com/file/d/11CUfc40F71Y3vU1PzAqf7i70UfaKQoem/view>

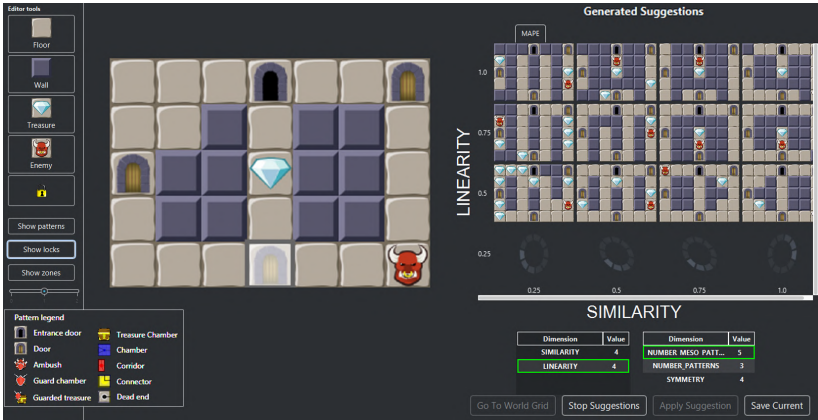


Figure 3: The room editor screen in EDD. The left pane contains all the options for manually editing the room displayed at the center-left of the screen. The right section displays the procedurally generated suggestions.

- calculating paths between any pair of passable tiles located in any connected room. Paths are automatically calculated following one of the following heuristics: *fastest* returns the shortest path, *rewarding* returns the path that traverses the highest number of treasure tiles, *less danger* provides a path with the fewest number of enemies, whereas *more danger* does the opposite.

The designer is required to select one of the added rooms as the *initial room*, which is the room used by the player to enter the dungeon (for the first time). This selection can be modified unlimited times. The *initial room* is used by EDD to calculate the feasibility of the dungeon. A dungeon is considered feasible when there is at least one path between the *initial room* and any other passable tile in every room. Rooms and doors that aren't reachable from the *initial room* are highlighted in red, so that they can be easily identified by the designer. This feasibility constraint ensures that all passable tiles are accessible, avoiding the possibility of accidentally creating unreachable areas.

The mixed-initiative workflow in EDD

The starting screen in EDD is the dungeon editor screen, shown in Figure 2. Every new room is empty (composed solely of floor tiles) when created and is placed detached from the dungeon graph. After manually connecting the room to the dungeon with at least one connection, the designer has the option to populate the room using the room editor screen (Figure 1). This screen can be reached in two different ways:

1. directly: by double-clicking or zooming in (by using the mouse steering wheel or by pinching on the touchpad) on the room.
2. indirectly: by clicking on the "Start with our suggestions" button on the right pane (Figure 2), six procedurally generated suggestions are displayed on a separate screen. The selected suggestion is then opened in the room editor screen.

Figure 1 shows the room editor screen displaying a sample room with the dimensions 7x5 tiles. The left pane lists all the available options for manually editing the room. Manual editing is carried out by brush painting over the room with one of the available tile types: floor, wall, treasure, or enemy. There are two brush sizes (single tile, and five-tile cross shape), and control-clicking allows the designer to bucket paint all adjacent tiles of the same type. Brush painting with the lock button on preserves selected tiles in all the procedurally generated suggestions. A detailed description of all the options in this pane is included in [20, 21].

The right side of the screen displays the procedurally generated suggestions, by means of the Interactive Constrained MAP-Elites genetic algorithm (Part II). The "Generate/Stop Suggestions" button at the bottom toggles this algorithm on and off. Once started, the algorithm continuously populates the suggestions pane with new optimal individuals. The evolutionary process is fed with the manually edited room, so that every change in the room affects the generated suggestions. By clicking on "Apply Suggestion", the manually edited room is replaced by the selected suggestion, thus affecting the upcoming procedural suggestions. "Go To World Grid" takes the user back to the dungeon editor screen.

Interactive Constrained MAP-Elites

EDD uses a single-objective fitness function with a FI2Pop genetic algorithm where fitness is a weighted sum divided equally between (1) the inventorial aspect of the rooms, which relates to the placement of enemies and treasures in relation to doors and target ratios, and (2) the spatial distribution of the design patterns, which relates to the distribution between corridors and rooms, and the meso-patterns that those encompass. An in-depth explanation of EDD's fitness function can be found in [21, 22].

The overarching goal of MI-CC is to collaborate with the user to produce content, either to optimize (i.e. exploit) their current design towards some goal or to foster (i.e. explore) their creativity by surprising them with diverse proposals. By implementing MAP-Elites [18] and continuous evolution into EDD, our algorithm

can (1) account for the many dimensions that a user can be interested, (2) explore multiple areas of the search space and produce a diverse amount of high-quality suggestions to the user, and (3) still evaluate how interesting and useful the tile distribution is within a specific room. Henceforth, we name the presented approach **Interactive Constrained MAP-Elites** (IC MAP-Elites).

Illuminating Dungeon Populations with MAP-Elites

MAP-Elites explores the search space more vastly by separating certain interesting dimensions, that affect different aspects of the room such as playability or visual aesthetics, from the fitness function, using them to categorize rooms into niches (cells).

Dimensions

Dimensions in MAP-Elites are identified as those aspects of the individuals that can be calculated in the behavioral space, and that are independent of the fitness calculation. EDD offers the designer the possibility to choose among the following dimensions, two at a time:

Symmetry and Similarity. We choose Symmetry as a consideration of the aesthetic aspects of the edited room since symmetric structures tend to be more visually pleasing. Similarity is used to present the user variations of their design but still preserving their aesthetical edits. Symmetry is evaluated along the X and Y axes, backslash and front slash diagonal and the highest value is used as to how symmetric a room is. Similarity is calculated through comparing tile by tile with the target room. Formulas, information and support for both evaluations are explained in greater details at [21], where both of them were used as aesthetic fitness evaluations.

Number of Meso-patterns. The number of meso-patterns correlates to the type and amount of encounters the designer wants the user to have in the room in a more ordered manner. The considered patterns are the treasure room (tr), guard rooms (gr), and ambushes (amb). Meso-patterns associate utility to a set of tiles in the room, for instance, a long chamber filled with enemies and treasures could be divided into 2 chambers, the first one with enemies and the second one with treasures so the risk-reward encounter is more understandable for the player. Since we already analyze the rooms for all possible patterns, the number of meso-patterns is simply $\#MesoPat = tr, gr, amb \in AllPatterns$. Equation (4) presents the dimensional value, and since the used meso-patterns can only exist in a chamber, we normalize by the maximum amount of chambers in a room, which are of a minimum size of 3×3 , and results in $Max_{chambers} = \lfloor Cols/3 \rfloor * \lfloor Rows/3 \rfloor$.

$$D_{mesoPat} = \min \left\{ \frac{\#MesoPat}{Maxchambers}, 1.0 \right\} \quad (1)$$

Number of Spatial-patterns. By spatial-patterns we mean chambers (c), corridors (cor), connectors (con), and nothing (n). We identify the number of spatial-pattern relates to how individual tiles group (or not) together to form spatial structures in the room. The higher the amount of spatial-patterns the lesser tiles will be group together in favor of more individualism. For instance, a room with one spatial-pattern can be one with no walls and just an open chamber, while a room with a higher number of spatial-patterns would subdivide the space with walls, using tiles for more specific patterns. Equation (5) presents how we calculate the value for such a dimension. The number of spatial patterns is simply $\#SpatialPat = c, n, cor, con \in AllPatterns$, we then normalize it by the largest side of the room and multiply it by a constant value, determined as $K = 4.0$ through a process of experimentation since it resulted in a good estimation of the amount of spatial patterns in the room.

$$D_{spatialPat} = \min \left\{ \frac{\#SpatialPat}{\max \{Cols, Rows\} * K}, 1.0 \right\} \quad (2)$$

Linearity. Linearity represents the number of paths that exist between the doors in the room. This relates to the type of gameplay the designer would like the room to have by the distributions of walls among the room. Having high linearity in a room does not need to only be by having a narrow corridor between doors but could also be generated by having all doors in the same open space (i.e. the user would not need to traverse other areas) or by simply disconnecting all paths between doors. Equation (6) shows the linearity calculation. Due to the use of patterns, we calculate the paths between doors as the number of paths that exist from a spatial-pattern containing a door to another. Finally, this is normalized by the number of spatial patterns in combination with the number of doors and their possible neighbors.

$$D_{lin} = 1 - \frac{AllPathsBetweenDoors}{\#spatialPat + \#NeighborsPerDoor} \quad (3)$$

Continuous Evolution

EDD implements continuous evolution in two ways. First, the EA constantly updates the target room and configuration with the most recent version of the user's design, and once the suggestions are broadcasted, that room is incorporated without changes to the population of individuals in the corresponding cell. Secondly, by changing the dimension information and their granularity for the MAP-Elites, which can be done at any given time by the designer.

Provided that EDD already uses a FI2Pop, we took as a starting point the constrained MAP-Elites presented by Khalifa et al. [8], where the illuminating capabilities of MAP-Elites explore the search space with the constraints aspects of FI2Pop. This approach manages two different populations within each cell, a feasible and an infeasible one. Individuals move across cells when their dimension values change, or between the feasible and infeasible population according to their fulfillment of the feasibility constraint.

Algorithm 2 Interactive Constrained MAP-Elites

```

1: procedure IC-MAP-ELITES( $\{\{d_1, v_1\}, \dots, \{d_n, v_n\}\}$ )
2:    $target \leftarrow curEditRoom$  ▷ Always in background
3:   createCells( $\{\{d_1, v_1\}, \dots, \{d_n, v_n\}\}$ )
4:   for  $i \leftarrow 1$  to  $PopSize$  do
5:     add mutate( $target$ ) to  $population$ 
6:   CheckAndAssignToCell( $population$ )
7:   while true do ▷ start continuous evo
8:     for  $generation \leftarrow 1$  to  $publishGen$  do
9:       if  $dimensionsChanged$  then
10:         $previousPop \leftarrow cells_{pop}$ 
11:        createCells( $newDimensions$ )
12:        checkAndAssignToCell( $previousPop$ )
13:       repeat [for feasible & infeasible pop.]
14:         for  $i \leftarrow 1$  to  $ParentIteration$  do
15:            $curCell \leftarrow rndCell(cells)$ 
16:           add tournament( $curCell$ ) to  $parent$ 
17:            $offspring \leftarrow crossover(Parent)$ 
18:           checkAndAssignToCell( $offspring$ )
19:         sortAndTrim( $cells$ )
20:       broadcastElites() ▷ render elites
21:        $pop' \leftarrow cells_{population}$ 
22:       add mutate( $cells_{pop}$ ) to  $pop'$ 
23:       add  $target$  to  $pop'$ 
24:       checkAndAssignToCell( $pop'$ )
25:       sortAndTrim( $cells$ )
26: procedure CREATECELLS(DIMENSIONS)
27:   for each  $dim \in dimensions$  do
28:     add newCell( $dim_d, dim_v$ ) to  $cells$ 
29: procedure CHECK&ASSIGNTOCELL( $curPopulation$ )
30:   for each  $individual \in curPopulation$  do
31:      $individual_f \leftarrow evaluate(individual)$ 
32:      $individual_d \leftarrow dim(individual)$ 
33:     add  $individual$  to  $cell_{pop}(individual_d)$ 

```

Algorithm

The current evolutionary algorithm is depicted in Algorithm 3. Cells are first created based on the dimensions selected by the user and proceed to initialize the population based on the user's design, evaluate it and assign each individual to the corresponding cell. Before starting each generation, we check if the dimensions have

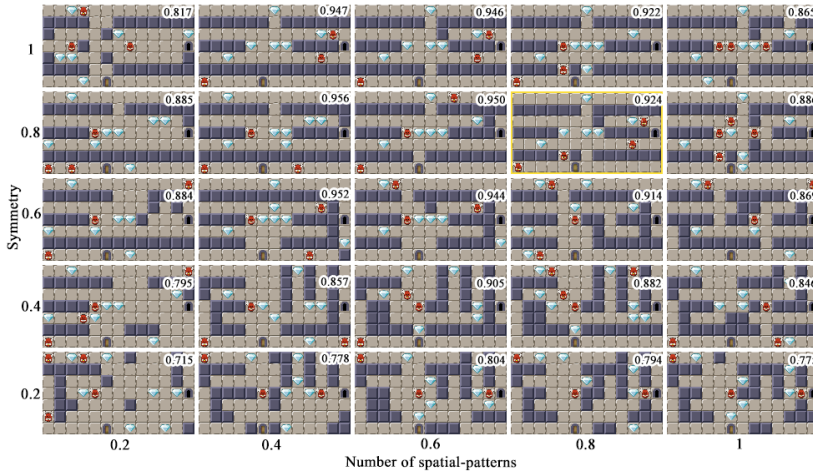


Figure 4: Rooms at generation 2090 targeting Number of spatial-patterns (X) and Symmetry (Y). Each cell displays (top-right) the fitness of the optimal individual in its related feasible population.

changed, and if so, recreate the cells and populate them with the previous individuals, and proceed through the evolutionary strategies. Selection is through tournament with a random number of competing parents and offspring are produced through a two-point uniform crossover with a chance of mutation. Offspring are placed in the correct cell and population after calculating their fitness and dimension's information. Finally, cells eliminate the low-performing individuals that over-cap their maximum capacity. Since interbreeding is not allowed, and can only happen indirectly (i.e. the offspring changing population and then used for breeding in consequent generations), the strategies are repeated for each of the population.

This procedure is repeated until the user decides to stop the algorithm. Meanwhile, the EA runs for n generations, and once it reaches the specified limit, it broadcasts the found elites. In order to push the exploration, we first mutate all the individuals from all the populations and cells (while retaining the previous population), and add them into the same pool together with the current edited room without changes. Finally, we evaluate and assign all the individuals to the correct cells, and cells that are over maximum capacity eliminates low-performing individuals.

Experiments

We ran a set of experiments to test the results from the IC MAP-Elites using all possible combinations of the five available dimensions using two dimensions at a time. All experiments were run using 13×7 rooms, the same room size as in *The Binding of Isaac* [31], a representative example of a dungeon based adventure game.

In each experiment, the initial population was set to 1000 mutated individuals distributed in feasible and infeasible populations in all cells which were set to a maximum capacity of 25 individuals each. The EA ran continuously, every 100 generations rendered the most prominent cells, and at each of the generations, it selected 5 parents per population among the different cells. Offsprings were produced through a two-point crossover, and were mutated with a 30% chance.

Designer Personas describes the results achieved and analyzes them in terms of the quality diversity of the suggestions obtained, the existing correlations found between each pair of dimensions, as well as the effects of integrating the MAP-Elites approach into a continuously evolving environment.

Results and Discussion

Figure 3 shows a grid containing the best found suggestions at generation 2090, while aiming for number of spatial-patterns at the X-axis and symmetry at the Y-axis with a granularity of 5. Each cell displays the optimal individual of the feasible population under a given pair of dimension values. The fitness score is displayed on the cells' top-right corner.

The fitness evaluation in IC MAP-Elites is quite lightweight in terms of computational cost, so that the grid of suggestions is completed in a matter of seconds. This is of key importance for successfully implementing continuous evolution, so that the influence of each manual change in the edited rooms is reflected in the suggestions almost instantly. The feeling of immediacy is further increased through updating cells as soon as a new optimal individual is produced and incorporated to the cell's underlying feasible population.

Results in Figure 3 are representative of the good quality diversity solutions produced by EDD. The average fitness across cells is 0.872, and the highest fitness is 0.956 (cell [0.4, 0.8]). No two rooms are the same. As intended, high levels of symmetry are displayed in the upper rows, gradually decreasing towards the bottom row. Similarly, rooms in the leftmost column contain lower amounts of spatial patterns, increasing towards the rightmost column. Lower amounts of spatial patterns translate into more open rooms with almost no corridors and one or two large adjacent chambers (as in cell [0.2, 0.2]), as opposed to highly pattern filled rooms that comprise intricate pathways converging at one or two small chambers (cell [1, 0.2]). Fitness values show that some dimension combinations are harder to optimize than others, so that the whole grid depicts a gradient landscape of the compatibility between each pair of dimensions.

The bottom-left corner in Figure 3 shows difficulties producing symmetric rooms

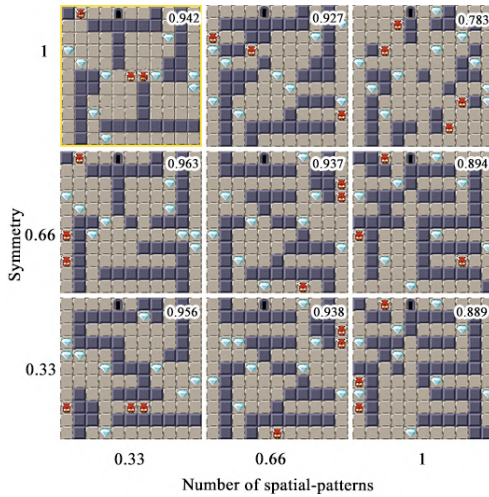


Figure 5: Rooms at generation 5303 targeting the same dimensions as in Figure 3, but with the size 11×11 instead.

with low amounts of spatial patterns, as opposed to rooms with many corridors (upper-right corner), which seem to favor the generation of symmetrical structures. The bottom row shows that aiming for low symmetry generally produces slightly less optimal results, whereas the top row shows that corridors are the most favorable spatial pattern for building symmetric rectangular rooms. Additional experiments (Figure 5) show that medium-large square rooms favor the appearance of chambers in combination with corridors for achieving symmetric rooms, thus revealing that squareness and size are important factors for the appearance of chambers in symmetric rooms.

Figure 6 contains the rooms generated at generation 7088 while targeting number of meso-patterns at the X -axis and symmetry at the Y -axis. The top-right cell is empty because its related feasible and infeasible populations are empty, that is, no individuals with value 1 for both dimensions have been found. The number of empty cells in the earlier generations 3722, 3875, and 5864 were 8, 7, and 2, respectively, indicating that some dimensional values for meso-patterns and symmetry take longer to converge. The continuous nature of IC MAP-Elites fills out the initially empty cells while the designer works with the already generated suggestions. The right half of the grid shows that a combination of small chambers and short corridors favors the appearance of multiple meso-patterns, such as treasure chambers, guarded chambers, and ambushes.

Figures 7 and 8 show how low valued linearity does not cope well with neither

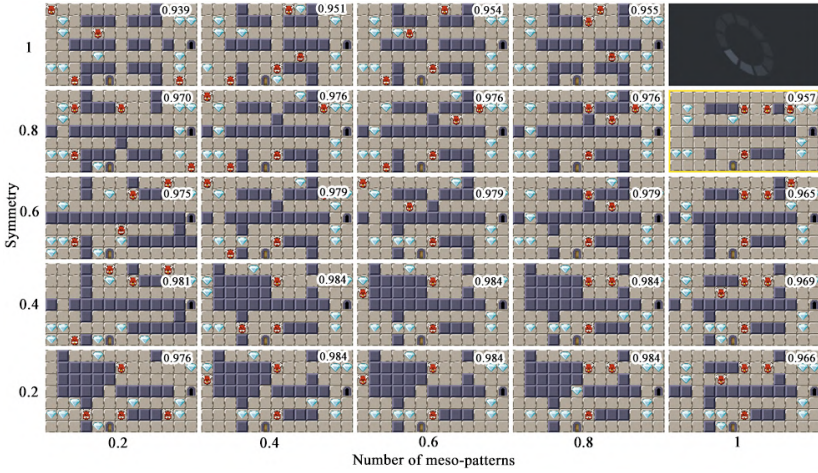


Figure 6: Rooms at generation 7088 targeting Number of meso-patterns at the X-axis and Symmetry at the Y-axis. The top-right cell shows that no optimal room could be generated under dimension values [1, 1].

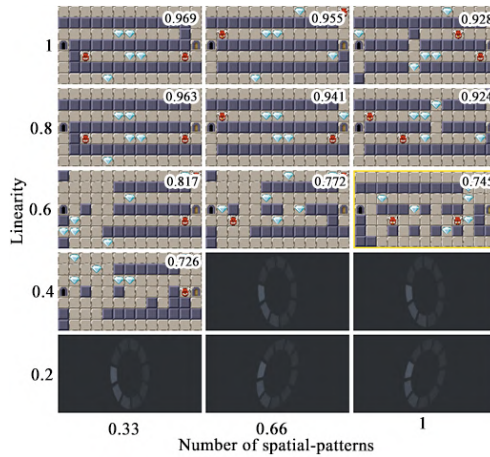


Figure 7: Rooms at generation 12545 targeting Number of spatial-patterns at the X-axis and Linearity at the Y-axis.

spatial- nor meso-patterns. High linearity tends to create one single pathway, either one long corridor or a wide chamber, that connects the doors in the room. Low linearity results in the opposite, scattering multiple small passages that increase the connectivity between doors but do not count neither as spatial- nor as meso-patterns.

Due to its nature, the performance of similarity in combination with other dimensions has been found to be very dependant on the characteristics already present in

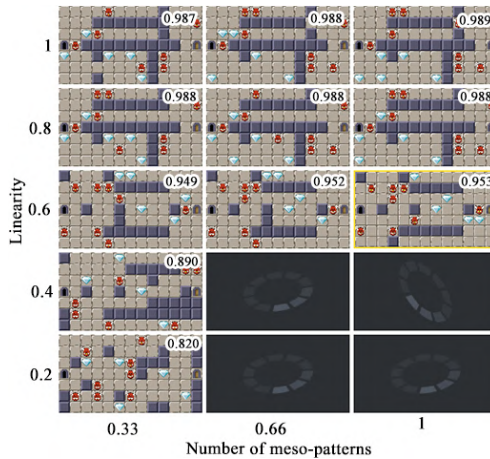


Figure 8: Rooms at generation 20348 targeting Number of meso-patterns at the X-axis and Linearity at the Y-axis.

the manually edited room. I.e., if this room is already highly symmetric, EDD has problems at preserving similarity while targeting low values of symmetry. This behavior is reported when combining similarity with the other dimensions.

Conclusions and Future Work

We have presented the Interactive Constrained MAP-Elites, a continuous implementation of MAP-Elites into the Evolutionary Dungeon Designer, creating a MI-CC tool where the users influence the EA through their design, as well as by choosing which dimensions to explore and the granularity of such.

The presented approach allows the designer to have a fast interaction with the EA through re-targeting and re-scaling the dimensions at will and at any moment. The continuous evolution fits perfectly to the mixed-initiative approach, providing a dynamic search that reacts on the fly to the different interactions of the user, as well as constantly offering new suggestions accordingly. Moreover, mixed-initiative fills the lapses between generations by inviting the designer to permeate the suggestions with custom aesthetics, challenges, paths, and other design decisions. Results show that this approach creates a very fluent workflow of mutual inspiration between designer and tool, yet offering highly customized quality diversity procedural suggestions.

Results also allowed us to study the compatibility between each pair of dimensions, spotting existing correlations among them and with the fitness function, as well as compatibility pitfalls that leave room for further analysis.

We aim to validate IC MAP-Elites with a user study, as well as to explore alternatives to visualize higher dimensions through the use of CVT-MAP-Elites [32] and Cluster MAP-Elites [33], analyze the effect of including more dimensions, and performing agent-based dungeon evaluation to improve the fitness calculation by incorporating automatic gameplay data.

Acknowledgement

The Evolutionary Dungeon Designer is part of the project *The Evolutionary World Designer*, which is supported by The Crafoord Foundation.

References

- [1] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018. 145
- [2] M. Toy, “Rogue,” 1980. 145, 146
- [3] D. Braben and I. Bell, “Elite,” 1984. 145
- [4] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 172–186, Sept. 2011. 145
- [5] N. Shaker, G. N. Yannakakis, J. Togelius, M. Nicolau, and M. O’neill, “Evolving personalized content for super mario bros using grammatical evolution.,” in *AIIDE*, 2012. 145
- [6] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the Galactic Arms Race video game,” in *2009 IEEE Symposium on Computational Intelligence and Games*, Sept. 2009. 145
- [7] J. Dormans, “Unexplored,” 2017. 145
- [8] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, pp. 1047–1054, ACM, 2018. 145, 146, 153
- [9] M. C. Green, G. A. B. Barros, A. Liapis, and J. Togelius, “Data agent,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, (New York, NY, USA), pp. 19:1–19:10, ACM, 2018. 145

- [10] A. Liapis, G. N. Yannakakis, and J. Togelius, “Designer modeling for sentient sketchbook,” in *Proc. IEEE Conf. Computational Intelligence and Games*, pp. 1–8, Aug. 2014. 145
- [11] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 145
- [12] N. Shaker, M. Shaker, and J. Togelius, “Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.,” in *AIIDE*, 2013. 145
- [13] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 145
- [14] T. Machado, A. Nealen, and J. Togelius, “Cicero: Computationally intelligent collaborative environment for game and level design,” in *3rd workshop on Computational Creativity and Games (CCGW) at the 8th International Conference on Computational Creativity (ICCC’17)*, 2017. 145
- [15] A. Liapis, G. N. Yannakakis, and J. Togelius, “Generating Map Sketches for Strategy Games,” in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS, Springer, 2013. 145
- [16] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, “Can computers foster human users’ creativity? theory and praxis of mixed-initiative co-creativity,” *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016. 145
- [17] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016. 145
- [18] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015. 146, 147, 150
- [19] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008. 146
- [20] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,”

- in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG '18, (New York, NY, USA), pp. 50:1–50:8, ACM, 2018. 146, 148, 150
- [21] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG '18, (New York, NY, USA), pp. 44:1–44:4, ACM, 2018. 146, 147, 148, 150, 151
- [22] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, (New York, NY, USA), pp. 74:1–74:10, ACM, 2017. 146, 148, 150
- [23] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Mixed-initiative procedural generation of dungeons using game design patterns,” in *Proc. IEEE Conf. Computational Intelligence and Games (CIG)*, pp. 25–32, Aug. 2017. 146, 148
- [24] C. W. Totten, *Level Design: Processes and Experiences*. CRC Press, 2017. 146
- [25] M. Barton, *Dungeons and Desktops: The History of Computer Role-playing Games*. A K Peters Ltd, 2008. 146
- [26] N. Brewer, “Going Rogue: A Brief History of the Computerized Dungeon Crawl.” accessed 2019, March 20. 146
- [27] R. Rutherford, “pedit5, alternately called The Dungeon,” 1975. 146
- [28] D. Worth, “Beneath Apple Manor,” 1978. 146
- [29] C. Bateman and R. Boon, *21st Century Game Design*. Charles River Media, 2006. 146
- [30] A. Rollings and E. Adams, *Andrew Rollings and Ernest Adams on Game Design*. New Riders Publishing, 2003. 146, 147
- [31] E. McMillen and F. Himsl, “The Binding of Isaac,” 2011. 154
- [32] V. Vassiliades, K. I. Chatzilygeroudis, and J. Mouret, “Scaling up map-elites using centroidal voronoi tessellations,” *CoRR*, vol. abs/1610.05729, 2016. 159

- [33] V. Vassiliades, K. Chatzilygeroudis, and J.-B. Mouret, “A comparison of illumination algorithms in unbounded spaces,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, (New York, NY, USA), pp. 1578–1581, ACM, 2017. 159

PAPER IV - LEARNING THE DESIGNER'S PREFERENCES TO DRIVE EVOLUTION

Alberto Alvarez and Jose Font

ABSTRACT

This paper presents the Designer Preference Model, a data-driven solution that pursues to learn from user generated data in a Quality-Diversity Mixed-Initiative Co-Creativity (QD MI-CC) tool, with the aims of modelling the user's design style to better assess the tool's procedurally generated content with respect to that user's preferences. Through this approach, we aim for increasing the user's agency over the generated content in a way that neither stalls the user-tool reciprocal stimuli loop nor fatigues the user with periodical suggestion handpicking. We describe the details of this novel solution, as well as its implementation in the MI-CC tool the Evolutionary Dungeon Designer. We present and discuss our findings out of the initial tests carried out, spotting the open challenges for this combined line of research that integrates MI-CC with Procedural Content Generation through Machine Learning.

PUBLISHED IN

Proceedings of the 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation, EvoApplications '20, Springer, 2020

LEARNING THE DESIGNER'S PREFERENCES TO DRIVE EVOLUTION

Introduction

As game production grows, so does the usage of computer-aided design (CAD) tools to develop various facets of games. CAD tools enable users to create new content or refine previously created content with the assistance of some type of technology that focuses on reducing the workload of the developer. Procedural Content Generation (PCG) denotes the use of algorithms to generate different types of game content, such as levels, narrative, visuals, or even game rules, with limited human input [1]. Search-based PCG is the subset of techniques whose approach generates content by using a search algorithm, a content representation mechanism, and a set of evaluation functions to drive the content creation process towards near-optimal solutions [2].

Mixed-initiative co-creativity (MI-CC) [3] is a branch of PCG through which a computer and a human user create content by engaging into an iterative reciprocal stimuli loop [4–9]. This approach addresses the design process with insight and understanding of the affordances and constraints of the human process for creating and designing games [10]. MI-CC helps designers to either optimize their current design towards a specific goal (thus exploiting the search space) or foster their creativity by proposing unexpected suggestions (exploring the search space). To these ends, diversity has been an important feature for the research community to focus on during the past decade, including novelty search [11], surprise [12], curiosity [13] and, more recently, quality-diversity approaches [14].

PCG through Quality-Diversity (PCG-QD) [15] is a subset of search-based PCG, which uses quality-diversity algorithms [16] to explore the search space and produce high quality and diverse suggestions. MAP-Elites [17] is a successful quality-diversity algorithm that maintains a map of good suggestions distributed along several feature dimensions. A constrained MAP-Elites implementation was presented by Khalifa et al. [14], combining MAP-Elites with a feasible-infeasible

(FI2Pop) genetic algorithm [18] for the procedural generation of levels for bullet hell games. The first implementation of a PCG-QD algorithm for MI-CC was presented by Alvarez et al. [19], elaborating on the combined MAP-Elites and FI2Pop approach by introducing a continuous evolution process that benefits from the multidimensional discretization of the search space performed in MAP-Elites.

In all the above MI-CC approaches, the designers play an active role in the procedurally generated content while struggling between the expressiveness of the automatic generation and the control that they want to exert over it [20]. Having this as motivation, this paper takes the work in [19] one step forward by adding an underlying interactive PCG via machine learning algorithm [21], the Designer Preference Model, that models the user's design style, to be able to predict future designer's choices and thus, driving the content generation with a combination of the designer's subjectivity and the search for quality-diverse content.

Previous work

Mixed-Initiative Co-Creativity

Similar to user or player modeling, designer modeling for content creation tools (CAD and MI-CC tools) was suggested by Liapis et al [22], where it is proposed the use of designers models that capture their styles, preferences, goals, intentions, and interaction processes. In their work, they suggest methods, indications, and advice on how each part can be model to be integrated into a holistic designer model, and how each game facet can use and benefit from designer modeling. Moreover, in [23] the same authors discuss their implementation of designer modeling and the challenges of integrating all together in their MI-CC tool, Sentient Sketchbook, which had a positive outcome on the adaptation of the tool towards individual "artificial" users.

Furthermore, Lehman et al [24] presented Innovation Engines that combine the capabilities and advantages of machine learning and evolutionary algorithms to produce novel 3D graphics with the use of Compositional Pattern-Producing Networks (CPPN) evolved with MAP-Elites, and evaluated by the confidence a deep neural network had on the models belonging to a specific object category.

Procedural Content Generation via Machine Learning

Summerville et al. [21] define Procedural Content Generation via Machine Learning (PCGML) as the generation of game content by models that have been trained on existing game content. The main approaches to PCGML are: autonomous content generation, content repair, content critique, data compression, and mixed-initiative

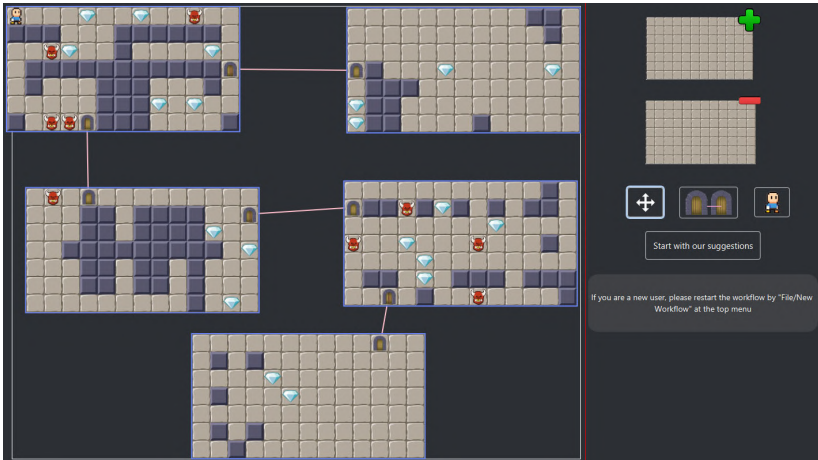


Figure 1: Screenshot of the dungeon editor screen in EDD, displaying a sample dungeon composed by five rooms.

design.

In the latter case and, as appointed by Treanor et al. [25], AI may engage with a human user participating in the creation of content, so that new gameplay emerges from this shared construction. This emerging relationship between the user and the AI system, when implemented through a trained machine learning algorithm, has the potential to reduce user frustration, error, and training time. This is due to the capacity of a machine learning solution to adapt to the design preferences of the user that interacts with the MI-CC tool by learning from the user-generated dataset of previous choices.

The Evolutionary Dungeon Designer

The Evolutionary Dungeon Designer (EDD) is an MI-CC tool for designers to build 2D dungeons. EDD allows designers to manually edit the overall dungeon and its composing rooms (see Figure 1), as well as to use procedurally generated suggestions either as inspiration to work on or as a finished design (see Figure 2). Both options fluently alternate during the creation process by means of a workflow of mutual inspiration, through which all manual editions performed by the user are fed into the underlying continuous Evolutionary Algorithm, accommodating them into the procedural suggestions. A detailed description of EDD and its features can be found in [20, 26–28].

Subsequent user studies [20, 28] carried out with game designers on EDD raised the following areas of improvement: (1) the designers struggled with EDD’s capability

of understanding the designer's intentions and preserving custom designs; (2) the tool was unable to generate aesthetically pleasing suggestions since the fitness function only accounted for functionality, but not aesthetics, of design patterns; (3) the designers wanted to keep certain manual editions from being altered by the procedural suggestions.

With the aims of addressing these limitations as well as fostering the user's creativity with quality-diverse proposals, EDD was improved with the Interactive Constrained MAP-Elites (IC MAP-Elites) [19], an implementation of MAP-Elites into the continuous evolutionary process in EDD. With this addition, the user drives the generation of procedural suggestions by modifying at any moment the areas of the search space where the evolution should put the focus on. This is done by selecting among the available dimensions: symmetry, similarity, design patterns, linearity, and leniency. Additionally, the designers have now the chance to limit the search space by locking map areas and thus preserving manually edited content.

This paper contributes by building on top of EDD's IC MAP-Elites, adding a data-driven Designer Preference Model that adapts and personalizes the design experience, as well as balances the expressivity of the tool and the controllability of the designer over the tool. Other researchers have pursued a similar goal by biasing the search space through having the user perform a manual selection after every given number of generations [11, 29, 30]. Nevertheless, this approach leads to an increase in user fatigue by repeatedly asking for user input and thus, stalling the evolutionary process until such input is received. Moreover, this staged process seems incompatible with the dynamic reciprocal workflow of MI-CC tools, where the focus is on the designer proactively creating content rather than passively browsing a set of suggestions.

The remaining sections of the paper are structured as follows: Section 3 describes the data-driven Designer Preference Model; Section 4 presents the initial experimental results, and Section 5 discusses the results and future lines of research of this novel approach.

Designer Preference Model

The Designer Preference Model is a data-driven intelligent system that learns the user's design style by training and testing over a continuously growing dataset composed of the user's actions and choices while operating EDD. The underlying evolutionary algorithm (EA) uses this model to assess the generated suggestions according to the predicted preference of the designer. This is a complementary assessment to EDD's original fitness function, which evaluates individuals first

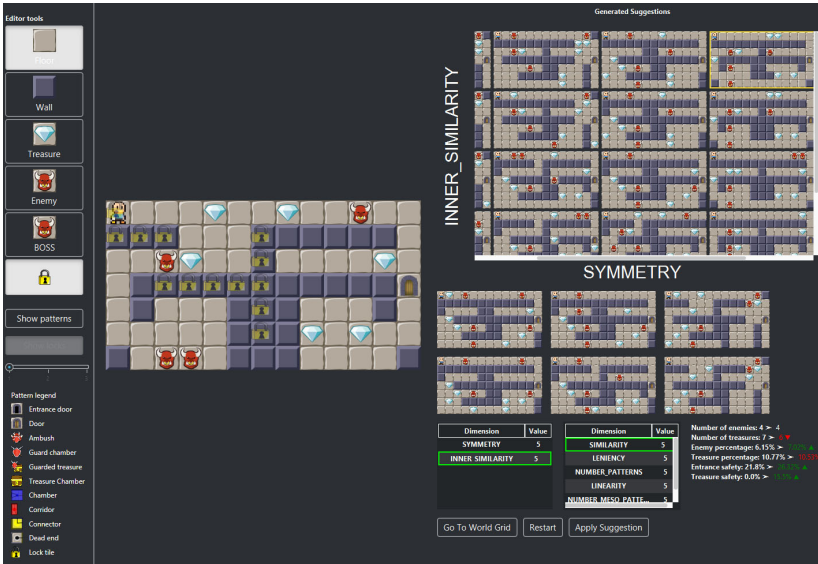


Figure 2: The room editor screen in EDD. The top-right pane shows the suggestions provided by the IC MAP-Elites algorithm. Below are the six top-ranked suggestions by the Designer Preference Model. The left pane contains the manual edition features.

based on the presence and distribution of spatial and meso-patterns (Figure 3), and then based on their degree of adaptation to the user-selected quality-diversity dimensions [19]. The relevance of the Designer Preference Model gradually increases over EDD’s fitness function as long as the model gains confidence in its assessments.

Model Update and Usage

The proposed model is a relatively small neural network M with as many input neurons as the number of tiles composing each room, two hidden layers (100 and 50 neurons respectively), and six output neurons, one per each discrete preference value assigned to the individuals by the designer. When the designer starts EDD, the neural network is created with random initialization and without any prior training (i.e. cold start). While the designer creates and modifies rooms, on the background, the EA produces and presents individuals to the designer using the MAP-Elite’s cells (Figure 2), while it adapts to the designer’s design. Following a proactive learning approach [31], anytime the designer chooses one suggestion to replace her current design, a training session is requested for a model M with a dataset S created with the current cells and their populations based on the designer’s chosen suggestion. The loop, depicted in figure 4, can be described in the following two

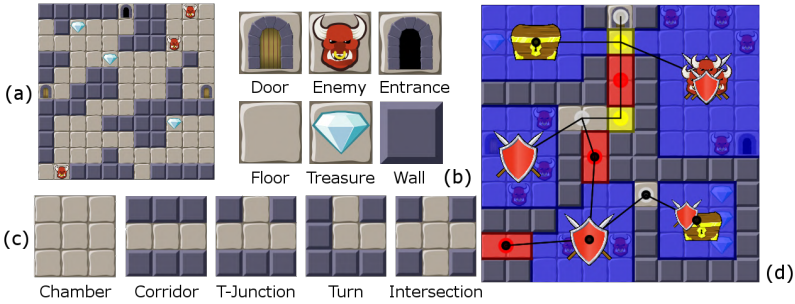


Figure 3: A sample room in EDD (a) composed by tiles (b), spatial patterns (c) and meso-patterns (d). Detailed descriptions for these components can be found in [27].

steps:

Dataset creation:

The designer chooses a suggestion to replace her current design, which in turn, requests a training session using all the current individuals (i.e. the elites and the rest of the feasible populations) to create a new dataset to train the model closer to the “actual” preference of the designer. As shown in figure 4.b, an ad-hoc matrix is created, based on the position of the applied suggestion, to calculate the estimated preference, starting with the applied suggestion (1.0 preference value), and reducing the preference value by 0.2 per each step that was taken away from the applied suggestion in the matrix until a minimum of 0.0.

Once all the individuals are given an estimated preference value based on their grid position by the ad-hoc matrix, they are all used to compose a general dataset S where each individual is transformed to match the network input. Finally, we divide the set into a training set (90%) and test set (10%) with the same label distribution. Through this process, we end up having a maximum of $M \times N \times feasible_population$ tuples, which relates to the granularity of each presented dimension times the maximum amount of feasible individuals per cell.

Training and usage:

The model is then trained for a limited set of epochs (i.e. 20 epochs) and later incorporated into the evolutionary loop to further evaluate individuals. As mentioned above, the model tries to slowly fit towards the designer’s preference, and as it becomes more confident in predictions, the more weight W_1 it has in the final fitness of an individual. Confidence is calculated based on the output of the softmax layer, which limits the output of all the neurons into the range 0 to 1, as

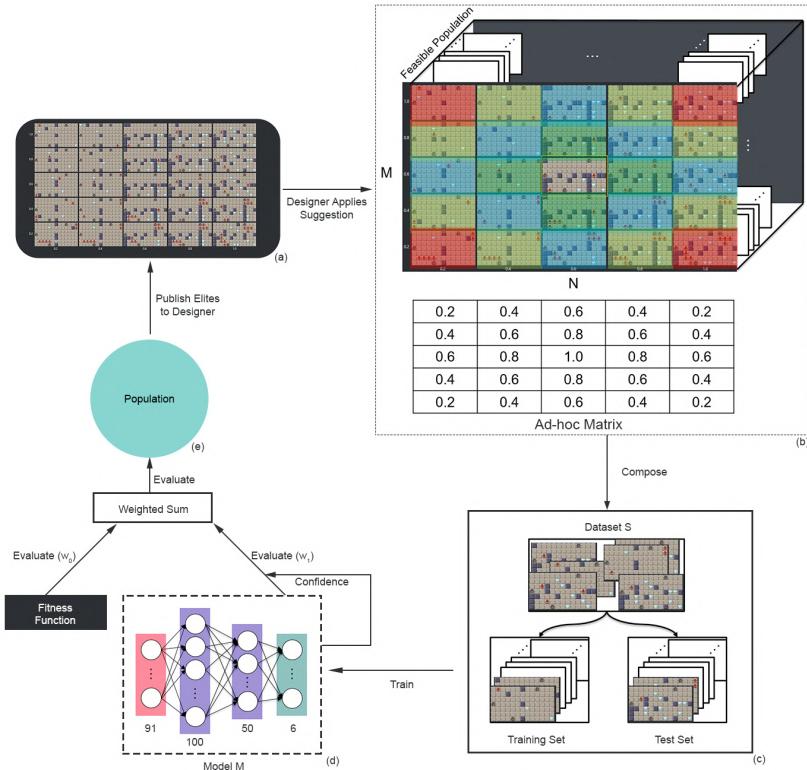


Figure 4: Overview of the Designer Preference Model integrated into the fitness function of EDD. Elites are published and shown to the designer in a grid fashion (a), and once the designer chooses and applies one of the suggestions, an ad-hoc matrix is created based on the position of the selected suggestion to estimate the preference of suggestions (b). The ad-hoc matrix is then applied to all the elites in the grid, and the feasible populations within the EA cells to compose a general dataset S with rooms labeled by the estimated preference. The composed dataset S is then subdivided into a training set (90%) and test set (10%), both with the same label distribution (c). The dataset is used to train a model M , which is a relatively small neural network, for 20 epochs (d). The model is then used to evaluate the population of the EA together with the current fitness function in a weighted sum, with the weight of the model M conditioned by the confidence of the network (e).

the sum of all the neurons' output must be 1.0. This characteristic of the softmax layer enables us to interpret the results as the probabilities for each of the classes. For instance, if the network predicts that an individual is going to be preferred to the designer with a 1.0 preference with a probability of 0.9, it means that the remaining 0.1 is distributed among the other output classes, and as a consequence, the network has high confidence. The resulting weights (Eq. 1) and weighted sum (Eq. 2) to evaluate each of the individuals in the EA were the following:

$$\begin{aligned} w_1 &= \min(M_{conf} \cdot M_{TestAcc}, 0.5), \\ w_0 &= 1.0 - w_1 \end{aligned} \quad (1)$$

$$weightedSum = (w_0 \cdot objective) + (w_1 \cdot predicted_{pref}) \quad (2)$$

Finally, the loop continues and the model awaits for the next training session that will be triggered the next time that the user applies a suggestion. In the meantime, the trained model is used as part of the combined individual evaluation process.

Evaluation

Model performance, integration, and setup

We conducted a set of experiments to test the extent to which the Designer Preference Model learns from the user-generated data and fits into the previously existing MI-CC workflow in EDD. These experiments also aimed for finding the hyperparameter configuration for the model that better suited its goals.

This resulted in a fully connected neural network with two hidden layers with 100 and 50 neurons respectively. Bigger and deeper networks, as well as longer training epochs, did result in higher accuracy but it was not worth the time-complexity/accuracy tradeoff since it obstructed the dynamic and high-paced workflow of the tool. Finally, the network had six output nodes related to the different preference values a suggestion could have (i.e. from 0.0 to 1.0 in 0.2 intervals, both ends inclusive) with a softmax layer, which was used to account for the confidence on the network.

Additionally, we decided to train the model's network under independent episodes every time the designer applied a suggestion using the most up-to-date data (the dataset that was created each time a selection was applied). We evaluated and through experimentation later discarded a more continuous approach, since continuously training between episodes led to the generation of large noisy datasets that distorted the training process.

As a result, the Designer Preference Model is smoothly integrated into EDD's workflow. User-wise, it runs in a completely transparent way, neither breaking the reciprocal stimuli loop nor slowing down the performance of the EA in a perceptible way.

User Study

A user study was also conducted to collect preliminary results that assess the relevance of the Designer Preference Model. We aimed for gathering feedback from game designers on how the model would be used, as well as their perception of the adaptive capabilities of the model.

Fifteen game design students (i.e. novice designers) participated in the study; all of them were introduced to all the features of the tool and were tasked to create a dungeon with interconnected rooms for as long as they were satisfied with their design. At the end of each test session, the participants were asked to fill a brief questionnaire assessing their understanding of the suggestions, its usability, pros, and constraints.

For the purposes of the user study and to test the new model's assessment capabilities in contrast to EDD's original fitness function, we presented the suggestions as displayed in Figure 2. The top-right pane displays EDD's IC-MAP-Elites as described in [19]. The bottom-right pane shows a smaller grid displaying the top ranked individuals assessed by the Designer Preference Model. As the designer applied the top suggestions, the lower grid would get trained with the expected preference, as explained in section II and, as a consequence, the lower grid would become more adapted.

This system was designed to validate the hypothesis that users would prefer to make use of the suggestions in the bottom-right pane in the long run, after the Designer Preference Model had been trained a sufficient amount of times, thus gaining confidence in its assessment. A total of 105 rooms were created and the designers applied 43 times suggestions to their designs, with most of the cases happening once the designers had manually created most of the dungeon. Unfortunately, this did not generate enough activity in EDD's procedural content generation system to be able to draw accurate conclusions from the study.

Open Problems and Future Work

This paper presents the first MI-CC tool with quality-diversity that explores the usage of a data-driven designer preference model, and its implementation into the EA loop as a complementary evaluation of individuals. Through this model, we

searched to cope with some of the limitations presented in previous work, mainly, the user fatigue when queried to choose solutions for the EA, and the stalling of the evolutionary process, thus, adapting the control of the user in the search-space to the dynamic workflow of MI-CC tools.

In this section, we present the multiple challenges that arose when trying to use the designer preference model from our first experiments and preliminary study and the open areas for active research. Through our user study, we were able to test the behavior of our preference model adapted to each of the designers and the performance of such in the wild. While the model, in general, was less used than expected, it was indeed able to learn to certain extent characteristics of the preferred suggestions.

Dataset

The dataset S created each discrete step the designer applied a suggestion, had a set of intrinsic attributes that while positive and interesting to learn from, they could have been counterproductive and could potentially explain the low and fluctuating accuracy of the model. Firstly, as mentioned in section Designer Preference Model, each generated dataset had a maximum number of samples of $M \times N \times feasible_population$, capped to 625 samples in our study, which might not be enough data to accurately learn or would require more training epochs, which ultimately would result in overfitting. This aligns with the open problems presented in [21], where the authors discuss that games will always be constrained by the amount of data, and even though we can generate many samples with our EA, it still might not be enough to cope with the amount of data that ML-approaches require.

Secondly, by taking advantage of the grid visualization of the MAP-Elites, we also inherited the behavioral relation among the different elites, and consequently, each independent training session would intrinsically represent such relation. While our objective was indeed to learn this behavior relationship, which could reveal interesting relations and perspectives by the model, the differences that each pair of behavioral dimensions have could potentially disrupt the whole model between training sessions. For instance, if we train with symmetry and similarity as dimensions, and subsequently change them to symmetry and leniency, what before could be 0.8 in preference in the dataset (i.e. a neighbor of the previously applied suggestion), could now be 0.0 in preference for this dataset, since the pair of dimensions would sort individuals completely different.

Finally, the fact that we automatically assigned an estimated preference value to all

individuals based on their grid position, and as pointed out in the previous point, relations could fluctuate dramatically, which could arise a potential issue with the dataset. For instance, a challenge with estimating the preference can be observed in the aesthetic aspects of the rooms, where two rooms can be quite aesthetically similar (i.e. have a single different tile) and yet, due to the way we assign the preference values to train, have a very different preference, thus, enabling confusion in the model. Nevertheless, we did not want the assigned preference value to be based on the similarity between suggested rooms since what the model would end up just learning is to classify based on aesthetic similarity. Therefore, there would not be any need to train any model and through just composing a similarity table and comparing new rooms to the ones already included we would probably achieve the same result.

Preference modality

We chose the suggestion grid of the MAP-Elites as an inflection point for the training of the model since it felt more appropriate and natural to the workflow of the tool, and more of a pointer to the actual preference of a designer. The suggestion grid is a reflection of the EA search for quality solutions and having the designer proactively choosing solutions that were interesting for them seemed like an indicator of the preference and interest of the designer.

Based on when the designers actually started applying suggestions and their reason why, indicates that they were not as representative of the preference of the designer as expected. Instead, suggestions were seen as an in-between step to help shape the final room, after creating a first draft of the room and before actually reaching a satisfactory room. This opens up the investigation on what design processes or combinations of processes could be captured to accurately represent the designers' preferences with higher fidelity.

Firstly, we need to consider the level of the designer that is using the tool. The design process, the objectives when designing, the vision on what to do, and the ideas on what to design and what is expected from an interactive tool as ours, could vary quite drastically between designer levels, as it is concluded in [9]. Considering our previous studies with game designers that are more experienced and the one done for this study, we realize that novice designers come with many different ideas that they would like to try, as well as experimenting with very different designs, which in turn means that their preferences and intentions change in very short periods. Understanding this, and adding it as a constraint on the design of preference models is vital since we would want to recognize this key changes to

probably discard the model and start fresh since what the model had learned might not be useful anymore.

Secondly, choosing what and when to gather information to create the model is a key aspect. Besides the EA suggestions on the designer's design, we could use the designer's history of changes through their design as well as their current designs. In our case, constantly analyzing the composition of the dungeon and the rooms could bring some insight on the stage of the design process of the designer, which could be used to further understand what to use, if we should keep using the same model, and how to train.

It might even be relevant to have a set of models per set of rooms that have some qualitative similarities to avoid confusion in the model, and updating the model that is relevant to the specific objectives of the designer. In counterpart, this would break the aspect of generalization (i.e. learning the preference of the designer throughout their design process) that could enable us to learn more from the designer.

Dynamic-Dynamic System vs. Dynamic-Static System

In our experiments, we designed a system where the model would move through the solution space (i.e. the preference-space of the designer) as the designer moves as well, which we call a dynamic-dynamic system. In such a system, the designers drift in many dimensions as they develop, understand better the tool, get deeper in the creative process, have different objectives, and such on. Further, designers might have drifted quite drastically in between training sessions, which ultimately makes the dynamic model harder to move with the designers, resulting in a deficient model.

Therefore, we can conclude that to have some stability and be more robust to an ever-changing designer and creative process, we need some part of the approach to be static. Yet, the designer will never stop being a dynamic component, thus, it is the model that needs to be static. An exciting and interesting open area of research is then in the notion of community models, which would be models fed with several designers' designs, clustered together by their qualitative similarities creating archetypes of designers or archetypes of designs. Such a set of group models would adapt to the dynamic designer by placing the models in the solution space, where a designer instead of drifting together with their model, they would traverse such a space of models as she drifts through the many dimensions of her creative process.

Future Work

Taking as a starting point the big amount of data (i.e. handmade rooms) collected from all the user studies done to date, and as abovementioned, we believe that a community model formed through clustering is a more realistic model. The envisioned system would follow exactly the same approach and core concept presented in this paper, i.e. a model that as it becomes more confident on the preference of the designer, the more weight it has to evaluate newly generated individuals by the IC-MAP-Elites, as a complementary evaluation to the objective function.

Such a system could be created by using the data of each designer (i.e. a list of created rooms), then those could be arranged in different clusters that would represent archetypical designers or archetypical designs. From this point, we would have a foundation from which we would categorize new designers and we could, on the one hand, create a model from the data in the cluster and start adapting it to the current designer, avoiding the cold start problem. On the other hand, we could as well just keep trying to assign the designer, based on her designs, to different clusters, using each cluster as a model to infer what the “community” of designers would prefer, and since, the designer is part of that community at the moment, what she would prefer. Therefore, creating a model that could be more robust for evaluating designers’ preferences by means of having more or less stable clusters that designers could navigate as they go deeper into the design process.

Furthermore, we could go a step further and conceptualize a layered model that on the top layer could represent the community models of the designers, and on the bottom layer, specific designer’s models. The bottom layer would then be created in a more classical training session outside our MI-CC tool, with the designer being queried a set of models and she explicitly labeling what she likes and whatnot. Such a model could be used to communicate the expected design style and preference among a group of designers working together or to train new designers based on senior designers’ preferences, intentions, and style.

We would also like to explore different steps on the tool where we could collect relevant and crucial data of the designers that could bring us a step closer to a more accurate model of their preferences. Furthermore, accounting for the designer level could have a very impactful result on an effective model, and on how we handle them and their relevance.

Finally, exploring and using different representations of the data, such as images of the rooms in a Convolutional Neural Network (CNN), or qualitative and more processed information of the room (e.g. tiles density, sparsity, and amount, room

complexity, connected rooms information, etc.) is an interesting future line. We believe that CNNs could perform better but required even larger amounts of data, and creating 625 images of the suggestion (i.e. our maximum number of data tuples) and then training the model could be cumbersome and have a significant impact on the workflow.

Conclusion

In this paper, we have presented the Designer Preference Model, which is a data-driven system that learns an individual designer's preference through the designer's proactive choosing of generated suggestions without disrupting the continuous reciprocal workflow in MI-CC. We implemented our approach in the Evolutionary Dungeon Designer, a Quality-Diversity MI-CC tool, where designers can create dungeons and rooms while the underlying evolutionary system provides suggestions adapted to their current design.

We used the model as a complementary evaluation system to the fitness function of the suggestions in a weighted sum, where the model gained more weight as it became more confident and performed better. Therefore, we aimed at better assessing these provided suggestions with the use of the Designer Preference Model, for them to be interesting and preferable but still usable for designers.

Through our experiments and preliminary studies on using the model to adapt to different designers, we identified a set of challenges and open areas for active research that integrates MI-CC with PCG through Machine Learning. Those challenges relate to the amount of user data needed to accurately learn from the user's preferences, what type of data is needed from the process, the cold start problem, the seldom collection of data to train, the quality of the dataset, and the designer-model setup. Moreover, we wanted to come closer to machine teaching [32] approaches where the human provides fewer data points but with higher quality (i.e. the necessary data to correctly learn) rather than classic approaches to ML (i.e. offline training with a substantial amount of data). In our approach, while the designer has the decision on when to train the algorithm and to a certain extent, with what data to train, we are still missing certain granularity to empower designers to give the right information to the algorithm.

The combination of MI-CC tools with PCG through Machine Learning is a promising area of research that has the potential to enhance content creation. Specifically, designer modeling and our approach to model the designer's preference can have a great impact on the creative process of designers by considering their preferences, intentions, and objectives into the loop, by adapting the workflow to

their requirements, or by smoothing the communication among various designers. Finally, by adding the preference model as a complementary evaluation to the generated suggestions of the evolutionary algorithm, we can give more control, to a certain extent, to the designers over the evaluation of the individuals. In consequence, we can generate higher quality suggestions that better fit a specific designer.

Acknowledgements

The Evolutionary Dungeon Designer is part of the project *The Evolutionary World Designer*, supported by The Crafoord Foundation.

References

- [1] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016. 167
- [2] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018. 167
- [3] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 167
- [4] N. Shaker, M. Shaker, and J. Togelius, “Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.,” in *AIIDE*, 2013. 167
- [5] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 167
- [6] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, “Pitako-recommending game design elements in cicero,” in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, IEEE, 2019. 167
- [7] A. Liapis, G. N. Yannakakis, and J. Togelius, “Generating Map Sketches for Strategy Games,” in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS, Springer, 2013. 167
- [8] M. Guzdial, N. Liao, and M. Riedl, “Co-creative level design via machine learning,” in *Joint Proceedings of the AIIDE 2018 Workshops co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2018)*, Edmonton, Canada, November 13-14, 2018., 2018. 167

- [9] P. Lucas and C. Martinho, “Stay awhile and listen to 3buddy, a co-creative level design support tool,” in *Proceedings of the Eighth International Conference on Computational Creativity, Atlanta, Georgia, USA, June 19-23, 2017*. (A. K. Goel, A. Jordanous, and A. Pease, eds.), pp. 205–212, 2017. 167, 177
- [10] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, “Can computers foster human users’ creativity? theory and praxis of mixed-initiative co-creativity,” *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016. 167
- [11] J. Lehman and K. O. Stanley, “Abandoning objectives: Evolution through the search for novelty alone,” *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, 2011. 167, 170
- [12] D. Gravina, A. Liapis, and G. Yannakakis, “Surprise search: Beyond objectives and novelty,” in *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO ’16*, (New York, NY, USA), pp. 677–684, ACM, 2016. 167
- [13] C. Stanton and J. Clune, “Curiosity search: Producing generalists by encouraging individuals to continually explore and acquire skills throughout their lifetime,” *PLOS ONE*, vol. 11, pp. 1–20, 09 2016. 167
- [14] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, pp. 1047–1054, ACM, 2018. 167
- [15] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, “Procedural content generation through quality diversity,” in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, 2019. 167
- [16] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016. 167
- [17] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015. 167
- [18] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008. 168

- [19] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, 2019. 168, 170, 171, 175
- [20] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, 2018. 168, 169
- [21] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, “Procedural content generation via machine learning (pcgml),” *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018. 168, 176
- [22] A. Liapis, G. Yannakakis, and J. Togelius, “Designer modeling for personalized game content creation tools,” in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19, pp. 11–16, AI Access Foundation, 2013. 168
- [23] A. Liapis, G. N. Yannakakis, and J. Togelius, “Designer modeling for sentient sketchbook,” in *2014 IEEE Conference on Computational Intelligence and Games*, pp. 1–8, Aug 2014. 168
- [24] J. Lehman, S. Risi, and J. Clune, “Creative generation of 3d objects with deep learning and innovation engines,” in *Proceedings of the 7th International Conference on Computational Creativity*, 2016. 168
- [25] M. Treanor, A. Zook, M. P. Eladhari, J. Togelius, G. Smith, M. Cook, T. Thompson, B. Magerko, J. Levine, and A. Smith, “Ai-based game design patterns,” in *Proceedings of the 10th International Conference on the Foundations of Digital Games 2015*, Society for the Advancement of Digital Games, 2015. 169
- [26] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, 2018. 169
- [27] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Mixed-initiative procedural generation of dungeons using game design patterns,” in *Proc. IEEE Conf. Computational Intelligence and Games (CIG)*, pp. 25–32, 2017. 169, 172

- [28] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG ’17, (New York, NY, USA), pp. 74:1–74:10, ACM, 2017. 169
- [29] J. Secretan, N. Beato, D. D’Ambrosio, A. Rodriguez, A. Campbell, and K. Stanley, “Picbreeder: Evolving pictures collaboratively online,” in *Proceedings of Computer Human Interaction Conference*, pp. 1759–1768, 04 2008. 170
- [30] A. Liapis, G. N. Yannakakis, and J. Togelius, “Adapting models of visual aesthetics for personalized content creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012. 170
- [31] P. Donmez and J. G. Carbonell, “Proactive learning: cost-sensitive active learning with multiple imperfect oracles,” in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 619–628, ACM, 2008. 171
- [32] P. Y. Simard, S. Amershi, D. M. Chickering, A. E. Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, and J. Wernsing, “Machine teaching: A new paradigm for building machine learning systems,” 2017. 180

PAPER V - INTERACTIVE CONSTRAINED MAP-ELITES: ANALYSIS AND EVALUATION OF THE EXPRESSIVENESS OF THE FEATURE DIMENSIONS

Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius

ABSTRACT

We propose the Interactive Constrained MAP-Elites, a quality-diversity solution for game content generation, implemented as a new feature of the Evolutionary Dungeon Designer (EDD): a mixed-initiative co-creativity tool for designing dungeons. The feature uses the MAP-Elites algorithm, an illumination algorithm that segregates the population among several cells depending on their scores with respect to different behavioral dimensions. Users can flexibly and dynamically alternate between these dimensions anytime, thus guiding the evolutionary process in an intuitive way, and then incorporate suggestions produced by the algorithm in their room designs. At the same time, any modifications performed by the human user will feed back into MAP-Elites, closing a circular workflow of constant mutual inspiration. This paper presents the algorithm followed by an in-depth evaluation of the expressive range of all possible dimension combinations in several scenarios, and discusses their influence in the fitness landscape and in the overall performance of the procedural content generation in EDD.

PUBLISHED IN

IEEE Transactions on Games, IEEE, 2020.

INTERACTIVE CONSTRAINED MAP-ELITES: ANALYSIS AND EVAL- UATION OF THE EXPRESSIVENESS OF THE FEATURE DIMENSIONS

Introduction

Procedural Content Generation (PCG) refers to the generation of game content with none or limited human input [1], where game content could be anything from rules and narrative, to levels, items, and music. While PCG has been a factor in game development since trailblazing games like *Rogue* [2] and *Elite* [3], it has only been a popular academic research topic for little more than a decade. Search-based PCG designates the use of a global search algorithm, such as an evolutionary algorithm to search content space [4].

Part of PCG's appeal is the promise to produce game art and content faster and at a lower cost, as well as enabling innovative content creation processes such as player-adaptive games [5–7], data-driven content generation [8,9], and mixed-initiative co-creativity [10]. Mixed-initiative co-creativity (MI-CC), a concept introduced by Yannakakis et al. [11], refers to the approach of using a creation process through which a computer and a human user provide and inspire each other in the form of iterative reciprocal stimuli. Examples of MI-CC systems are *Pitako* [12], *Ropossum* [13], *Tanagra* [14], *CICERO* [15], and *Sentient Sketchbook* [16].

MI-CC aligns with the principles of lateral thinking and creative emotive reasoning: the processes of solving seemingly unsolvable problems or tackling non-trivial tasks through an indirect, non-linear, creative approach [10]. Additionally, MI-CC provides insight on the affordances and constraints of the human process for creating and designing games [1].

A key mechanism in MI-CC approaches is to present suggestions to users, and these suggestions must be of high quality but also be sufficiently diverse. So-called quality-diversity algorithms [17] are very well suited for this, as they find solutions

that have high quality according to some measure but are also diverse according to other measures [18]. The Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [19] is a suitable algorithm for this kind of problem. Khalifa et al. [8] presented constrained MAP-Elites, a combination MAP-Elites with the feasible-infeasible concept from the FI2Pop genetic algorithm [20], and applied this to procedurally generating levels for bullet hell games. Another recent implementation of MAP-Elites has been used to produce small sections of Super Mario Bros levels called *scenes*, addressing specific game mechanics [21].

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool for generating dungeons for adventure games using a FI2Pop evolutionary approach [22–25]. This research was presented in [26], introducing *Interactive Constrained MAP-Elites* (IC MAP-Elites), a combination of Constrained MAP-Elites with interactive evolution, in the shape of a continuous evolutionary process that takes advantage of MAP-Elites’ multidimensional discretization of the search space into cells. In [26], we analyzed the effects of using quality-diversity in procedurally generating dungeons, as well as the effects of continuous evolution and dimension customization.

This paper contributes with a thorough evaluation of the expressive range of EDD’s MAP-Elites through all possible dimension combinations in several scenarios, as well as by extending the feature dimensions to two new dimensions, *Inner similarity*, which adds a similarity measure independent of the aesthetics, and *Leniency*, which strives to measure the subjective challenge score of a level. Following recent research on how to evaluate procedural content generators and, in particular, quality-diversity approaches [27–29], we present the results from new experiments with the objective to evaluate the expressive range of all dimensions in pairs, as well as to analyze how the generated and unique solutions relate to all the dimensions included in the search space, and assess IC MAP-Elites feasibility and adaptability to create dungeons and adventure levels.

Previous Work

Map-Elites for illuminating search spaces

Quality-diversity algorithms are algorithms which search a solution space, not just for the single best solution, but for a set of diverse solutions which are high performing. MAP-Elites maintains a map of good solutions [19] and is a well-known quality-diversity algorithm. The map is divided into a number of cells according to one or more feature dimensions. In each cell, a single solution is kept. At every update, an offspring is generated based on one or more existing solutions. That offspring is then assigned to a cell based on its feature dimensions,

which might or might not be the same as the cell(s) its parent(s) occupy. If the new offspring has a higher fitness than the existing solution in that cell, it replaces the previous item in the cell. This process results in a map of solutions where each cell contains the best found solution for those particular feature dimensions.

Evaluation of Procedural Content Generators

Shaker, et al. [30], argues that ultimately the evaluation of content generators is to verify that they fulfil their design goals. In order to be able to understand or modify a generator it is important to visualize its content space. However, it is seldom enough to look at a single individual piece of content, but rather it is vital to examine the frequency the different features appear, or the amount of variety the features demonstrate. Previous attempts of doing this are termed expressivity measures [31, 32] and have, for instance, explored difficulty measures [33]. Other approaches incorporate tool assisted parameter exploration due to its effect on the content space [27, 34].

Evolving Dungeons as a Whole, Room by Room

EDD is a MI-CC tool that allows a human designer to create a 2D dungeon and the rooms it is composed of. The designer is able to manually edit both the dungeon by placing and removing rooms, and the individual rooms by editing the tiles that each room consists of. EDD's underlying evolutionary algorithm provides procedurally generated suggestions, and is driven through the use of game design micro- and meso- patterns. A detailed description of all EDD's features, including the use of game design patterns, can be found in [22–25], where [22, 24] analyze and discuss the mixed-initiative system in EDD and [23, 25] focus on the procedural content generator.

In this section we present the latest version of EDD¹, which includes significant improvements based on the outcomes from the qualitative analysis discussed in [22]. The dungeon is now represented as a graph of interconnected rooms of any given size between 3×3 and 20×20 tiles. The smallest allowed dungeon is composed by two rooms with one connection to each other. Rooms and connections can now be added and removed at any time. Connections are marked with door tiles.

The designer marks one room as the *initial room* for feasibility calculation: a dungeon is feasible when there is at least one path between the *initial room* and any other passable tile in the dungeon. Rooms and doors that are unreachable from the *initial room* are highlighted in red, so that they can be easily identified by the

¹Available for download at <https://github.com/mau-games/eddy>

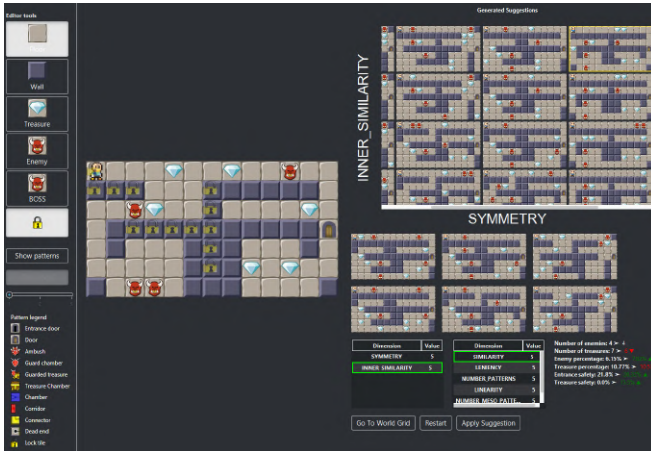


Figure 1: The room editor screen in EDD. The left pane contains all the options for manually editing the room displayed at the center-left of the screen. The right section displays the procedurally generated suggestions.

designer.

The starting screen in EDD is the dungeon editor screen. Every new room is empty (composed solely of floor tiles) when created and is placed detached from the dungeon graph. After manually connecting the room to the dungeon with at least one connection, the designer has the option to populate the room using the room editor screen (Figure 1). This screen can be reached by either double-clicking on the room or by clicking on the "Start with our suggestions" button, which will present six procedurally generated suggestions to the designer to start editing from.

Figure 1 shows the room editor screen displaying a sample room with the dimensions 7×13 tiles. The left pane lists all the available options for manually editing the room by brush painting with one of the available tile types (floor, wall, treasure, or enemy) and one of the two brush sizes (single tile and five-tile cross shape). Control-clicking allows the designer to bucket paint all adjacent tiles of the same type. Brush painting with the lock button on preserves selected tiles in all the procedurally generated suggestions. A detailed description of all the options in this pane is included in [22, 23].

The right side of the screen displays the procedurally generated suggestions, by means of the Interactive Constrained MAP-Elites (IC MAP-Elites) genetic algorithm (see Part II). When the designer accesses the room editor screen, IC MAP-Elites starts and continuously populates the suggestions pane with elites. The evolutionary process is fed with the manually edited room (i.e. target room), so that

every change in the room affects the generated suggestions. By clicking on "Apply Suggestion", the manually edited room is replaced by the selected suggestion, thus affecting the upcoming procedural suggestions. "Restart" restarts evolution, and "Go To World Grid" takes the user back to the dungeon editor screen.

Interactive Constrained MAP-Elites

The overarching goal of MI-CC is to collaborate with the user to produce content, either to optimize (i.e., exploit) their current design towards some goal or to foster (i.e., explore) their creativity by surprising them with diverse proposals. By implementing MAP-Elites [19] and continuous evolution into EDD, our algorithm can (1) account for the multiple dimensions that a user can be interested in, (2) explore multiple areas of the search space and produce a diverse amount of high-quality suggestions to the user, and (3) still evaluate how interesting and useful the tile distribution is within a specific room. Henceforth, we name the presented approach **Interactive Constrained MAP-Elites** (IC MAP-Elites).

EDD uses a single-objective fitness function (shown in Equation (1)) with a FI2Pop genetic algorithm, where fitness is a weighted sum divided equally between (1) the inventorial aspect of the rooms and (2) the spatial distribution of the design. $f_{inventorial}$ is the evaluation of the aggregated and normalized quality of treasures, enemies, and doors (inventorial patterns). $f_{spatial}$ refers to the quality and distribution of chambers, i.e. open areas in the room and corridors (both categorized as spatial patterns), and the meso-patterns that are created within chambers and their quality. Quality refers to the positioning, safety, composition, and relation between patterns. The fitness adapts to the user's current design, automatically informing target ratios and distributions to be used as targets. In-depth evaluation of EDD's fitness function, as well as discussion and explanation of the quality of each inventorial, spatial, and meso pattern, can be found in [23–25].

$$f_{fitness}(r) = \frac{1}{2}f_{inventorial}(r) + \frac{1}{2}f_{spatial}(r) \quad (1)$$

Furthermore, IC MAP-Elites adds interactive and continuous evolution to the Constrained MAP-Elites presented by Khalifa et al. [8]. This is done through an adaptive fitness function (based on the designer's design) that adapts the content generation, and by enabling the designer to flexibly change the feature dimensions and the granularity of the cells. It also adapts the usability of MAP-Elites to generate dungeon and adventure levels in an MI-CC system, which gives more control to the designers over non-intuitive parameters and aspects of MAP-Elites, while providing a richer set of high-performing and diverse suggestions.

Illuminating Dungeon Populations with MAP-Elites

MAP-Elites explores the search space more vastly by separating interesting feature dimensions, that affect different aspects of the room, such as playability or visual aesthetics, from the fitness function, to categorize rooms into cells.

In this subsection, we firstly present all the current feature dimensions identified and implemented in EDD, secondly, we explain the transition from fixed evolution to continuous evolution, and lastly, we introduce and outline the IC MAP-Elites algorithm.

Dimensions

Dimensions in MAP-Elites are identified as those aspects of the individuals that can be calculated in the behavioral space, and that are independent of the fitness calculation. These are crucial in MAP-Elites, as they represent the discretized dimensions where individuals will be retained as the space is explored. EDD offers the designer the possibility to choose among the following dimensions, two at a time (examples of rooms generated using the dimensions are shown in Figure 2c):

Symmetry (Sym) and Similarity (Sim). We choose Symmetry as a consideration of the aesthetic aspects of the edited room since symmetric structures tend to be more visually pleasing for the user and relate to fairness distribution and human-made structures [23,35,36]. Similarity is used to present the user variations of their design but still preserving their aesthetical edits. Symmetry is assessed using only impassable tiles (i.e., walls) and evaluated along the X and Y axes and diagonals. The highest value is normalized by the total amount of walls and used as the symmetry score (Equation (2)). Similarity is calculated by comparing tile by tile with the target room (Equation (3)). In-depth descriptions and evaluations can be found in [23], where both dimensions were used as aesthetic fitness evaluations.

$$D_{sym} = \frac{highestSymmetricValue}{totalWalls} \quad (2)$$

$$D_{sim} = \frac{totalTiles - notSimilarTiles}{totalTiles} \quad (3)$$

Number of Meso-patterns (NMP). The number of meso-patterns correlates to the type and amount of encounters the designer wants the user to have in the room in a more structured manner. The considered patterns are the treasure room (tr), guard rooms (gr), and ambushes (amb). Meso-patterns associate utility to a set of tiles in the room, for instance, a long chamber filled with enemies and treasures could be divided into 2 chambers, the first one with enemies and the

second one with treasures so the risk-reward encounter is more understandable for the player. Since we already analyze the rooms for all possible patterns, the number of meso-patterns is simply $\#MesoPat = tr, gr, amb \in AllPatterns$. Equation (4) presents the dimensional value, and since the used meso-patterns can only exist in a chamber, we normalize by the maximum amount of chambers in a room, which are of a minimum size of 3×3 , and results in $Max_{chambers} = \lfloor Cols/3 \rfloor \cdot \lfloor Rows/3 \rfloor$.

$$D_{NMP} = \min \left\{ \frac{\#MesoPat}{Max_{chambers}}, 1.0 \right\} \quad (4)$$

Number of Spatial-patterns (NSP). By spatial-patterns we mean chambers (c), corridors (cor), connectors (con), and nothing (n). We identify the number of spatial-pattern relates to how individual tiles group (or not) together to form spatial structures in the room. The higher the amount of spatial-patterns the lesser tiles will be group together in favor of more individualism. For instance, a room with one spatial-pattern can be one with no walls and just an open chamber, while a room with a higher number of spatial-patterns would subdivide the space with walls, using tiles for more specific patterns. Equation (5) presents how we calculate the value for such a dimension. The number of spatial patterns is simply $\#SpatialPat = c, n, cor, con \in AllPatterns$, we then normalize it by the largest side of the room and multiply it by a constant value, determined as $K = 4.0$ through a process of experimentation.

$$D_{NSP} = \min \left\{ \frac{\#SpatialPat}{\max \{Cols, Rows\} \cdot K}, 1.0 \right\} \quad (5)$$

Linearity (Lin). Linearity represents the number of paths that exist between the doors in the room. This relates to the type of gameplay the designer would like the room to have by the distributions of walls among the room. Having high linearity in a room does not need to only be by having a narrow corridor between doors but could also be generated by having all doors in the same open space (i.e. the user would not need to traverse other areas) or by simply disconnecting all paths between doors. Equation (6) shows the linearity calculation. Due to the use of patterns, we calculate the paths between doors as the number of paths that exist from a spatial-pattern containing a door to another. Finally, this is normalized by the number of spatial patterns in combination with the number of doors and their possible neighbors.

$$D_{lin} = 1 - \frac{AllPathsBetweenDoors}{\#spatialPat + \#NeighborsPerDoor} \quad (6)$$

Inner Similarity (IS). Inner similarity compares the target room to one generated, considering only the distribution and ratios of micro-patterns in both rooms rather than any aesthetic criteria. Specifically, we look into the density (*den*) and sparsity (*spa*) of enemies (*en*), treasures (*tre*), and walls (*wal*) in the target room (R_{tg}) in comparison with the generated rooms (R_{gen}). To calculate the density and sparsity of each micro-pattern, we first clustered all micro-patterns of the same kind based on the distance within the room (i.e. *distance* = 1). We then use these clusters to calculate the density (Equation (7)) using as density threshold $\theta = 4$ for treasures and enemies, and $\theta = 6$ for walls, and calculate the sparsity (Equation (8)). Finally, we calculate the difference between each distribution in R_{tg} and R_{gen} , linearly combining all the values into the IS measure as in Equation (9).

$$den(x) = \frac{\sum_{i=1}^{|clusters|} \min \left\{ 1.0, \frac{|cluster_i|}{\theta_x} \right\}}{|clusters|} \quad (7)$$

$$spa(x) = \frac{\sum_{i=1}^{|clusters|} \sum_{j=1, j \neq i}^{|clusters|} \frac{Dist(cluster_i, cluster_j)}{|room|}}{|clusters| \cdot (|clusters| - 1)} \quad (8)$$

$$D_{IS} = \sum_{i=1}^{micropats} |den(R_{gen}) - den(R_{tg})| + |spa(R_{gen}) - spa(R_{tg})| \quad (9)$$

Leniency (Len). Leniency calculates how challenging a room is at any given point. It is based on the amount of enemies and treasures that are in a room, their density and sparsity calculated as in eq. (7) and (8), respectively, and how safe the doors are (i.e. entry/exit points) calculated as in [24]. We base our calculation in the idea that rooms are less lenient the more enemies they contain as well as how they are distributed, counterbalanced by the number of treasures as they reward players. We calculate the dimension value for each room as shown in (Equation (12)), which uses a combination of precomputed non lenient (Equation (10)) and lenient (Equation (11)) values.

$$nonLenientValues = w_0 \cdot \log_{10}(|en| \cdot spa(en)) + w_1 \cdot \log_{10}(|en| \cdot den(en)) + w_2 \cdot (1.0 - door_{safety}) \quad (10)$$

$$LenientValues = \frac{1}{2} \log_{10}(|tre| \cdot spa(tr)) + \frac{1}{2} \log_{10}(|tre| \cdot den(tr)) \quad (11)$$

$$D_{len} = 1.0 - (nonLenientValues - (\frac{1}{2} \cdot LenientValues)) \quad (12)$$

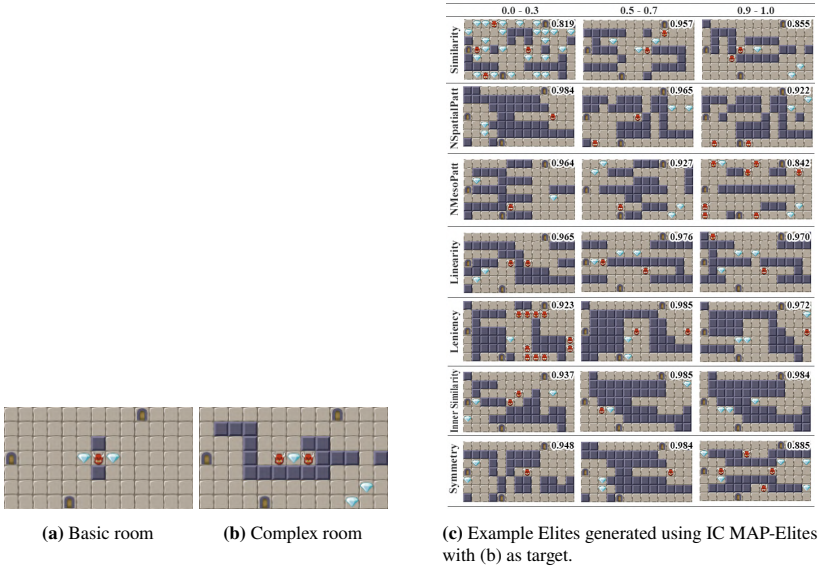


Figure 2: (a) and (b) represent target rooms used in the experiments. Each row in (c) represents an independent run of the algorithm using the dimension specified to the left. Each column splits the dimension score into three intervals. Each cell displays (top-right) the fitness of the optimal individual in its related interval.

Continuous and Interactive Evolution

Since EDD already uses a FI2Pop, we took the Constrained MAP-Elites, presented by Khalifa et al. [8], as a starting point. The illuminating capabilities of MAP-Elites explore the search space with the constraints aspects of FI2Pop. This approach manages two different populations, a feasible and an infeasible one, within each cell. Individuals move across cells when their dimension values change, or between the feasible and infeasible population according to their fulfillment of the feasibility constraint.

As discussed by Takagi, interactive evolution is a way to improve the capabilities of Evolutionary Algorithms (EA) by having humans in-the-loop to subjectively evaluate individuals. This hybrid approach has proven to reach better and more

adaptive results but at the expenses of user's fatigue and user's understanding of the EA and the given problem [37]. However, in EDD and the IC MAP-Elites, the user does not directly evaluate individuals; instead, IC MAP-Elites adapts the fitness and search based on different interactions the designer has with the algorithm.

The designer can interact with the crossover step by locking tiles [23], and with the dimensions and cells by changing the dimensions and granularity for the MAP-Elites, enabling IC MAP-Elites to focus on different regions of the generative space. Furthermore, IC MAP-Elites constantly updates the target room and configuration with the most recent version of the designer's design. Once the suggestions are broadcasted, that room is incorporated without changes to the population of individuals in the corresponding cell.

This adaptability feature and different designer's indirect interactions with IC MAP-Elites, enables the implementation of continuous and interactive evolution, as well as allowing the designer to focus solely in the design of the room, while IC MAP-Elites adapts to the new designs.

Algorithm

IC-MAP-Elites is depicted in Algorithm 3. Cells are first created based on the dimensions selected by the user and proceed to initialize the population based on the user's design, evaluate it and assign each individual to the corresponding cell. Before starting each generation, we check if the dimensions have changed, and if so, recreate the cells and populate them with the previous individuals, and proceed through the evolutionary strategies. We first select uniformly random which cell to choose parents from, and then we select 5 parents through tournament-selection. Offspring are produced through a two-point uniform crossover operation with a 30% chance of mutation. Offspring are placed in the correct cell and population after calculating their fitness and dimension's information. Finally, cells eliminate the low-performing individuals that over-cap their maximum capacity. Since interbreeding is not allowed, and can only happen indirectly (i.e. the offspring changing population and then used for breeding in consequent generations), the strategies are repeated for each of the populations.

IC MAP-Elites runs for n generations, and once it reaches the specified limit, it broadcasts the found elites. In order to foster the exploration, we first mutate all the individuals from all the populations and cells (while retaining the previous population), and add them into the same pool together with the current edited room without changes. Finally, we evaluate and assign all the individuals to the correct cells, and cells that are over maximum capacity eliminates low-performing

Algorithm 3 Interactive Constrained MAP-Elites

```
1: procedure IC MAP-ELITES( $\{d_1, v_1\}, \dots, \{d_n, v_n\}$ )
2:    $target \leftarrow curEditRoom$  ▷ Always in background
3:   createCells( $\{d_1, v_1\}, \dots, \{d_n, v_n\}$ )
4:   for  $i \leftarrow 1$  to  $PopSize$  do
5:     add mutate( $target$ ) to  $population$ 
6:   CheckAndAssignToCell( $population$ )
7:   while true do ▷ start continuous evo
8:     for  $generation \leftarrow 1$  to  $publishGen$  do
9:       if  $dimensionsChanged$  then
10:         $previousPop \leftarrow cells_{pop}$ 
11:        createCells( $newDimensions$ )
12:        checkAndAssignToCell( $previousPop$ )
13:       repeat [for feasible & infeasible pop.]
14:         for  $i \leftarrow 1$  to  $ParentIteration$  do
15:            $curCell \leftarrow rndCell(cells)$ 
16:           add tournament( $curCell$ ) to  $parent$ 
17:            $offspring \leftarrow crossover(Parent)$ 
18:           checkAndAssignToCell( $offspring$ )
19:         sortAndTrim( $cells$ )
20:       broadcastElites() ▷ render elites
21:        $pop' \leftarrow cells_{population}$ 
22:       add mutate( $cells_{pop}$ ) to  $pop'$ 
23:       add  $target$  to  $pop'$ 
24:       checkAndAssignToCell ( $pop'$ )
25:       sortAndTrim( $cells$ )
26: procedure CREATECELLS(DIMENSIONS)
27:   for each  $dim \in dimensions$  do
28:     add newCell( $dim_d, dim_v$ ) to  $cells$ 
29: procedure CHECK&ASSIGNTOCELL( $curPopulation$ )
30:   for each  $individual \in curPopulation$  do
31:      $individual_f \leftarrow evaluate(individual)$ 
32:      $individual_d \leftarrow dim(individual)$ 
33:     add  $individual$  to  $cell_{pop}(individual_d)$ 
```

individuals. This procedure is repeated until the user decides to stop the algorithm.

Performance Evaluation across Dimensions

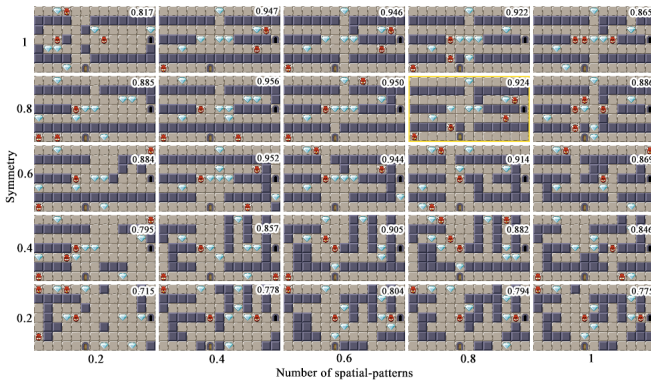


Figure 3: Rooms at generation 2090 targeting Number of spatial-patterns (X) and Symmetry (Y). Each cell displays (top-right) the fitness of the optimal individual in its related feasible population.

First, we ran a set of experiments to test the results from the IC MAP-Elites using all possible combinations of the available dimensions using two dimensions at a time. All experiments were run using 13×7 rooms, the same room size as in *The Binding of Isaac* [38], a representative example of a dungeon-based adventure game. In each experiment, the initial population was set to 1000 mutated individuals distributed in feasible and infeasible populations in all cells which were set to a maximum capacity of 25 individuals each. IC MAP-Elites ran continuously, and every 100 generations rendered the elites of each cell. At each generation, it selected 5 parents per population among uniformly random chosen cells. Offspring were always produced through a two-point crossover and had a 30% chance of being mutated, which would randomly alter one tile in the level.

Figure 3 shows a grid containing the best found suggestions at generation 2090, while aiming for number of spatial-patterns at the X-axis and symmetry at the Y-axis with a granularity of 5. Each cell displays the optimal individual of the feasible population under a given pair of dimension values. The fitness score is displayed on the cells' top-right corner.

The fitness evaluation in IC MAP-Elites is quite lightweight in terms of computational cost, which enables the grid of suggestions to be completed in a matter of seconds. This is of principal importance for successfully implementing continuous evolution, so that the influence of each manual change in the edited rooms is reflected in the suggestions almost instantly.

Table 1: Comparison of the avg. explored space and avg. fitness of the generated individuals in different evaluations.

Evaluation	◇	△	○
IC MAP-Elites: Avg. dimensions in pairs	0.89	36.46%	52.4%
IC MAP-Elites: All Dimensions	0.78	51.7%	N/A
Objective-based EA	0.92	22.48%	N/A

◇ Avg. Fitness △ Avg. coverage across all dimensions

○ Avg. coverage in respective dimension pair

Results in Figure 3 are representative of the good quality diversity solutions produced by EDD. The average fitness across elites is 0.872, and the highest fitness is 0.956 (cell [0.4, 0.8]). No two rooms are the same. As intended, high levels of symmetry are displayed in the upper rows, gradually decreasing towards the bottom row. Similarly, rooms in the leftmost column contain lower amounts of spatial patterns, increasing towards the rightmost column. Lower amounts of spatial patterns translate into more open rooms with almost no corridors and one or two large adjacent chambers (as in cell [0.2, 0.2]), as opposed to highly pattern filled rooms that comprise intricate pathways converging at one or two small chambers (cell [1, 0.2]).

Fitness values show that some dimension combinations are harder to optimize than others, so that the whole grid depicts a gradient landscape of the compatibility between each pair of dimensions. The bottom-left corner shows difficulties producing symmetric rooms with low amounts of spatial patterns, as opposed to rooms with many corridors (upper-right corner), which seem to favor the generation of symmetrical structures.

Additional similar experiments can be found in [26], where we combined other pairs of dimensions and analyzed and discussed the correlation and limitations found in them. However, in the following section, we present an extension of those evaluations through a more exhaustive and in-depth assessment of IC MAP-Elites.

Expressive Range Analysis

We ran a second set of experiments analyzing the expressive range [31] of the IC MAP-Elites using the 21 possible combinations of dimension pairs. We followed the same setup as presented in the previous section, and every 100 generations, we collected the unique generated individuals' data, which was aggregated in all the expressive ranges and individually plotted in Figure 7. Through this, between 150 to 2001 individuals were produced every 100 generations. By exploring the expressive range and conducting a comparative analysis among the various

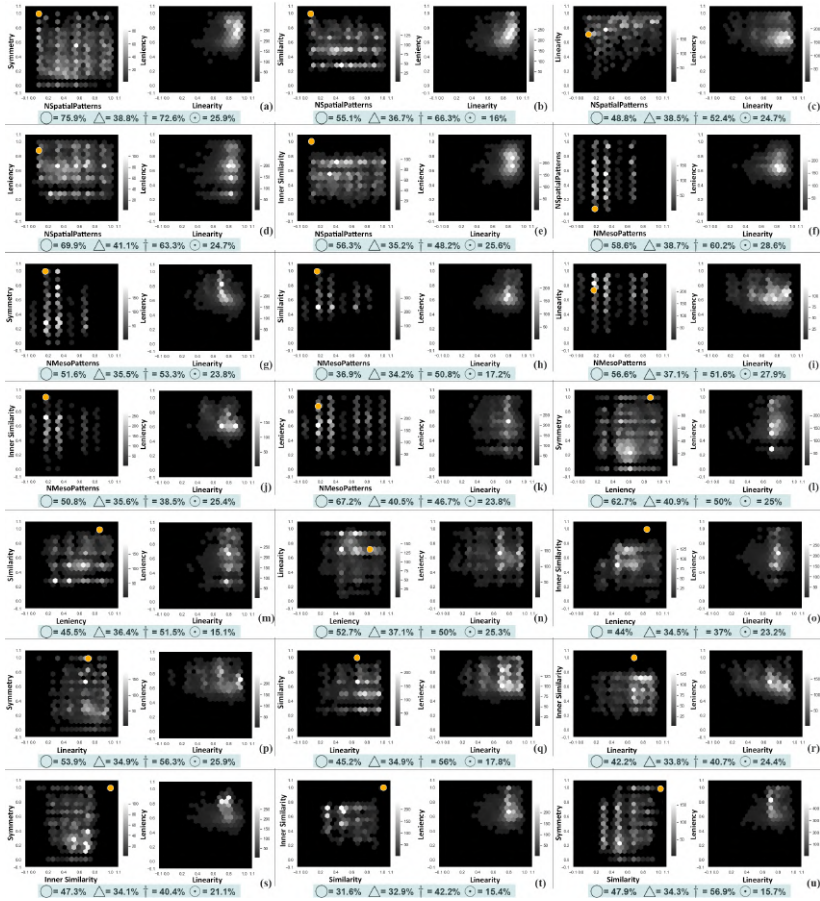


Figure 4: Expressive range of the 21 possible combinations (from a to u) of dimensions picked in pairs. Each subfigure is composed of two plots: (left) an evaluation based on a given pair of dimensions; (right) the same pair of dimensions but evaluated in terms of Linearity and Leniency scores. Each hexagon relates to a dimensional score, whereas a lighter hue indicates a higher number of unique individuals generated under that particular score. All runs used Figure 2a as the target room, and its dimensional score is highlighted with an orange mark. Under each subfigure, we present data particular to the specific tested pair of dimensions. ○, †, and ⊙ represent the coverage percentage in the respective pair of dimensions when running IC MAP-Elites with the subfigure’s pair of dimensions (○), IC MAP-Elites with all dimensions in the search (†), and when using objective-based EA (⊙). △ represents the avg. coverage percentage across all dimensions when running IC MAP-Elites with the subfigure’s pair of dimensions.

dimension combinations, we intend (1) to assess IC MAP-Elites, its exploration and exploitation capabilities, and the feature dimensions, (2) to analyze how different dimension combinations affect the search for QD content, and (3) to identify bias in the search space.

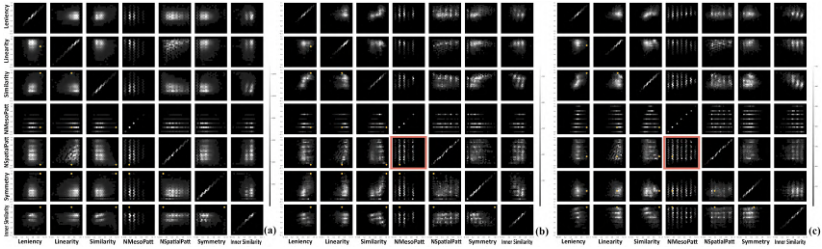


Figure 5: In-detail run showing how all dimensions relate to each other in alternative scenarios. In (a) IC MAP-Elites ran for 2000 generations using the same target room as in Figure 4 (Figure 2a), but using all the dimensions in the search, which results in 78125 cells. (b) Shows how all dimensions were explored when using Figure 4f (NMP-NSP) as pair of dimension. In (c) we used the same dimensions as in (b) but we changed the target room to Figure 2b. The dimensional score of the respective target room in each subfigure (Figure 2a for a and b, and Figure 2b for c) is highlighted with an orange mark. In b and c, the dimensions used for the experiment are highlighted with a red border.

Moreover, in Table 1 we present a comparison of the avg. diversity and avg. quality of the generated individuals between different approaches to support our evaluation. The approaches consist of IC MAP-Elites picking dimensions in pairs (avg. of the aggregated results), IC MAP-Elites using all dimensions at the same time in the search, and the objective-based EA that was used in EDD previous to IC MAP-Elites. When using a pair of dimensions in IC MAP-Elites, the search explores in avg. 52.4% in the respective pair of dimensions (individual results are presented in Figure 4) and the individuals generated have an avg. fitness of 0.89. In comparison, when using all dimensions rather than only pairs, the search explores 51.7% across all dimensions, 15% more than when using pairs, but with a drop in the avg. fitness of the population (0.78), which is discussed further in Part II and shown in Figure 6. As expected and concluded before by Mouret and Clune [19], objective-based EA generated individuals with an avg. high fitness (0.92), but with low diversity (22.48%). This means that the search had a narrower focus, and the generated levels did not differ much from each other. Objective-based ran for 5000 generations, but it stagnated and stopped generating novel levels after 13 generations i.e., stopped exploring, generating a total of 1669 levels. Conversely, IC MAP-Elites kept generating novel levels until stopped, but considerably less after 1000 generations.

Figure 4 shows the expressive range of the IC MAP-Elites with each letter referring to a unique pair of dimensions tested, and the subfigure divided into two different plots. In the left plot, we evaluate the setup based on the used pair of dimensions with each hexagon placed in relation to their dimensions' score. The hue of each of them is connected to the number of unique suggestions generated. Likewise,

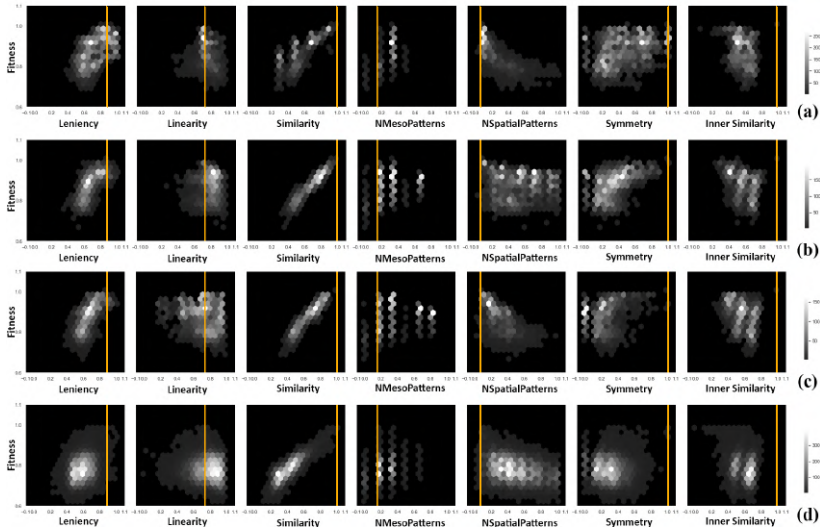


Figure 6: Relation between dimension score and fitness score. (a), (b), and (c) show results using, respectively, Similarity and Symmetry, NMP and NSP, and NMP and Linearity. (d) was run using all the 7 dimensions. All runs used Figure 2a as target room, and its dimensions score are denoted with an orange line across the different plots.

in the right plot, we evaluated the setup based on its linearity and leniency score, which is used to compare the setups' expressiveness. All the setups were run for 5000 generations, using the same target room, which is shown as an orange marker (Figure 2a).

In Figure 4, it is shown that with IC MAP-Elites, it is explored a substantial area of the generative space (denoted in the figure with \bigcirc under each subfigure) rather than just exploiting the area around the target room, depicted as an orange marker. On average, in all the independent runs in their respective dimensions, the search explores around 52% of the space, filling at least half of the map with high-performing elites averaging 0.89 (as shown in Table 1). It can also be observed that the dense areas of the search space (i.e., where the algorithm exploited the most) are distant from the target room's scores, and most of them are sparse throughout the search. This indicates that using IC MAP-Elites and a pair of dimensions at a time helps the distribution of the search while exploiting promising areas, and the search is less likely to be biased towards creating levels similar to the target.

Nevertheless, when using both leniency and linearity to compare the performance of the dimensions' pairs, it is shown that they are underexplored and within the same range (0.4-1.0) when not using them as dimensions. This points towards the search having difficulties getting out of other dimensions' local optima, especially

since the densest search area is within the target room (this is denoted with Δ under each subfigure).

Alternative Scenarios

In Figure 5, we examine how the algorithm would vary its dimensions' exploration and exploitation in two different scenarios. (a) Using the same target room as in all the cases in Figure 4 but using all the possible dimensions in the search space (i.e. 7), and (c) using NMP and NSP (see Section II) as dimensions in the search but changing the target room to Figure 2b. To draw a better comparison, we added (b), which shows how all dimensions were explored when using Figure 4f (NMP-NSP) as pair of dimension.

When using all the dimensions (a), IC MAP-Elites can explore a substantial area of the search space in each of the dimensions (51.7%), which in total is 15% more searched space than when using only a pair of dimensions on average. This is expected since all the dimensions are now acting as archives. However, as it is noted in Figure 4 under each subfigure, the actual explored space in the respective pairs (\circ) is most of the time greater than what using all dimensions explore in the respective pair (\dagger).

It can also be observed that when using NMP and NSP as dimensions Figure 5b and c, regardless of the target room, IC MAP-Elites manages to still search a good amount of space (in avg. 38.7% and 43% among all dimensions, respectively). We suspect that this is because the range between low and high scores in the NMP or NSP dimensions produces very different rooms, as it can be seen in Figure 2c in their respective rows.

Moreover, when comparing (a) and (b), it is noticeable that while (a) explores a greater area than (b) (in general, 15.24% more), it seems to be recurrently generating the same type of individuals (i.e. depicted with the hue of the hexagon) while in (b), the dense areas for most of the dimensions are sparser, especially when matching the pair of dimensions used for evaluation. Finally, it should be noted that while these three plots are comparable in their diversity search, they differ in the number of elites they store during the search, with an archive of 78125 cells for (a), and 25 cells for (b) and (c).

Fitness Evaluation

Figure 6 shows the relation of the fitness with the explored individuals in each dimension in 4 independent runs. (a, b, c) Were runs using dimensions in pairs with the same data and dimensions as in Figure 4 (u), (f), and (h), respectively.

(d) Was run using all the dimensions in the search space with the same data as in Figure 5 (a). There is an evident high correlation between Similarity scores and fitness across all the subfigures, which is expected since our fitness value is highly dependant on the target's ratios. In contrast, IS is not even close to match the fitness curve of Similarity rather there are high-performing individuals along the dimension, even when IS calculates similarity using ratios, densities, and sparsities of the target's micropatterns to calculate the score of individuals.

Moreover, our experiment shows that when using specific dimensions (Figure 6a-c), we achieve a relatively better search (i.e. find more diverse and high-quality individuals) in those dimensions, while still being able to explore the rest of dimensions. For instance, when not using NSP as a feature dimension such as in (a) or (c), the NSP dimension is fully explored but generating no high-quality individuals, meanwhile, when using NSP as a feature dimension as in (b), the search can find individuals in the same range as in (a) or (c) but with a higher fitness. Similar results can be seen in the rest of the dimensions where the search uses specific dimensions, for instance, in (a) exploring diverse and higher quality individuals in Similarity, or in (c) in Linearity.

The most interesting result can be seen in (d), where we used all dimensions in the search. This allows for a vast search of diverse individuals in all dimensions, but at the same time it seems to exploit sub-optimal areas. On the other hand, when using only a pair of dimensions as in (a), (b) and (c), the search remained dense in high-quality individuals in all dimensions.

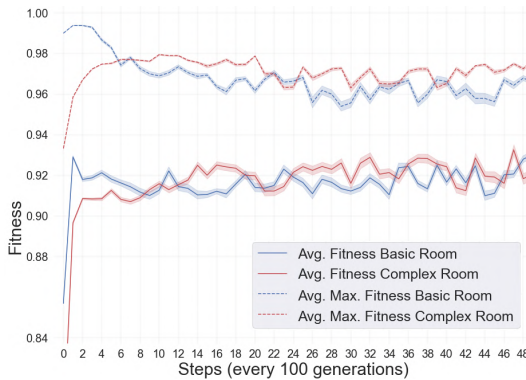


Figure 7: Fitness evolution per 100 generations throughout all the runs. Blue represents the fitness of the basic room (Figure 2a), and Red represents the fitness of the complex room (Figure 2b).

Furthermore, figure 7 shows the aggregated fitness over time of all dimension pairs (21). The avg. max fitness and the avg. total fitness are depicted as a dashed line

and a continuous line, respectively. Both are surrounded by a low opacity thicker line representing the confidence interval. The graph shows that across all pairs of dimensions, the avg. fitness of novel generated levels is very high regardless of the target room and the generation, with a minor fluctuation between 0.92 and 0.94. In addition, the graph further supports what is presented in Figure 6 a, b, and c, with most of the individuals generated in high score regions of the space in relation to fitness, but also showing that this is stable throughout the generations. The stable fitness

This stationary overall high fitness is expected from MAP-Elites as it aims at constantly generating high-performing diverse individuals. This diversity goal, combined with an adaptive fitness dependant on the target room, makes it harder to generate levels with maximum fitness. Figure 7 clearly shows this; depending on the fitness, there are scores in dimensions or pairs of dimensions that are less compatible with the fitness evaluation. Yet, what has been shown thus far in the multiple expressive range analysis and especially in Figure 6a-c, is that IC MAP-Elites is able to generate high-performing and diverse levels.

Conclusions and Future Work

In this paper, we have done an in-depth evaluation of IC MAP-Elites by analyzing the expressive range of all the possible pairs of dimensions, their relation to other dimensions and the fitness. Our results indicate specific dimensions in level generation, such as when using NMP, NSP, LENIENCY, or Symmetry, that foster greater exploration. The exploration is not only fostered in their respective dimensions (in avg. 54%, 61%, 57%, 57%, respectively, when used) but also in all the others due to the diverse individuals generated within the respective dimensions as shown in Figure 2c. As observed on Figure 6, aesthetic feature dimensions such as *Similarity* and *Symmetry* have an impact when used or not in the search. When not using *Symmetry*, the search does not explore high levels of symmetry, disregarding to some extent that aesthetic feature in favor of exploring the other feature dimensions. Moreover, *Similarity* has a high correlation with fitness as observed in Figure 6 and depending on the designer's objective, this might affect positively or negatively since there will be a strong bias towards highly similar individuals. In contrast, *is* seems to be more robust in the fitness landscape and the exploration of other dimensions because it captures the properties of the target room rather than its aesthetics.

Regarding the bias in the exploration when using IC MAP-Elites together with continuous and adaptive evolution, our results show that the generated content is highly diverse with dense areas along most of the searched space, which is shown

as well in Table 1. This means that due to the diversity pressure imposed by IC MAP-Elites, the search is unlikely to be biased towards creating content that is similar in the feature dimensions' scores of the target room. Yet IC MAP-Elites adapts to the target room and generates high-performing individuals along the rest of the space in the other dimensions, especially in those explicitly used in the search.

To further assess the algorithm, we ran an experiment using all possible dimensions (Figure 5a) rather than specific pairs to observe the exploration and exploitation of the algorithm when not using specific pair of dimensions. As expected, it explores a substantial area of the search space in all dimensions (in avg. 51.7%) but surprisingly, the search results in the exploitation of sub-optimal individuals in all dimensions with an avg. fitness of 0.78 as shown in Figure 6d. While using pair of dimensions results on 15.24% reduced exploration, the exploitation seems to be fairly spread among the explored space, visible in Figure 5b and c, and the density related to fitness is focused on high-performing individuals Figure 6a-c. This points towards that there are difficulties in (1) **fully exploring** the space when using a pair of dimensions even when the exploitation is distributed and focused on high-performing individuals, and in (2) **exploiting the promising areas** of the search when using a higher range of dimensions even when the space is vastly explored.

In addition, based on our experiments, such difficulties are exacerbated since the exploration stagnates and keeps exploiting the same areas after ~1000 generations, yet finding novel individuals. This happens regardless of the number of dimensions, which dimensions, or the target room. Our findings point to challenges in the *selection step* of IC MAP-Elites, which selects cells uniformly random. Exploring different methods for the selection of cells and individuals is a promising future step. For instance, Gravina et al. [29] explored how four divergent search algorithms guide cells' selection. benefit MAP-Elites standard selection method.

Furthermore, preliminary experiments seem to indicate that some dimensions, which are more explorative (e.g., NMP, NSP, Leniency, Symmetry), are more robust to changes in the target room, exploring similar areas of the search space regardless of the target, which can be observed in Figure 5 (b) and (c). This points towards dimensions that would make the algorithm more robust to changes, reinforcing its adaptability feature. Further experiments are needed to analyze how different dimensions are better at adapting to continuous changes in the target room, which would also indicate better stability in the search. Along these lines, further evaluation is needed with human designers to assess and explore whether

IC MAP-Elites is beneficial for the MI-CC workflow and interaction.

Currently, IC MAP-Elites is only used in EDD on a per room basis. Future work should consider the whole dungeon for the evolution procedure and further develop for the generation of complete dungeons using holistic metrics to evaluate the dungeon.

In conclusion, based on our results, IC MAP-Elites generate more diverse levels while retaining high-quality among them, which would result in richer and more options and suggestions to designers. Our experiments show that which dimensions are used significantly impact the search space, fostering the search of high-quality individuals within the selected dimensions while not discouraging exploration in other dimensions. This means that by editing their levels and choosing which dimension IC MAP-Elites should focus on, the designers will be given more meaningful choices and interactions.

References

- [1] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018. 189
- [2] M. Toy, “*Rogue*.” Game [PC], 1980. 189
- [3] D. Braben and I. Bell, “*Elite*.” Game [PC], 1984. 189
- [4] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 172–186, Sept. 2011. 189
- [5] N. Shaker, G. N. Yannakakis, J. Togelius, M. Nicolau, and M. O’neill, “Evolving personalized content for super mario bros using grammatical evolution.,” in *AIIDE*, 2012. 189
- [6] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the Galactic Arms Race video game,” in *2009 IEEE Symposium on Computational Intelligence and Games*, Sept. 2009. 189
- [7] J. Dormans, “*Unexplored*.” Game [PC], 2017. 189
- [8] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, ACM, 2018. 189, 190, 193, 197
- [9] M. C. Green, G. A. B. Barros, A. Liapis, and J. Togelius, “DATA agent,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, (New York, NY, USA), pp. 1–10, ACM, 2018. 189
- [10] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, “Can computers foster human users’ creativity? theory and praxis of mixed-initiative co-creativity,” *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016. 189

- [11] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, pp. 1–8, 2014. 189
- [12] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, “Pitako-recommending game design elements in cicero,” in *2019 IEEE Conference on Games (CoG)*, IEEE, 2019. 189
- [13] N. Shaker, M. Shaker, and J. Togelius, “Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.,” in *AIIDE*, 2013. 189
- [14] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 189
- [15] T. Machado, A. Nealen, and J. Togelius, “Cicero: Computationally intelligent collaborative environment for game and level design,” in *3rd workshop on Computational Creativity and Games (CCGW) at the 8th International Conference on Computational Creativity (ICCC’17)*, pp. 1–8, 2017. 189
- [16] A. Liapis, G. N. Yannakakis, and J. Togelius, “Generating Map Sketches for Strategy Games,” in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS, Springer, 2013. 189
- [17] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016. 189
- [18] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, “Procedural content generation through quality diversity,” in *2019 IEEE Conference on Games (CoG)*, 2019. 190
- [19] J. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” 2015. 190, 193, 203
- [20] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008. 190
- [21] A. Khalifa, M. C. Green, G. Barros, and J. Togelius, “Intentional computational level design,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO ’19, (New York, NY, USA), pp. 796–803, ACM, 2019. 190

- [22] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 190, 191, 192
- [23] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 190, 191, 192, 193, 194, 198
- [24] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG ’17, (New York, NY, USA), ACM, 2017. 190, 191, 193, 196
- [25] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Mixed-initiative procedural generation of dungeons using game design patterns,” in *Proc. IEEE Conf. Computational Intelligence and Games (CIG)*, 2017. 190, 191, 193
- [26] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 190, 201
- [27] M. Cook, S. Colton, J. Gow, and G. Smith, “General analytical techniques for parameter-based procedural content generators,” in *2019 IEEE Conference on Games (CoG)*, Aug 2019. 190, 191
- [28] M. Cook, J. Gow, and S. Colton, “Towards the automatic optimisation of procedural content generators,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, Sep. 2016. 190
- [29] D. Gravina, A. Liapis, and G. N. Yannakakis, “Blending notions of diversity for map-elites,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’19, (New York, NY, USA), pp. 117–118, ACM, 2019. 190, 208
- [30] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016. 191
- [31] G. Smith and J. Whitehead, “Analyzing the expressive range of a level generator,” in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, (New York, NY, USA), ACM, 2010. 191, 201

- [32] A. Summerville, "Expanding expressive range: Evaluation methodologies for procedural content generation," in *AIIDE*, pp. 116–122, 2018. 191
- [33] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, "A comparative evaluation of procedural level generators in the mario ai framework," in *Proceedings of the 9th International Conference on the Foundations of Digital Games*, FDG '14, pp. 1–8, 2014. 191
- [34] M. Cook, J. Gow, and S. Colton, "Danesh: Helping bridge the gap between procedural generators and their output," in *Proceedings of the 7th Workshop on Procedural Content Generation*, FDG, pp. 1–16, 2016. 191
- [35] J. Mariño and L. Lelis, "A computational model based on symmetry for generating visually pleasing maps of platform games," in *AIIDE*, 2016. 194
- [36] A. Liapis, G. N. Yannakakis, and J. Togelius, "Adapting models of visual aesthetics for personalized content creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012. 194
- [37] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001. 198
- [38] E. McMillen and F. Himsl, "*The Binding of Isaac*." Game [PC], 2011. 200

PAPER VI - TO MAKE SENSE OF PROCEDURALLY GENERATED DUNGEONS

Simon Tolinsson, Alexander Flodhag, Alberto Alvarez, and Jose Font

ABSTRACT

With the growth of procedural content generation in game development, there is a need for a viable generative method to give context and make sense of the content within game space. We propose procedural narrative as context through objectives, as a useful means to structure content in games. In this paper, we present and describe an artifact developed as a sub-system to the Evolutionary Dungeon Designer (EDD) that procedurally generates objectives for the dungeons created with the tool. The quality of the content within rooms is used to generate objectives, and together with the distributions and design of the dungeon, main and side objectives are formed to maximize the usage of game space and create a proper context.

PUBLISHED IN

Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play ACM, 2020.

TO MAKE SENSE OF PROCEDURALLY GENERATED DUNGEONS

Introduction

Procedural content generation (PCG) has found itself in the spotlight within game development with games such as Minecraft [1], No Man's Sky [2], and Spelunky [3], improving replayability, reducing the developers' workload, and fostering the designers' creativity [4–6]. However, some type of narrative or context is required to make sense of the PCG content when implemented into the game space [7]. An example of narrative is objectives. If there is an objective within the content, for example, finding a sacred gem in a dangerous dungeon, then that creates interaction between the user and the content, which creates the needed context.

The Evolutionary Dungeon Designer (EDD) is a mixed-initiative design tool used for creating and generating dungeons [8]. This paper gathers the first step towards implementing a sub-system for EDD, which procedurally generates objectives for dungeons. The current sub-system gathers and continuously adapts to the designer's dungeon to place different objectives based on the rooms' content. Through this, the designer focuses on creating the dungeon while seamlessly, they are provided with the different generated objectives. We evaluate the artifact's utility, quality, and efficacy based on how well the objectives represent the layout of the dungeon with experimental scenarios.

Related Work

The Evolutionary Dungeon Designer

EDD is a mixed-initiative tool for designers to create dungeons as a set of interconnected tile-based rooms [8]. Each tile in a room can be modified to represent different types of paths, obstacles or rewards, and are used to form inventorial (Fig. 1.b) or spatial (Fig. 1.c) micro patterns. These micro patterns can be further combined to form meso patterns (Fig. 1.d) such as treasure or

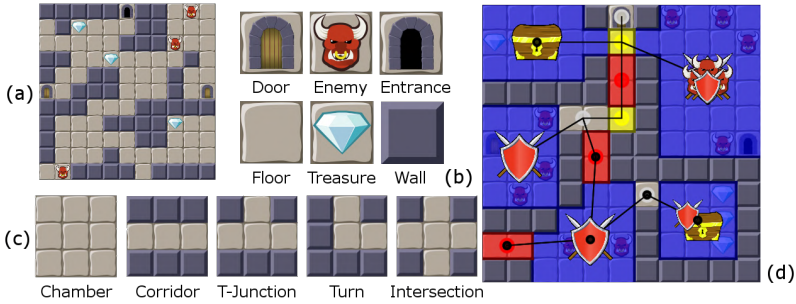


Figure 1: The main components in EDD. (a) A basic room, (b) different placeable tiles, (c) micro patterns and (d) meso patterns [10].

guarded rooms. Furthermore, as the designer creates rooms, EDD dynamically offers procedurally generated room suggestions through the Interactive Constrained MAP-Elites, using such patterns as evaluation and continuously adapting to the designer’s design [9].

Procedural Generation of Game Narrative

Interactivity and narrative have conflicting demands [11]. With narrative, the author decides the direction of the flow, while interactivity turns to the player for motive power. Straying from the author’s path may make for a less satisfying story, but restricting the player’s freedom of actions will have the same effect on the game. But game designers are not only storytellers, but they also sculpt and design game worlds and spaces.

Generated content needs context in the game space. A lack of context may negatively affect user experience, with the content being perceived as empty or meaningless [7]. The limitations of a story are related to the quest combinations available. By understanding the structure of quests, we can also understand the limits and potential of these kinds of games and how to create rich, open game worlds and tell interesting stories within them [12].

The common factor with objectives in games is to provide the player a reason to further progress through the game [7]. When generating content for game space, narrative, or context, needs to be generated as well. In action-adventure games, the level design is essential, and when procedurally generating levels for these games, it is best to break down the generation process in two steps, one for generating game space and one for generating missions [13].

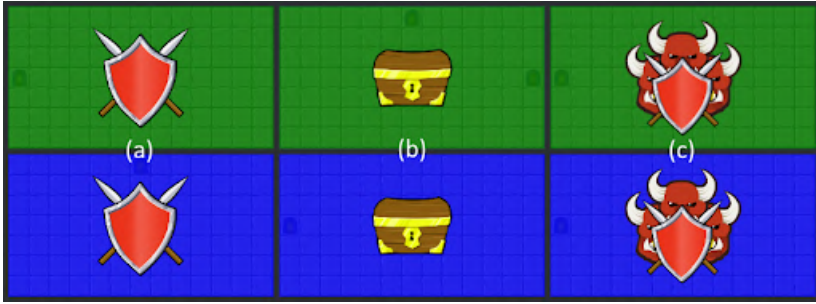


Figure 2: Every type of dungeon objective as both main objective (green) and side objective (blue). The icons represent the different types of objectives (a) “Defeat the enemies”, (b) “Find the treasure” and (c) “Defeat the boss”.

Charbitat bases narrative generation on sets of tiles, which partitions the game space and creates a graph that keeps track of the player’s position. Through this, the system evaluates new possible objectives to generate that would suit better. This evaluation takes in mind previous objectives and actions done by the player, resulting in a more adaptive experience while also increasing the replayability [7].

Procedural narrative generation is often approached split into two tasks, plot and space, either automatically or manually generated [13–16]. The plot is defined as a set of events with an overall structure that represents both the temporal ordering and the causal relations between the events. Space includes the characters, settings, props, and anything which is present either physically or abstractly in the space of the narrative. By generating space, they also generate context for it, thus creating a unique narrative for each possible outcome of the generative process.

Generating Objectives for Dungeons

Using the room’s meso patterns and their qualities, each room is assigned an objective described in Figure 2: defeat the enemies, find the treasure, defeat the boss, except the initial room, which is always excluded to avoid placing objectives where the player enters the dungeon. When all rooms have been assigned an objective, we calculate the number of objectives N_{obj} needed for the dungeon based on its size and layout. Furthermore, the number of objectives is calculated as $N_{obj} = \max(1, DE + ((R - DE)/K))$, where DE is the number of dead ends, R is the total number of rooms, and K is an adjusting variable. High values for K lower the number of objectives per normal (non-dead end) rooms. The designer can trigger the objective generation at any time in EDD by pressing a *toggle objectives* button.

All objectives are then sorted due to their relevance. The most relevant objective is set as the only main objective of the dungeon. Side objectives are subsequently assigned in descendant relevance order until the amount of objectives needed is reached. The sorting algorithm for objective relevance evaluation calculates the following metrics:

Dead end.

Dead ends have a higher chance of hosting interesting content for the player [7]. Therefore, to not make the outskirts of the dungeon feel meaningless, dead ends are prioritized locations for objective placement.

Distance to the player.

Placing all objectives close to the start position, preventing full space exploration, would break the players' immersion [11]. Therefore, large distances (measured in rooms) from the player are encouraged when placing objectives.

Connectivity.

Rooms connected to many rooms have a higher chance of the player passing through them than those with fewer connections. Therefore, to foster the exploration of these rooms, rooms with fewer connections to other parts of the dungeon will be prioritized to have an objective.

Quality.

The objectives are based on the existing meso patterns in the room. Each meso pattern receives a quality score [17], that when combined, create the quality of the room.

Objectives are then sorted in sequential steps. The policy for, any given pair of objectives, choosing the more relevant one, is:

1. Return the objective that is set in a dead end. If both are, or neither of them is, then
2. return the objective with the largest distance to the player. If both lie at the same distance, then
3. return the objective with lower connectivity. In case of a tie, then
4. return the objective with the highest quality score.

Results and Discussion

We have carried out a total of 14 simulations in EDD for testing the objective generation with a representative set of layouts, room sizes, and content. In all figures and due to its importance in the calculations, the initial room is highlighted in yellow. Out of experimentation, K was set to 4, meaning that one additional objective is generated per every 4 normal rooms in the dungeon.

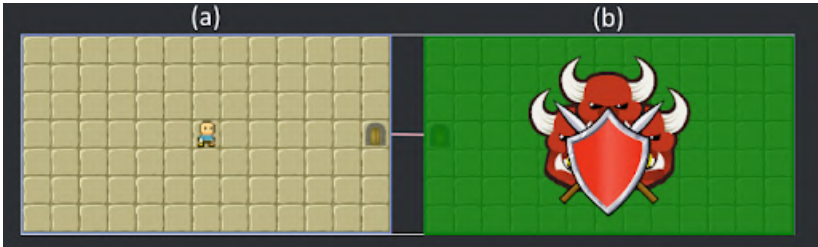


Figure 3: The most simplistic layout of a dungeon with a main objective (green).

Figure 3 shows the simplest scenario (two empty interconnected rooms), where the dead end (b) turns into a "Defeat the boss" main objective, leaving the initial room (a) as is.



Figure 4: Small single-path dungeon layouts.

Figure 4 shows two different small sequential scenarios. In simulation (a), the initial room is the leftmost one, and the two rightmost rooms become the side objective and the main objective. This specific order makes use of all the available game space. However, in simulation (b), the initial room is the second to the left. The system adapts to this change by placing the side objective to the left of the start position to use both dead ends and maximizing the game space usage.

Figure 5 shows two small scenarios without any dead ends. Only one objective (the



Figure 5: Small circular dungeons.

main one) is placed under these configurations. In both cases, to utilize most of the dungeon layout, the room on the opposing side of the initial room gets assigned with the main objective.

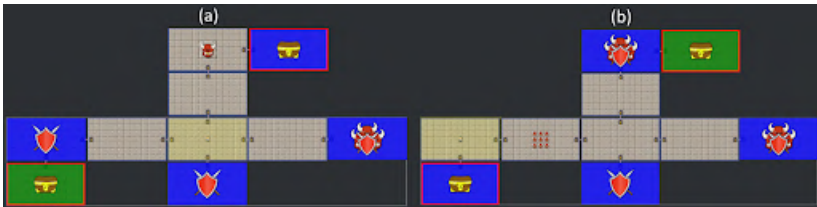


Figure 6: Large dungeon layouts with several dead ends.

Figure 6 shows results in two larger scenarios with several dead ends. In both cases, the main objective is placed in the furthest dead end from the start position, and side objectives with identical distance and connectivity scores are chosen according to their quality score. Notice how a similar layout in (b) places main and side objectives differently based on a different start position, trying to maximizing space usage.

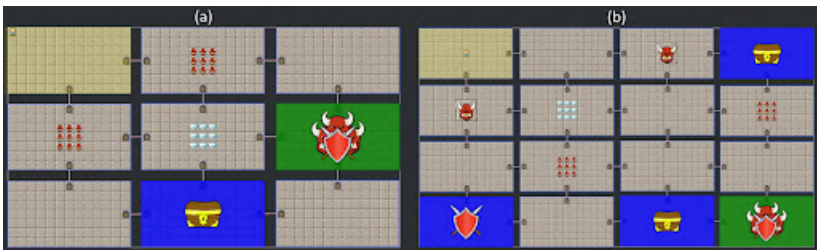


Figure 7: Two large circular dungeon layouts without dead ends.

Figure 7 generates objectives for larger circular dungeons with no dead ends and (a) no content in any of the corners, and (b) corner rooms with meso patterns. In

(a), the lack of content in the corner rooms makes the system choose objectives in the neighboring rooms to the furthest corner. Being both equally distant from the initial room, the “Defeat the boss” has a higher quality and is then marked as the main goal. In (b), the corners of the dungeon layout contain content, and the system makes use of this to generate objectives in every corner of the dungeon to maximize the usage of game space. The remaining non-corner “Find the treasure” side objective is prioritized over the other rooms without an objective based on its distance from the initial room.



Figure 8: Large circular dungeons layout with dead ends.

Figure 8 introduces large circular dungeons with dead ends. In (a), the initial room is part of the circular center of the dungeon layout, and the system generates objectives in the various dead ends of the layout to utilize the game space. In addition, there is a final side objective of type “Defeat the boss” which is prioritized because of both its distance from the initial room and its lower connectivity.

In (b), the system adapts to the placement of the initial room in the dead end that hosted the main objective in (a). The main objective is relocated to another dead end, being one side objective now placed inside the inner circular structure of the layout. “Defeat the boss” is still a side objective since it is connected to fewer rooms than the remaining non-objective rooms, exhibiting the relevance order introduced in section Generating Objectives for Dungeons.

The dungeon layout in Figure 9 is a variation of the layout in Figure 8, now showing how the system reacts to different types and sizes of dungeon layouts with minor changes to the dungeon content. In simulation (a), we changed the different rooms’ sizes to show that the system does not consider the size as a factor when evaluating and assigning objectives. Therefore, the objectives in 9.a) are nearly identical to 8.b). The only difference is that the “Defeat the enemies” objective in the circular structure is removed (see the red-bordered rooms in 8.b) and 9.a)). The reason is that the two middle rooms in the circular structure are now combined into one big room, thus reducing N_{obj} from 5 to 4. In (b), the big room in the middle is now

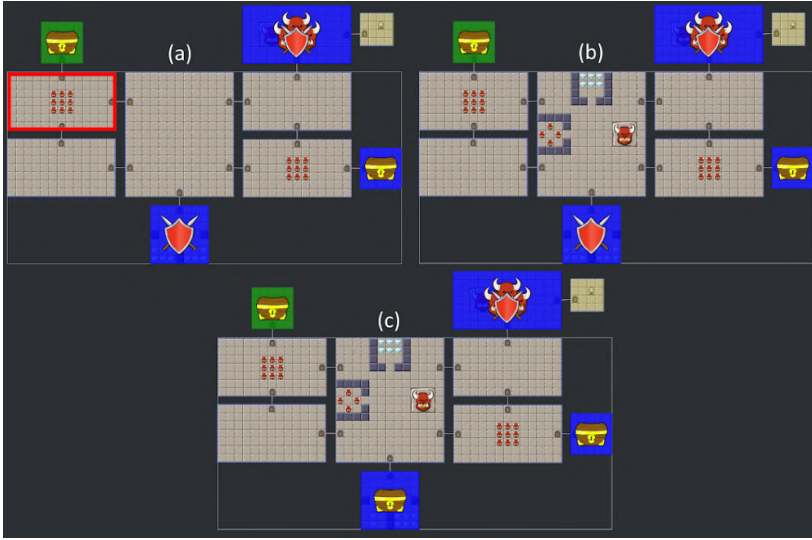


Figure 9: A large circular dungeon layout with dead ends, rooms of different sizes and varying content.

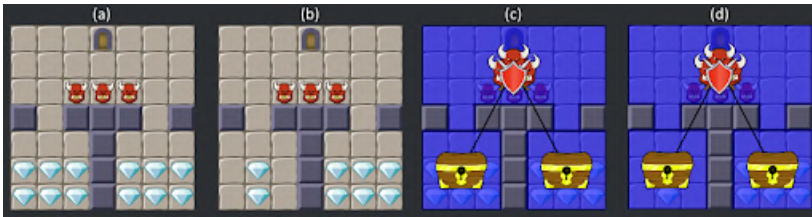


Figure 10: Two different room designs shown with and without meso patterns toggled. (a) and (c) as well as (b) and (d) are the same design. The two room designs represent the bottom room from Figure 9.b) and c), respectively.

filled with several meso patterns to show that the system does not prioritize the amount of content in a room when evaluating the dungeon objectives.

In (c), we showcase how the system generates objectives based on the content in the dungeon, showing how the designer is in control of what objectives are created. The difference between (b) and (c) is the side objective at the bottom room of the dungeon, depicted in Figure 10.a) and b), respectively. Figure 10.c) and d) are the meso pattern representation for Figure 10.a) and b), respectively. The meso pattern distribution is the same, but the quality of the room as a guarded treasure pattern is lower than its quality as a treasure room. Therefore, swapping between a) and b) alters the nature of the objective in the room, and a "Find the treasure" objective is placed at the bottom of 9.c) instead.

Conclusions and Future Work

We have developed a sub-system integrated into EDD that generates suitable objectives based on dungeon layouts created in a mixed-initiative environment. This integration has been carried out in a harmonic way [18] with EDD's already existing functionalities. The developed artifact also enhances the mixed-initiative creative loop in EDD, and helps the designer to visually validate their creation in terms of narrative.

These contributions open a promising line of research on procedural narrative in mixed-initiative environments. The next steps will be adding a coherent story that ties all objectives together, as well as articulating them by means of "quest givers" that offer different starting points for each objective. Ultimately, we will make use of all these pieces to engineer a procedural narrative generator that intertwines story, objectives, characters, and map. We would conduct a user study to validate the resulting worlds with game designers and players.

References

- [1] Mojang Studios, “*Minecraft*.” Game [PC], May 2009. Microsoft Studios, Washington, US. Last played October 2019. 217
- [2] Hello Games, “*No Man’s Sky*.” Game [PC], August 2016. Sony Interactive Entertainment, Tokyo, Japan. Last played March 2017. 217
- [3] Derek Yu and Mossmouth, LLC, “*Spelunky*.” Game [PC], December 2008. Mossmouth, California, US. Last played December 2019. 217
- [4] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural content generation in games*. Springer, 2016. 217
- [5] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the Galactic Arms Race video game,” in *2009 IEEE Symposium on Computational Intelligence and Games*, Sept. 2009. 217
- [6] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, (New York, NY, USA), pp. 50:1–50:8, ACM, 2018. 217
- [7] A. Calvin and N. Michael, “The quest in a generated world,” in *DiGRA གྷ - Proceedings of the 2007 DiGRA International Conference: Situated Play*, The University of Tokyo, September 2007. 217, 218, 219, 220
- [8] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, 2019. 217
- [9] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *arXiv: 2003.03377*, 2020. 218

- [10] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, "Assessing aesthetic criteria in the evolutionary dungeon designer," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG '18, (New York, NY, USA), pp. 44:1–44:4, ACM, 2018. 218
- [11] H. Jenkins, "Game design as narrative," *Computer*, vol. 44, no. 53, pp. 118–130, 2004. 218, 220
- [12] E. Aarseth, "From hunt the wumpus to everquest: introduction to quest theory," in *International Conference on Entertainment Computing*, pp. 496–506, Springer, 2005. 218
- [13] J. Dormans and S. Bakkes, "Generating missions and spaces for adaptable play experiences," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 216–228, 2011. 218, 219
- [14] B. Kybartas and R. Bidarra, "A survey on story generation techniques for authoring computational narratives," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 239–253, 2016. 219
- [15] A. M. Abuzurairq, A. Ferguson, and P. Pasquier, "Taksim: A constrained graph partitioning framework for procedural content generation," in *2019 IEEE Conference on Games (CoG)*, pp. 1–8, 2019. 219
- [16] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, "Toward supporting stories with procedurally generated game worlds," in *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2011*, pp. 297–304, 2011. 219
- [17] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Towards pattern-based mixed-initiative dungeon generation," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG '17, (New York, NY, USA), pp. 74:1–74:10, ACM, 2017. 220
- [18] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, "Orchestrating game generation," *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2018. 225

PAPER VII - QUESTGRAM [QG]: TOWARD A MIXED-INITIATIVE QUEST GENERATION TOOL

Alberto Alvarez, Eric Grevillius, Elin Olsson, and Jose Font

ABSTRACT

Quests are a core element in many games, especially *role-playing* and *adventure* games, where quests drive the gameplay and story, engage the player in the game's narrative, and in most cases, act as a bridge between different game elements. The automatic generation of quests and objectives is an interesting challenge since this can extend the lifetime of games such as in *Skyrim*, or can help create unique experiences such as in *AI Dungeon*. This work presents Questgram [Qg], a mixed-initiative prototype tool for creating quests using grammars combined in a mixed-initiative level design tool. We evaluated our tool quantitatively by assessing the generated quests and qualitatively through a small user study. Human designers evaluated the system by creating quests *manually*, *automatically*, and through *mixed-initiative*. Our results show the Questgram's potential, which creates diverse, valid, and interesting quests using quest patterns. Likewise, it helps engage designers in the quest design process, fosters their creativity by inspiring them, and enhance the level generation facet of the Evolutionary Dungeon Designer with steps towards intertwining both level and quest design.

PUBLISHED IN

The 16th International Conference on the Foundations of Digital Games (FDG), ACM, 2021.

Questgram [Qg]: Toward a Mixed-Initiative Quest Generation Tool

Introduction

Defining quests and related concepts have been the focus of considerable research, where quests have been related to tasks, challenges, rewards, or as a storytelling device adding nuances to what a quest is [1–4]. Most games have some quest driving the game’s plot and gameplay. Adventure games, action-adventure games, and role-playing games (RPG) are among the main genres using quests [5], where most of these genres take place or containing some type of dungeon such as *The Legend of Zelda*, *Skyrim*, or *The Binding of Isaac*. Dungeons as game content can be defined as a single level or set of levels containing enemies, treasures, hidden passages, puzzles, decorations, or Non-playable characters (NPC), thus creating space that allows the player to explore the unknown areas [6]. Dungeons are a popular level design, especially within PCG [7–9], where it has been present ever since the 1970s in games such as *Rogue*.

The increasing usage of Procedural Content Generation (PCG) in both research and industry [9, 10] has shown successful results regarding the efficiency of the game development process [11] but also to generate a big amount of variation in games, increasing their replayability [12]. PCG can generate game content quickly such as missions and levels [13], content adapted to players [14], or data-driven generation [15, 16]. Narrative and quest generation as objectives and goals has also been the focus of PCG [17–19], where the aim has been to capture and use quest concepts and patterns to approach the generation, such as the work by Trenton et al. [3], Kreminski and Wardrip-fruin [20], Smith et al. [21] or Doran and Parberry [1].

Nevertheless, much of the content is still best made by humans, especially when subjective evaluations are needed [7]. To cope with this, one could use a mixed-initiative approach. Mixed-initiative Co-creativity (MI-CC) was introduced by Yannakakis et al., where both human and AI co-create and -design some game facet

with a proactive initiative [22]. MI-CC has been explored mainly for level design in tools such as the Sentient Sketchbook [23], Tanagra [24], Morai Maker [25], or the Evolutionary Dungeon Designer (EDD) [26]. EDD lets the user design interconnected room in a dungeon while receiving room suggestions adapted to their creation. Questgram is implemented in EDD, taking advantage of its level design capabilities and mixed-initiative approach.

This research takes the quest analysis, quest patterns, and quest grammars identified by Doran and Parberry [1], implements it in EDD, adapts it in a mixed-initiative approach for the creation of quest sequences, and extends it to work with level generation. The designer can create quests by adding manually available quest actions in a quest sequence and receive suggestions from the quest grammar that they might use to continue the quest, replace some part of the quest, or get inspiration to continue their quest. The available quest actions are related to the current dungeon layout. If modifying such dungeon renders invalid current quest parts, the designer is prompted to fix the quest manually or using actions suggested by the system. The system was evaluated quantitatively by assessing the diversity and incidence of quest actions, and qualitatively through a user study evaluating the experience, usability and suitability of the system.

Related Work

Howard defined quests as "... a conceptual bridge that can help to join together many two-part or binary pairs [...] these include game and narrative, gaming and literature, technology and mythology and meaning and action [5]". Howard argues that quests unify both meaning and action, where meaning hails from strategic actions with thematic, narrative, and personal implications; and actions being those that are meaningful for the player on the level of ideas, personal ambitions, benefits to society, and spiritual authenticity [5]. Aarseth describes quests as concrete and attainable goals, and such can be hierarchic, concurrent, serial, or a combination of those. Further, Aarseth describes three basic quest types *Time-*, *Place-*, and *Objective-oriented*, which can also be combined to form seven different quest types [27]. Questgram is based on the quest analysis and proposed grammar by Doran and Parberry, constructed and extracted from analyzing over 750 quests from four RPGs and where they defined quests as a task given to the player that challenges them to complete some goals in exchange for some reward [1]. While helpful to understand quests as a whole, these definitions create a sense of ambiguity over different concepts surrounding quests. Yu et al. [4] proposed a generic quest definition in games that aims at unifying related concepts that appear in most of other's work, clearing ambiguity and easing it's use in PCG

quest generation tools. Formally, they define a quest as $Q = \langle T, \leq, R \rangle$, where a quest Q is a partially ordered set \leq of tasks T to be done to receive one or more rewards from a set R , which usually are in-game items.

Story and Quest Generation

Quests are fundamental elements in most games, driving the plot and player actions and providing goals and tasks to engage players with the game and the narrative. Doran and Parberry analyzed quests in four RPGs and found nine different "motivations" from NPC's, which resulted together with a specific strategy in a "verb-noun" pair, for example, "steal supplies" or "attack enemy". They used this grammar to generate quests while the user chose between nine identified NPC motivations [1]. Based on Doran and Parberry's action classification, Breault et al. developed an engine capable of creating quests similar to human-made ones, and since the engine generates quests based on the world state at the time of generation, the creation of possible quests increases as the game progresses [2].

One key characteristic in games is that they are interactive, and as such, can present choices to players. However, quests do not tend to provide such choice, especially in RPGs; rather, it is common that they are limited as a series of steps to follow. An interesting approach is Questbrowser [28], a quest design brainstorming tool where the designer can query the system for ideas, alternatives, and possibilities on elements or concepts that foster designers' creativity and help make quests playable (i.e., adding choice for players). However, presenting choices to players could create competing objectives for designers as they want to impose their narrative but, at the same time, want to create adaptable experiences for players.

Planning algorithms are a common technique to compose stories and quests meaningfully and with some partial-ordering [29, 30], focusing and optimizing character believability together with multi-agent systems [31, 32], replicating common quests and quest patterns [1, 3, 33], or identifying fundamental units and assembling them based on various pre-conditions [20, 34]. Kreminski and Wardrip-Fruin [20] mapped and compared multiple storylets-based systems and proposed the use of storylets, which are discrete, atomic, and recombinable narrative chunks, to assemble narratives based on a set of preconditions to create different narrative structures. Storylets were used by Garbe et al. in the StoryAssembler to generate dynamic narratives, which attempts to create a valid story with a planner using a set of provided storylets and storytelling goals that the planner uses as objective [34].

Moreover, Questgram functions within EDD's level design tool, which means both would function in relation to each other. Kybartas and Bidarra discussed the

relation between plot and space, focusing on the degree of automation for story elements. This resulted in six categories: *automated space*, *constrained space*, *space simulation*, *space modification*, and *manual space that builds a gradient between automatic and manual generation* [35]. Ashmore and Nitsche investigated a player-centric quest generation, where the progression through level generation is achieved with a "key and lock" structure, which results in a bridge between the generated space and the quests [36].

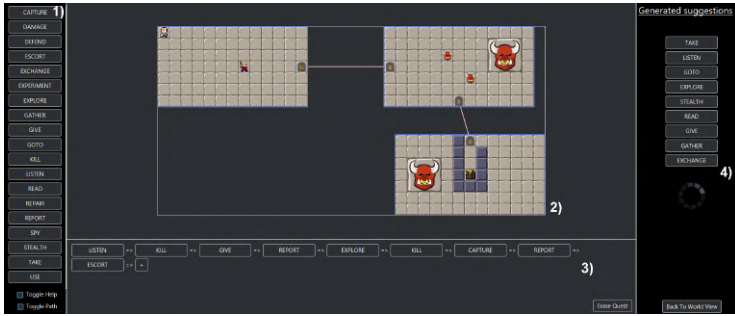
Dormans and Bakkes used two grammars to generate both missions and game space, where the latter was informed by the first. Missions are generated using graph grammars, creating a non-linear structure suited for exploration, while extended shape grammar generates the corresponding space required [13]. Further, Flodhag et al. use the information from levels co-created in *EDD* and categorize them based on the meso-patterns within them to present a set of main and side objectives to designers in their dungeon [18]. Hartsook et al. explore the creation of complete RPGs from a story created by either a computational system or human-authored and a set of player preferences. Their approach creates and represents game worlds as transition graphs based on a story composed of plot points, player's playstyle preferences, and designer constraints [37].

Mixed-Initiative Co-Creativity

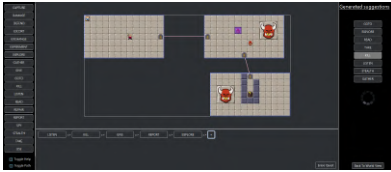
MI-CC is a paradigm where both humans and AI have a proactive initiative in the collaboration to co-create some creative content [22, 38]. Both human and AI leverage on each other's strengths to achieve the task and continuously negotiate to determine roles; thus, collaborating as a team [39]. One critical aspect of MI-CC systems is the link between these systems and theories of computational and human creativity, where a main focus of MI-CC is on fostering human's creativity while reducing their workload [40, 41].

The *Sentient Sketchbook* is an MI-CC tool for the co-creation of strategy games where the designer focused on creating low-resolution sketches, and the computational designer suggested variations generated with different evolutionary algorithms [23]. Cicero is an MI-CC system that helps designers create complete games using a recommender system and A-Priori to suggest what content might be added next regarding sprites, mechanics, rules, or interactions [42]. Another interesting MI-CC system is *Why Are We Like This? (WAWLT)* where two players can develop a story transcript while supported by an AI system with tools to inspect the story world and with suggestions to direct the plot [43].

EDD is an MI-CC system where the designer can create interconnected rooms



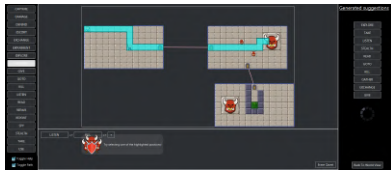
(a) Overview of the GUI used for the design of quests in EDD. 1) The possible quest actions, 2) the dungeon created thus far by the designer, 3) the quest sequence, and 4) the suggestions from the grammar.



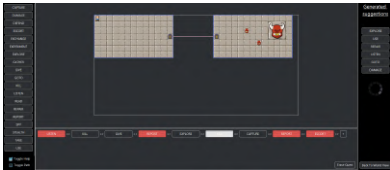
(b) An example quest sequence and the user attempting to select a suggested quest action



(c) Example of two empty and connected rooms, where most prerequisites for quest actions are not met.



(d) Example of the provided help for designers. A pop-up informs them of events, and in cyan, the A* path.



(e) Error in the quest sequence since the designer erased a room containing quest action tiles.

Figure 1: Visualization of the GUI used for creating quest sequences and different states

that compose a dungeon while receiving a set of diverse suggestions using the IC MAP-Elites evolutionary algorithm driven by level design patterns and considering the designer’s current design. The designer can interact with the suggestion system by locking tiles, editing their design, and selecting and interacting with hyper-parameters of IC MAP-Elites [26, 44]

Quest Generation

Questgram is a quest generation tool that lets the designer compose one long sequence of quest actions to create an overarching objective for the dungeon they are creating. These quest actions are based on the quest analysis and classification and produced grammar by Doran and Parberry [1]. Questgram builds on top of EDD extending its level design and generation capabilities with a mixed-initiative

Table 1: displaying the actions together with Doran and Parberry’s [1] prerequisites and how the actions and the previously mentioned prerequisites have been implemented in EDD. This indirectly explains the “unlocking” - describing what tiles that must be placed for an action to be available. Note that “Goto” & “Explore” do not have any special tile prerequisites besides available floor.

Action	Prerequisites in [1]	Prerequisites in EDD
Capture	“Somebody is there”	A NPC or boss/enemy must be placed.
Damage	“Somebody or something is there”	An item or NPC must be placed.
Defend	“Somebody or something is there”	An item or NPC must be placed.
Escort	“Somebody is there”	A NPC must be placed.
Exchange	“Somebody is there, they and you have something”	A NPC and an item must be placed (requires two positions).
Experiment	“Something is there”	An item must be placed.
Explore	“none”	An available floor tile.
Gather	“Something is there.”	An item must be placed.
Give	“Somebody is there, you have something.”	A NPC and an item must be placed (requires two positions).
Goto	“You know where to go and how to get there.”	An available floor tile.
Kill	“Somebody is there.”	A boss/enemy must be placed.
Listen	“Somebody is there.”	A NPC must be placed.
Read	“Somebody is there.”	A NPC must be placed.
Repair	“Somebody is there.”	A NPC must be placed.
Report	“Somebody is there.”	A NPC must be placed.
Spy	“Somebody or something is there.”	A NPC or boss/enemy must be placed.
Stealth	“Somebody is there.”	A NPC or boss/enemy must be placed.
Take	“Somebody is there, they have something.”	A NPC and an item must be placed (requires two positions).
Use	“There is something there.”	An item must be placed.

quest editor and takes advantage of its mixed-initiative perspective and the level design system.

EDD was extended with some key elements such as a new quest editor view depicted in figure 1a, and two new generic tiles; an NPC acting as quest giver and target, and a quest item, which is the subject of many quests. These two new tiles were kept as generic as possible for future systems to have the responsibility of handling what type of NPC and object should replace those, similarly as with the other tiles in EDD such as the generic enemy, boss, and treasure. These tiles, together with the pre-existing enemy and boss tiles, have been intertwined with the actions, resulting in the "unlocking" mechanism of different quests, which can be observed in table 1. It must be noted that while Questgram integrates and utilizes the different features and tiles of EDD’s level generation facet, there is no integration of the new tiles and quests with EDD’s evolutionary algorithm IC MAP-Elites [26]. This is left for future work.

While games can be either linear, semi-open, or open, with branching narratives and the design structured by the types of quests featured in a game [27], with concepts such as kernels and satellites [45]; our approach only allows for the creation of a single overarching quest.

Table 2: displaying the grammatical rules. The columns marked with asterisks are identified as “motivations” by Doran and Parberry [1], but are used as a starting point for the quests. The “<>” indicates the next production rule to be taken, and actions without “<>” is the terminating action.

Production rules	Actions
knowledge*	["<get>","<go_to>","give"], ["<spy>"], ["<go_to>","listen","<go_to>","report"], ["<get>","<go_to>","use","<go_to>","give"]
comfort*	["<get>","<go_to>","give"], ["<go_to>","damage","<go_to>","report"]
reputation*	["<get>","<go_to>","give"], ["<go_to>","<kill>","<go_to>","report"], ["<go_to>","<go_to>","report"]
serenity*	["<go_to>","damage"], ["<get>","<go_to>","use","<go_to>","give"], ["<get>","<go_to>","use","capture","<go_to>","give"], ["<go_to>","listen","<go_to>","report"], ["<go_to>","take","<go_to>","give"], ["<get>","<go_to>","give"], ["<go_to>","damage","escort","<go_to>","report"]
protection*	["<go_to>","damage","<go_to>","report"], ["<get>","<go_to>","use"], ["<go_to>","repair"], ["<get>","<go_to>","use"], ["<go_to>","damage"], ["<go_to>","repair"], ["<go_to>","defend"]
conquest*	["<go_to>","damage"], ["<go_to>","<steal>","<go_to>","give"]
wealth*	["<go_to>","<get>"], ["<go_to>","<steal>"], ["repair"]
ability*	["repair","use"], ["<get>","use"], ["use"], ["damage"], ["<get>","experiment"]
equipment*	["repair"], ["<get>","<go_to>","give"], ["<steal>"], ["<go_to>","exchange"]
subquest*	["<go_to>"], ["<go_to>","<QUEST>","go_to"]
go_to	["explore"], ["<learn>","go_to"]
learn	["<go_to>","<subquest>","listen"], ["<go_to>","<get>","read"], ["<get>","<subquest>","give","listen"]
get	["<steal>"], ["<go_to>","gather"], ["<go_to>","<get>","<go_to>","<subquest>","exchange"]
steal	["<go_to>","stealth","take"], ["<go_to>","<kill>","take"]
spy	["<go_to>","spy","<go_to>","report"]
capture	["<get>","<go_to>","capture"]
kill	["<go_to>","kill"]

Quest Actions

Doran and Parberry identified 19 different actions to be used as quest actions [1], which we implemented and where each has its contextual prerequisites to be able to add them. In some cases, we have decided regarding if the actor represented in the action is friendly, e.g., NPC, or hostile, based on the tool's nature and the levels created. These actions are available for both the grammar and the designer to create as many steps as wanted in the overarching quest. The actions, their original prerequisite, and the domain-specific prerequisite are depicted in table 1.

These actions can be added one after each other in any order by the designers allowing for combinations and quests outside of the possible grammar seen in table 2. Besides manual creation, the designer can instead pick a suggested action from the generated actions from the right panel, which offers the next action to be added to the quest. After deciding these options, the user will need to press the "+" button on the bottom panel, which will add the action to the quest sequence.

Quest Grammar

The system employs a generative grammar, specifically Lindenmayer Systems (L-Systems) [46] to generate the different set of quests using the production rules depicted in table 2. The production rules are divided into two categories: 1) motivations for NPCs to start a quest such as *knowledge* where the focus would be to create quests with more passive actions or *reputation* where the focus would be to kill some enemy to gain reputation with some NPC. 2) Non-motivation rules related to the development of quests (i.e., non-terminal symbols) such as "go_to" or "get". In table 2 non-terminal symbols are represented with "<" and ">", and terminal symbols simply list the action.

The system can be used for generating complete quests on its own, which select one of the NPC motivations as an axiom or with the designer in a mixed-initiative approach. As a mixed-initiative approach, the designer can manually create a quest sequence while the system uses the current quest sequence to suggest a set of valid quest items to the designer to continue the quest or replace a current action. Given that the production rules are invariable, there can be situations where the system cannot generate quests based on the quest sequence. This would result in the designer receiving feedback that the quest is not compatible with the grammar itself, giving the designer suggestions on how to continue and overcome this limitation.

Quest actions are suggested to continue the current sequence and append a new action at the end, or they can be used to replace an action in any position of the current sequence. For both, the system continuously produce quests using the

grammar and filters out those that do not match the designer's sequence up to the position where they wish to change a quest action. In this way, the designer can choose suggestions for either continuing and finishing the quest or replacing existing parts for other valid actions.

Workflow

The GUI for the quest generation system of EDD follows the same concept and design as the room editor for consistency. Placeable quest actions are at the left pane, and generated suggestions with the grammar are at the right pane. The whole dungeon can be seen on the center top pane, and at the center bottom, the designer can compose the quest. These parts can be explicitly seen in figure 1a.

The designer cannot add any quest action, which prerequisite has not been fulfilled yet as described in table 1 and exemplified in figure 1c. Quest actions need to be linked to some actual tile representation in the dungeon. For instance, a "KILL" action requires selecting an enemy, while the "GO TO" requires any floor tile. Therefore, once the designer adds a new quest action, they must choose which tile is linked to this, presented to the designer in green. Similarly, when a quest action is suggested, the system randomly picks an available tile shown in purple to the designer as shown in figure 1b.

Moreover, the sequence panel is displayed in fig. 7.4. The panel displays the actions the user has selected. To add a sequence to the list, the user needs to manually select an action and its desired position or select a suggested action. Both of these options require the user to press the "+" button manually. Similarly, each quest action in the sequence is clickable and interchangeable. If a quest action in the sequence is selected, the designer can exchange it by selecting the quest action desired from the action panel or from the newly suggested actions, or remove it.

Finally, the designer can toggle two different types of assistance. The first focuses on informing the designer of changes in the sequence due to a manually placed or automatically generated, and informs the designer of the tile that needs to be selected. The second assistance shows A* paths between the target tile of a selected quest action and the next target tile. Both assistance is depicted in figure 1d.

Evaluation

Questgram has undergone a two-fold evaluation, top down (expressivity analysis) and bottom up (user study), as suggested by Shaker et al [7].

Expressive Range Analysis

Expressive Range Analysis visualizes the expressivity and diversity of the generator and measures variations in the generated content according to specific metrics [47]. In our case, these metrics are quest length and actions. With them, we visualize each action's probability to be included in a quest of any given length and the existing dependencies between the actions and the grammar productions.

We ran the grammar using the dungeon seen in Figure 1a and created 100000 quests with a maximum length of 50 quest actions, although the system could create on average 146 long quests. We chose 50 quest actions because of the dungeon's size and what it could offer and because creating quests with more than 50 subsequent actions are highly unlikely to find in commercial games.

Figure 2 shows the results obtained from three different perspectives. Figure 2a is a heatmap that displays the chance (in %) for every quest action (row) to appear in a quest of a given length (column). This shows the most frequent quest actions for every quest length up to 50. E.g., quests with length 1, meaning that the complete quest sequence is composed of only one action, "Repair" is that action in 60% of the 100000 generated quests, "Damage" in 20%, and "Use" in another 20%. On the other hand, if the quest is of length 5, the quest contains "Explore" 46% of the time, "Take" 10%, "Kill" 7.8%, "Report" 7%, "Stealth" 6.2%, "Give" 5.2%, followed by much lower values for the remaining actions.

Figure 2b presents the chance (in %) for every action (row) to appear at any step of a quest (column), regardless its length. This heatmap shows how frequently a specific action is chosen at a given quest step and how this frequency varies as the quest length increases. For instance, on step 3, "Explore", "Take", "Gather", "Go_To", and "Report" are the most common quests actions. However, moving forward to step 20, "Explore", "Take", "Gather", and "Report" become less frequent, while "Go_To", "Listen", "Read", and "Give" become more common.

Finally, Figure 3 show the most commonly generated subsequences, with a minimum size of 3, that were produced over the 100000 generated quests.

Experiment Discussion

Results show "Explore" as the most common action among the generated quests. Its dominance ranges from short to long quests (Figure 2a), though it is noticeable how its chance to appear significantly drops down, from 87% to 24%, in the later stages of a quest (Figure 2b). The main cause for this high frequency of appearance might be that "Explore" has a quite easily fulfilled prerequisite: an available floor

tile. While most of the other actions require NPCs, items, or both, the existence of available floor tiles is several times higher than any of those elements. "Go_To" is the other action with such a simple prerequisite, and its chance to appear is also high. As opposed to "Explore", it raises from 0% to 28% in the later quest steps. Though both actions imply space exploration, "Explore" is more commonly used in the early stages of a quest, when the map remains uncharted, whereas "Go_To" gets used more in the later stages, where some map locations have been already visited. It is also remarkable that the first action in 87% of the quests is "Explore", while the only other actions that appear in the first step (in a much lower degree) are "Use", "Damage", and "Repair". No other actions are used as quest starters. This "Explore" and "Go_To" dominance can also be observed in table 2, where "Go_To" appears in 77% of the production rules, sometimes more than once per production, and "Explore" has a 50% chance to appear per "Go_To".

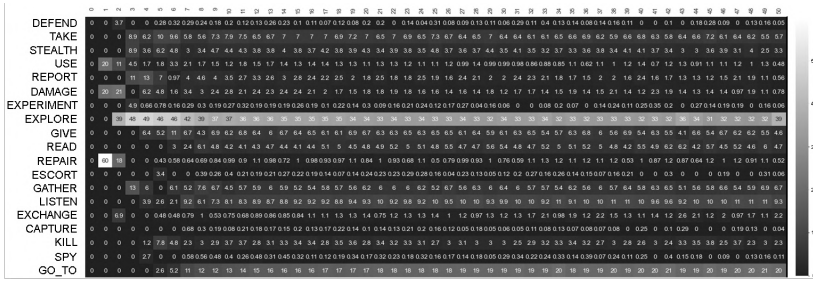
The appearance rate of the combat-related actions, "Damage" and "Kill", is relatively low, though their peak rates are located in shorter quests (Figure 2a). "Damage" has a 21% chance to appear in quests of length 3, whereas "Kill" has its peak at 7.8% in 5-step quests. This can be extended to other actions such as "Use", "Give", "Repair", "Gather", and "Exchange", suggesting that this subset of actions is much more likely to appear in short, quickly solvable quests. Nevertheless, all of them still appear in longer quests at stable rates, though movement actions are much more predominant in the long run.

Some actions are very underrepresented regardless of quest length or step number, as is the case for "Defend", "Report", "Experiment", "Escort", "Capture", and "Spy". This implies that these actions have very little chance to be suggested at any quest step, so it is more likely to end up in a quest if manually added by the designer. A future evaluation of the utility of these actions seems interesting in light of these results.

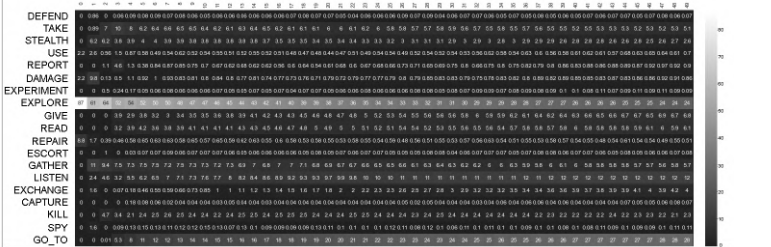
Finally, Figure 3 indicates a clear bias in the grammar towards exploration ("Explore" and "Go_To"), as one or both appear, at least once, in any of the most commonly generated subsequences. The actual relevance of these dominant actions should also be evaluated for the grammar's future development.

User Study

Six participants tested our tool following three pre-designed tasks and questionnaires to evaluate Questgram's usability, functionality, and usability. They were all given a document describing the study's purpose and aim, a brief introduction to EDD, and the interview overview. The users were then asked to complete three tasks



(a) Chance (in %) for every quest action (row) to appear in a quest of a given length (column).



(b) Chance (in %) for every action (row) to appear at the Nth step of a quest (column), regardless of its length.

Figure 2: Results from the Expressive Range Analysis

that covered the tool’s functionality and different approaches to creating quests. The tasks were to 1) manually create a quest, 2) automatically create a quest, and 3) create a quest through mixed-initiative. They were also asked to create a dungeon that suited their preferences and objectives before creating quests. The questionnaire consisted of 17 closed-ended questions, and the rest were open-ended. The interview began with a questionnaire with six questions about the users’ background and experience within game development and finish with questions about their experience and opinions on the tool. Both the questionnaire and interview followed guidelines described by Oates [48].

The participants were selected through convenience sampling and were game developers working in game and level design (2) and game development alumni (4), without any experience with EDD or mixed-initiative tools. Five out of six have played dungeon/adventure games and have developed some game with quests and missions, while only two out of six have developed dungeon style games.

Manual Quest Creation

Participants reported that the tool was easy to use, clear, intuitive, and while simple and basic, it had enough building blocks for them to create their objective quest.

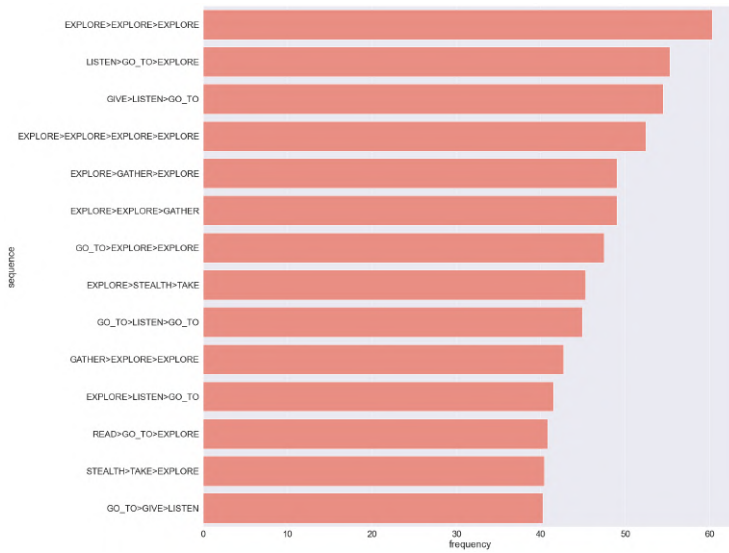


Figure 3: Most commonly generated subsequences, with a minimum size of 3.

Positive feedback was also given regarding the UI, integration with the rest of EDD functionalities, clarity of the quest action concerning the quest, and making the tool overall more interesting. However, some participants expressed confusion when quest sequences became too large as they would have preferred to separate the quests into sections or subquests. Another concern expressed was the inability to change the order of already created quests without redoing the whole quest sequence.

Automatic Quest Creation

Some participants reported that the system showed potential and a good addition to the manual creation, especially for the creation of side quests such as how similar systems work in *Skyrim*, and for learning how to use the tool as some kind of tutorial. Nevertheless, most participants remarked the system as random and illogical regarding random tile picks connected with quest actions as the system picked farther away NPC and targets with no purpose. In addition, participants felt that a system like this complicated the creation and took the freedom of creating their world and ideas.

Mixed-Initiative Quest Creation

The participants described the system as helpful and useful; pointing out that the main advantages and potentials were related to when they reached an inspiration

blockage as the designer could get inspired by the suggestions; to allow designers to focus on key parts of a quest sequence, and the speed gained to create quests "in just a matter of moments." While most feedback was positive, there were still concerns among participants regarding the system's cohesiveness as it could feel hard to make the suggestions cohesive with what the participant had in mind. Nevertheless, a majority of the participants experienced that both the manual and automatic complemented each other.

Automatic Suggestions

Participants generally described the automatic suggestions as useful to gain inspiration, keep the quest creation diverse, and learn what could be created, rather than useful to replace or add to their work. For instance, one participant said that "[the system] suggested to capture a monster which had thought about killing. The "capture" option might be more interesting and might have been an option I had otherwise overlooked." Similarly, another participant pointed out that the automatic suggestions "... were useful in getting inspiration for quests, and to learn the program and what kinds of quests I was actually able to make." Usually, designers have a predetermined idea on how they would like the narrative to unfold. However, based on the responses we received, the system gave a different perspective to the users on what they could create or how they could continue a quest, which makes the tool useful for brainstorming quest design similar to tools such as Questbrowser [28], albeit constrained to the possible quest actions.

Still, participants preferred to use the system as inspiration rather than effectively incorporating changes to the quests. This was mainly due to the suggestion not feeling cohesive enough with what participants created until then, and the random tiles picked for the quest action. For instance, receiving a "GO TO" suggestion to a random tile on the opposite side of the level and not near the player or the previous quest action.

Quest Actions

Quest actions and their goals were perceived as easy to understand and suitable for the type of game they were creating, except for "experiment," "stealth," and "spy," since they felt ambiguous and not clear for some participants. Fulfilling some quest actions' prerequisites was somewhat obscure, and some participants needed to go through trial and error to gain access to the quest action. However, once they fulfilled the prerequisites, it was clear and made sense in the context.

Usability

All participants described the tool as a useful addition for game developers when developing dungeon games but with different arguments and situations on when it would be useful. For instance, to create mundane quests in games with a grinding flow such as *Diablo*, to fast-prototype ideas and systems to give insights into what it might be possible and what might work, and to complement the creation of content and quest-design. Some participants expressed that making the game playable is a must to get the full benefit from the tool.

Creativity

Most of the participants reported that they experienced increased creativity when using mixed-initiative creation. The participants described that it helped when they got stuck, and it showed different alternatives and routes they did not previously consider. Further, one participant explained that they could make more creative decisions and not "staying safe" and adding "extra steps" without any effort. For instance, using a spy before a kill, thus prolonging the sequence by an extra action. Two participants highlighted that while they did not experience increased creativity, they saw the tool as useful when no new ideas are there given the possible "out of the box" suggestions.

Overall Experience and Missing Features

Some participants expressed that the tool was useful for someone in the gaming community, but it would be hard to grasp for someone unfamiliar with the concept. Further responses were that it was simple to work with, felt scalable, and the software's recommendation felt useful when designing a quest. Additional response from one participant, who, despite having no prior experience in either creating or playing dungeon games, said it was a fun introduction to the genre and that it was easy to learn and understand. Further, they explained that the simple workflow inspired them to continue creating. In addition, further responses were that the software was quick, feature-rich, simple to use, and got their creativity flowing.

In general, the participants expressed that more interaction with the quest sequence itself, such as changing the order of subsequences, adding quest actions in arbitrary parts, having separate quests, or knowing which quests were manually and automatically, would improve the system considerably.

User Study Discussion

Since we use the work by Doran and Parberry [1] as a base for the quest generation, this research indirectly tests their quest patterns and their applicability into a

mixed-initiative tool. We leveraged on these quest patterns similar as others have on quest patterns [3, 21], and how EDD leverages on level design patterns for the level generation [49, 50]. The use of quest patterns greatly improves the communication with the designers as they can use concepts they feel comfortable with and relate better to the content they create. All of our participants confirmed the previous statement by pointing out how the tool was straightforward, easy to understand with quests actions to be found in any other game type, and easy to use even if they had never used a mixed-initiative tool.

Furthermore, while relevant, the suggestions by the system felt impractical mainly and according to participants because the system randomly assigned tiles for the suggested quest action, which limited the tool's perceived usability. Another reason is the use of abstract quest actions. On the one hand, this allows us to disconnect the system from specific implementations and gameplay functionalities of the quests, creating and representing a more generic system. On the other hand, this resulted in a lack of thematic and concrete elements such as NPC roles or defined plot lines and plot elements to follow. These make it harder for designers to contextualize their creations and why the system recommends specific quest actions.

Nevertheless, the system was helpful for creativity support and design aid as the suggestions were used as an inspiration to what was possible and what to do next rather than using the actual suggestion. Participants only applied suggestions when the next step felt mundane, and the system suggested a logical position. Leveraging on the human designer for deciding location, while the system provides the quest actions that would follow a more typical quest based on the quest patterns would probably compose a better collaboration.

Some feature suggestions such as separating quests or reordering the quest sequence would have improved the user experience considerably, as after manually placing just a few quest actions, the quest became harder to approach. Even if the system was tested to generate 50 quest actions as explained in section Expressive Range Analysis, this might be impractical for human designers. An interesting suggestion was to use color tags or similar to understand which agent (Human or machine) created the quest. Then, designers could also use this as a way to understand decisions made by the system and for the system to create a model of the current designer's quests. This could also be used for a collaborative tool where several designers interact with each other and a centralized system; for a more crowdsourced approach such as the one proposed by Charity et al. [51].

Conclusions and Future Work

This paper presents Questgram [Qg], a quest generation tool with a mixed-initiative approach integrated into the Evolutionary Dungeon Designer. Questgram lets a human designer co-create an overarching quest that fits in a dungeon level, as the dungeon is being developed in the designer. Both map and quest are designed in parallel and with the suggestions provided by EDD. Quests make use of Doran and Parberry's quest structure as production rules in a grammar so that all quests are well-formed with respect to the grammar and the level landscape. We show results from a two-fold evaluation, an expressive range analysis, and a user study.

The expressive range analysis shows several dominant quest actions and structures, though all types of actions could be generated at a wide range of quest lengths. The mixed-initiative approach was positively met by the user study participants, along with the manual creation. However, automatic creation and automatic suggestions received a mixed response, mainly because of a random placement of position on the actions and the use of abstract quest actions. The tool's overall response was positive, and a majority of the participants reported increased creativity while using the tool. Many participants expressed its usability to gain inspiration, as a solution to inspiration blockages, and as a resource-efficient tool for game developers to use. None of the testers noticed the dominance of some actions detected in the expressive range analysis.

This inspirational use points towards the need to explore other fundamental and more useful ways to establish effective MI-CC workflows where systems can adapt and be effectively employed and used. For instance, some interesting future paths would be to explore the creation of more adaptive collaboration that considers the designer's style or to give more autonomy to the AI to have more participation in the creative process and its effects. Within this, one interesting area is the one of eXplainable AI for Designers [52] where the goal is to achieve system explainability to improve the collaboration and interaction between human and AIs.

This research sets the first step toward intertwined story and level mixed-initiative generation on EDD, and future work could be to incorporate quest elements as input in the level generation process so that quests and levels reciprocally influence their generative processes [53]. Furthermore, adding semantic evaluation on the generated suggestions would allow Questgram to generate interconnected quests that make sense as subsequent parts of an overarching story plot involving game elements, as well as adding a natural-language generation layer to enhance quests with the automatic generation of detailed descriptions and narratives. Another interesting future research would be to create designer models to adapt quest

suggestions to the designer's particular style [54, 55]. Finally, more extensive user studies will be conducted to analyze further the tool's usability and intuitiveness.

References

- [1] J. Doran and I. Parberry, “A prototype quest generator based on a structural analysis of quests from four mmorpgs,” in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, PCGames ’11, (New York, NY, USA), ACM, 2011. 231, 232, 233, 235, 236, 237, 238, 245
- [2] V. Breault, S. Ouellet, and J. Davies, “Let conan tell you a story: Procedural quest generation,” *Entertainment Computing*, vol. 38, p. 100422, 2021. 231, 233
- [3] M. Trenton, D. Szafron, J. Friesen, and C. Onuczko, “Quest patterns for story-based computer games,” in *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE’10, p. 204–209, AAAI Press, 2010. 231, 233, 246
- [4] K. Yu, N. R. Sturtevant, and M. Guzdial, “What is a quest?,” in *Proceedings of the Intelligent Narrative Technologies Workshop at AIIDE*, 2020. 231, 232
- [5] J. Howard, *Quests: Design, Theory, and History in Games and Narratives*. CRC Press, 2008. 231, 232
- [6] S. Dahlskog, J. Togelius, and S. Björk, “Patterns, dungeons and generators,” in *Proceedings of the 10th International Conference on the Foundations of Digital Games, FDG 2015, Pacific Grove, CA, USA, June 22-25, 2015* (J. P. Zagal, E. MacCallum-Stewart, and J. Togelius, eds.), Society for the Advancement of the Science of Digital Games, 2015. 231
- [7] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016. 231, 239
- [8] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018. 231

- [9] A. Liapis, “10 years of the pcg workshop: Past and future trends,” in *Proceedings of the 11th Workshop on Procedural Content Generation, FDG*, 2020. 231
- [10] R. Lavender, “The zelda dungeon generator: Adopting generative grammars to create levels for action-adventure games,” 2016. 231
- [11] C. Pedersen, J. Togelius, and G. N. Yannakakis, “Modeling player experience for content creation,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, pp. 54–67, 2010. 231
- [12] G. Smith, “Understanding procedural content generation: A design-centric analysis of the role of pcg in games,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’14*, (New York, NY, USA), p. 917–926, Association for Computing Machinery, 2014. 231
- [13] J. Dormans and S. Bakkes, “Generating missions and spaces for adaptable play experiences,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 216–228, 2011. 231, 234
- [14] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the Galactic Arms Race video game,” in *2009 IEEE Symposium on Computational Intelligence and Games*, Sept. 2009. 231
- [15] M. C. Green, G. A. B. Barros, A. Liapis, and J. Togelius, “Data agent,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, (New York, NY, USA), ACM, 2018. 231
- [16] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, “Procedural content generation via machine learning (pcgml),” *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018. 231
- [17] S. Mason, C. Stagg, and N. Wardrip-Fruin, “Lume: A system for procedural story generation,” in *Proceedings of the 14th International Conference on the Foundations of Digital Games*, (New York, NY, USA), Association for Computing Machinery, 2019. 231
- [18] A. Flodhag, S. Tolinsson, A. Alvarez, and J. Font, “To make sense of dungeons: Developing procedural narrative for edd with the help of macro patterns,” in *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts, CHI PLAY ’20 Extended Abstracts*, (New York, NY, USA), ACM, 2020. 231, 234

- [19] P. Ammanabrolu, W. Broniec, A. Mueller, J. Paul, and M. Riedl, “Toward automated quest generation in text-adventure games,” in *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*, (Tokyo, Japan), pp. 1–12, Association for Computational Linguistics, 29 Oct.–3 Nov. 2019. 231
- [20] M. Kreminski and N. Wardrip-Fruin, “Sketching a map of the storylets design space,” in *Interactive Storytelling* (R. Rouse, H. Koenitz, and M. Haahr, eds.), (Cham), pp. 160–164, Springer International Publishing, 2018. 231, 233
- [21] G. Smith, R. Anderson, B. Kopleck, Z. Lindblad, L. Scott, A. Wardell, J. Whitehead, and M. Mateas, “Situating quests: Design patterns for quest and level design in role-playing games,” in *Interactive Storytelling* (M. Si, D. Thue, E. André, J. C. Lester, J. Tanenbaum, and V. Zammitto, eds.), (Berlin, Heidelberg), pp. 326–329, Springer Berlin Heidelberg, 2011. 231, 246
- [22] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 232, 234
- [23] A. Liapis, G. N. Yannakakis, and J. Togelius, “Sentient sketchbook: Computer-aided game level authoring,” in *In Proceedings of ACM Conference on Foundations of Digital Games, 2013. In Print*, 2013. 232, 234
- [24] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 232
- [25] M. Guzdial, N. Liao, and M. Riedl, “Co-creative level design via machine learning,” in *Joint Proceedings of the AIIDE 2018 Workshops co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2018)*, Edmonton, Canada, November 13-14, 2018., 2018. 232
- [26] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *IEEE Transactions on Games*, pp. 1–1, 2020. Preprint available: <https://arxiv.org/abs/2003.03377>. 232, 235, 236
- [27] E. Aarseth, “From hunt the wumpus to everquest: introduction to quest theory,” in *International Conference on Entertainment Computing*, pp. 496–506, Springer, 2005. 232, 236

- [28] A. Sullivan, M. Mateas, and N. Wardrip-Fruin, "Questbrowser: Making quests playable with computer-assisted design," in *Proceedings of Digital Arts and Culture*, DAC '09, 2009. 233, 244
- [29] R. M. Young, S. G. Ware, B. A. Cassell, and J. Robertson, "Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives," *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, vol. 37, no. 1-2, pp. 41–64, 2013. 233
- [30] J. Porteous, *Planning Technologies for Interactive Storytelling*, pp. 393–413. Singapore: Springer Singapore, 2017. 233
- [31] M. O. Riedl and R. M. Young, "An objective character believability evaluation procedure for multi-agent story generation systems," in *Intelligent Virtual Agents* (T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist, eds.), (Berlin, Heidelberg), pp. 278–291, Springer Berlin Heidelberg, 2005. 233
- [32] M. O. Riedl and R. M. Young, "Story planning as exploratory creativity: Techniques for expanding the narrative search space," *New Generation Computing*, vol. 24, pp. 303–323, Sep 2006. 233
- [33] V. Breault, S. Ouellet, and J. Davies, "Let conan tell you a story: Procedural quest generation," 2018. 233
- [34] J. Garbe, M. Kreminski, B. Samuel, N. Wardrip-Fruin, and M. Mateas, "Storyassembler: An engine for generating dynamic choice-driven narratives," in *Proceedings of the 14th International Conference on the Foundations of Digital Games*, FDG '19, (New York, NY, USA), Association for Computing Machinery, 2019. 233
- [35] Q. Kybartas and R. Bidarra, "A survey on story generation techniques for authoring computational narratives," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 239–253, 2016. 234
- [36] A. Calvin and N. Michael, "The quest in a generated world," in *DiGRA གྷ - Proceedings of the 2007 DiGRA International Conference: Situated Play*, The University of Tokyo, September 2007. 234
- [37] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, "Toward supporting stories with procedurally generated game worlds," in *2011 IEEE Conference on Computational Intelligence and Games, CIG 2011, Seoul, South Korea*,

August 31 - September 3, 2011 (S. Cho, S. M. Lucas, and P. Hingston, eds.), pp. 297–304, IEEE, 2011. 234

- [38] A. Liapis, G. Smith, and N. Shaker, “Mixed-initiative content creation,” in *Procedural content generation in games*, pp. 195–214, Springer, 2016. 234
- [39] J. F. Allen, “Mixed-initiative interaction,” in *Intelligent Systems, Trends and Controversies section*, pp. 14–16, 1999. 234
- [40] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, “Can computers foster human users’ creativity? theory and praxis of mixed-initiative co-creativity,” *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016. 234
- [41] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 234
- [42] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, “Pitako-recommending game design elements in cicero,” in *2019 IEEE Conference on Games (CoG)*, IEEE, 2019. 234
- [43] M. Kreminski, M. Dickinson, M. Mateas, and N. Wardrip-Fruin, “Why are we like this?: The ai architecture of a co-creative storytelling game,” in *International Conference on the Foundations of Digital Games*, FDG ’20, (New York, NY, USA), Association for Computing Machinery, 2020. 234
- [44] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 235
- [45] E. Aarseth, “A narrative theory of games,” in *Proceedings of the International Conference on the Foundations of Digital Games*, FDG ’12, (New York, NY, USA), p. 129–133, ACM, 2012. 236
- [46] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Berlin, Heidelberg: Springer-Verlag, 1996. 238
- [47] G. Smith and J. Whitehead, “Analyzing the expressive range of a level generator,” in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, (New York, NY, USA), ACM, 2010. 240

- [48] J. Oates, “Researching information systems and computing,” in *SAGE*, 2013. 242
- [49] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG ’17, (New York, NY, USA), ACM, 2017. 246
- [50] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 246
- [51] M. Charity, A. Khalifa, and J. Togelius, “Baba is y’all: Collaborative mixed-initiative level design,” 2020. 246
- [52] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, Aug 2018. 247
- [53] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, “Orchestrating game generation,” *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019. 247
- [54] A. Liapis, G. Yannakakis, and J. Togelius, “Designer modeling for personalized game content creation tools,” in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19, pp. 11–16, AI Access Foundation, 2013. 248
- [55] A. Alvarez, J. Font, and J. Togelius, “Designer modeling through design style clustering,” *Arxiv*, 2020. Preprint available: <https://arxiv.org/abs/2004.01697>. 248

PAPER VIII - ASSESSING THE EFFECTS OF INTERACTING WITH MAP-ELITES

Alberto Alvarez, Jose Font, Steve Dahlskog, and Julian Togelius

ABSTRACT

MAP-Elites has been successfully applied to the generation of game content and robot behaviors. However, its behavior and performance when interacted with in-co-creative systems is underexplored. This paper analyzes the implications of synthetic interaction for the stability and adaptability of MAP-Elites in such scenarios. We use pre-recorded human-made level design sessions with the Interactive Constrained MAP-Elites (IC MAP-Elites). To analyze the effect of each edition step in the search space over time using different feature dimensions, we introduce Temporal Expressive Range Analysis (TERA). With TERAs, MAP-Elites is assessed in terms of its adaptability and stability to generate diverse and high-performing individuals. Our results show that interactivity, in the form of design edits and MAP-Elites adapting towards them, directs the search process to previously unexplored areas of the fitness landscape and points towards how this could improve and enrich the co-creative process with quality-diverse individuals.

PUBLISHED IN

The 17th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE), AAAI, 2021.

ASSESSING THE EFFECTS OF INTERACTING WITH MAP-ELITES

Introduction

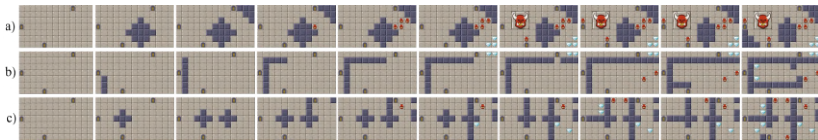


Figure 1: Sequences of rooms used in the three scenarios, targeting (a) low leniency, (b) high linearity, (c) and high meso-pattern concentration respectively. The leftmost and rightmost rooms correspond to the start and end rooms in each sequence, while intermediate steps are shown in between, limited to eight due to space restrictions.

Mixed-initiative co-creativity (MI-CC) [1], is a human-AI collaborative approach where both human and computer have a proactive role in the creation of content [2]. Recent research shows the importance of using quality-diversity (QD) algorithms [3,4] to better drive the evolutionary process in complex search spaces by generating stepping stones that barely resemble the optimal solution [5]. A popular QD implementation in recent research is MAP-Elites [6], which has been applied to generate levels for bullet hell games [7], dungeon levels for MI-CC generation of adventure games [8], and levels for puzzle games [9].

The rising interest of the evolutionary computation and computational intelligence in games research community in PCG, MI-CC and MAP-Elites, calls for improving the ways for evaluating these novel approaches. Some of the main problems in mixed-initiative tools are the inadequate consideration of the costs and benefits for every automated action, as well as failing to spot the opportunities for users to guide the invocation of the automated services [10].

The area of eXplainable AI for Designers (XAID) [11] strives for achieving system explainability necessarily built on *understandings of both algorithmic properties of the underlying AI techniques and the needs of human designers*. Similarly, Compton reflects on the grokloop [12]: the creative feedback MI-CC loop where a user builds

a hypothesis, modifies a system, evaluates the result, and then updates the system. Shorter grokloops improve the overall performance of the mutual inspiration, and attempting to shorten this loop implies a clear understanding and interpretation of the relationship between each user action and the changes that it triggers in the system.

MAP-Elites have shown excellent results at generating QD individuals in games [13, 14], and offline adaptation based on its generated repertoire [15, 16]. However, MAP-Elites generation capabilities have been mostly evaluated in non-interactive scenarios and based on its final result even when used in interactive situations [8, 9]. This results in a lack of research to assess the effects and consequences of interacting with MAP-Elites, and the adaptability and stability properties of MAP-Elites in dynamic scenarios.

Therefore, the contributions of this paper are two-fold. On the one hand, we present Temporal Expressive Range Analysis (TERA) as a novel way to analyze interactive PCG. TERAs allow us to inspect and analyze the changes in the expressive range over a defined period, which in our case, are design editions. On the other hand, using TERAs, we explored and analyzed how population dynamics react and adapt to constant changes in the IC MAP-Elites for level generation of 2D adventure games. IC MAP-Elites is evaluated using simulated pre-recorded design sessions with different design goals that display the algorithm's stability and adaptability properties and benefits. Our results show that IC MAP-Elites stably encounters high-performing solutions while adapting to changes in the design, and by doing this, regions of the search space which previously seemed inaccessible are opened for exploration.

Background

The Evolutionary Dungeon Designer (EDD) is a MI-CC tool to co-create 2D dungeons in the style of the seminal game *The Legend of Zelda* [17]. Designers manually edit the dungeon structure as well as the interior of every room in it. EDD constantly offers tailored room suggestions on the fly that designers may decide to incorporate to their designs at any moment.

In EDD, the system analyzes the level-design patterns (i.e., micro- and meso-patterns) that exist in each room, calculating and utilizing their quality to assess rooms. Micro-patterns are the building blocks in a design, which in EDD are categorized as *spatial micro-patterns*: chamber, corridor, intersections, connector; and *inventorial micro-patterns*: enemy, treasure, and door. On the other hand, Meso-patterns are defined as the relation between micro-patterns or other meso-

patterns, and by the composition between inventorial micro-patterns and spatial micro-patterns. Meso-patterns are used to identify structures in the room that join together a set of micro-patterns and can be: *ambush*, *guard chamber*, *treasure chamber*, and *guarded treasure*. All patterns are shown in figure 2, and further information and discussion can be found in [18, 19].

Methods for Evaluating PCG

Recent research focuses on methods for evaluating procedural algorithms. The work in [20] evaluates fitness, offspring and selection for five MAP-elite methods, whereas [21] shows how users can improve the generator with the aid of automatic parameter tuning and, consequently, evaluates the effect that it has on the generator. The work is continued in [22] where two analytical techniques, smoothness and co-dependence, were introduced to analyze the impact of a parameter change and its effect on a generative system; all integrated in Danesh [23]. Liapis et al. [24] did a similar evaluation as the one in this paper, using artificial agents simulating designer’s choices of suggestions to evaluate and display properties of their designer’s model.

Previously suggested evaluation methods include a top-down approach [25, 26], called the *expressive range analysis* (ERA) which refers to the idea of exploring and visualizing the content space. Summerville [27] proposes techniques for visually assessing and analyzing procedural systems, with other means of visual assessment including analysis of generative space and individual procedural artefacts.

Variants of MAP-Elites

MAP-Elites, a quality-diversity (QD) algorithm, seeks to *illuminate a behavior space* by trying to find the best solutions across a feature-dimension grid [6].

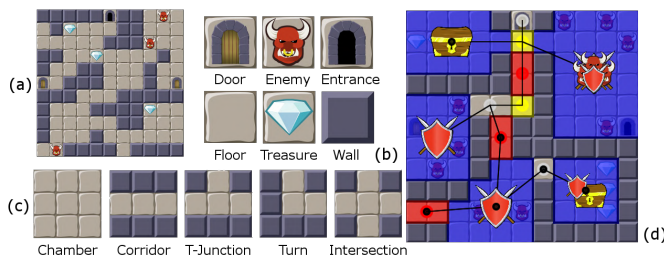


Figure 2: Main components in EDD. (a) Basic room, (b) different tiles, (c) micro-patterns and (d) meso-patterns

Feature	Definitions
Similarity (Sim)	Aesthetic (tile-by-tile) similarity between a generated level and the designer’s design.
Inner Similarity (IS)	Different tiles’ sparsity and density similarity between a generated level and the designer’s design.
Symmetry	Room’s aesthetic symmetry.
Leniency (Len)	Challenge based on enemies and treasures.
Linearity (Lin)	Paths that exist connecting entry points in a level.
#Meso-Patterns (Meso)	Amount of meso-patterns that exist within a level. This is a discrete dimension rather than continuous.
#Spatial-Patterns (Spa)	Amount of spatial-patterns that exist within a level.

Table 1: Level design based dimensions used in EDD with IC MAP-Elites.

Some versions skip the grid in favour of voronoi tessellation to decide which elite individuals to keep in the map [28]. Other works combine the effective adaptive search of Covariance Matrix Adaptation Evolution Strategies with a map of elites, yielding large improvements for real-valued representations in terms of both objective value and number of elites discovered [13]. ME-MAP-Elites [29] creates a set of emitters to focus on different optimization processes that are active at different generations, generating higher performing and diverse individuals.

Constrained MAP-Elites [7] combines divergent search with a two-population approach to constraint satisfaction, taken from the FI-2Pop algorithm [30]. Constrained MAP-Elites has been used as the basis for subsequent experiments, e.g., to find sets of levels implementing diverse game mechanics [31]. This algorithm was later combined with interactive evolution to yield the aforementioned Interactive Constrained MAP-Elites [14]. Moreover, MAP-Elites has been shown to be robust at adapting to changing conditions after running the algorithm thanks to its generated behavioral repertoire. This was proposed and tested in the intelligent trial-and-error algorithm [15, 16]. Related work extended MAP-Elites with *Adaptive Sampling* and *Drifting-Elites* to be more robust in noisy environments and domains where the fitness and behavior evaluation might be stochastic such as games [32].

Approach

IC MAP-Elites is a variation of Constrained MAP-Elites that incorporates adaptive mechanisms and implements continuous and interactive evolution [14]. Our evaluation described in the following section is applied to EDD, which implements IC MAP-Elites, allowing us to evaluate the effects and dynamics of interacting with MAP-Elites.

EDD’s IC MAP-Elites implementation uses a single-objective weighted fitness function with a FI2Pop genetic algorithm [30]. Individuals are deemed infeasible when they contain unreachable areas from any of the room’s entry points and are evaluated based on how many unreachable tiles they have. Feasible individuals are evaluated as the following equally divided weighted sum:

$$f_{fitness}(r) = \frac{1}{2}f_{inventorial}(r) + \frac{1}{2}f_{spatial}(r) \quad (1)$$

This evaluation is adaptive, meaning that the tile’s ratios, patterns, and balance between chambers and corridors are related to the target and collected by MAP-Elites after every modification to the target. $f_{inventorial}$ calculates the quality of all inventorial micro-patterns in relation to the current edited room, and $f_{spatial}$ calculates both the quality of spatial micro-patterns and the distribution and composition of tiles in the room described by the meso-patterns.

Seven level-design related feature dimensions are implemented in EDD. The designer can pick dimension pairs at a time and change the dimensions’ granularity. When the designer changes dimensions, IC MAP-Elites seamlessly reshape the cells and move around the current elites, allowing the designer to switch between features. The seven features are briefly described in table 1, although an extensive discussion can be found in [14].

Experiment Setup

We have conducted a series of experiments on EDD to analyze the *adaptability* and *stability* of IC MAP-Elites, as well as the effects of the interaction for MAP-Elites and the user. *Stability* relates to the steady generation of high-performing individuals, while gradually growing the search and stably covering the generative space at each edit step. *Adaptability* relates to the ability of the search to adapt to changing conditions, adjusting the search to the new goals, while still generating high-performing individuals. Both features, relate to the notions of evolvability [33], the ability of the search to generate creative individuals in problems with changing conditions.

Dimensions	Low Leniency Scenario		
	◇	†	○
Len-Meso	*56.6%	*19.46±5.55	0.96±0.013
Len-Spa	71.4%	*19.34±7.979	0.94±0.016
Lin-Meso	78.7%	43.5±3.76	0.92±0.013
Lin-Spa	*57.8%	28.28±4.109	0.92±0.011
Meso-Spa	78.9%	27.15±7.315	0.93±0.012
Sym-Len	68.1%	21.54±5.88	0.97±0.014
Sym-Spa	83.4%	24.87±10.334	0.95±0.014
Sim-Lin	58.4%	30.78±2.602	*0.9±0.012
Sim-Meso	62.3%	27.98±3.385	*0.9±0.014
Sym-IS	70.8%	21.63±3.877	0.96±0.011
Average.	67.09±3.35	25.75±2.48	0.93±0.01
Dimensions	High Linearity Scenario		
	◇	†	○
Len-Spa	72.6%	*16.31±3.035	0.91±0.006
Lin-Meso	67.2%	40.79±3.221	0.85±0.015
Lin-Spa	62.7%	33.55±0.963	*0.84±0.013
Sym-Len	72.6%	20.5±2.646	0.94±0.008
Sym-Lin	84.3%	37.94±2.103	0.87±0.013
Sym-Spa	85.5%	18.94±4.366	0.92±0.011
Sim-Len	*58.7%	20.93±2.174	0.87±0.011
Sim-Lin	62.3%	27.78±1.424	*0.84±0.014
Sim-Meso	68%	25.6±1.946	*0.84±0.011
Sym-IS	82.8%	21.24±3.005	0.94±0.011
Average.	71.8±3.12	24.52±2.81	0.89±0.01
Dimensions	High Meso-Pattern Scenario		
	◇	†	○
Len-Lin	*56.3%	32.49±1.536	0.92±0.007
Lin-Spa	*55.7%	30.47±1.959	0.9±0.007
Sym-Len	65.7%	*22.73±5.025	0.96±0.009
Sym-Lin	83.4%	44.4±1.019	0.93±0.007
Sym-Meso	83.6%	38.45±5.136	0.96±0.01
IS-Meso	90.2%	34.82±3.749	0.95±0.009
Sim-IS	59.9%	*21.16±2.237	0.91±0.009
Sim-Lin	66.3%	36.29±1.585	*0.89±0.008
Average.	72.34±4.12	29.88±2.84	0.93±0.01

Table 2: Results from the three metrics in all scenarios across the relevant dimension pairs. ◇ relates to coverage, † relates to average coverage per step, and ○ average population fitness throughout all generations. Higher scores per column are highlighted in bold. ★ marks the lower values. Confidence intervals are shown for each average value (†, and ○), and in the last row we show the average of all the 21 dimensions per metric and scenario.

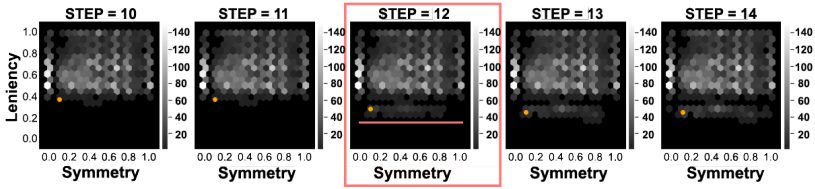


Figure 3: Aggregated TERA using Sym and Len following the low leniency scenario (fig. 1.a). Highlights how the design introduces new generative region to the algorithm. In red (step 12) it is highlighted when the design enters a new region of the space and MAP-Elites is then able to generate individuals in that region, explained in detail in Case 1.

We recorded three different design sessions, called scenarios, where we manually designed, step by step, dungeon rooms with specific target design goals. They are shown in Figure 1: a) a boss room - identified as a **low leniency** design goal; b) a linear room with specific paths and targets - identified as a **high linearity** design goal, and finally, c) a room where each chamber within is usable - identified as a **high meso-pattern** design goal. We chose these design goals as they represent key metrics with clear distinct representative goals and design styles that one might create in a dungeon.

The experiments consist on running these pre-recorded scenarios separately on EDD, step by step with a lapse of 100 evolutionary generations between steps. Each scenario implies 21 evolutionary runs, one per each pair of feature-dimension, where the dimensions are (table 1): leniency (len), linearity (lin), spatial patterns (spa), meso-patterns (meso), symmetry (sym), similarity (sim), and inner similarity (IS). Each edition (i.e., design step) is included as-is in the population and used as a target in the fitness function by IC MAP-Elites, updating corridor, chamber, and inventorial ratios affecting the quality of micro and meso patterns. Every 100 generations we collect the novel generated individuals to later analyze how the search and the fitness landscape vary after each design step. Step after step, we measure how the explored generative space grows, as well as how distribution and concentration of elites together with the manually edited room traverse the generative space.

In all the experiments, the initial population was set to 1000 mutated individuals. All cells were set to a maximum capacity of 25 individuals each. In every generation, we selected 5 cells random, and 5 parents per cell through tournament selection. The random selection followed a uniform distribution. Offspring were produced through a two-point crossover and a 30% mutation chance. Using this setup, between 150 to 2001 individuals were produced every 100 generations, with an average of 373 unique individuals generated every 100 generations throughout all runs.

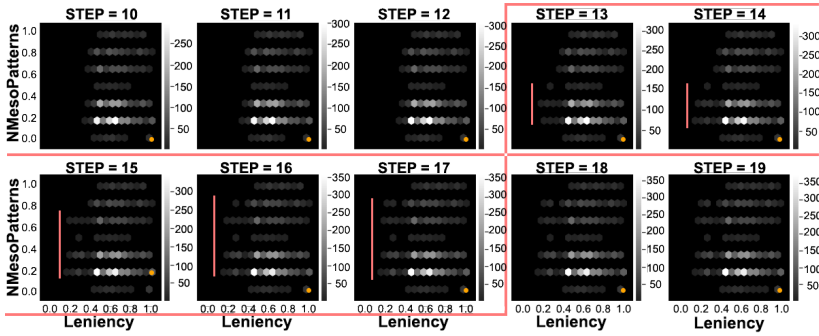


Figure 4: Aggregated TERA using Len and Meso following the high linearity scenario (fig. 1.b). Highlights subtle discovery of new generative regions. In red (step 12), this subtle discovery is highlighted, explained in detail in Case 2.

Metrics

All our experiments are evaluated and analyzed following the same procedure and metrics, focusing in the novel generated individuals. In particular, we calculated the *coverage* (\diamond), the *average coverage per step* (\dagger), and *average fitness* (\circ). *Coverage* relates to the percentage of space covered by the search in total and is calculated as the cumulative amount of covered hexagons at the final step divided by the maximum amount in our experiments. *Average coverage* relates to the average coverage per step, i.e., how much of the space is covered at each design step in average, calculated as the cumulative coverage per step divided by the amount of design editions. Finally, *average fitness* calculates the average individual fitness in the search throughout all generations.

Results and Analysis

Table 2 shows an extract of all test results. The results were filtered to only show the pair of dimensions with the lowest or highest values in different metrics. This means that some pair of dimensions are not shown since all their metric values were in between lower and higher scores. Each subtable represents one of the three design scenarios, each of them displaying the metrics described above. The higher and lower scores per column are highlighted in bold and with \star , respectively. Confidence intervals are shown per value when using averages. In general, table 2 shows IC MAP-Elites' stability to cover the space while encountering high-performing individuals; thus, it is able to adapt to the new editions which change the fitness landscape. Coverage (\diamond), supported by average coverage per step (\dagger) and average fitness (\circ), shows that in average the algorithm keeps exploring and generating novel and high-performing individuals rather than sporadically generating them.

The symmetry and spatial pattern (Sym-Spa) dimension pair explore and cover on average more of the space across all scenarios than others (\diamond). In general, when using either *sym* or *spa*, MAP-Elites is pressured to generate content that maximizes the utility of walls since both use walls as a core building block. Similarly, *Meso* and *IS* are two other dimensions that perform well with others, especially *Meso*. However, *Meso* requires combination of shapes with walls and correct placement of tiles; thus, in principle involving more complex operations to achieve high dimensional values.

Overall the average population fitness (\circ) is very high with 0.915 in average (of a maximum 1.0) with a narrow confidence interval ± 0.01 . Symmetry and leniency (Sym-Len) scored the highest in all scenarios while Similarity and Linearity (Sim-Lin) scored the lowest. This shows a stable behavior in the IC MAP-Elites as that the average quality of individuals is high even when large portions of the generative space are explored (high diversity), and regardless of the dimension pair chosen.

IC MAP-Elites works in constant interaction with the edited design. Each step reshapes the fitness landscape to a certain extent, which could hinder the evolutionary process. However, per step, an avg. of 27.34% of the space is covered by generating novel individuals (i.e., not encountered previously in the population). This, coupled with the relatively narrow confidence intervals, means that the search is constantly exploring the space, diversifying and encountering new spaces. However, as it will be exemplified through the cases, the implicit explorative and exploitative mechanisms of MAP-Elites might not be enough to explore new regions, which can be introduced interactively to MAP-Elites.

The following cases examine TERAs following the different scenarios presented in figure 1. Different cases will use either an aggregated TERA step by step or a non-aggregated version showing each step's specific scores in the evaluated dimensions. Each figure's caption and respective case will indicate what type of TERA is used. Non-Aggregated TERAs show the delta maps in the search, meaning where the search has focused and the space covered for a specific step. On the other hand, aggregated TERAs show the density of generated individuals over all steps and the coverage up to the specific step. In each TERA, we also show as an orange dot where the current design is in relation to the used and tested dimensions. Case 1 and 2 examine aggregated TERAs, while case 3 examines non-aggregated TERAs.

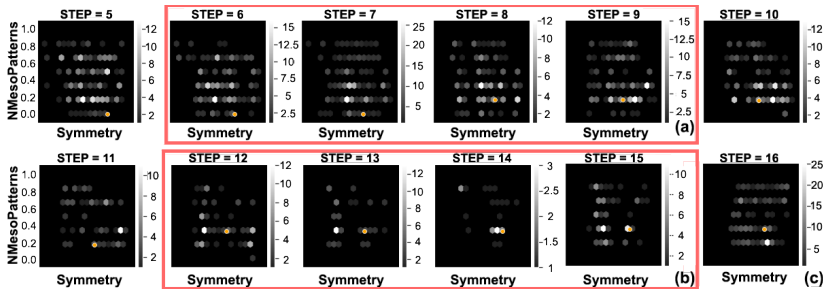


Figure 5: Non-aggregated TERA using Sym and Meso following the high meso-pattern level scenario (fig. 1.c). Highlights overall properties of interacting with MAP-Elites.a, b, and c, represent the main areas of focus explained in detail in Case 3.

Case 1 - Design Opens New Regions of the Generative Space

In this case, we analyze the interaction with IC MAP-Elites through examining the generative space when using Sym and Len as dimensions (figure 3) following the low leniency scenario depicted in figure 1.a. The scenario aimed to gradually increase the room’s challenge while dividing it into two clear and connected areas.

Table 2 shows that this pair of dimensions do not have the best scores except for the *fitness* (○). This indicates that these dimensions are able to stably find high-performing individuals (above the avg.) while adapting to the new designs exploring an average of 21.54 and a total of 68.1%. Moreover, in fig 3, for several steps, half of the generative space is completely unexplored, which could indicate that in those regions, dimensions would be mutually exclusive.

However, at step 12 (highlighted in red in fig 3), when reaching a low leniency score, the design enters an unexplored region of the generative space, which subsequently enables IC MAP-Elites to search and generate high-performing individuals in the new region. Furthermore, there is a significant rise in the number of unique individuals generated with a high concentration on the new region, spreading over the already explored space.

Case 2 - Subtle Changes in the Design Reflected in the Generation of MAP-Elites

Case 1 showed that by entering an unexplored region of the generative space, the designer could show possibilities for the algorithm and influence the search, yet more subtle guidance is possible. Figure 4 presents such a case. We focus on the high linearity scenario (fig. 1.b), where the goal was to create a single

narrow corridor between the top and left door and add some objective at the bottom entrance. Through this, we not only aimed at high linearity but also tried to promote other main characteristics such as the open chamber/corridor balance, or the combination of meso-patterns.

Similar to the previous case, in figure 4, it is shown that for many steps, the search does not explore a big part of the generative space. In this case, the room does not move either in the generative space as the changes are not affecting the dimensions. However, between steps 13 to 17, a new region of the generative space is filled (highlighted in red in fig 4), which indicates that even if changes in the room do not have a direct influence by moving in the generative space, they still can foster exploration in new regions. These main steps are visible in fig. 1.b, subfigure 5, and 6 from the left. Specifically, lower leniency regions are generated once the room is divided into a representative corridor and a big open area with a treasure meso-pattern. These stepping-stones gave the needed “building blocks” to MAP-Elites to cross and mutate until the new generative space was explored. Moreover, similarly to the previous case, the algorithm generates significantly more novel individuals during this time (around 531 novel individuals), and the search covers 20% of the space, exploiting the new region, akin to Novelty search behavior [34].

Case 3 – Exploring Multiple Properties

In this case, we analyze the TERA of unique individuals generated each step using Sym and Meso as dimension pairs (fig. 5). We heed to the high meso-pattern level scenario (fig. 1.c), where we subdivided the room into small open chambers with clear objectives. Overall, in fig. 5 two aspects stand out; firstly, the generative space is explored more at the early steps, as there are fewer constraints from the edited design. Secondly, the generated individuals seem to follow the path taken by the design in the generative space. Supported by the other cases, this indicates that the design can filter the search and point regions of interest for the IC MAP-Elites.

Moreover, opposite to case 1 and as a consequence of filtering the generative space, when the design leaves low scoring regions, the algorithm rapidly disregards creating individuals in those spaces. For instance, the bottom region after step 7 (fig. 5.a). Further, as the room is changing but without any type of influence in the searched dimensions e.g., steps 12 to 15 (fig. 5.b), MAP-Elites has difficulties exploring the space. A similar challenge was discussed by Alvarez et al. [14], where their experiments showed that the search got to a plateau after 1000 generations due to the MAP-Elites lacking the incentive to explore. Even if their experiments

focused on static environments, this case gives further evidence that minimal changes to the design and lack of influence in the generative space, conditions the exploration of space and the generation of novel individuals.

Lastly, at step 16 (fig. 5.c), we encounter a similar situation as with fig. 4; where a design edition enables the needed “building blocks” for MAP-Elites. In this case, it was triggered by the forming of a dead-end chamber pattern, which enables even more meso-patterns to be used and discovered. Rather than finding a new region of the generative space, this time, the search gets rebooted, and therefore, explores all regions generating novel individuals.

Discussion

Our evaluation shows that IC MAP-Elites has a high degree of adaptability to dynamic environments, adapting the generated content to the design process and design goals while stably generating high-performing and diverse solutions. For MI-CC systems and interactive approaches as in EDD, this is especially relevant and important. The fitness function adapts to the current design; thus, adaptability and stability go hand in hand. Furthermore, the deployment of an MI-CC approach in a scenario as the ones presented would benefit both Map-Elites and the human designer. On the one hand, it enables MAP-Elites to explore more of the generative space while producing quality solutions. On the other hand, users would have more control over the suggestions as they seamlessly influence and guide the search and generation with their design, in a similar approach to Anderson et al., who explored explicit human guidance in the automated solution search [35].

We also observe that when using Linearity as a dimension, IC MAP-Elites performed quite stably in all our scenarios regardless of the design traversing around the generative space or not, which indicates that Linearity is more robust and stable and more agnostic and independent from the design. These characteristics are beneficial in certain cases, but based on the results presented in table 2, this stability comes at the expense of adaptability and higher fitness scores.

Furthermore, Alvarez et al. [14] presented an analysis of IC MAP-Elites in a static scenario, where on average the covered space of MAP-Elites after 5000 generation was 52.4% using pair of dimensions and 51.7% using all seven dimensions. Our results show a clear advantage for MAP-Elites when used continuously and interactively with an avg. coverage of 70.9%. However, when the design remains still in the space defined by the dimensions, exploration is hindered; thus, what dimensions are used and how the design maps to them is crucial.

Finally, we used and introduced TERAs to analyze the dynamic behaviors in

generative systems and algorithms, and observe the effects of changes over time in the expressive range based on the edition steps. We used two variations, non-aggregated TERA, which shows the delta maps of the search, and aggregated TERA, showing the search density and aggregated results. TERAs are generic and could be used with other generative system to evaluate their dynamics by simply defining a pair of features and a step period such as design editions, amount of generations, or whenever a suggestion is applied. TERAs can also be used to spot key and non-trivial steps or changes that have an effect in the search, which can help to understand more in-depth the sensibility of the algorithm and the system.

Conclusions and Future Work

This paper analyzes and evaluates the benefits of dynamically interacting with quality-diversity algorithms, specifically, the IC MAP-Elites. We have examined the adaptability and stability of MAP-Elites in relation to 21 dimension pairs highlighting different characteristics and properties through different simulated design scenarios. We examined key metrics when exploring the generative space, as depicted in table 2, and conducted three different case studies that highlighted different dynamics with the algorithm.

While our results show several MAP-Elites' properties and promising ways to improve the MI-CC workflow, further evaluation is needed with human users to assess these properties in-the-wild and evaluate the interactive dynamic between humans and algorithms. To further highlight the importance of interaction, it would be interesting to analyze and compare with MAP-Elites disabling adaptive mechanisms (i.e., rendering the algorithm static and agnostic to changes) and with other non QD algorithms. Likewise, another interesting project for future work, would be to use TERAs to compare IC MAP-Elites with other co-creative systems using other algorithms such as reinforcement learning [36,37] or constraint solving algorithms [38].

Finally, a promising step is to analyze MAP-Elites together with surrogate designer models that capture preference, style, and design processes [39–41], and how these influence the properties discussed in this paper. For instance, Designer Personas [41] could be used to explore how the user's design moves through the space, identifying possible paths, and analyzing if changes, i.e., moving between style clusters, connect to key moments in the search.

References

- [1] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 257
- [2] A. Liapis, G. N. Yannakakis, and J. Togelius, “Generating Map Sketches for Strategy Games,” in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS, Springer, 2013. 257
- [3] J. K. Pugh, L. B. Soros, and K. O. Stanley, “Quality diversity: A new frontier for evolutionary computation,” *Frontiers in Robotics and AI*, vol. 3, p. 40, 2016. 257
- [4] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, “Procedural content generation through quality diversity,” in *2019 IEEE Conference on Games*, 2019. 257
- [5] A. Gaier, A. Asteroth, and J.-B. Mouret, “Are quality diversity algorithms better at generating stepping stones than objective-based search?,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 115–116, ACM, 2019. 257
- [6] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *preprint arXiv:1504.04909*, 2015. 257, 259
- [7] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, ACM, 2018. 257, 260
- [8] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 257, 258

- [9] M. Charity, A. Khalifa, and J. Togelius, “Baba is y’all: Collaborative mixed-initiative level design,” in *2020 IEEE Conference on Games (CoG)*, pp. 542–549, 2020. 257, 258
- [10] E. Horvitz, “Principles of mixed-initiative user interfaces,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’99*, (New York, NY, USA), p. 159–166, Association for Computing Machinery, 1999. 257
- [11] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, Aug 2018. 257
- [12] K. Compton, “Casual creators: Defining a genre of autotelic creativity support systems.” PhD Dissertation, University of California Santa Cruz, 2019. 257
- [13] M. C. Fontaine, J. Togelius, S. Nikolaidis, and A. K. Hoover, “Covariance matrix adaptation for the rapid illumination of behavior space,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference, GECCO ’20*, p. 94–102, 2020. 258, 260
- [14] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *IEEE Transactions on Games*, pp. 1–1, 2020. 258, 260, 261, 267, 268
- [15] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, “Robots that can adapt like animals,” *Nature*, vol. 521, pp. 503–507, May 2015. 258, 260
- [16] M. Gonzalez-Duque, R. Berg Palm, D. Ha, and S. Risi, “Finding game levels with the right difficulty in a few trials through intelligent trial-and-error,” in *Proceedings of the IEEE Conference on Games*, 2020. 258, 260
- [17] Nintendo R&D4, “*The Legend of Zelda*.” Game [NES], 1986. Nintendo, Kyoto, Japan. Last played December 2017. 258
- [18] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG ’17*, (New York, NY, USA), ACM, 2017. 259

- [19] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 259
- [20] D. Gravina, A. Liapis, and G. N. Yannakakis, “Blending notions of diversity for map-elites,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’19, (New York, NY, USA), pp. 117–118, ACM, 2019. 259
- [21] M. Cook, J. Gow, and S. Colton, “Towards the automatic optimisation of procedural content generators,” in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, Sep. 2016. 259
- [22] M. Cook, S. Colton, J. Gow, and G. Smith, “General analytical techniques for parameter-based procedural content generators,” in *2019 IEEE Conference on Games (CoG)*, Aug 2019. 259
- [23] M. Cook, J. Gow, G. Smith, and S. Colton, “Danesh: Interactive tools for understanding procedural content generators,” *IEEE Transactions on Games*, pp. 1–1, 2021. 259
- [24] A. Liapis, G. N. Yannakakis, and J. Togelius, “Designer modeling for sentient sketchbook,” in *2014 IEEE Conference on Computational Intelligence and Games*, pp. 1–8, Aug 2014. 259
- [25] G. Smith and J. Whitehead, “Analyzing the expressive range of a level generator,” in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, (New York, NY, USA), ACM, 2010. 259
- [26] B. Horn, S. Dahlskog, N. Shaker, G. Smith, and J. Togelius, “A comparative evaluation of procedural level generators in the mario ai framework,” in *Proceedings of the 9th International Conference on the Foundations of Digital Games*, FDG ’14, 2014. 259
- [27] A. Summerville, “Expanding Expressive Range: Evaluation Methodologies for Procedural Content Generation,” in *Proceedings of the 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2018. 259
- [28] V. Vassiliades, K. I. Chatzilygeroudis, and J. Mouret, “Scaling up map-elites using centroidal voronoi tessellations,” *CoRR*, vol. abs/1610.05729, 2016. 260

- [29] A. Cully, “Multi-emitter map-elites: Improving quality, diversity and convergence speed with heterogeneous sets of emitters,” in *Proceedings of the 2021 Genetic and Evolutionary Computation Conference*, GECCO ’21, 2021. 260
- [30] S. O. Kimbrough, G. J. Koehler, M. Lu, and D. H. Wood, “On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *European Journal of Operational Research*, vol. 190, no. 2, pp. 310–327, 2008. 260, 261
- [31] M. Charity, M. C. Green, A. Khalifa, and J. Togelius, “Mech-elites: Illuminating the mechanic space of gvg-ai,” in *International Conference on the Foundations of Digital Games*, pp. 1–10, 2020. 260
- [32] N. Justesen, S. Risi, and J.-B. Mouret, “Map-elites for noisy domains by adaptive sampling,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO ’19, (New York, NY, USA), pp. 121–122, Association for Computing Machinery, 2019. 260
- [33] S. Doncieux, G. Paolo, A. Laflaquière, and A. Coninx, “Novelty search makes evolvability inevitable,” in *Proc. of the Genetic and Evolutionary Computation Conf.*, GECCO ’20, 2020. 261
- [34] A. Liapis, G. N. Yannakakis, and J. Togelius, “Constrained novelty search: A study on game content generation,” *Evolutionary Computation*, vol. 23, no. 1, pp. 101–129, 2015. 267
- [35] D. Anderson, E. Anderson, N. Lesh, J. Marks, B. Mirtich, D. Ratajczak, and K. Ryall, “Human-guided simple search,” in *AAAI/IAAI*, pp. 209–216, 2000. 268
- [36] O. Delarosa, H. Dong, M. Ruan, A. Khalifa, and J. Togelius, “Mixed-initiative level design with rl brush,” in *10th International Conference on Artificial Intelligence in Music, Sound, Art and Design, EvoMUSART 2021 held as Part of EvoStar 2021*, pp. 412–426, Springer, 2021. 269
- [37] M. Guzdial, N. Liao, and M. Riedl, “Co-creative level design via machine learning,” in *Proceedings of the 5th EXAG Workshop at AIIDE*, 2018. 269
- [38] I. Karth and A. M. Smith, “Addressing the fundamental tension of pcgml with discriminative learning,” in *Proceedings of the 14th International Conference on the Foundations of Digital Games*, FDG ’19, (New York, NY, USA), Association for Computing Machinery, 2019. 269

- [39] A. Liapis, G. Yannakakis, and J. Togelius, “Designer modeling for personalized game content creation tools,” in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19, pp. 11–16, AI Access Foundation, 2013. 269
- [40] A. Alvarez and J. Font, “Learning the designer’s preferences to drive evolution,” in *Applications of Evolutionary Computation*, vol. 12104 of *Lecture Notes in Computer Science*, (Cham), pp. 431–445, Springer International Publishing, 2020. 269
- [41] A. Alvarez, J. Font, and J. Togelius, “Designer modeling through design style clustering,” *Arxiv*, 2020. Preprint available: <https://arxiv.org/abs/2004.01697>. 269

PAPER IX - DESIGNER MODELING THROUGH DESIGN STYLE CLUSTERING

Alberto Alvarez, Jose Font, and Julian Togelius

ABSTRACT

We propose modeling designer style in mixed-initiative game content creation tools as archetypical design traces. These design traces are formulated as transitions between design styles; these design styles are in turn found through clustering all intermediate designs along the way to making a complete design. This method is implemented in the Evolutionary Dungeon Designer, a research platform for mixed-initiative systems to create adventure and dungeon crawler games. We present results both in the form of design styles for rooms, which can be analyzed to better understand the kind of rooms designed by users, and in the form of archetypical sequences between these rooms, i.e., Designer Personas.

PUBLISHED IN

IEEE Transactions on Games, IEEE, 2022.

DESIGNER MODELING THROUGH DESIGN STYLE CLUSTERING

Introduction

How can we best build a system that lets a human designer collaborate with procedural content generation (PCG) algorithms to create useful and novel game content? Collaboration between AI and humans to co-design and co-create content is a major challenge in AI, and the main focus of Mixed-Initiative Co-Creativity [1,2]. These systems' objectives are to foster creativity and provide seamless proactive collaboration; ultimately enabling a colleague relation and collaboration as described by Lubart [3]. However, there needs to be an understanding between the human designer and the AI system about what needs to be designed, ideally even a shared goal.

Reaching such a shared understanding is a hard task, even when both collaborators share significant cultural and professional backgrounds. However, we can take steps towards the goal of shared understanding. One idea is to train a supervised learning model on traces of other collaborative creation sessions and try to predict the next step the human would take in the design process. The main problem with this is that people are different, and creators will want to take different design actions in the same state. Another problem is what to do in design states that have not been encountered in the training data. To remedy this, it has been proposed to train multiple models, predicting the next step for different designer "personas" (akin to procedural personas in game-playing [4]). However, for such a procedure to be effective, we need to have sufficient training data. The more different designer personas there are, the more training data is necessary.

One way of overcoming this problem could be to change the level of abstraction at which design actions are modeled and predicted. Instead of predicting individual edits, one could identify different styles or phases of the artifact being created and model how a designer moves from one to another. To put this concretely in the context of designing rooms for a Zelda-like dungeon crawler [5], one could

classify room styles depending on whether they were enemy onslaughts, complex wall mazes, treasure puzzles, and so on. One could then train models to recognize which types of rooms a user creates in which order. By clustering sequences of styles, we could formulate designer personas as archetypical trajectories through style space rather than as sequences of individual edits. For example, in the context of creating a dungeon crawler, some designers might start with the outer walls of the rooms and then populate it with NPCs, whereas another type of designer might first sketch the path they would like the player to take from the entrance to the exit and then add parts of the room outside the main path. These designer models could then be combined with search-based [6] or other procedural generation methods [7] to suggest ways of getting to the next design style from the current one.

In this paper, we present a method to create and identify designer personas as archetypical paths through style space and provide a prototype implementation of it. For this, we use the Evolutionary Dungeon Designer (EDD), a research platform for exploring mixed-initiative creation of adventure and dungeon crawler content [8,9]. Data from 48 users designing game levels with the tool following different goals and styles have been used to fit the models. Based on this data, we clustered room styles to identify a dozen distinct types of rooms. To understand the typical progress of designers and validate the clustering, we visualize how typical design sessions traverse the various clusters. We also perform frequent sequence mining on the design sessions to find a small handful of designer personas.

Background

Player modeling, the ability to recognize general socio-emotional and cognitive/behavioral patterns in players [10], has been appointed by the game research community as an essential process in many aspects of game development, such as designing of new game features, driving marketing and profitability analyses, or as a means to improve PCG and game content adaptation. Player modeling frequently relies on data-driven and ML approaches to create such models from user data or user-generated gameplay data [4, 11–13].

Modeling users and players can have different goals and use multimodal data. User data can be used to understand and enable behavior for agents in games. For instance, Melhart et al. model a user's *Theory of Mind*, finding that players' perception of an agent's frustration is more a cognitive process than an affective response [11]. Alvarez and Vozaru explored personality-driven agents, evaluating how observers judged and perceived agents using their personality data from their personality tests when encountering multiple situations [14]. Gameplay data can show player behavior, as well as help developers understand their player base. Melhart et al.

used gameplay data from *Tom Clancy's The Division* to find predictors of player motivation [12], and Drachen and Canossa used playing behavior data from *Tomb Raider Underworld* and identified four types of players as behavior clusters, which provide relevant information for game testing and mechanic design [15].

Furthermore, the combination of Machine Learning (ML) with PCG has led to the rise of Procedural Content Generation via Machine Learning (PCGML), defined as the generation of game content by models that have been trained on existing game content [16]. PCGML has been used for autonomous content generation, content repair, content critique, mixed-initiative design, or content adaptation. A promising PCGML usage is in the area of content adaptation, where using player and user models are essential to adapt the generated content [17–19].

Content adaptation can take place as players play or use the content online or offline, building models from collected data. For instance, Duque et al. adapt and adjust the difficulty of generated content as players play the game using bayesian optimization [17]. Summerville et al. model players automatically and implicitly by learning from video traces; generating levels that correspond to the latent player models [20]. Player models can also be used to enhance and adapt design tools, specifically MI-CC tools. Migkotzidis and Liapis use player models as surrogate models to generate content assisting game designers in the creation of more relevant content for specific players [21]. Similarly, Holmgård et al. use player personas based on player archetypes as content critics to help designers adapt their content to different archetypes [4]. Their work on player personas is similar to our proposed work, yet instead of personas based on player archetypes, we propose personas based on design style.

Designer-centric Perspective

Mixed-initiative co-creativity (MI-CC) [1], is the subset of PCG algorithms where human users and AI systems engage in a constant mutual inspiration loop towards the creation of game content [22–26]. Understanding the designer's behavior and experience, as well as predicting their intentions is key for mixed-initiative creative tools while aiming to offer in real-time user-tailored procedurally generated content.

While player modeling is key for generating content adapted to players, adapting tools, systems, and AI methods for designing games requires a shift towards a designer-centric perspective, focusing on Designer Modeling. *Designer Modeling*, akin to player modeling, refers to the creation of models of either individual designers or groups of designers informed by how they create various types of content. Liapis et al. [27, 28] introduced designer modeling for personalized

experiences when using computer-aided design tools, with a focus on the integration of such in automatized and mixed-initiative content creation. The focus is on capturing the designer's style, preferences, goals, intentions, and iterative design process to create representative models of designers.

EXplainable AI (XAI) is an emergent research field that holds substantial promise for improving model explainability while maintaining high-performance levels [29, 30]. Explanations should be aligned with the users' understanding to not hinder the usability of systems, as demonstrated by Nourani et al. [31], who discuss the effects of meaningful and meaningless explanations to users of an AI interactive systems.

Zhu et al. [32] proposed the field of eXplainable AI for Designers (XAID) as a human-centered perspective on MI-CC tools. This work discusses three principles of mixed-initiative, *explainability*, *initiative*, and *domain overlap*, where the latter focuses on the study of the overlapping creative tasks between game designers and black-box PCG systems in mixed-initiative contexts. This work deems of high relevance the inclusion of data-driven and trained artifacts to facilitate a fluent bi-directional communication of the internal mechanisms of such a complex co-creative process in which *the designer provides the vision, the AI provides capabilities, and they merge that into the creation*. Mapping the designer's internal model to the AI's internal model is suggested as a meaningful way for creating a common ground that establishes a shared language that enables such communication. Our method, and designer modeling in general, aims to develop this designer's internal model to enhance MI-CC tools by adapting the AI's functionality towards designers' needs and aligning to their objectives.

Moreover, Guzdial et al.'s [33] discuss the insufficiency of current approaches to PCGML for MI-CC, as well as the need for training on specific datasets of co-creative level design. Guzdial et al. work on the mixed-initiative Morai Maker [34] shows the relevance of exploring the ways designers and AI interact towards co-creation, identifying four human-AI relationships (friend, collaborator, student, and manager), as well as the different ways they impact on the designer-user experience. Our study advocates for the importance of designer modeling through ML as the generation of surrogate models of designer styles by training on existing designer-generated data, aiming for an improvement in quality and diversity in computational creativity and, in particular, MI-CC tools.

The Designer Preference Model in EDD

EDD is an MI-CC tool where designers can create dungeons and rooms; meanwhile, a PCG system analyzes their design and proposes suggestions to the designer [9,

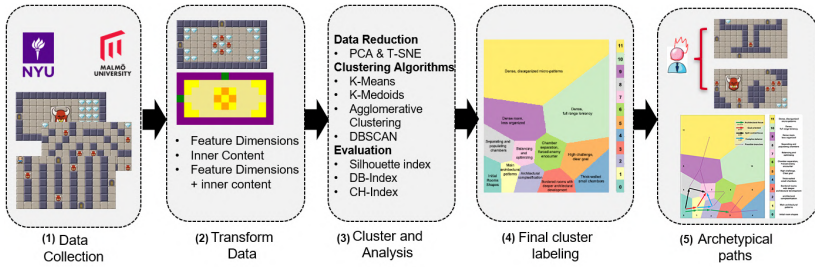


Figure 1: The stages of the design style clustering development: (1) Data was first collected through two user studies. (2) Then, using the design sequences, the data was processed into five different datasets, one using the room images, a second using the tiles information, and three using tabular information. (3) A data reduction technique was applied to different datasets, and then they were clustered and internally evaluated. (4) The clusters were formed, picked from the best performing methods, and labeled based on the data points within each cluster. The clusters were evaluated by visualizing how a typical design session traverse the various clusters, and K-Means (K=12) was chosen as the final approach. (5) Finally, using this final approach all the sequences were clustered and archetypical paths were identified.

35]. EDD uses *Interactive Constrained MAP-Elites* (IC MAP-Elites) [8], an evolutionary algorithm that combines Constrained MAP-Elites [36] with interactive and continuous evolution.

The work presented in [37] introduced the Designer Preference Model, a data-driven solution that learns from user-generated data based on their choices while using EDD. Both systems constantly interact and depend on each other, so that the Designer Preference Model learns from the selected suggestions, and IC MAP-Elites uses the Designer Preference Model as a designer’s surrogate model to complement the fitness evaluation of new individuals. The results showed the need for stability and robustness in the data-driven model, to counterbalance the highly dynamic designer’s creative process.

Concepts and Definitions

Our work draws from ideas, concepts, and definitions introduced by Liapis et al., such as the core designer model loop when using CAD tools, what can be modeled: preferences, style, goals, processes, and their definition, and particularly, the use of designer modeling as an individual or collective model [27]. We support our approach on the idea of style as a particular type of designer’s preference, and that a collective model can be used to form a stable and static design space, which after being interacted with by designers, can be adapted towards them.

Design Style

There exist many different styles when creating content, especially levels, that designers can create and adapt to accomplish their goals and the experiences they want for players. On a general level, *Design Style* encompasses the creative process from conceptualization, prototyping, reflection, adaptation, especially when following different processes or constraints during collaboration. Taking a more concrete and operational level, *Design Style* can be analyzed as overarching goals that different designers have when creating a dungeon. For instance, dungeons in games such as *Zelda* [5] or *The Binding of Isaac* [38], represent a particular playing style planned by the designer. In the former, low tempo, exploring the dungeon, and secret rooms define the style of the dungeons, whereas in the latter, high tempo, optimizing time and resources, small rooms, and in general high-challenge define the dungeons.

While interesting and relevant to understanding the designers' holistic design process and the expected player experience, *Design Style* can also be discussed on an individual room basis. Rooms have their own set of characteristics and styles that can be identified and modeled to understand their design process. Some would prefer to create the room's architecture first to then create the goals within, whereas others would like to place strategic objectives around and then create the architecture around it or alternating between both. Even with such a division, how to reach those design styles is not straightforward and does not require the same strategy, which also shows the preference and style of individual designers. For instance, if the goal is to create a challenge to reach a door, the designer could create a room with a substantial number of enemies, create a concentrated high-challenge in the center of the room, or divide the room into smaller choke areas. Therefore, in this paper, we take a simplified view of *Design Style* and treat it as the style designers follow to create a room, informed by the individual steps each has taken connected to their preferences and goals.

Room Style Clustering

This paper presents an approach and method for the implementation of designer personas: an analysis of designer style clustering to isolate archetypical paths that can later be used to build ML surrogate models of archetypal designers. Such models would allow for an stable and static design and solution space to be traversed by designers, adapting it to them as they explore their creative process.

The proposed system builds on top of EDD's Designer Preference Model and preliminary results [37], expanding it to classify the designers' designs based

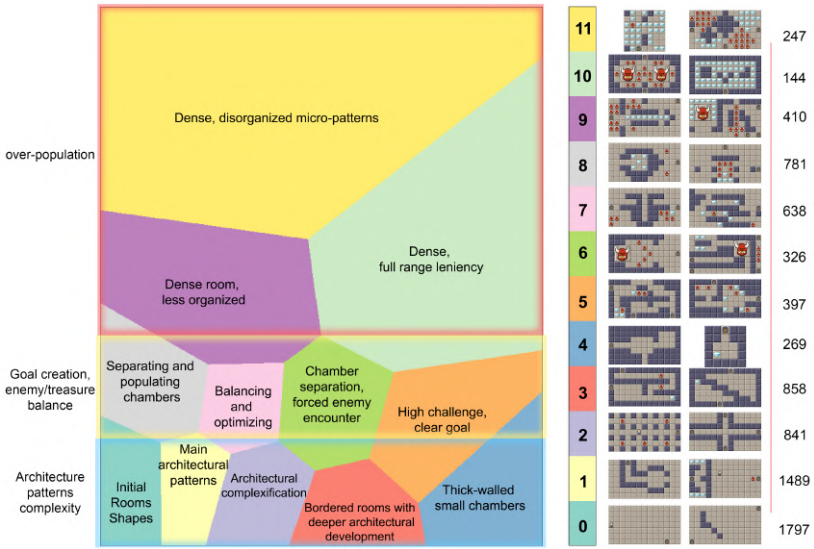


Figure 2: Best resulting cluster set. K-Means (K=12), using the Tiles Dataset. While it scores slightly less in the internal indices than other setups, a qualitative analysis successfully gives us more granularity by subdividing the main bottom clusters, to label and cluster the design process of designers. Sample rooms belonging to each cluster are displayed on the right, next to the total number of rooms in the cluster.

on clusters developed using previously hand-made design sequences by expert and non-expert designers. Figure 1 illustrates our approach in five sequential stages, from data collection to experimentation and results. The first four stages are explained in the following subsections, whereas Section Designer Personas shows the experimental results.

Data Collection

We conducted two user studies where participants were tasked with designing and creating a dungeon with interconnected rooms without restrictions, using all

Table 1: Best performing setups based on their internal validation and visualization of clustered data points.

Algorithm	Data	K	◇	□	△	ARI	ARI(R)
K-Means	Tiles-PCA	9	0.43	0.73	9437.23	0.941	0.581
★K-Means	Tiles-PCA	12	0.41	0.77	9436.57	0.935	0.775
K-Means	Dimensions-PCA	12	0.43	0.73	7738	0.84	0.876
K-Means	Combined-PCA	10	0.41	0.73	8455.34	0.931	0.866
Agglomerative compl.	Tiles-PCA	6	0.501	0.64	4221.34	0.719	0.419

◇ Silhouette Score □ Davies Bouldin Index △ Calinski-Harabasz Index

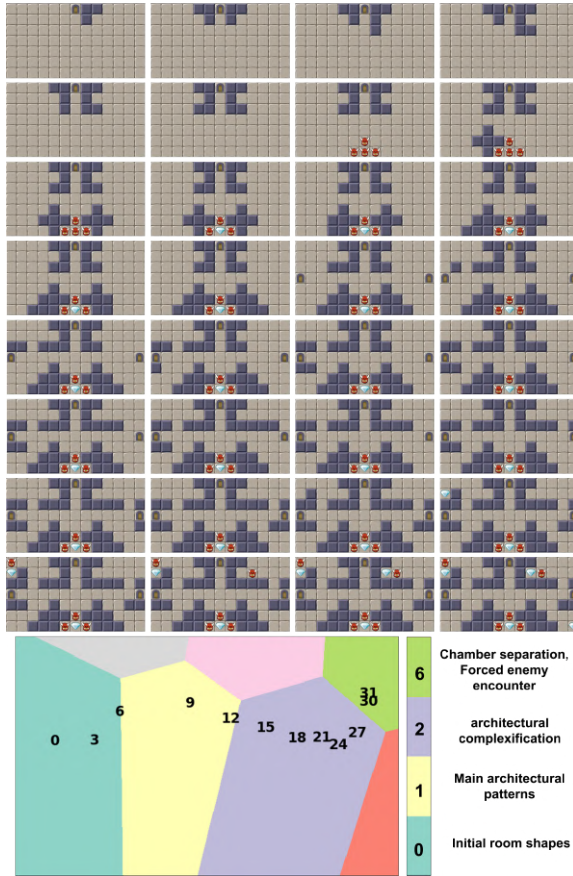


Figure 3: Example of a room’s design step sequence and its clustering. At the top, we present the design steps of one of the rooms in the dataset starting at the top left with the first design step. At the bottom, it is the actual trajectory of the design in the cluster space. Numbered and in black (every 3 steps), it is shown how each step of the design process is clustered by our approach

the available tiles i.e. floor, wall, treasure, enemy, and boss tiles. All participants were introduced to the tool before the design exercise. User-generated data was gathered during the complete design session, creating a new data entry every time the designer edited the dungeon. In total, we had 40 participants, 25 of these (i.e. NYU participants) were industry or academic researchers within the Games and AI field, and the other 15 (i.e. MAU participants) were game design students. This resulted in a diverse dataset composed of 180 unique rooms like the ones depicted in Figure 1, that was pre-processed and clustered in the subsequent stages.

Dataset pre-processing

From the 180 unique rooms, we extracted and used each room's design step sequence, from their initial design to the more elaborated end-design, to compose a richer dataset that could capture the design process of a designer rather than focusing on the end-point. This resulted in a dataset with 8196 data points. Moreover, five different copies of the dataset were created to analyze and compare the performance of the clustering stage using the following image pre-processing methods:

1. **Room:** No pre-processing. Room images are fed into the next stage as they were created by the designer, with a resolution of $1300 \times 700 \times 3$, corresponding to width, height, and RGB (3 color channels).
2. **Tiles:** Each room tile type is mapped to a single-color pixel and the rooms are simplified to a pixel-tile based representation, as shown in the second stage of Figure 1. The dimensions are downscaled to $13 \times 7 \times 3$.
3. **Dimensions:** Each room is described by its five IC MAP-Elites feature dimension values, excluding the similarity scores: LINEARITY, LENIENCY, #MESOPATTERNS, #SPATIALPATTERNS, and SYMMETRY. A complete description of these features can be found in [39].
4. **Inner Content:** Each room is described by 12 values, related to the count, sparsity, and density of the enemy, treasure, floor, and wall tiles contained in it.
5. **Combined:** A combination of the **Dimensions** and **Inner Content** methods.

Clustering and Analysis

We used the Scikit-learn machine learning toolset [40] to run all setups, data reduction and clustering algorithms, and to internally evaluate the clusters. To obtain the best set of clusters, we ran different setups with the above datasets. The data was reduced to two meaningful dimensions with two different data reduction algorithms, Principal Component Analysis (PCA) and T-Distributed Stochastic Neighbor Embedding (T-SNE). For both dimension reduction algorithms, we fit the algorithms with each individual dataset, setting to two principal components and in the case of T-SNE using PCA as an initializing algorithm, and transforming the data into a *pca_dataset* and *tsne_dataset*. Each two-dimensional point in the new datasets represents a step in the sequences described above. Preliminary tests without dimensionality reduction using the same distance metrics were conducted for all datasets except for **Room** and **Tiles** since the amount of dimensions made it computationally infeasible. These did not result in good space partition, achieved

worse internal indices, and produced disjointed clusters. For detailed information, see the supplementary material.

All the resulting datasets were then clustered using K-MEANS, K-MEDOIDS, AGGLOMERATIVE CLUSTERING, and DBSCAN. K-Means was initialized using the standard `k-means++` implementation in `scikit-learn`, which initializes all centroids distant from each other. K-Medoids was initialized similarly, using the standard `k-medoids++`, and tested using the *cosine*, *euclidean*, and *manhattan* distances. Agglomerative clustering is a hierarchical clustering approach using a bottom-up approach implemented in `scikit-learn` using four different linkage criteria for comparing data points: *Ward*, *Complete*, *Average*, and *Single*. Finally, DBSCAN clusters points based on density separated by low-density areas; thus, DBSCAN automatically finds k based on two parameters, ϵ describing the maximum distance between points and *min_samples* describing the minimum amount of samples within a group to be considered a cluster. K-Means, K-Medoids, and Agglomerative clustering were tested using multiple K values ranging from 3 to 13, and DBSCAN was tested with several ϵ values ranging from 0.3 to 1.0, and *min_samples* ranging from 2 to 9.

Since we lack a labeled dataset (i.e. ground truth) for cluster validation, we evaluated the results from all setups using the internal indices below. However, good values in internal indices are indicative and do not imply the best information retrieval, which is why we also manually inspect the clusters and the rooms grouped together.

- **Silhouette Score** (\diamond): The Silhouette Score shows how similar a data point is to the cluster it is associated with, through calculating the difference between the distance from the point to the points in the nearest cluster and the distance to the points in the actual cluster. The value is bounded from -1 to +1, with values closer to +1 indicating a good separation of the clusters, and closer to -1 meaning that some points might belong to another cluster.
- **Davies-Bouldin Index** (\square): The DB-index is the ratio between the within-cluster distances and between-clusters distances. With this, we can have an insight into the average similarity of clusters with their closest cluster. The value is bounded from 0 to +1, where values closer to 0 relate to clusters that are farther apart from each other and less dispersed, thus, this index is more crucial when we have more dense representations.
- **Calinski-Harabasz Index** (Δ): The CH-index is another index related to the density of the clusters and how well separated they are. The score is the ratio between the within-cluster dispersion (compactness) and the between-cluster

dispersion (separation). The CH-index is positively unbounded, and the higher the score the better.

- **Adjusted Rand Index (ARI):** *ARI* computes the similarity between two set of clusters. We tested our setups by using 70% of the data as training set to fit the cluster algorithm and the other 30% to analyze how the fitted clusters clustered unseen data. Then, using our setups fitted with the full dataset, we compared these 30% test set using *ARI*, giving us an estimated performance value with unseen data. Additionally, we computed $ARI(R)$ by splitting the data based on the room sequences rather than on the individual design steps to remove any relation between training and test set.

Cluster Labelling

Table 1 shows the best performing setups according to their internal indices scores and manual qualitative analysis of the clusters, rooms clustered together, and sequence analysis (further explained in section Designer Personas). The algorithm selection process followed a systematic analysis of each algorithm with each dataset from the lowest to the highest hyper-parameter value for each algorithm. We focused on the setups that both scored the best in the different indices (as a combination) and using the *elbow method*, which is a common heuristic in clustering to select the best k . We filtered our results by $k \geq 6$ as it gave more meaningful partitions and better groups overall. Some setups with lower k values gave good overall internal indices scores but logically, tended to make big clusters and supersets. For instance, K-Means ($k=3$) using **Tiles** dataset scores $\diamond = 0.55$, $\square = 0.66$, and $\triangle = 9049.65$. However, the three clusters acted as supersets, dividing the space into an upper "over-population" cluster, and into two smaller bottom partitions, one to the left accounting for the architecture of the rooms and one to the right focused on challenge and leniency. By using $k \geq 6$, we guarantee that at least the big clusters would be divided into meaningful subsets such as the ones shown in figure 2.

When using the **Dimensions** and **Combined** datasets, the clusters perform well in certain indices than when using the **Tiles** dataset with certain algorithms. However, when analyzing the resulting setups, they missed a clear relationship between the clustered rooms, which was exacerbated when analyzing sequences on these setups, where they missed continuity between clusters and either jump around clusters or had close to no movement between them. When using these datasets and analyzing the sequences, we notice that there are sequences where the cluster set is robust to seemingly meaningless changes, whereas jumps occurred when something important changed (or was added), and in other sequences, the clusters are very sensitive to changes. We believe this is because similar rooms (one or a

few steps in difference) could have a big impact on several dimensions within these datasets, such as a room becoming more symmetric and fully linear by simply adding a few walls. This, combined with the lack of natural continuity in many of the analyzed examples, negatively affected their evaluation.

Conversely, given that we are creating tile-based rooms and dungeons, the **Tiles** dataset had more representative features, which when used, generally performed better in the evaluated internal indices, and the clusters meaningfully separated the data. Similar robustness and beneficial behavior as when using the **Dimensions** and **Combined** datasets was also encountered while showing a clear continuity between designs that supports its usability (further discussed in section Designer Personas). Figure 2 shows the best-resulting cluster set found among all the experiments run. K-Means (K=12) with the **Tiles** dataset was chosen as the best cluster set. While K-Means (K=9) performs better in all indices, K=12 gave us more granularity within the bottom clusters. In our case, this was more important, especially since the difference was minimal, as the bottom area of the design space is denser than others containing much of the early design steps and their refinement.

In the figure, we have plotted on top of the clusters the labels describing in general, the content that is within them. Labels were defined based on the rooms that were clustered together and surrounding clusters' content. The following is a description of the clusters and rooms clustered together:

0. Empty-Initial rooms: This cluster relates mostly to the initial designs made by the designers. These designs are from completely empty rooms to initial work-in-progress structures.

1. Main architectural shapes: Similar to other clusters within the same layer, this cluster relates to the development and definition of main architectural patterns that are somewhat symmetric.

2. Architectural complexification: This cluster relates mostly to the complexification of wall structures by having dense wall chunks, representative architectural patterns, or symmetrical patterns.

3. Bordered rooms with deeper architectural development: This cluster relates mostly to rooms with an added wall border by the designer, and where the focus is to shape chambers and develop more visual structures.

4. Thick-walled small chambers: This cluster to the far right region in fig. 2, relates to more highly-linear, confined and maze-like rooms.

5. High challenge, clear goal: This cluster relates to well-shaped rooms with clear

wall structures and goals, towards more challenge.

6. Chamber separation with forced enemy encounter: This cluster relates to rooms that are in the process of a clear segmentation into corridors and chambers, and that enforce to some extent enemy encounters for the player.

7. Balancing and optimizing: This cluster contains a mix between corridors and chambers within rooms with a focus on balancing rooms and optimizing their design towards certain goals.

8. Separating and populating chambers: This cluster relates to the process of separating rooms into distinct chambers, focusing on the center of the room, and starting to populate rooms with enemies and treasures.

9. Dense, less organized: This cluster contains rooms that still have a certain objective but are moving towards more disorganized distributions of micro-patterns in relation to their density.

10. Dense, full range leniency: Focusing on density as the other two clusters within the same layer, this cluster relates to rooms that are in the full range of leniency from very rewarding treasure rooms to very challenging boss rooms.

11. Dense, disorganized micro-patterns: This cluster contains the extreme rooms that contain a high density of tiles, other than floor-tiles, without a clear structure or objective for the player.

Moreover, besides the local relation between clusters, the clusters are implicitly divided in three layers on the Y-axis. From bottom to top, (a) architectural patterns complexity, relating to clusters composed of rooms with clearer or complex shapes created with walls, from empty rooms to chambers. (b) Goal creation, enemy/treasure balance, with clusters containing the strategic addition of enemies and treasures to establish objectives in the room for the player. In terms of EDD, these rooms are composed of more meso patterns. And (c), over-population, which relates to clusters filled with less organized and dense rooms where the addition of enemy or treasure does not necessarily need to follow any clear objective. Identifying the designer in a layer, and the path they have taken to get there could show meaningful information in the design process. For instance, the intentions of the designer or in what phase of the design process they are at the moment; i.e. trying the tool or observing how the tool reacts or scraping their current goal towards a new goal within the room.

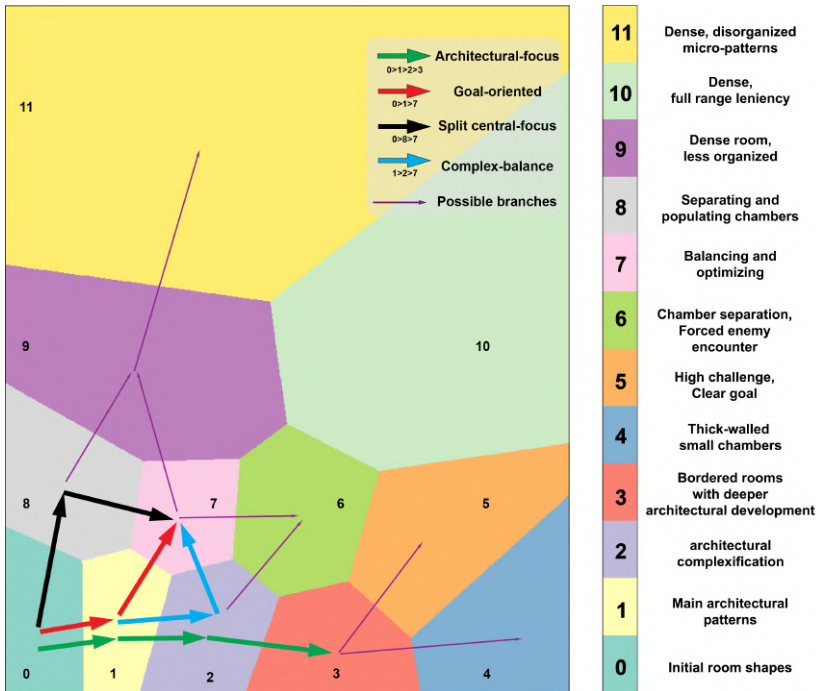


Figure 4: Final and common designer trajectories. The archetypal paths are represented with thick arrows, and were calculated using the frequencies of subsequences from 180 diverse rooms. Each color represent a unique trajectory; with green the ARCHITECTURAL-FOCUS, with red the GOAL-ORIENTED, with black the SPLIT CENTRAL-FOCUS, and with blue the COMPLEX-BALANCE. Finally, thinner purple arrows extending from clusters traversed by the archetypal paths show the multiple possible branches that an archetypal path can deviate or extend to.

Designer Personas

Once we created, evaluated, and labeled the room style clusters, we were able to cluster and visualize the paths of a typical design session. Figure 3 presents an example of the design sessions, where we cluster each step of the design. This sequential process revealed that there is an interesting continuity between room style clusters, even capturing when a designer applied one of the procedural suggestions due to bigger steps in the room style clusters. Further, through this process, we could understand the progress of designers in their design process and represent their design style and trajectory in relation to the traversed room style clusters rather than individual editions.

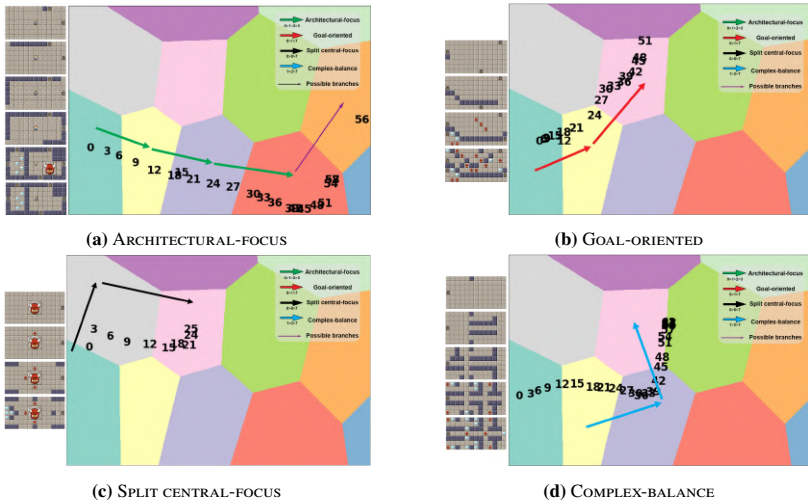


Figure 5: Example archetypical paths from different frequent sequences used to create the clusters. To the left of each subfigure, we present each key step in the trajectory i.e. when the design entered a new room style cluster. (a) presents the ARCHITECTURAL-FOCUS where the focus is firstly on creating the structural design of the rooms; the design process jumps back and forth suddenly to room style cluster 5 (one of the possible branches) due to the designer adding a boss, and removing it immediately. (b) presents the GOAL-ORIENTED where the design focus on a minimal architecture complexity and mix between adding structural changes and enemies/treasures. (c) shows the SPLIT CENTRAL-FOCUS where, intentionally, the designer creates a center obstacle with a boss and build around it. Finally, (d) presents the COMPLEX-BALANCE; the designer focuses on building complex, uncommon structures first and then add some goal with enemies and treasures, taking advantage of the spaces.

Unique Trajectories

Using the room style clusters in Figure 2, we clustered the design session of all the 180 designs. This resulted in trajectories where each individual design step was associated to some room style cluster. For instance, the design session in fig. 3 resulted the trajectory: $Trajectory = \{0>0>0>0>0>0>1>1>1>1>1>2>6>6>6\}$. We then converted these trajectories to their unique trajectory, which would convert the previous *Trajectory* into *Unique* = $\{0>1>2>6\}$, where the first and last element of the sequence are respectively, the starting- and end-points, with all the unique intermediate steps in between.

These unique trajectories varied in the starting point, length, and end-point, however, when analyzing the trajectories we identified common patterns among them. To gather the common patterns from the trajectories, we applied the Generalized Sequential Pattern (GSP) algorithm, which locates frequent subsequences in the analyzed trajectories. For instance, given three trajectories (a) $\{8>1>2>6>9\}$, (b)

{8>1>2>6>10}, and (c) {0>1>2>6}, none of these is a perfect match in its entirety, but GSP can spot that subsequences {1>2>6}, {1>2}, {2>6}, among others, appear with frequency = 3.

After doing a preliminary analysis, we identified some steps that we classified as “border designs”: steps that are borderline between two room style clusters. These *border designs* disrupted the sequence pattern mining with noise in the unique trajectories, specifically when these *border designs* entered a different room style cluster for just a few steps. Therefore, we filtered them out by applying a threshold $\theta = 3$, so that all subsequences inside one room style cluster with less than θ steps are removed from the main sequence. For instance, the sample trajectory {0>0>0>0>8>8>8>7>8} turns into {0>8} instead of {0>8>7>8}. Through this, we were able to reduce the noise and the search space, obtaining meaningful and frequent patterns.

Archetypical Paths through Style Space

In Figure 4, we present the archetypical paths, represented as thicker arrows to denote direction, which show the most frequent paths taken by designers either through their whole design process or as the initial meaningful steps. From all the collected unique trajectories, we have identified 4 main archetypical paths, labelled, ARCHITECTURAL-FOCUS, GOAL-ORIENTED, SPLIT CENTRAL-FOCUS, and COMPLEX-BALANCE. In addition, we have numbered each archetypical path for easier visualization and referencing. In the figure, we also include thinner purple arrows pointing to different room style clusters from several of the room style clusters that are part of the main paths. These are *possible branches* presented in the unique trajectories and added based on their frequency. Through these possible branches, the design of an archetypical session, can vary and extended or deviate the final design. Each archetypical path is defined and explained as follows:

Architectural-focus

The path followed by this archetype focuses first on designing the architecture of the room with walls. Through this, the design focuses on shaping the visual patterns, chambers, and corridors to give a clear space for adding goals and objectives with enemies and treasures. The sequence is denoted with a green arrow in Figure 4, and following the sequence {0>1>2>3}.

Goal-oriented

Design processes following this archetypical path, create the rooms in a more standard way, combining simpler symmetric wall structures with distributed

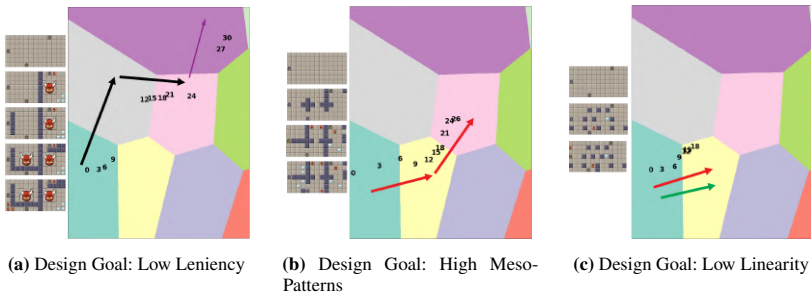


Figure 6: Example sequences created with specific design goals orthogonal to the designer personas. (a) and (b) are categorized as SPLIT CENTRAL-FOCUS and GOAL-ORIENTED, respectively. While (c) is uncategorized (with two possible designer personas).

placement of enemies and treasures. Thus, rather than focusing extensively on an individual part of the room, the rooms have an initial structure and then they are populated with some specific goal. The sequence is denoted with a red arrow in Figure 4, and following the sequence $\{0>1>7\}$.

Split central-focus

This archetypical path focuses on designing rooms with obstacles placed in the center of the room in the shape of enemies, treasures, or wall structures that clearly split the room into different areas. The design process is less organized than the other archetypes since it searches to achieve the split goal with any of the available tiles. The sequence is denoted with a black arrow in Figure 4, and following the sequence $\{0>8>7\}$.

Complex-balance

This archetypical path focuses on building complex symmetric shapes with a clear objective for the player and adapting the spaces with a balance of enemies and treasures. In general, the rooms created following this path are more unique and typically balanced. The sequence is denoted with a blue arrow in Figure 4, and following the sequence $\{1>2>7\}$.

Furthermore, using these archetypical paths, we can then categorize certain room style clusters as key clusters based on their contribution to the paths, their frequency, and their usage. Most of the paths go through or end in cluster 7 (“Balancing and optimizing”) and cluster 1 (“Main architectural patterns”), which relate to rooms that have a more explicit mix between corridors and small chambers, and more clear architecture. The rooms in those clusters are or shaped as end rooms, as in the case of cluster 7, or architecturally shaped to be “optimized” to a specific goal

e.g. a dense bordered room. Similarly, most of the sequences start from cluster 0 ("Initial room shapes"), with 134 out of the 180 designs, which correlates to the type of designs encountered in that clusters. Thus, it is understandable that most of the archetypical paths pass through these three clusters.

Nevertheless, it is the steps in-between that creates a clear differentiation between the archetypical paths, which is the benefit of observing the design process as a whole in the clustered room style space. For instance, in fig. 4, it can be observed that *SPLIT CENTRAL-FOCUS* starts in the same room style cluster as three other paths, and tentatively ends in the same room style cluster as three other. However, the designs following *SPLIT CENTRAL-FOCUS* are more different from the other trajectories, since it enters a room style cluster that is denser with several tile types in principle, and where designers seem to have a clearer goal.

Moreover, in figure 5, we present examples of each of the designer personas by visualizing the sequence of steps done in representative design sessions, showing what these paths would look like in practice. Each visualization of a designer persona has the key design steps to the left, where each image is in a sequence: the first is the first edition of the designer, the last is the final edition, and the in-between represent entering a new room style cluster.

Figure 5a shows the *ARCHITECTURAL-FOCUS*, where the designer first created the border of the room with a clear chamber division. As the designer adds and subsequently removes the boss, the design jumps to cluster 10, which is one of the possible branches, adding a high challenge.

Figure 5b shows the *GOAL-ORIENTED*, where the designer sketched the main shape of the room followed by alternating between enemies, treasures, and walls to adjust the room towards the goal for the player. In this example, the designer ends the design close to cluster 9, with a disorganized placement of tiles and a less aesthetical room, but forming small choke areas balancing the placement of enemies and treasures.

Figure 5c shows the *SPLIT CENTRAL-FOCUS*, where the designer directly started by adding a boss in the center of the room and using this as a reference point, shaped the rest of the room. Figure 5d shows the *COMPLEX-BALANCE*, where the designer focused on creating an uncommon structure and followed by adding enemies and treasures symmetrically, with clear individual areas for the player to approach.

Finally, further analyzing figure 5, it can also be observed an interesting dual tendency of the designers in the archetypical paths. This dual tendency is to either focus on the aesthetic configuration of the room based on what is perceived in the

editor exemplified the personas: ARCHITECTURAL-FOCUS and SPLIT CENTRAL-FOCUS, and to focus on the player experience exemplified the personas: GOAL-ORIENTED and COMPLEX-BALANCE. Nevertheless, both are not mutually exclusive, instead this illustrates adequately the dualistic role the designer has when using the tool and designing rooms. That of creating an aesthetically pleasing object as it is seen in the editor, and that of creating an experience.

Preliminary Evaluation

We recorded three design sessions with specific design goals orthogonal to the archetypical trajectories to preliminary evaluate our approach. These are shown in figure 6 with the design goals: figure 6a) Low Leniency (i.e., challenging room), figure 6b) high meso-pattern level (i.e., large quantity of guard, treasure, enemy, and ambush rooms), and figure 6c) low linearity (i.e., several paths between doors). Both figure 6a and figure 6b fitted two of the archetypical trajectories, SPLIT CENTRAL-FOCUS and GOAL-ORIENTED. Figure 6c shows a design achieving its goal but identified as an early design, aligned with two possible designer personas. This last example shows the situations we expect to normally encounter during design sessions, namely, uncertainty and adaptation. If the designer continued elaborating the level's architecture, they would align with the ARCHITECTURAL FOCUS else they would probably align with the GOAL-ORIENTED.

While these results requires a larger study that analyses and verifies the different possibilities and properties of using Designer Personas, especially, in-the-wild; this evaluation helps to understand and visualize how unseen design sessions map and align to possible archetypical paths regardless of the designer's design goal.

Discussion

Our work draws on many of Liapis et al.'s [27, 28] ideas, concepts, and goals but differs on the tool and type of game being created and the methods to create designer models. These differences strengthen the importance and usefulness of designer modeling and highlight this designer-centric perspective's holistic and generic properties.

The archetypical paths presented in sec. Archetypical Paths through Style Space relate to EDD and the options that exist within it. However, while important for EDD's development, this is used to demonstrate what can be done and as an exemplification of the method to model designers. Our method focuses on designer modeling through clustering the design space and the room style based on level design step sequences, resulting in archetypical paths. We leverage a collective

dataset to build a collective cluster model of a set of designers and utilize this to mitigate, to some extent, problems regarding lack of individual designer data and continuous model adaptation [37]. Our method allows us to better understand, cluster, categorize and isolate designer behavior. A virtual model of the designer's style could allow to better drive the search process for procedurally generating content that is valuable for designers and aligned with the designers' intentions, observed as well by Liapis et al. [27].

We expect that work on similar domains and tools working with level design could discuss these archetypical paths in relation to what designers create. They can build on and adapt the proposed method to model designers to their respective domains, identifying valuable and applicable designer personas. Then, these could be used to understand what designers create and to design adaptive systems and better Human-AI collaborations and interactions. The resulting Designer Personas have the potential to be used in many different scenarios. For instance, as objectives for a search-based approach to enable a more style-sensitive system, to evaluate the fitness of evolutionary generated content, or to train PCG agents via Reinforcement Learning [7].

Conclusions

This paper presents a novel method and meaningful steps towards designer modeling through an experiment on archetypical design trajectories analysis in an MI-CC environment. Through this, we characterize several representative design styles as designer personas. We have first run and compared several clustering setups to find the best partitioning of the room style using the design step sequence of the collected 180 unique rooms, ending in 8196 data points, and resulting in a set of twelve cohesive and representative room style clusters. We have then mapped these 180 design sequences in terms of these clusters, applying frequent sequence mining to find four frequent and unique designer styles, with related common sub-styles. As a result, we have presented a roadmap of design styles over a map of data-driven room style clusters.

Recognizing the designers' current style and the path taken so far, which would indicate a possible designer persona, would open the possibility for recognizing their intentions, preferences, and goals. This traced roadmap of designer personas could let a content generator anticipate a designer's next moves without heavy computational cost, just by identifying their current location on the map and offering content suggestions that lie in the most promising clusters to be visited next. Conversely, it would also identify designers who do not follow a certain path, i.e., deviating from the pattern, trying to understand their objective through their

design style. Finally, we aim at implementing our approach in a functional system within EDD to assess and validate the benefits and usability of these adaptable systems with human designers.

References

- [1] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-initiative co-creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 279, 281
- [2] A. Liapis, G. Smith, and N. Shaker, “Mixed-initiative content creation,” in *Procedural content generation in games*, pp. 195–214, Springer, 2016. 279
- [3] T. Lubart, “How can computers be partners in the creative process: Classification and commentary on the special issue,” *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 365 – 369, 2005. Computer support for creativity. 279
- [4] C. Holmgård, M. C. Green, A. Liapis, and J. Togelius, “Automated playtesting with procedural personas through mcts with evolved heuristics,” *IEEE Transactions on Games*, vol. 11, pp. 352–362, Dec 2019. 279, 280, 281
- [5] Nintendo R&D4, “*The Legend of Zelda.*” Game [NES], 1986. Nintendo, Kyoto, Japan. Last played January 2022. 279, 284
- [6] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 172–186, Sept. 2011. 280
- [7] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, “Pcgrl: Procedural content generation via reinforcement learning,” in *Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE’20*, pp. 95–101, AAAI Press, 2020. 280, 298
- [8] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 280, 283

- [9] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proceedings of the 12th International Conference on the Foundations of Digital Games*, FDG ’17, (New York, NY, USA), ACM, 2017. 280, 282
- [10] R. Thawonmas, J. Togelius, and G. N. Yannakakis, “Artificial general intelligence in games: Where play meets design and user experience,” *NII Shonan Meeting No. 130*, 2019. 280
- [11] D. Melhart, G. N. Yannakakis, and A. Liapis, “I feel i feel you: A theory of mind experiment in games,” *KI - Künstliche Intelligenz*, vol. 34, pp. 45–55, Mar 2020. 280
- [12] D. Melhart, A. Azadvar, A. Canossa, A. Liapis, and G. N. Yannakakis, “Your gameplay says it all: Modelling motivation in tom clancy’s the division,” in *2019 IEEE COG*, pp. 1–8, Aug 2019. 280, 281
- [13] A. Canossa and G. Smith, “Towards a procedural evaluation technique: Metrics for level design,” in *The 10th International Conference on the Foundations of Digital Games*, p. 8, sn, 2015. 280
- [14] A. Alvarez and M. Vozaru, “Perceived behaviors of personality-driven agents,” in *Violence | Perception | Video Games: New Directions in Game Research* (F. A. Igarzábal, M. S. Debus, and C. L. Maughan, eds.), pp. 171–184, transcript Verlag, 2019. 280
- [15] A. Drachen, A. Canossa, and G. N. Yannakakis, “Player modeling using self-organization in tomb raider: Underworld,” in *2009 IEEE Symposium on Computational Intelligence and Games*, pp. 1–8, Sep. 2009. 281
- [16] A. Summerville, S. Snodgrass, M. Guzdial, C. Holmgård, A. K. Hoover, A. Isaksen, A. Nealen, and J. Togelius, “Procedural content generation via machine learning (pcgml),” *IEEE Transactions on Games*, vol. 10, no. 3, pp. 257–270, 2018. 281
- [17] M. G. Duque, R. B. Palm, and S. Risi, “Fast game content adaptation through bayesian-based player modelling,” in *IEEE Conference on Games (CoG)*, 2021. 281
- [18] J. Togelius, R. De Nardi, and S. M. Lucas, “Towards automatic personalised content creation for racing games,” in *2007 IEEE Symposium on Computational Intelligence and Games*, pp. 252–259, 2007. 281

- [19] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011. 281
- [20] A. Summerville, M. Guzdial, M. Mateas, and M. Riedl, "Learning player tailored content from observation: Platformer level generation from video traces using lstms," *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 12, no. 2, pp. 107–113, 2016. 281
- [21] P. Migkotzidis and A. Liapis, "Susketech: Surrogate models of gameplay as a design assistant," *IEEE Transactions on Games*, pp. 1–1, 2021. 281
- [22] M. Charity, A. Khalifa, and J. Togelius, "Baba is y'all: Collaborative mixed-initiative level design," in *2020 IEEE Conference on Games (CoG)*, pp. 542–549, 2020. 281
- [23] T. Machado, D. Gopstein, A. Nealen, and J. Togelius, "Pitako-recommending game design elements in cicero," in *2019 IEEE Conference on Games (CoG)*, IEEE, 2019. 281
- [24] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.," in *AIIDE*, 2013. 281
- [25] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 281
- [26] A. Liapis, G. N. Yannakakis, and J. Togelius, "Generating Map Sketches for Strategy Games," in *Proceedings of Applications of Evolutionary Computation*, vol. 7835, LNCS, Springer, 2013. 281
- [27] A. Liapis, G. Yannakakis, and J. Togelius, "Designer modeling for personalized game content creation tools," in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19, pp. 11–16, AI Access Foundation, 2013. 281, 283, 297, 298
- [28] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for sentient sketchbook," in *2014 IEEE Conference on Computational Intelligence and Games*, pp. 1–8, Aug 2014. 281, 297
- [29] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018. 282

- [30] F. Doshi-Velez and B. Kim, “Considerations for Evaluation and Generalization in Interpretable Machine Learning,” in *Explainable and Interpretable Models in Computer Vision and Machine Learning* (H. J. Escalante, S. Escalera, I. Guyon, X. Baró, Y. Güçlütürk, U. Güçlü, and M. van Gerven, eds.), pp. 3–17, Cham: Springer International Publishing, 2018. 282
- [31] S. M. Mahsan Nourani, Samia Kabir and E. D. Ragan, “The effects of meaningful and meaningless explanations on trust and perceived system accuracy in intelligent systems,” in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 2019. 282
- [32] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation,” in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, Aug 2018. 282
- [33] M. Guzdial, N. Liao, and M. Riedl, “Co-creative level design via machine learning,” in *Joint Proceedings of the AIIDE 2018 Workshops co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2018), Edmonton, Canada, November 13-14, 2018.*, 2018. 282
- [34] M. Guzdial, N. Liao, J. Chen, S.-Y. Chen, S. Shah, V. Shah, J. Reno, G. Smith, and M. O. Riedl, “Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019. 282
- [35] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, 2018. 282
- [36] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, ACM, 2018. 283
- [37] A. Alvarez and J. Font, “Learning the designer’s preferences to drive evolution,” in *Proceedings of the 23rd European Conference on the Applications of Evolutionary and bio-inspired Computation*, EvoApplications ’20, 2020. 283, 284, 298

- [38] E. McMillen and F. Himsl, “*The Binding of Isaac.*” Game [PC], 2011. McMillen, Edmund and Himsl, Florian, California, US. Last played December 2021. 284
- [39] A. Alvarez, J. M. M. Font Fernandez, S. Dahlskog, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *IEEE Transactions on Games*, pp. 1–1, 2020. 287
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 287

PAPER X - TROPETWIST: TROPE-BASED NARRATIVE STRUCTURE GENERATION

Alberto Alvarez and Jose Font

ABSTRACT

Games are complex, multi-faceted systems that share common elements and underlying narratives, such as the conflict between a hero and a big bad enemy or pursuing a goal that requires overcoming challenges. However, identifying and describing these elements together is non-trivial as they might differ in certain properties and how players might encounter the narratives. Likewise, generating narratives also pose difficulties when encoding, interpreting, and evaluating them. To address this, we present TropeTwist, a trope-based system that can describe narrative structures in games in a more abstract and generic level, allowing the definition of games' narrative structures and their generation using interconnected tropes, called narrative graphs. To demonstrate the system, we represent the narrative structure of three different games. We use MAP-Elites to generate and evaluate novel quality-diverse narrative graphs encoded as graph grammars, using these three hand-made narrative structures as targets. Both hand-made and generated narrative graphs are evaluated based on their coherence and interestingness, which are improved through evolution.

PUBLISHED IN

Proceedings of the 18th Procedural Content Generation Workshop at the Foundations of Digital Games (FDG), ACM, 2022.

TROPETWIST: TROPE-BASED NARRATIVE STRUCTURE GENERATION

Introduction

There exists a plethora of games¹, with diverse genres and each containing a different set of gameplay mechanics, audio, level, graphic, and narrative facets. The creation and combination of these facets make game development a hard task, commonly involving a diverse group of developers [1]. Likewise, the generation of these facets in conjunction has been categorized as one of the biggest and most challenging tasks within computational creativity [2, 3]. However, games share common elements and underlying narratives, but it is non-trivial how to identify these, how to define and analyze these games structurally, or what type of common underlying structures exist; pointed out as well by [4, 5].

Among the different facets, narrative stands out in games as it helps to create meaning, make sense of situations, and make games [stories] recognizable [6–9]. Narrative structures can be used to describe how an experience or story is to be developed as argued by Barthes [10], and to create an abstract representation based on the narrative structure instead of a temporal and partially-ordered sequence of events [11]. Common narrative structures used in many domains are Aristotle’s drama structure, which subdivides a story into *exposition*, *climax*, and *resolution* or Propp’s analysis on the morphology of russian folktale, which revealed a common structure among them, denoted as Propp’s 31 “narremes” [12].

This paper presents *TropeTwist*, a preliminar system that uses Tropes [13, 14] extracted from TvTropes [15, 16] as patterns and fundamental units, which when combined can compose structures further representing other composed tropes. Common narrative structures can be identified and defined using *TropeTwist*. *TropeTwist* can define generic aspects of a story, leading to the identification of events, roles, and narrative elements, as well as a novel way to form narratives.

¹For instance, currently there are more than 68k games in steam <https://store.steampowered.com/search/?category1=998>.

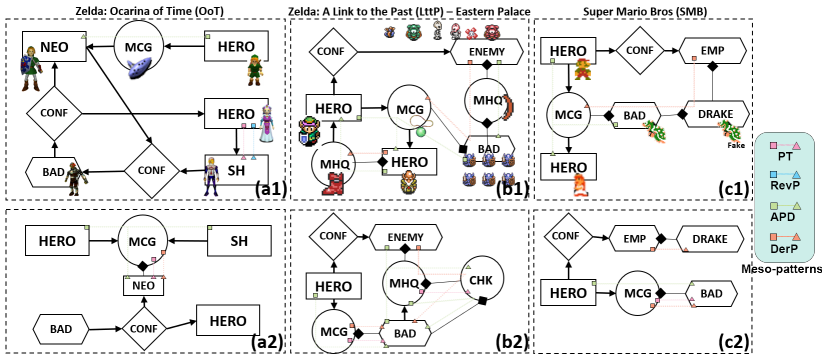


Figure 1: Proof-of-concept narrative structures of existing games (top row) created using TropeTwist with the available nodes (table 1). Bottom row shows exemplar elites generated with MAP-Elites using the respective top row narrative structure as root. Color matching squares, lines, and triangles denote different meso-patterns in the structures. Squares and triangles are the start and end of a meso-pattern, respectively.

As a proof-of-concept, we built, analyzed, and described structurally three game examples shown in figure 1, top row.

We propose graph grammars as indirect encoding of narrative graphs and the use of the Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [17] to generate novel variations (shown in figure 1, bottom row) using the proof-of-concept examples as roots. Simultaneously, we propose metrics to evaluate the resulting narrative graphs’ coherence, cohesion, and interestingness. Our preliminary results show that we can produce more interesting structures retaining coherence based on our metrics.

Related Work

Propp [12] analyzed Russian folktales identifying their fundamental structure in 31 steps. His work contributed to the identification of core elements, the proposal of actions and events as *functions* and narrative atoms, and roles that are recurrent within the folktales. Propp emphasized that these 31 *functions* and their arrangement were the structure and what gave meaning to the story discourse. Barthes [10] proposed three intertwined and progressively integrated levels in narrative work: *functions*, *actions*, and *narration*. His work is characterized by the proposal of fundamental narrative units in the *function* level to better assess and identify structures in a narrative. Furthermore, Baikadi and Cardona-Rivera [18] further discuss these fundamental units as *narremes* encoding narrative state and how they could be combined to narrative structures. Their work, similar to TropeTwist, proposes a graph structure of interconnected *narremes*. However, they defined

narrative axes like Barthes, where each connection between *narremes* means a change along a narrative axis. In games, the narrative is usually directed by quests, which Aarseth [19] discusses as a central element in games to make sense of other elements, and which are defined by Yu et al. as a form of structure, dividing the story into achievable rewards and partially ordered set of tasks [20].

Furthermore, the generation of narratives, stories, and quests using a variety of techniques such as planning algorithms [21, 22], grammars [23, 24], or machine learning [25, 26], is a growing and important field within games research and narrative research in general [8, 20, 27, 28]. One typical approach for the generation of content and stories is the use of patterns representing different elements such as level design patterns [9, 29], quest patterns and common quests in games [30, 31], or identifying fundamental units and assembling them based on various pre-conditions [32, 33]. A particular type of pattern is tropes, which are concepts that are recurrently used in transmedia storytelling [14, 15]. Horswill [34] focused on constructing an expressive language that could encode plot tropes as story fragments, composing a database of fragments combined sequentially with a planner. Similarly, Thompson et al. [13] used the idea of tropes as story bits where a system would construct valid stories from users' defined story bits with pre-and post-conditions. TropeTwist uses the idea of tropes for nodes and patterns in structures and encodes and represents these as a graph. *Scheherazade* is a system that can capture narrative structures by encoding and annotating narrative texts, which introduced the Story Intention Graph model, a formal and expressive representation of narratives [35].

Moreover, we use graph grammars and grammar recipes to generate structures. This approach is similar to how Dormans and Bakkes [36] generate missions and space using a “key and lock” structural idea. Our approach uses MAP-Elites, a quality-diversity algorithm that uses behavioral dimensions that are orthogonal to the objective function to store diverse individuals in a grid [17]. Evolutionary algorithms are a popular approach in PCG to generate diverse type of content [37], but not as much for narrative content. MAP-Elites have been used to generate content in different game facets such as levels [38, 39], mechanics [40], or enemy behavior [41].

Assessing narratives is a complex and non-trivial task. The goal is to create a narrative that is both syntactically correct (e.g., coherent and consistent) and semantically rich (e.g., novel and interesting) [42–44]. Perez y Perez and Ortiz [45] proposed a model to evaluate interestingness based on novelty and correct story recount, with emphasis on the story's opening, closure, and dramatic tensions.

Table 1: Tropes included and used in TropeTwist, extracted from [15].

Name	Symbol	Definition
Hero	HERO	A protagonist character.
Five-man band	5MA	Group composed by up-to-five archetypical characters.
The chosen one	NEO	Specific hero chosen as the one.
Superhero	SH	Specific hero with unique abilities.
Conflict	CONF	Non-specific problem to overcome between characters.
Enemy	ENEMY	A nemesis to the hero.
Empire	EMP	Collective enemy with the ambition of conquering the world.
Big bad	BAD	Specific enemy, which is the ultimate cause for all the bad.
Dragon	DRAKE	Specific enemy, which is the right hand of BAD.
Plot device	PLD	A feature or element that drives the plot forward.
Chekhov's gun	CHK	PLD relevant to the story
MacGuffin	MCG	PLD with irrelevant nature to drive the story.
May help in quest	MHQ	PLD important to resolve a conflict.

Szilas et al. [46] discuss interestingness as a paradox dramatic situation with obstacles and conflicts, albeit applicable to stories as successive events. Yet, to approach subjective measurements such as interestingness, most research turns towards having human evaluation [47, 48] or using such to form human models to be used as surrogate models [49, 50].

Building narrative structures with tropes

In storytelling, a trope [14] is a convention or figure of speech that the storyteller assumes to be recognizable by the audience. TvTropes is an online wiki that compiles and describes several thousand tropes in many sorts of media [15]. As exemplified by [16], tropes could be interconnected in graph-like structures, called story molecules, to succinctly depict the structure behind a narrative.

TropeTwist

TropeTwist elaborates on the concept of story molecule to represent narratives using graph-like structures of interconnected tropes, called narrative graphs (NG). NGs encode narrative structures in an abstract level that show and define the game's narrative structure and certain abstract properties such as key items, roles, relations, or main events. Table 1 shows all the included tropes to be used as nodes. Nodes are depicted (fig 1) with shapes specific to their trope base type: heroes (rectangle), conflicts (diamond), enemies (hexagon), and plot devices (circle). HERO is the base pattern of 5MA, NEO, and SH. ENEMY is the base pattern of EMP, BAD, and DRAKE. PLD is the base pattern of CHK, MCG, and MHQ.

Nodes in a narrative graph are necessarily interconnected by either unidirectional or bidirectional edges (with one or both arrowheads) or by entailment edges (with a single diamond head). Given nodes A and B, $A \diamond\text{---} B$, reads as "A entails B," whereas $A \rightarrow B$ denotes a relationship from A to B, and $B \rightarrow A$ the opposite. $A \leftrightarrow B$ denotes a reflexive relationship between A and B. As an example, $\text{HERO} \rightarrow \text{CONFLICT} \rightarrow \text{EMP}$ denotes a hero who is in conflict against an empire-type enemy, whereas $\text{HERO} \leftrightarrow \text{CONFLICT}$ denotes a hero who is in conflict with themselves. $\text{EMP} \diamond\text{---} \text{DRAKE} \diamond\text{---} \text{NEO}$, denotes an empire that entails a dragon enemy that, once beaten, will lead to the appearance of a chosen one hero, creating some causal links. The system is ambiguous by design. We take advantage of the ambiguity for 1) the generation of new structures (fewer constraints), 2) removing the focus on details by designers to let them focus on the overarching picture, and 3) for other systems to define and interpret these abstract properties.

Furthermore, interconnecting tropes can give rise to other tropes and patterns, described in the following section. The nodes and their respective trope and pattern were chosen from a subset of tropes in generic categories such as heroes or plot devices. These categories were inspired and chosen based on tropes from TVTropes, the division by James Harris [16], and previous research such as Propp's morphology [12] or Greimas' actantial model [51].

Trope Patterns

Tropes and interconnected tropes (i.e., subgraphs) give rise to different types of patterns. These patterns can be **micro-patterns**, encapsulating a single trope node, **meso-patterns**, often composed by more than one micro-pattern with special meaning, and **auxiliary patterns**, denoting graph problems. We calculate the relative tropes and patterns' quality within an NG and use this to assess the general quality of the graph. These qualities are proxies for certain characteristics among

the defined patterns that are used to evaluate the graphs, but they do not capture any story quality; especially, since we are only defining structures. When generating narrative graphs from a root (explained in section Evolving Narratives with Graph Grammars), the quality of a narrative graph becomes relative to the root, henceforth, the “root graph” (RG). In the following descriptions, we will use **EG** referring to the “evaluated graph” we are calculating the pattern’s quality (the generated individual), and **RG** to refer to the relative and root graph. When using subscript “pat,” we refer to the current pattern that is evaluated.

For most patterns, we calculate three general qualities (indicated when used) that add to the quality of the pattern. $G_q(pattern)$ relates to the *Generic* quality of patterns in EG, which calculates the general occurrence of a pattern within EG compared to its occurrence in RG, calculated in eq. 1. $R_q(pattern)$ relates to the *Repetition* quality of patterns, which calculates if a trope is unique in EG ($R_q(pattern) = 1$) or its ratio among the same base pattern. Lastly, $I_q(pattern)$ relates to the *Involvement* quality of patterns in EG, which calculates the amount of associations a pattern has with **structure patterns**. Involvement means that the pattern is either *source* or *target* in a structure and is calculated as the ratio of structure pattern involvement by the structure pattern count in EG. These three metrics incentivize graphs with similar amount and type of nodes than RG, minimal repetitions, and more involvement.

$$G_q(pattern) = 1.0 - |RG_{pat} - EG_{pat}| / \max(RG_{pat}, EG_{pat}) \quad (1)$$

Micro-Patterns

Micro-patterns are the fundamental unit in the system, which aims at categorizing different sets of the individual patterns that are shown in table 1. Micro-patterns are single nodes and the basic building block that, when interconnected, allows the detection of meso-patterns.

Structure Pattern (SP) is any type of trope that would give some structural definition to a narrative, whether this being a conflict, specific act, or a part in a dramatic arc (e.g., climax). Currently, the only type of structure trope is the **CONFLICT (CONF)** trope, which represents the most basic structural interaction. The quality SP_q is calculated as the equally weighted linear combination of:

$$SP_q = G_q(SP) + I_q(SP) \quad (2)$$

Character Pattern (CP): are identified as nodes within the narrative that could be

either the player, possible ally or enemy NPCs, or simple enemies. In TropeTwist, it is distinguished between heroes and villain patterns, and these are commonly used as **sources** or **targets** (or both) of other patterns, and on a few special occasions to denote a relation to another character. The quality CP_q is calculated per group (heroes and villains), and it is the equally weighted linear combination of:

$$CP_q = G_q(CP) + R_q(CP) + I_q(CP) \quad (3)$$

Plot Device Pattern (PDP) is described as the element within the narrative that moves it forward, as a goal, object, or dramatic element to be used or encountered by any of the characters. The quality PDP_q is calculated as the equally weighted linear combination of:

$$PDP_q = G_q(PDP) + R_q(PDP) \quad (4)$$

Meso-Patterns

Meso-patterns are the features that emerge in the narrative from dynamically combining micro-patterns and, on some occasions, these with other meso-patterns. They are always composed of more than one pattern denoting some spatial, semantic, or usability relationship within the narrative graph. We identified a subset of Tropes (extracted from TVTropes [15]) that requires or works as the combination between more fundamental units. For instance, the *reveal pattern* relates to the “Good all along” or “evil all along.”

Conflict Pattern (ConfP) is a type of structure pattern composed by a conflict node (Con), a source s node, and a target t node, which are both CPs and usually a hero and a villain or the same character as s and t . For instance, the subgraph HERO \rightarrow CONFLICT \rightarrow EMP, indicates that a hero CP has a conflict with an enemy CP. A conflict node can be used indefinitely to define several ConfP. A ConfP is also either EXPLICIT or IMPLICIT. EXPLICIT conflicts are explicitly encoded in the graph and directed from s to t passing through the conflict trope. On the other hand, IMPLICIT conflicts relates to the conflicts from t (or derivatives) to s (or derivatives) that are not encoded in the graph. For instance, the previous example is an EXPLICIT conflict from HERO to EMP, and at the same, the EMP has an IMPLICIT conflict with the HERO. The quality $ConfP_q$ is calculated as the equally weighted linear combination of:

$$ConfP_q = G_q(ConfP) + R_q(ConfP) \quad (5)$$

Derivative Pattern (DerP) defines a relationship between tropes connected by “entails” connections ($\diamond\text{---}$). Therefore, a DerP contains a list of patterns connected by entails, named derivatives. DerP starts from a root micro-pattern and continue until no more “entail” connections are encountered, effectively establishing a hierarchy from the root derivative to the rest. By design, the patterns within a DerP have a local and temporal order and a causal relationship. For instance, in the subgraph $EMP \diamond\text{---} DRAKE \diamond\text{---} NEO$, engaging with the *EMP*, entails both the conflict with *DRAKE* and the appearance of *NEO*. This means that only by overcoming the *DRAKE*, *NEO* will appear - as a new hero or the evolution of another. The quality $DerP_q$ is calculated (eq. 6) based on its $G_q(DerP)$, the ratio of derivatives within the DerP among the total amount of derivatives across all DerPs in EG ($ratio\theta_q$), and the derivatives’ diversity.

$$DerP_q = G_q(DerP) + ratio\theta_q + \frac{\sum_{i=0}^{len(DerP_{der})} DerP_{der_i,basepat}}{len(DerP_{der})} \quad (6)$$

Reveal Pattern (RevP) connects two independent CPs as one, meaning that character A was, in fact, always character B, and vice-versa. This pattern identifies confusion and surprise within an EG, as, for instance, a villain could have been, in fact, “Good All Along”². In practice, a RevP is identified as a villain or hero connected with a unidirectional connection (\rightarrow) to another hero or villain. As a consequence, all existing conflicts between them would become *fake*. $RevP_q$ is calculated based on its $G_q(RevP)$, the number of reveals in EG in relation to characters, and the number of fake conflicts given the specific reveal.

$$RevP_q = G_q(RevP) + \frac{len(EG_{RevP})}{len(EG_{CP})} + \left(1.0 - \frac{\sum_{i=0}^{len(EG_{conf})} \begin{cases} 1, & \text{if } RevP \in x_i \\ 0, & \text{otherwise} \end{cases}}{len(EG_{conf})} \right) \quad (7)$$

Active Plot Device Pattern (APD) operationalize and integrate PDPs within a narrative since PDPs only describe an abstract goal or target. In practice, an APD is identified as PDPs that have at least one incoming connection, and optionally, one single outgoing connection. These limitations are added to limit the effect of a PDP within a narrative. APD_q is measured based on its $G_q(APD)$, and the APD’s usability, calculated based on the sum of incoming and outgoing connections

²<https://tvtropes.org/pmwiki/pmwiki.php/Main/GoodAllAlong>

divided by half of the nodes in EG depicted as $bal\gamma_q$, penalizing APDs for not using all their connections.

$$APD_q = G_q(APD) + bal\gamma_q \quad (8)$$

Plot Points (PP) are key events within the EG, identified as discrete moments given some pattern. The derivatives within a *DerP*, *RevP*'s source, and PDPs that are *APD* are considered as plot points. PP_q is measured based on the number of PPs within RG ($G_q(PP)$), and the number of PPs within EG in relation to the number nodes within it ($Balance_q(PP)$).

$$PP_q = G_q(PP) + Balance_q(PP) \quad (9)$$

Plot Twist (PT) takes advantage of plot points to identify those that could have a bigger impact on the narrative. In practice, *PTs* consider the source of *RevP*, derivatives from *DerP* that are a different micro-pattern than the root of the *DerP* (except PDPs), and *APDs* that are connected to other *APDs*. For instance, in the subgraph: EMP \diamond — DRAKE \diamond — NEO, given that NEO is a different micro-pattern than root EMP (Hero and Villain, respectively), NEO will be identified as a *Plot Twist* as it alters the “natural” order in the *DerP*. PT_q is based on the number of *PTs* within RG ($G_q(PT)$), the *PT*'s involvement in EG, and the balance of *PTs* based on the PPs in EG. Involvement varies depending on the associated pattern to *PT*. When a *PT* is associated with a *RevP*, involvement is calculated as how much the structure changes based on that (i.e., how many fake conflicts are created). When it is related to *DerP*, involvement is calculated as how different the pattern is and its order within the derivatives. Finally, when it is related to *APD*, involvement is based on how usable the *APD* is within the narrative based on incoming and outgoing connections.

$$PT_q = G_q(PT) + I_q(PT_{assoc_pat}) + \frac{len(EG_{pt})}{len(EG_{pp})} \quad (10)$$

Auxiliary Patterns

Auxiliary patterns denote problems in the graph and sub-optimal or impractical nodes and connections within a graph. They are classified into *Nothing*, which are nodes that are not identified as part of a meso-pattern; and *Broken Link*, which are outgoing connections from a node that are not used or do not lead to any pattern.

Proof-of-Concept

TropeTwist can be used to represent different narrative structures and parts of games. To test and show TropeTwist's expressiveness, we chose to form three different narrative graphs representing different games shown in figure 1, top row: *Zelda: Ocarina of Time* (Zelda:OoT) [52], *Zelda: A Link to the Past* (Zelda:LttP) [53] - eastern palace, and *Super Mario Bros* (SMB) [54]. They represent different games from different genres (fig. 1.a and 1.b are adventure-dungeon games, and 1.c is a platformer), and represent different game's phases; in the case of fig. 1.a and 1.c, both represent the main structure of the game, while 1.b, represents a specific area and sequence of the game.

Figure 1.a represents a simplified overarching narrative structure from *Zelda: OoT*. The ocarina of time, given by Zelda to Link, is defined as a McGuffin (MCG) that, when collected by "young link," allows him to go forward in time to "adult link," the chosen one (NEO). This, in turn, enables explicit conflicts between hero and enemy characters, which represents the main loop of the game. The structure shows two factions, a set of heroes and the BAD.

Figure 1.b represents the structure and plot points from the eastern palace in *Zelda: LttP*. All palaces in *A Link to the Past* follow a very similar structure and sequence. The HERO's goal is to get the "Pendant of Courage" (MCG). However, the MCG derives from ENEMY and BAD, so the HERO must overcome them to achieve his goal. The structure shows a causal and linear narrative that could be used to identify elements that need to appear before others, similar to the work by Dormans and Bakkes [36].

Figure 1.c represents the overarching narrative structure of *SMB*. In *SMB*, the objective of Mario (HERO) is to rescue Peach (HERO) from Bowser (BAD). To do this, the player goes through a series of platform worlds that always end in a "Fake Bowser" (DRAKE). The player must continue until encountering the "Real Bowser" (BAD), which then would enable the player to get to their objective (MCG).

Evolving Narratives with Graph Grammars

We use the Constrained MAP-Elites [41], and adapt it to work with graph grammars, evolve production rules, and adapt the evolution towards a target similar to [39]. Constrained MAP-Elites adds feasible-infeasible two populations to each cell, effectively evolving sub-populations per cell. An individual's phenotype is a narrative graph, and its encoding genotype is the production rules of a graph grammar. A graph grammar is a context-free grammar whose productions add, remove, and modify nodes and edges of a graph. Our implementation uses the

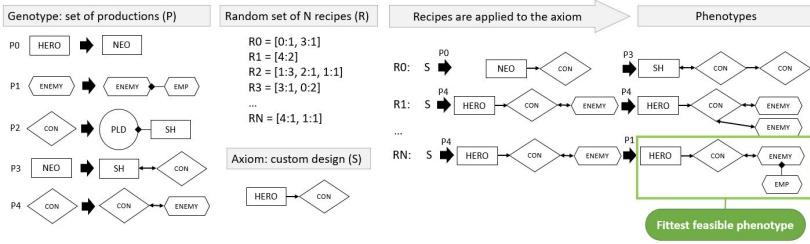


Figure 2: sample complete process from an individual's genotype to the phenotype.

tropes listed in Table 1 as nodes, and the three available connection types as edges (\rightarrow , \leftrightarrow , \diamond). Graph grammars do not apply rules sequentially; instead, every individual does a random sampling of the rules in their genotype to produce *recipes* to generate graphs. *Recipes* describe the rules' order and repetition, and their size is limited by the amount of production rules as minimum and the minimum plus five as maximum. *Recipes* do not have repetitions within them, i.e., if rule 1 is added at step 2, subsequent addition would simply add to the number of times that rule will be applied at step 2. Their size is limited by the number of production rules as minimum and up to five more design samples as maximum. Figure 2 shows a sample complete process from an individual's genotype (i.e., rules) to the phenotype (i.e., narrative graph).

Individuals move between the feasible and infeasible population depending on the feasibility constraint. NGs are deemed infeasible if the nodes are not fully connected or if there exists a conflict pattern with more than one self-conflict. Infeasible individuals are evaluated based on how close they are to be fully connected and not having any inadequate self-conflict. The fitness function assesses NGs that are deemed feasible based on their coherence (equation 3), which we use to assess how correct, coherent, and in general, syntactically correct the narrative graphs are. Coherence aims at maximizing an equally weighted sum between cohesion and consistency. Cohesion refers to the link between elements that hold together to form some group. In our implementation, it focuses on minimizing the number of auxiliary patterns by calculating the proportion of *Nothing* and *Broken Link* among all patterns in NG. A consistent NG should be regular and free of contradictions. Thus, we calculate *consistency* (eq. 2) as the collective quality of micro-patterns since they are the building blocks, and conflicts' goodness based on the number of fake conflicts. Thus, we aim at maximizing the quality of micro-patterns and minimizing contradictions created by meso-patterns.

$$f_{consistency} = \frac{\sum_{i=0}^{len(ng_{micro})} i_{qual}}{len(ng_{micropat})} - \frac{len(ng_{fakeConfP})}{len(ng_{confP})} \quad (11)$$

$$f_{coherence} = f_{consistency} + (1.0 - f_{cohesion}) \quad (12)$$

Furthermore, MAP-Elites uses behavioral dimensions in a grid shape to retain and foster diversity throughout generations. We use the following two dimensions to evaluate the diversity:

Step. Step (eq. 4) calculates the Levenshtein distance [55] between two narrative graphs, taking into consideration the number and type of nodes and connections. Step is normalized using step threshold $\theta = 11$ determined through a process of experimentation, which does not consider steps farther than θ , avoiding the generation of too dissimilar graphs.

$$D_{step} = \min(lev_{EG,RG}(|EG|, |RG|), \theta) \quad (13)$$

Interestingness (int). We aim at measuring the semantic quality of a narrative graph. A narrative graph can be syntactically correct and coherent yet lack a good semantic quality and do not evoke interest for designers or players. Therefore, we leverage **plot point**, **plot twist**, and **active plot device** patterns to measure the *interestingness* of the NGs. The nature of *interestingness* creates pressure on the fitness function since the incidence of the three meso-patterns could (if overused) “degenerate” the narrative; thus, decreasing its coherence. D_{int} is calculated as the weighted sum ($w_0 = 0.4$, $w_1 = 0.2$, $w_2 = 0.4$) of the normalized cumulative quality of **APDs**, **PPs**, and **PTs** within an NG (eq. 5).

$$D_{int} = w_0 \times \frac{APD_q}{\#APD} + w_1 \times \frac{\#PP_q}{\#PP} + w_2 \times \frac{PT_q}{\#PT} \quad (14)$$

Experiments

We conducted a series of experiments to evaluate and analyze how the system could evolve NGs into quality-diverse and valid narrative structures. We evolved the three manually constructed narrative graphs shown in figure 1, top row. They were used as root graphs and axioms in the EA, and we used *interestingness* and *step* as behavioral dimensions. We did 5 MAP-Elites runs per narrative graph, ran each for 500 generations, and set the initial population to 1000 randomly created individuals. The initial population is generated by randomly creating between

Table 2: Comparative results between root graphs and generated elites (shown in fig. 1)

Graph	Cohesion	Consistency	Coherence (fitness)	Interestingness
RG (fig 1.a1)	1.0	0.66	0.825	0.61
Elite (fig 1.a2)	1.0	0.76	0.875	0.73
RG (fig 1.b1)	1.0	0.75	0.87	0.38
Elite (fig 1.b2)	1.0	0.91	0.95	0.55
RG (fig 1.c1)	1.0	0.77	0.88	0.4
Elite (fig 1.c2)	1.0	0.85	0.92	0.52

two and five production rules. Each feasible and infeasible population per cell has 25 individuals. Each individual is limited to test 10 recipes regardless of the chromosome size. Offspring were produced either by selecting either the left-side or right-side of a random production rule and exchanging them or with a 50% mutation chance. If an offspring was generated by mutation, there was a 10% chance to add or remove a production rule and a 90% to modify in various ways existing production rules.

We calculated the *coverage*: how much of the constrained search space is explored (i.e., constrained by the behavioral dimensions); the avg. fitness and the avg. interestingness. All experiments had little variation regarding these metrics, and got in avg. 23.5% coverage (24.9%, 21.4%, and 24.2%, respectively), 0.79 fitness (0.76, 0.8, 0.8, respectively), and 0.37 interestingness (0.39, 0.37, 0.36, respectively). These results exemplify both the hard task of generating narrative graphs and exploring the possibility space, and the seemingly competing qualities of coherence (i.e., fitness) and interestingness.

Furthermore, in figure 1, bottom row, it is shown three different example elite narrative graphs, generated from their respective root graphs on the top row and with each individual evaluation shown in table 2. The root graphs have a cohesion of 1.0 since none of them have unused nodes or connections and have similar mid-high consistency values because of using generic nodes (e.g., HERO or ENEMY), repeating them, and low involvement in structures by characters. In the case of fig 1.a1, the **RevP** from HERO to SH creates some fake conflicts, which affect the consistency but also boost the interestingness value of the narrative graph. Both fig 1.b1 and 1.c1, are evaluated similarly with low interestingness; c1 involves a simplistic and linear structure, and b1, while in principle more complex, is also a relatively linear structure with no **PTs**.

Furthermore, all the exemplar elites have better *consistency*, *coherence*, and *interestingness* than the respective root graph. In figure 1.a2, the graph has been

reduced towards a bottleneck, **RevP** (HERO \rightarrow SH) is removed, and MCG is added as the objective for SH, which could point towards competition or cooperation to enable NEO. Such a change gives more *consistency* to the graph while seemingly reducing its *interestingness*, but this relation and the \diamond - connection between MCG and NEO increase its *interestingness*. In figure 1.b2, the narrative has more interaction between **Plot Devices**, and the BAD has a more active role. Particularly, the fact that now HERO \rightarrow MCG \diamond - BAD and MHQ \diamond - CHK \diamond - BAD could enable and force the HERO towards two main objectives before overcoming the boss, which is reflected in the higher *Interestingness*. Finally, in figure 1.c2, the narrative did not change much (only four steps away), yet the graph is seemingly better, and the narrative could be very different. The graph has broken the loop which connected DRAKE \diamond - BAD, and could point towards a side objective. Further, the connection between BAD and MCG has been reversed; thus, the HERO does not need to face the BAD to get the MCG, rather reaching the MCG will have as a consequence the emergence of the BAD. Finally, BAD is no longer connected to EMP and DRAKE; thus, BAD could be its own enemy faction, in this case, complexifying the narrative and creating more challenge.

Discussion and Limitations

The trope-graph representation in TropeTwist allows for a quick definition of narrative structures. They are, by design, ambiguous, do not encode temporal information besides causal chains, and are, to some extent, generic, which makes structures relatively simple to develop but more complex to interpret. These design decisions make the system encode less rich information than others, such as Scheherazade [35], but allow the structure to be interpreted in multiple ways. For instance, the generated graphs could equally describe different stories, and the interpretation given in this paper is just one of many. Thus, the system effectively shifts the complexity from the structure to the “interpreter.” While the generated structures could already serve as inspiration for users, an interpreter could provide alternative interpretations that could be guided by or learned from users, which is part of our future work.

Furthermore, the metrics proposed and developed here were used to tune and evaluate the graph outputs without humans in the loop. However, they do not stand in or replace human judgment. The metrics are estimated heuristics mainly based on the graph functionality and relation among patterns. Most of them are related to a “root graph,” which is a preliminary step for making TropeTwist interactive and have humans-in-the-loop. We aim to develop a mixed-initiative version of TropeTwist, where metrics depend on the designer’s creation. This would, in turn,

allow the designer to steer the MAP-Elites search, generating content adapted to them [56], and for MAP-Elites to assist designers with ideation proposing varied structures.

Conclusions and Future Work

In this paper, we have presented *TropeTwist*, a system that interconnects tropes and trope patterns to describe narrative structures. We demonstrated through three proof-of-concept structures the system's expressiveness to describe games with diverse genres and mechanics, and different game phases. Further, we illustrated how we could generate novel structures from the three proof-of-concept structures using MAP-Elites, improving them on our metrics.

Tropes could be seen as something to avoid when exploring creativity, mainly due to the possibility of showing unoriginal views by definition. However, a set of combined tropes, patterns, and structures could give rise to novel combinations that express the wanted structure. Similarly, identifying, visualizing, and defining the tropes and patterns and doing "twists" with them; thus, transforming something typical into atypical is the goal with TropeTwist.

The narrative structures show essential aspects of how the story will develop and lead, and important components such as events, conflicts, or roles. However, to further operationalize these structures, it is necessary other systems that make use of them, such as quest [24, 57] or plot [58] generators. Another interesting future work would be to explore the multi-faceted nature of games [3] and combine this type of system with generators that focus on other facets such as level design [29, 59] or game mechanics [40, 60].

Generating novel narrative structures resulted in interesting variations, but the system could not exploit all the advantages of MAP-Elites. Our results point towards difficulties exploring the space, possibly because *coherence* and *interestingness* are to some extent competing objectives. Therefore, we aim at extending TropeTwist towards a mixed-initiative co-creative system [61], and with that, evaluate with human participants. Given that our metrics are dependant on the designed graph; then, we could constantly adapt the content generation and have adaptive models, for instance, of interestingness, based on the user's creation similar to [29, 62].

References

- [1] J. Blow, “Game development: Harder than you think,” *Queue*, vol. 1, pp. 28–37, 02 2004. 309
- [2] A. Liapis, G. N. Yannakakis, and J. Togelius, “Computational game creativity,” in *Proceedings of the fifth International Conference on Computational Creativity (ICCC 2014)* (H. Toivonen, S. Colton, M. Cook, and D. Ventura, eds.), 2014. 309
- [3] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, “Orchestrating game generation,” *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019. 309, 323
- [4] E. Aarseth and P. Grabarczyk, “An ontological meta-model for game research,” in *DiGRA & #3918 - Proceedings of the 2018 DiGRA International Conference: The Game is the Message*, DiGRA, July 2018. 309
- [5] M. Vozaru, *The Game Situation: An object-based game analysis framework*. PhD Thesis, IT University of Copenhagen, 2022. Published: PhD Dissertation, IT University of Copenhagen. 309
- [6] M. Mateas and A. Stern, “Integrating plot, character and natural language processing in the interactive drama façade,” in *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE-03)*, vol. 2, 2003. 309
- [7] E. Aarseth, “A narrative theory of games,” in *Proceedings of the International Conference on the Foundations of Digital Games, FDG '12*, (New York, NY, USA), p. 129–133, ACM, 2012. 309
- [8] Q. Kybartas and R. Bidarra, “A survey on story generation techniques for authoring computational narratives,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, no. 3, pp. 239–253, 2016. 309, 311

- [9] S. Tolinsson, A. Flodhag, A. Alvarez, and J. Font, “To make sense of procedurally generated dungeons,” in *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play, CHI PLAY '20*, (New York, NY, USA), p. 384–387, Association for Computing Machinery, 2020. 309, 311
- [10] R. Barthes, “An introduction to the structural analysis of narrative,” *New Literary History*, vol. 6, no. 2, pp. 237–272, 1975. Translated by Lionel Duisit. 309, 310
- [11] N. Szilas, “Structural models for interactive drama,” in *Proceedings of the 2nd International Conference on Computational Semiotics for Games and New Media*, 2002. 309
- [12] V. Propp, L. Scott, and L. Wagner, *Morphology of the Folktale: Second Edition*. American Folklore Society Bibliographical and Special Series, University of Texas Press, 1975. 309, 310, 313
- [13] M. Thompson, J. Padget, and S. Battle, “Governing narrative events with tropes as institutional norms,” in *Artificial Life and Intelligent Agents* (P. R. Lewis, C. J. Headleand, S. Battle, and P. D. Ritsos, eds.), (Cham), pp. 133–137, Springer International Publishing, 2018. 309, 311
- [14] P. García-Sánchez, A. Velez-Estevez, J. Julián Merelo, and M. J. Cobo, “The simpsons did it: Exploring the film trope space and its large scale structure,” *PLOS ONE*, vol. 16, pp. 1–28, 03 2021. 309, 311, 312
- [15] C. Richmond and D. Schoentrup, “Tv tropes.” <https://tvtropes.org/pmwiki/pmwiki.php/Main/Tropes>, 2004. Accessed: 2020-03-24. 309, 311, 312, 315
- [16] J. Harris, “Periodic table of storytelling.” <https://jamesharris.design/periodic/>, 2016. Accessed: 2020-03-24. 309, 312, 313
- [17] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015. 310, 311
- [18] A. Baikadi and R. E. Cardona-Rivera, “Towards finding the fundamental unit of narrative: A proposal for the narreme,” in *Proceedings of the Third Workshop on Computational Models of Narrative*, pp. 42–44, European Language Resource Association, 2012. 310

- [19] E. Aarseth, “From hunt the wumpus to everquest: introduction to quest theory,” in *International Conference on Entertainment Computing*, pp. 496–506, Springer, 2005. 311
- [20] K. Yu, N. R. Sturtevant, and M. Guzdial, “What is a quest?,” in *Proceedings of the Intelligent Narrative Technologies Workshop at AIIDE*, 2020. 311
- [21] M. O. Riedl and R. M. Young, “Story planning as exploratory creativity: Techniques for expanding the narrative search space,” *New Generation Computing*, vol. 24, pp. 303–323, Sep 2006. 311
- [22] R. M. Young, S. G. Ware, B. A. Cassell, and J. Robertson, “Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives,” *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, vol. 37, no. 1-2, pp. 41–64, 2013. 311
- [23] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, “Toward supporting stories with procedurally generated game worlds,” in *2011 IEEE Conference on Computational Intelligence and Games, CIG 2011, Seoul, South Korea, August 31 - September 3, 2011* (S. Cho, S. M. Lucas, and P. Hingston, eds.), pp. 297–304, IEEE, 2011. 311
- [24] A. Alvarez, E. Grevillius, E. Olsson, and J. Font, “Questgram [qg]: Toward a mixed-initiative quest generation tool,” in *Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG’21*, (New York, NY, USA), Association for Computing Machinery, 2021. 311, 323
- [25] P. Tambwekar, M. Dhuliawala, L. J. Martin, A. Mehta, B. Harrison, and M. O. Riedl, “Controllable neural story plot generation via reward shaping,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019* (S. Kraus, ed.), pp. 5982–5988, ijcai.org, 2019. 311
- [26] J. van Stegeren and J. Myundefinedliwiec, “Fine-tuning gpt-2 on annotated rpg quests for npc dialogue generation,” in *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, FDG’21, (New York, NY, USA), Association for Computing Machinery, 2021. 311
- [27] P. Gervas, “Computational approaches to storytelling and creativity,” *AI Magazine*, vol. 30, p. 49, Jul. 2009. 311

- [28] M. P. Eladhari, P. L. Lopes, and G. N. Yannakakis, “Interweaving story coherence and player creativity through story-making games,” in *Interactive Storytelling : 7th International Conference on Interactive Digital Storytelling, ICIDS 2014, Singapore, Singapore, November 3-6, 2014, Proceedings*, no. 8832 in Lecture Notes in Computer Science, pp. 73–80, 2014. 311
- [29] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained map-elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 311, 323
- [30] M. Trenton, D. Szafron, J. Friesen, and C. Onuczko, “Quest patterns for story-based computer games,” in *Proceedings of the Sixth AAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE’10*, p. 204–209, AAAI Press, 2010. 311
- [31] J. Doran and I. Parberry, “A prototype quest generator based on a structural analysis of quests from four mmorpgs,” in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, PCGames ’11*, (New York, NY, USA), ACM, 2011. 311
- [32] M. Kreminski and N. Wardrip-Fruin, “Sketching a map of the storylets design space,” in *Interactive Storytelling* (R. Rouse, H. Koenitz, and M. Haahr, eds.), (Cham), pp. 160–164, Springer International Publishing, 2018. 311
- [33] J. Garbe, M. Kreminski, B. Samuel, N. Wardrip-Fruin, and M. Mateas, “Storyassembler: An engine for generating dynamic choice-driven narratives,” in *Proceedings of the 14th International Conference on the Foundations of Digital Games, FDG ’19*, (New York, NY, USA), Association for Computing Machinery, 2019. 311
- [34] I. Horswill, “Dear leader’s happy story time: A party game based on automated story generation,” in *Proceedings of the EXAG Workshop at AIIDE*, 2016. 311
- [35] D. Elson, “DramaBank: Annotating agency in narrative discourse,” in *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, (Istanbul, Turkey), pp. 2813–2819, European Language Resources Association (ELRA), May 2012. 311, 322
- [36] J. Dormans and S. Bakkes, “Generating missions and spaces for adaptable play experiences,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 3, pp. 216–228, 2011. 311, 318

- [37] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne, “Search-based procedural content generation: A taxonomy and survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 172–186, Sept. 2011. 311
- [38] M. Charity, A. Khalifa, and J. Togelius, “Baba is y’all: Collaborative mixed-initiative level design,” 2020. 311
- [39] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *IEEE Transactions on Games*, 2020. 311, 318
- [40] M. Charity, M. C. Green, A. Khalifa, and J. Togelius, “Mech-elites: Illuminating the mechanic space of gvg-ai,” in *International Conference on the Foundations of Digital Games*, FDG ’20, (New York, NY, USA), Association for Computing Machinery, 2020. 311, 323
- [41] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, ACM, 2018. 311, 318
- [42] J. Rowe, S. McQuiggan, J. Sabourin, D. Marcey, and J. Lester, “Storyeval: An empirical evaluation framework for narrative generation,” pp. 103–110, 01 2009. 311
- [43] C. Hargood, D. Millard, and M. Weal, “Measuring narrative cohesion: A five variables approach,” in *Narrative and Hypertext Workshop at Hypertext 11 (05/06/11)*, June 2011. 311
- [44] L. Castricato, S. Frazier, J. C. Balloch, and M. Riedl, “Fabula entropy indexing: Objective measures of story coherence,” *CoRR*, vol. abs/2104.07472, 2021. 311
- [45] R. Y. Pérez and O. Ortiz, “A model for evaluating interestingness in a computer-generated plot,” in *Proceedings of the fourth International Conference on Computational Creativity (ICCC 2013)*, 2013. 311
- [46] N. Szilas, S. Estupiñán, and U. Richle, “Qualifying and quantifying interestingness in dramatic situations,” in *Interactive Storytelling* (F. Nack and A. S. Gordon, eds.), (Cham), pp. 336–347, Springer International Publishing, 2016. 312

- [47] M. Kreminski, M. Dickinson, and N. Wardrip-Fruin, “Felt: A simple story sifter,” in *Interactive Storytelling* (R. E. Cardona-Rivera, A. Sullivan, and R. M. Young, eds.), (Cham), pp. 267–281, Springer International Publishing, 2019. 312
- [48] P. Lankoski, “Models for story consistency and interestingness in single-player rpgs,” in *Proceedings of International Conference on Making Sense of Converging Media*, AcademicMindTrek ’13, (New York, NY, USA), p. 246–253, Association for Computing Machinery, 2013. 312
- [49] M. Sharma, S. Ontañón, C. R. Strong, M. Mehta, and A. Ram, “Towards player preference modeling for drama management in interactive stories,” in *Proceedings of the Twentieth International Florida Artificial Intelligence Research Society Conference, May 7-9, 2007, Key West, Florida, USA* (D. Wilson and G. Sutcliffe, eds.), pp. 571–576, AAAI Press, 2007. 312
- [50] B. Li and M. O. Riedl, “An offline planning approach to game plotline adaptation,” in *Proc. AIIDE’10*, 2010. 312
- [51] A. J. Greimas, *Structural Semantics: An Attempt at A Method*. University of Nebraska Press, 1984. 313
- [52] Nintendo R&D4, “*The Legend of Zelda: Ocarina of Time*.” Game [N64], 1998. Nintendo, Kyoto, Japan. Last played December 2011. 318
- [53] Nintendo R&D4, “*The Legend of Zelda: A Link to the Past*.” Game [SNES], 1991. Nintendo, Kyoto, Japan. Last played December 2011. 318
- [54] Nintendo R&D1 and Intelligent Systems, “*Super Mario Bros.*” Game [SNES], September 1985. Nintendo, Kyoto, Japan. Last played August 2020. 318
- [55] V. I. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions and Reversals,” *Soviet Physics Doklady*, vol. 10, p. 707, Feb. 1966. 320
- [56] A. Alvarez, J. Font, S. Dahlsgog, and J. Togelius, “Assessing the Effects of Interacting with MAP-Elites,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, pp. 124–131, AAAI, Oct. 2021. 323
- [57] P. Ammanabrolu, W. Broniec, A. Mueller, J. Paul, and M. Riedl, “Toward automated quest generation in text-adventure games,” in *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*, (Tokyo, Japan), pp. 1–12, Association for Computational Linguistics, 29 Oct.–3 Nov. 2019. 323

- [58] P. Ammanabrolu, E. Tien, W. Cheung, Z. Luo, W. Ma, L. J. Martin, and M. O. Riedl, "Story realization: Expanding plot events into sentences," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7375–7382, AAAI Press, 2020. 323
- [59] A. Sarkar and S. Cooper, "Dungeon and platformer level blending and generation using conditional vaes," in *Proceedings of the IEEE Conference on Games (CoG)*, 2021. 323
- [60] M. C. Green, A. Khalifa, P. Bontrager, R. Canaan, and J. Togelius, "Game mechanic alignment theory and discovery," 2021. 323
- [61] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity," in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 323
- [62] P. Migkotzidis and A. Liapis, "Susketech: Surrogate models of gameplay as a design assistant," *IEEE Transactions on Games*, pp. 1–1, 2021. 323

PAPER XI - STORY DESIGNER: TOWARDS A MIXED-INITIATIVE TOOL TO CREATE NARRATIVE STRUCTURES

Alberto Alvarez, Jose Font, and Julian Togelius

ABSTRACT

Narratives are a predominant part of games, and their design poses challenges when identifying, encoding, interpreting, evaluating, and generating them. One way to address this would be to approach narrative design in a more abstract layer, such as narrative structures. This paper presents Story Designer, a mixed-initiative co-creative narrative structure tool built on top of the Evolutionary Dungeon Designer (EDD) that uses tropes, narrative conventions found across many media types, to design these structures. Story Designer uses tropes as building blocks for narrative designers to compose complete narrative structures by interconnecting them in graph structures called narrative graphs. Our mixed-initiative approach lets designers manually create their narrative graphs and feeds an underlying evolutionary algorithm with those, creating quality-diverse suggestions using MAP-Elites. Suggestions are visually represented for designers to compare and evaluate and can then be incorporated into the design for further manual editions. At the same time, we use the levels designed within EDD as constraints for the narrative structure, intertwining both level design and narrative. We evaluate the impact of these constraints and the system's adaptability and expressiveness, resulting in a potential tool to create narrative structures combining level design aspects with narrative.

PUBLISHED IN

Proceedings of the Foundations of Digital Games (FDG), ACM, 2022.

STORY DESIGNER: TOWARDS A MIXED-INITIATIVE TOOL TO CREATE NARRATIVE STRUCTURES

Introduction

Games are multifaceted content that intertwines gameplay, mechanics, audio, level, graphics, and narrative facets [1]. Narrative has been linked as a key facet to connect different components in games such as level design [2], and to create meaningful interactions with depth and context [3,4]. Thus, narrative in games has been a focus of study [5–7] and its generation has been approached in different ways and with different techniques [8–12].

Patterns have been a common approach to narrative and other facets. The focus then has been on extracting common narrative aspects to ease the identification, encoding, and generation of narrative in different forms [13–18]. However, it remains a challenge to define certain narrative aspects more aligned with the structure and overarching goals of the game given what type of content is generated, as well as using these to design and compare among games. One approach would be to change the abstraction level at which the narrative is designed. Instead of focusing on the details, quests, or plot, one could focus on the structure. Narrative structures can be used to describe how a story is to be developed, as argued by Barthes [19], and to create an abstract representation that reveals common structures among them, such as Propp's 31 "narremes" [20]. One approach to generate narrative structures is TropeTwist [14], which uses tropes, narrative conventions found across many media types [21, 22], as patterns to design these structures.

This paper presents Story Designer, a mixed-initiative co-creative (MI-CC) narrative structure tool built on top of the Evolutionary Dungeon Designer (EDD) using the TropeTwist system. Story Designer uses tropes as building blocks for designers to compose complete narrative structures by interconnecting them in graph structures called narrative graphs. Story Designer lets designers create narrative graphs and

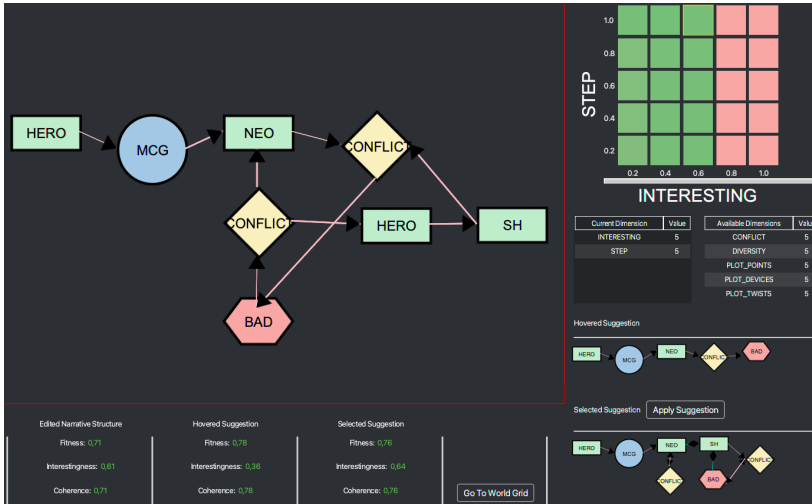


Figure 1: The Story Designer screen in the Evolutionary Dungeon Designer. In the center, there is the main narrative graph being edited by the designer. To the right, the suggestion grid using the Interactive Constrained MAP-Elites (IC MAP-Elites), the possible dimensions to be used, and two inspected suggestions. At the bottom, Story Designer presents some extra information regarding fitness, interestingness, and coherence, for the designer's convenience.

assist them with a suggestion grid that uses the Interactive Constrained MAP-Elites (IC MAP-Elites) [23]. By having an MI-CC system to design narrative structures, designers could ideate and prototype their structures while the system adapts and suggest novel narratives, making use of patterns, optimizing coherence, and situating the narrative structures along dimensions of interest for designers. At the same time, IC MAP-Elites can take advantage and use the designer's structure as a proxy to evaluate subjective characteristics such as interestingness, which has been the subject of several studies [24–26].

As Story Designer is implemented in EDD and based on the link between level design and narrative, we make use of the designed dungeon to create constraints over the narrative generation, effectively intertwining both facets. We assess Story Designer with four controlled and simulated experiments, three premade structures of different games, and one step by step design that showcases the possibilities within the system. All experiments were tested with and without level design constraints and using a pair of dimensions and all dimensions during the search. Our results indicate that IC MAP-Elites have consistency and stability in generating content and that delimiting the search space with additional level constraints, while limiting the diversity and generation of complex structures, guides better the search.

Related Work

There is by now a large body of research on procedurally generating various types of game content [27]. While the literature on PCG in general is far too voluminous to survey here, it should be noted that PCG methods of different kinds have been developed for a wide variety of content, not just game levels. Narrative, quests, and plots have been generated using different approaches such as planning [28], grammars [13,29], machine learning [11], and patterns [14,15,18]. Further, several approaches have been proposed to generate multiple facets of games, in particular level geometry together with rules, music, lighting, sound etc [1,10,30–35]. More relevantly to the current project, several papers have proposed ways of co-generating narrative and levels [3,29,36,37].

In tandem with research on automatically and autonomously generating game content and narrative, there has been a considerable amount of work “mixed-initiative” systems, which allow a human designer to co-create content with algorithms. In the domain of level generation for games, a number of systems have been developed that allow a human to receive suggestions, feedback, or constraints from an AI systems. These include systems for co-creating platform game levels [38], strategy maps [39], and certain aspects of narrative [40,41].

The core algorithm employed in the current paper is MAP-Elites, a quality-diversity algorithm that seeks to illuminate a space of possible problem solutions [42]. While essentially a type of evolutionary algorithms, MAP-Elites, like other quality-diversity algorithms, do not seek to find a single best solution but rather a set of solutions that vary along certain specified measures. The measures define a grid, where each cell is the best solution that has been found within certain values of the measures. These measures can be defined in many ways; for game levels, they might include the density of a level, its difficulty for a particular type of agent, its symmetry etc. MAP-Elites has been used in multiple recent AI-based game design systems [43–47].

Story Designer

Story Designer is a new system integrated in EDD, which presents a visual interface for mixed-initiative narrative structure generation. It makes extensive use of the TropeTwist system as foundation to build narrative graphs and assess them by identifying trope patterns. The user manually designs a story structure by adding and interconnecting nodes in a graph, which seeds an evolutionary algorithm (EA) that generates story structure suggestions that can be incorporated into the user’s design. This continuous co-creative design process implements the Interactive

Constrained MAP-Elites (IC MAP-Elites) approach presented in [44], providing quality-diverse suggestions across several feature-dimensions.

Story Designer is interconnected with the level design facet in EDD. This means that the narrative graphs that can be developed and that can be generated and suggested are constrained by the content that exists in the levels. For instance, if the designer adds two NPCs besides the Hero, then the system could at most, use three character nodes to represent them, or if the designer adds a boss enemy and a quest item, this would mean that the boss enemy could be represented as one of the villain nodes (e.g., Enemy, Big Bad, or Dragon) and the quest item as a possible Plot Device.

TropeTwist

TropeTwist [14] is a system that uses tropes [21, 22, 48, 49], narrative conventions easily recognizable by the audience, as patterns that combine to compose narrative structures. These structures define generic aspects of a story, leading to the identification of events, roles, and other relevant narrative elements arranged as nodes in an interconnected narrative graph. By having all this elements in a graph, entire narratives are encoded using graph grammars, to then procedurally generate novel narrative variations by means of a MAP-Elites algorithm that considers several narrative evaluation metrics, such as interestingness, coherence, and cohesion.

Nodes in a narrative graph represent tropes. Interconnected tropes create other composite tropes and patterns, that can be identified as subgraphs of a complete narrative graph. These patterns can be **micro-patterns** encapsulating a single trope node, **meso-patterns**, often composed by more than one micro-pattern with a specific meaning, and **auxiliary patterns**, identifying structural gaps in the graph. For a detailed definition of all tropes and patterns, please refer to [14]. Here we present a comprehensive summary:

- Micro-patterns are the fundamental narrative unit in the system, encapsulating tropes in building blocks to create complex narrative structures. These are classified into structure patterns (SP), that articulate the story elements (i.e. Conflict), character patterns (CP) (i.e. heroes and villains), and plot device patterns (PDP), that move the story forwards towards a particular goal (i.e. the MacGuffin).
- Meso-patterns may emerge from the combination of micro-patterns and other meso-patterns, denoting spatial, semantic, and usability relationship within the

narrative graph.

1. The *Conflict Pattern (ConfP)* ties a conflict node to two other nodes representing both parties in a conflict (i.e. HERO → CONFLICT → EMP, a hero is in conflict with the Empire).
 2. The *Derivative Pattern (DerP)* defines relations of entailment between other nodes, called derivatives. These derivatives acquire a local and temporal order, and a causal relationship. I.e the former conflict connected to EMP \diamond — DRA \diamond — NEO, means that the hero engages the Empire, which entails both a conflict with the Dragon (*DRA*) and the appearance of the Chosen One (*NEO*).
 3. The *Reveal Pattern (RevP)* connects two independent CPs as one, meaning that character A was, in fact, always character B, and vice-versa. This pattern turns all existing conflicts between them into *fake* conflicts.
 4. The *Active Plot Device Pattern (APD)* triggers a PDP and integrates it in the the narrative, since PDP are passively described and lack any start condition.
 5. *Plot Points (PP)* are key discrete narrative events. The derivatives within a *DerP*, the source of a reveal pattern, as well as active plot devices are considered plot points.
 6. A *Plot Twist (PT)* identifies those plot points that could change the natural flow of the narrative. I.e. in EMP \diamond — DRA \diamond — NEO, NEO is identified as a plot twist since its nature (heroic) is opposed to that of the first node EMP (villainous), which alters the natural order of the connecting derivative pattern.
- Auxiliary patterns spot and encapsulate those areas in the graph that don't contain meaningful narrative information. *Nothing* highlights nodes that are not identified or part of any meso-pattern; whereas *Broken Link* marks outgoing connections from any node that do not lead to any pattern.

Workflow

Story Designer is integrated in EDD as a separate view (Figure 1) that can be accessed anytime from the dungeon editor. The use starts with a minimal sample narrative graph HERO → CONFLICT → ENEMY in the manual edition pane (center). This graph can be extended by adding nodes from the node context menu that pops up with a right-click on an empty space. Node are arranged by type for

the sake of clarity, and an option to automatically re-arrange the graph is shown at the end of the menu. Right-clicking on an existing node border will pop up the edge context menu, that allows the user to create a new connection or to delete the selected node. Existing connections are deleted by left-clicking on them.

In a way similar to EDD's room editor [44], as the user edits the narrative graph manually, this graph is fed into the underlying evolutionary algorithm that procedurally generates on the fly alternative narrative graphs in the suggestions pane (right). The top-right corner shows the feature-dimension matrix, whose cells are colored depending on the fitness of the fittest elite contained in it, ranging from dark red (no elite yet), to dark green (optimal fitness). The elite in the selected cell of the matrix is displayed in the bottom-right corner. Hovering the mouse above a cell displays its elite's graph above the selected one, which allows the user to compare several graphs at a glance.

Evolving narrative structures with Graph Grammars

The underlying evolutionary algorithm in Story Designer is an adapted version of IC MAP-Elites [23] to evolve grammars. In Story Designer, an individual's phenotype is a narrative graph, and its encoding genotype is a graph grammar. A graph grammar is a context-free grammar whose productions add, remove, and modify nodes and edges to a graph.

An individual's genotype is the production rules of the grammar, which are deterministic i.e., a production rule (or pattern) only matches one production. Given that the graph grammar does not need to be applied sequentially until terminal nodes are reached, every individual does a random sampling of the rules in their genotype to produce *recipes*. *Recipes* simply describe the order of rules to be applied (sequentially) and the amount of times they will be applied. *Recipes* do not have repetitions within them i.e., if rule 1 is added at step 2, subsequent addition would simply add to the amount of times that rule will be applied at step 2. The internal parts of the EA works exactly as in TropeTwist, but now it is extended to use all the capabilities of IC MAP-Elites, namely, the continuous adaptive evolution aspect [14, 23].

Moreover, thanks to continuous evolution, the EA constantly incorporates the most recent version of the user's design to the population of individuals in the corresponding cell of the feature-dimension matrix. The designer can switch between dimensions at any given time, as well as changing their granularity. IC MAP-Elites manages two different populations within each cell: a feasible and an infeasible one. Individuals move across cells when their dimension values change,

or between the feasible and infeasible population according to their fulfillment of the feasibility constraint. Narrative graphs are deemed infeasible if they are not fully connected (i.e., all nodes can be reached from an arbitrary starting point), if there exist a conflict pattern within the graph with more than one self-conflict, or if level design constraints are enabled, the narrative graph violates any level design constraint. Infeasible individuals are evaluated (equation 1 in a weighted sum ($w_0 = 0.5$, $w_1 = 0.25$, $w_2 = 0.25$) based on how close they are to being fully connected and to remove inadequate self-conflicts, while trying to maximize the graph's cohesion.

$$f(infeasible) = w_0 \times f(cohesion) + w_1 \times \frac{\#!reachable_{V(NG)}}{|V(NG)|} + w_2 \times \frac{\#!valid_{NG(selfconf)}}{|V(NG)|} \quad (1)$$

Generated narrative graphs that are deemed feasible, are evaluated on their coherence (equation 3), which is used to assess how correct, coherent, and in general, syntactically correct the narrative graphs are. Coherence aims at maximizing an equally weighted sum between cohesion and consistency (eq. 2). Cohesion refers to the link between elements that hold together to form some group, which in Story Designer means the minimization of auxiliary patterns (*Nothing* and *Broken Link*) within the narrative graph. Consistency means that the narrative graph should be regular and free of contradictions, aiming at maximizing the quality of micro-patterns and minimize contradictions created by meso-patterns (contradictions can affect the consistency fitness up to $w_0 = 0.3$). For a more detailed explanation of how the EA works internally and the different fitness functions, we refer to the TropeTwist paper [14].

$$f_{consistency} = \frac{\sum_{i=0}^{len(ng_{micro})} i_{qual}}{len(ng_{micropat})} - w_0 \times \frac{len(ng_{fakeConfP})}{len(ng_{confP})} \quad (2)$$

$$f(coherence) = f(consistency) + (1.0 - f(cohesion)) \quad (3)$$

Behavior Dimensions for Graph Grammars

Dimensions in MAP-Elites are a key component for the search space to be delimited, and are identified as those aspects of the individuals that can be calculated in the behavioral space, and that are independent of the fitness calculation. In Story Designer, the designer is able to pick two dimensions at a time to facilitate

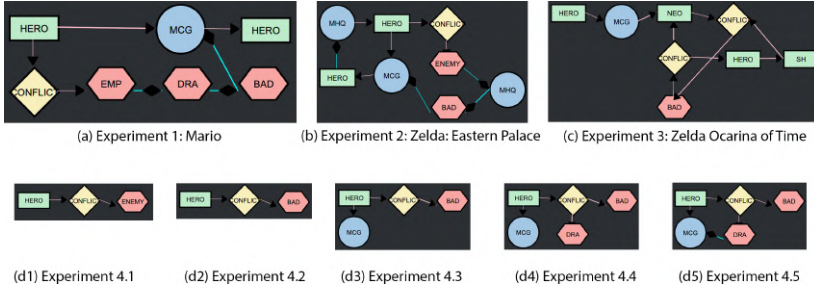


Figure 2: Narrative graphs used for the experiments, constructed and designed in Story Designer. When experiment 4 is discussed, the narrative graph referred is Experiment 4.5 as that is the design’s final step.

visualization, and all dimensions, when needed, are limited using a threshold $\delta = 5$. TropeTwist implemented *Interestingness* and *Step* as behavior dimensions when using MAP-Elites to generate novel narrative graphs. *Step* is calculated as the Levenshtein distance between two narrative graphs, taking into account the amount of nodes and connections and their type (eq. 4). *Interestingness* make use of the APDs, Plot Points, and Plot Twists that are present in a narrative graph to assess an approximate semantic evaluation since those represent some type of variation in the graph (eq. 5). Given *Interestingness* is a highly subjective measurement, we rely on those patterns since they calculate their quality based on the current narrative graph and the one being edited by the designer.

$$D_{step} = \frac{lev_{a,b}(|a|, |b|)}{\theta} \tag{4}$$

$$D_{int} = w_0 \times \frac{APD_q}{\#APD} + w_1 \times \frac{\#PP_q}{\#PP} + w_2 \times \frac{PT_q}{\#PT} ng \tag{5}$$

Furthermore, we have extended TropeTwist with five more dimensions relevant to the narrative structure design process, to give more choice to designers and experiment with other dimensions in the search space:

Diversity (div). Diversity measures the variety of [base] trope types within a narrative structure. Currently, there exist four base trope types, *Hero* (h), *Villain* (v), *Structure* (s), *Plot Devices* (pd). Diversity takes into account the tropes that also extend these base tropes. Thus, D_{div} collects all tropes within a graph, and increase a counter for each of the base trope type ($NG_{base} = h, v, s, pd \in NG$), normalized by the max amount of base trope types depicted in Eq. 6:

Table 1: Level constraints used per Experiment. Constraints were chosen based on the maximum amount of elements needed to design the narrative structure in the system.

Constraining elements	Exp. 1	Exp. 2	Exp. 3	Exp. 4
Heroes	2	2	4	2
Enemies	2	2	1	2
Quest Items	2	3	1	2

$$D_{div} = \frac{NG_{base}}{\#Trope_{base}} \quad (6)$$

Conflict (confs). Since we already calculate all patterns within a narrative graph, conflict simply calculates the amount of *explicit* conflict patterns ($\#NG_c = C_{exp} \in allPatterns$) that exist within a narrative graph normalized by a conflict threshold $\omega = 5$. We use ω to avoid stimulating the generation of narrative graphs with a massive amount of conflicts, which could create noise in the evolution and focus on the conflicts rather than other tropes and patterns. D_{confs} is then calculated as $\frac{\#NG_c}{\omega}$.

Plot points (pp). Plot points measures the amount of plot points within a narrative graph ($\#NG_{pp} = pp \in allPatterns$) and normalize it by δ . Given that plot points are dynamically assessed based on other patterns and combination of tropes, we limit the dimension with δ to avoid losing coherence in favor of generating more plot points. D_{pp} is calculated as $\frac{\#NG_{pp}}{\delta}$.

Plot Twist (pt). Plot twist measures the amount of plot twists within a narrative graph ($\#NG_{pt} = pt \in allPatterns$) and normalize it by δ . Plot twists relate to special situations within a narrative graph where the somewhat abrupt change in the tropes or combination of tropes could alter the narrative and create a surprise effect. Therefore, we limit the dimension with δ to avoid “degenerating” narratives with too many twists. D_{pt} is calculated as $\frac{\#NG_{pt}}{\delta}$.

Plot devices (pd). Plot devices measure the amount of active plot devices within a narrative graph ($\#NG_{pd} = apd \in allPatterns$) and normalize it by δ . Plot devices create targets and goals within a narrative, and active plot devices operationalize these in the narrative graph associating them with multiple tropes; thus, similar to pt , we limit with δ to avoid “degenerating” the narrative. D_{pd} is calculated as $\frac{\#NG_{pd}}{\delta}$.

Table 2: Average values from the experiments using Interestingness and Step as dimensions. Values in bold represent the best values in the specific experiment between using or not using level constraints. ★ represents the best values across experiments within their specific condition (using or not level constraints).

Experiment	With level constraint				No level constraint			
	Avg. coverage	Avg. Uniques	Avg. fitness	Avg. Int	Avg. coverage	Avg. Uniques	Avg. fitness	Avg. Int
Experiment 1	20.7%±2.8	132±37.6	0.82±0.01★	0.35±0.02	24.2%±2.2	209.6±27.8	0.8±0.01★	0.36±0.03
Experiment 2	19.1%±0.9	161.8±20.2	0.82±0.02	0.37±0.02	21.4%±2	290.6±27.2★	0.8±0.01★	0.37±0.01
Experiment 3	20.6%±3.9	173.6±55.5★	0.78±0.02	0.39±0.03★	24.9%±1.9	244.2±15.9	0.76±0.02	0.39±0.02★
Experiment 4	22.5%±4.1★	156.4±53.6	0.8±0.04	0.34±0.02	27.8%±2.2★	267.8±30.3	0.8±0.02	0.33±0.03
Experiment 4.1	6.5%±3	49±31.9	0.67±0.09	0.07±0.02	9.8%±2.6	83.8±40.3	0.72±0.04	0.13±0.05
Experiment 4.2	4.3%±2.7	23.8±16.7	0.68±0.07	0.12±0.08	6.8%±2.6	52±28.2	0.75±0.04	0.15±0.05
Experiment 4.3	10.9%±3.7	62.4±26.9	0.75±0.03	0.27±0.04	15.4%±5.4	124±52.9	0.79±0.03	0.32±0.05
Experiment 4.4	15%±2.1	94.8±19.5	0.79±0.04	0.29±0.06	14.9%±3.8	116.2±41	0.85±0.03	0.35±0.05
Experiment 4.5	14.1%±4.6	82.8±37.9	0.83±0.04	0.33±0.03	14.6%±3.2	93.2±36.9	0.85±0.02	0.32±0.04

Table 3: Experiments using all possible dimensions (7 dimensions) as behavioral dimensions in the MAP-Elites search. Coverage relates to the pair Interestingness-Step for comparison with Study 1. Values in bold represent the best values in the specific experiment between using or not using level constraints. ★ represents the best values across experiments within their specific condition (using or not level constraints).

Experiment	With level constraint				No level constraint			
	Avg. coverage	Avg. Uniques	Avg. fitness	Avg. Int	Avg. coverage	Avg. Uniques	Avg. fitness	Avg. Int
Experiment 1	33.2%±1.5	496.2±132.3	0.75±0.05★	0.32±0.03	36.7%±1.8	1257.8±165.3	0.75±0.02★	0.33±0.02
Experiment 2	29%±3.4	760.8±100.9★	0.72±0.01	0.32±0.02	27.8%±2	1221.8±229.3	0.72±0.01	0.33±0.01
Experiment 3	33.6%±1.5	658.2±48.2	0.71±0.01	0.34±0.01★	37.2%±1.7	1357.4±48★	0.68±0.01	0.36±0.01★
Experiment 4	35.2%±2★	690.4±325.4	0.71±0.03	0.29±0.02	38.3%±1.8★	1314.6±181.1	0.73±0.01	0.28±0.02
Experiment 4.1	6.5%±1.4	56.8±8.5	0.67±0.06	0.04±0.02	10.2%±1.5	118.6±43.5	0.65±0.06	0.07±0.03
Experiment 4.2	8.8%±1.9	74.8±19.9	0.59±0.04	0.07±0.02	9.9%±2.5	153.6±21.9	0.56±0.11	0.09±0.05
Experiment 4.3	16.4%±2.1	123.8±17.7	0.62±0.04	0.19±0.02	23%±1.5	377.2±81.2	0.61±0.07	0.25±0.02
Experiment 4.4	17.8%±3.5	155.8±25.5	0.65±0.06	0.21±0.05	25.8%±1.5	286.8±209.3	0.65±0.05	0.26±0.01
Experiment 4.5	23%±3.2	232.8±69.8	0.67±0.02	0.28±0.03	30.8%±3	891.8±202.3	0.67±0.04	0.3±0.01

Experiment Setup

We ran a set of experiments using different narrative graphs as starting points and level constraints to evaluate Story Designer, the use of TropeTwist with IC MAP-Elites, and its adaptability. Our goal is to analyze how IC MAP-Elites can adapt to the designer's narrative graph and how that affects the search space. At the same time, we explore how we can connect level and narrative and the effect using level design constraints have on the development and generation of narrative structures.

We ran each experiment 5 times, set the initial population to 1000 randomly created grammars, and each individual is limited to test 5 recipes regardless of the chromosome size. Offspring were produced either by selecting either the left-side or right-side of a random production rule and exchanging them or with a 50% mutation chance. If an offspring was generated by mutation, there was a 10% chance to add or remove a production rule and a 90% to modify existing production rules in various ways (e.g., removing, adding, or changing part the rule). When using level constraints, these were enforced as feasibility constraints, effectively setting individuals as infeasible if violating any constraint.

For each experiment, we used the dimension pairs **Interestingness-Step** and all dimensions for IC MAP-Elites to compare different space constraints that would be employed in an MI-CC system and a full space search across dimensions similar to Alvarez et al.'s work [23]. Experiments 1-3 consist of reusing the three proof-of-concept narrative graphs used by Alvarez and Font [14], and testing them in Story Designer; assessing the impact of level constraints and how the space is explored in comparison with previous results. Experiment 4 assesses the same as experiments 1-3 but focuses on evaluating the system's adaptability and how IC MAP-Elites respond to design changes, which enables different patterns to arise in the narrative structure. Experiment 4 is evaluated as a whole and step by step in the design process (5 core steps). For experiments 1-3, we ran each for 500 generations when using a pair of dimensions and for 250 when using all dimensions. For experiment 4, each step in the design is done after 50 generations; thus, we recorded data every 50 generations.

Metrics

All our experiments are evaluated and analyzed following the same procedure and metrics, focusing on the novel generated individuals and their average across the 5 runs. In particular, we focus on the *average coverage*, *average unique individuals*, *average fitness*, and *average interestingness*. *Average coverage* is the

cumulative coverage of the search space after a set of generations focused on the Step-Interestingness dimension pair. *Average uniques* is a simple count of how many novel individuals were created throughout the experiments. *Average fitness* calculates the average individual fitness in the search throughout all generations. Finally, *average interestingness* calculates the average individual interestingness in the search throughout all generations.

Narrative Graphs for Experiments

Figure 2 and table 1 show the target narrative graphs used in each experiment and their level design constraints, respectively. Experiments 1-3 use the proof-of-concept narrative graphs presented by Alvarez and Font [14] and experiment 4 uses a handmade narrative graph, exemplar of what a designer could create in Story Designer.

Experiment 1 represents the overarching narrative structure of Super Mario Bros. (SMB) [50]. Mario (HERO) has as objective to rescue Princess Peach (HERO) from Bowser (BAD), who keeps Peach as a prisoner until Mario beats it, creating a derivative and conditional relation between Bowser and Peach. Before reaching Bowser, Mario must face "fake Bowsers" (DRA). Experiment 2 represents the structure from the eastern palace in *Zelda: A Link to the Past* (Zelda:LtP) [51]. Link (HERO) has as a goal the "Pendant of Courage" (MCG), but in order to collect it, Link must face ENEMY and BAD since there is a derivative pattern connecting them to (MCG). All palaces in *A Link to the Past* follow a very similar structure and sequence. Experiment 3 represents a simplified overarching structure from *Zelda: Ocarina of Time* (Zelda:OoT) [52]. Young Link (HERO) has as a goal to collect/receive the Ocarina of Time (MCG), which enables the appearance of Adult Link (NEO). Achieving this goal creates conflicts between several heroes, Link and Zelda - Sheik (NEO, SH), and Gannondorf (BAD).

Experiment 4 was designed with the capabilities of Story Designer in mind; step by step as a designer would create the narrative structure. First, it starts with the default structure; a HERO has a CONFLICT with an ENEMY. Subsequently, the structure is changed to fine-tune the ENEMY to BAD and create a goal (MCG) for the HERO. Another enemy (DRA) is added, creating a side conflict for the HERO (i.e., BAD is by definition the "final boss"). Finally, the DRA is connected with the MCG with an entail connection, effectively making the DRA part of the game's main loop.

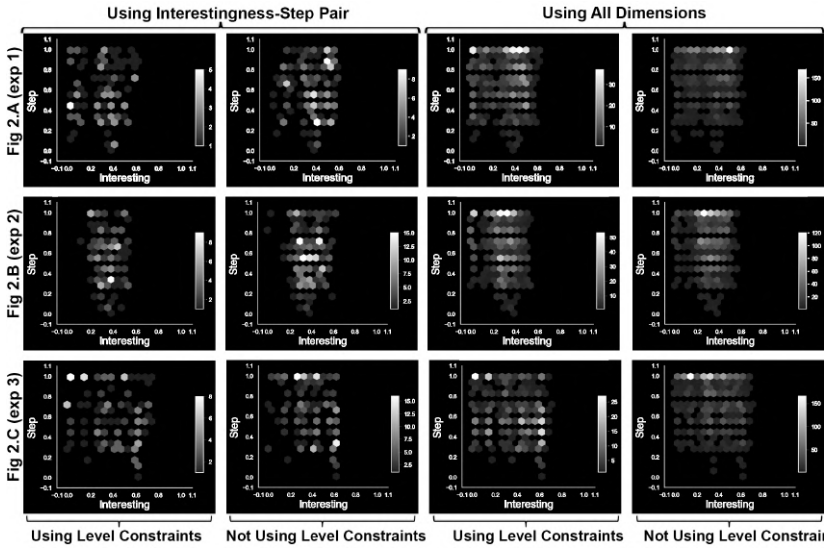


Figure 3: Rows show experiment 1-3, respectively. Each narrative graph can be seen in figure 2.a-c, respectively. The first two columns are using interestingness-step as dimensions, and the other columns are using all dimensions in the search.

Result and Analysis

In tables 2 and 3, we present the results based on our metrics for the four experiments. Table 2 uses interestingness and step as dimension for MAP-Elites, while Table 3 uses all dimension during search. To complement the analysis, figure 3 shows an exemplar expressive range analysis (ERA) for experiments 1-3 in the different configurations, and figure 4 shows an exemplar ERA for experiment 4 and an exemplar Temporal ERA (TERA) of the design steps. An ERA is an evaluation method to explore and visualize the expressiveness of an algorithm in content space [53]. TERA is an extension of ERA that allows the inspection and analysis of changes in expressiveness over a defined period, which, when used in a non-aggregated fashion, as in experiment 4, shows the delta maps of the search [43].

Analyzing and comparing the experiments show similar and consistent results across experiments regardless of using level design constraints or not, and using all dimensions or just a pair. Experiments 1-3 present consistent and stable results, similar among them in all metrics except coverage, which is more influenced by the specific graph and what type of information it provides, such as patterns, nodes, and connections.



Figure 4: Expressive Range Analysis (ERA) and Temporal ERA (TERA) for Experiment 4. The first row is the general results for experiment 4 (ERA), while the subsequent ones are the individual results per step taken in the narrative structure design (figure 2.d) (TERA). The first two columns are using interestingness-step as dimensions, and the other columns are using all dimensions in the search.

Experiment 4 shows MAP-Elites adaptability throughout the different design steps, especially visible in figure 4. In the first two steps (4.1 and 4.2), MAP-Elites exploration is limited due to the narrative graph’s simplicity. This is expected as the default narrative graph (HERO → CONFLICT → ENEMY) and the fine-tuned (i.e., ENEMY changed for BAD) has an interestingness score of 0 and, when used

as a target, hinders the exploration with or without level constraints. However, as the design progresses, MAP-Elites adapt. Minimal input into the graph (experiment 4.3, onwards) improves the search and interestingness following the design's trend. IC MAP-Elites maintain properties such as adaptability and stability shown before for level design generation, making it adequate for the evolution of grammars and narrative structs as well.

Experiment 4 also shows a concrete example of how the narrative graph would be used and designed by designers to change components in a game and enable different narrative structures. When put in context with the graphs for experiments 1-3, show relative diversity and expressiveness in the system. Experiment 4 and its steps show as well how the structure can relate to different "in-game" and level components, how, through the structure, designers can design main and side objectives, and how these could be approached. For instance, the DRA as a side conflict in the game and then incorporated as a main part of the game since to get the MCG, the HERO needs to face the DRA. That could then be used, in practice, to change, constrain, or adapt quests or part of the level design to be aligned with the structure.

Furthermore, both tables seem to show similar patterns when using level design constraints or not. Fitness and interestingness vary slightly (avg. +0.01 and -0.003, respectively), whereas coverage and unique individuals are worst (avg. 3.1%, 366.9, respectively).¹ The lower unique individuals are expected since the search space is more constrained; thus, individuals that would otherwise be feasible (i.e., fully connected graph and without inadequate self-conflicts) would become unfeasible with the level constraints. A similar analysis could be expected from the slightly higher or comparable fitness since the lesser the individuals that are generated, the lesser fitness' variance. However, this also shows a practical and possible way to intertwine and enforce inter-facet constraints, since when adding level design constraints to the narrative generation, due to the more delimited space, the search can be more guided and focused and still generate quality-diverse content [54, 55].

The results point towards IC MAP-Elites, due to its constrained features and adaptability properties, being agnostic to these inter-facet constraints, which allows and ease the incorporation of these constraints in the system without having a major impact on the development. The tradeoff is then clearly that the possibility of the system to search for more or more complex narrative structures when using constraints is reduced since they would most probably violate constraints.

When comparing the use of a pair of dimensions (interestingness and step) and

¹These values are a combination of both table 2, 3

all dimensions in the search regardless of having or not constraints, the difference is expected regarding coverage (avg. 11.2% more) and unique individuals (avg. 765.3 more) generation since MAP-Elites will be able to encounter and store elites in a bigger grid. However, the quality of the individuals is subpar in comparison with a pair of dimensions regarding fitness (avg. 0.08 more) and interestingness (avg. 0.04 more). This result is in line with [23], where their results, applied to level design, showed more coverage and individuals generated when using all dimensions but focusing on suboptimal parts of the space. Figures 3 and 4 show that the experiments explore similar spaces, sparser when using a pair of dimensions and denser when using all dimensions. When observing the heatmap intensity, the search focus is distributed across the search space when using a pair of dimensions, while when using all dimensions, the search is focused on high step levels.

Conclusions and Future Work

In this paper, we have presented the first iteration of *Story Designer* as a step towards a mixed-initiative co-creative system implementation of TropeTwist [14] to design narrative structures in EDD. The system allows the creation of narrative structures as narrative graphs that defines the overarching narrative, identifying characters, their roles and involvement, objectives, and core events. *Story Designer* also presents a step towards creating a holistic system, intertwining level design and narrative through simple level design constraints, effectively delimiting the search space of MAP-Elites with promising results. We analyzed and evaluated *Story Designer* and the impact of these level constraints through four simulated experiments; experiments 1-3 approach *Story Designer* in a more static scenario, while experiment 4 focuses on the step-by-step creation process.

Experiment 4 and, in general, the design process in *Story Designer* shows how the tropes, nodes, and connections, can be used to design a narrative structure step by step, changing the components of the narrative and how different elements in the game can be used and interpreted with simple changes. Defining conflicts among characters (thus, creating factions), defining primary and side objectives, as well as important elements in the narrative (e.g., plot devices), is a simple process. Changing these to adapt to the designer's goal is possible with minimal input. For instance, see the change from experiment 4.4 to 4.5 (fig. 2.d4,d5), where DRA passes from a side objective to a main part of the structure by creating an "entails" connection and forming a DerP meso-pattern (increasing the graph's interestingness score to 0.33). Equally important, the system preserves its properties and adapts to the narrative graph created, which could create a better experience for the designer. However, our evaluation was through simulated design sessions

(especially, experiment 4) highlighting properties and tradeoffs of the system. We aim at evaluating Story Designer with a user study to assess its usability, the expressiveness designers have when creating structures, and the experience intertwining and creating level design constraints.

Our next steps would be to continue the development of Story Designer to reincorporate the narrative structure into other facets and systems. Following a similar approach with constraints, narrative structures could constrain the search space for other facets, creating a feedback loop across facets and systems for a holistic approach. For instance, within EDD, narrative constraints could be reincorporated into both the level design facet [44] and the quest system to adapt main and side objectives [12].

References

- [1] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra, “Orchestrating Game Generation,” *IEEE Transactions on Games*, vol. 11, no. 1, pp. 48–68, 2019. 335, 337
- [2] E. Aarseth, “From Hunt the Wumpus to EverQuest: Introduction to Quest Theory,” in *Entertainment Computing - ICEC 2005* (F. Kishino, Y. Kitamura, H. Kato, and N. Nagata, eds.), vol. 3711, (Berlin, Heidelberg), pp. 496–506, Springer Berlin Heidelberg, 2005. Series Title: Lecture Notes in Computer Science. 335
- [3] C. Ashmore and M. Nitsche, “The Quest in a Generated World,” in *Proceedings of the 2007 DiGRA International Conference: Situated Play*, The University of Tokyo, Sept. 2007. 335, 337
- [4] Q. Kybartas and R. Bidarra, “A Survey on Story Generation Techniques for Authoring Computational Narratives,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 9, pp. 239–253, Sept. 2017. 335
- [5] E. Aarseth, “A Narrative Theory of Games,” in *Proceedings of the International Conference on the Foundations of Digital Games*, FDG ’12, (Raleigh, North Carolina), pp. 129–133, ACM, 2012. event-place: Raleigh, North Carolina. 335
- [6] M. P. Eladhari, P. L. Lopes, and G. N. Yannakakis, “Interweaving Story Coherence and Player Creativity through Story-Making Games,” in *Interactive Storytelling : 7th International Conference on Interactive Digital Storytelling, ICIDS 2014, Singapore, Singapore, November 3-6, 2014, Proceedings*, Lecture Notes in Computer Science, pp. 73–80, University of Malta, Msida, Malta, 2014. Issue: 8832. 335
- [7] K. Yu, N. R. Sturtevant, and M. Guzdial, “What is a Quest?,” in *Proceedings of the Intelligent Narrative Technologies Workshop at AIIDE*, 2020. 335

- [8] G. Smith, R. Anderson, B. Kopleck, Z. Lindblad, L. Scott, A. Wardell, J. Whitehead, and M. Mateas, “Situating Quests: Design Patterns for Quest and Level Design in Role-Playing Games,” in *Interactive Storytelling* (M. Si, D. Thue, E. André, J. C. Lester, J. Tanenbaum, and V. Zammito, eds.), (Berlin, Heidelberg), pp. 326–329, Springer Berlin Heidelberg, 2011. 335
- [9] P. Ammanabrolu, E. Tien, W. Cheung, Z. Luo, W. Ma, L. J. Martin, and M. O. Riedl, “Story Realization: Expanding Plot Events into Sentences,” in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7375–7382, AAAI Press, 2020. 335
- [10] M. C. Green, G. A. B. Barros, A. Liapis, and J. Togelius, “DATA Agent,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games, FDG ’18*, (New York, NY, USA), ACM, 2018. event-place: Malmö, Sweden. 335, 337
- [11] P. Tambwekar, M. Dhuliawala, L. J. Martin, A. Mehta, B. Harrison, and M. O. Riedl, “Controllable Neural Story Plot Generation via Reward Shaping,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019* (S. Kraus, ed.), pp. 5982–5988, ijcai.org, 2019. 335, 337
- [12] A. Alvarez, E. Grevillius, E. Olsson, and J. Font, “Questgram [Qg]: Toward a Mixed-Initiative Quest Generation Tool,” in *Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG’21*, (New York, NY, USA), Association for Computing Machinery, 2021. 335, 351
- [13] J. Doran and I. Parberry, “A Prototype Quest Generator Based on a Structural Analysis of Quests from Four MMORPGs,” in *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games, PCGames ’11*, (New York, NY, USA), ACM, 2011. event-place: Bordeaux, France. 335, 337
- [14] A. Alvarez and J. Font, “TropeTwist: Trope-based Narrative Structure Generation,” in *Proceedings of the 13th Workshop on Procedural Content Generation, FDG, 2022*. 335, 337, 338, 340, 341, 345, 346, 350

- [15] M. Trenton, D. Szafron, J. Friesen, and C. Onuczko, “Quest Patterns for Story-Based Computer Games,” in *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE’10, pp. 204–209, AAAI Press, 2010. event-place: Stanford, California, USA. 335, 337
- [16] M. Treanor, A. Zook, M. P. Eladhari, J. Togelius, G. Smith, M. Cook, T. Thompson, B. Magerko, J. Levine, and A. Smith, “AI-based game design patterns,” 2015. Publisher: Society for the Advancement of Digital Games. 335
- [17] M. Kreminski and N. Wardrip-Fruin, “Sketching a Map of the Storylets Design Space,” in *Proceedings of the International Conference on Interactive Digital Storytelling* (R. Rouse, H. Koenitz, and M. Haahr, eds.), vol. 11318, (Cham), pp. 160–164, Springer International Publishing, 2018. Series Title: Lecture Notes in Computer Science. 335
- [18] V. Breault, S. Ouellet, and J. Davies, “Let CONAN tell you a story: Procedural quest generation,” *Entertainment Computing*, vol. 38, p. 100422, May 2021. 335, 337
- [19] R. Barthes, “Introduction à l’analyse structurale des récits,” *Communications*, pp. 1–27, 1966. 335
- [20] V. Propp, L. Scott, and L. Wagner, *Morphology of the Folktale: Second Edition*. American Folklore Society Bibliographical and Special Series, University of Texas Press, 1975. 335
- [21] M. Thompson, J. Padget, and S. Battle, “Governing Narrative Events with Tropes as Institutional Norms,” in *Artificial Life and Intelligent Agents* (P. R. Lewis, C. J. Headleand, S. Battle, and P. D. Ritsos, eds.), vol. 732, pp. 133–137, Cham: Springer International Publishing, 2018. Series Title: Communications in Computer and Information Science. 335, 338
- [22] P. García-Sánchez, A. Velez-Estevez, J. Julián Merelo, and M. J. Cobo, “The Simpsons did it: Exploring the film trope space and its large scale structure,” *PLOS ONE*, vol. 16, p. e0248881, Mar. 2021. 335, 338
- [23] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, “Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions,” *IEEE Transactions on Games*, pp. 1–1, 2020. 336, 340, 345, 350

- [24] R. P. y. Pérez and O. Ortiz, “A Model for Evaluating Interestingness in a Computer-Generated Plot,” in *Proceedings of the Fourth International Conference on Computational Creativity, ICCCC 2013, Sidney, Australia, June 12-14, 2013* (M. L. Maher, T. Veale, R. Saunders, and O. Bown, eds.), pp. 131–138, computationalcreativity.net, 2013. 336
- [25] P. Lankoski, “Models for Story Consistency and Interestingness in Single-Player RPGs,” in *Proceedings of International Conference on Making Sense of Converging Media, AcademicMindTrek ’13, (New York, NY, USA)*, pp. 246–253, Association for Computing Machinery, 2013. event-place: Tampere, Finland. 336
- [26] J. P. Rowe, S. W. McQuiggan, J. L. Robison, D. R. Marcey, and J. C. Lester, “STORYEVAL: An Empirical Evaluation Framework for Narrative Generation,” in *AAAI Spring Symposium ’09*, p. 8, 2009. 336
- [27] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016. 337
- [28] R. M. Young, S. G. Ware, B. A. Cassell, and J. Robertson, “Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives,” *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, vol. 37, no. 1-2, pp. 41–64, 2013. 337
- [29] K. Hartsook, A. Zook, S. Das, and M. O. Riedl, “Toward supporting stories with procedurally generated game worlds,” in *2011 IEEE Conference on Computational Intelligence and Games (CIG’11)*, (Seoul, Korea (South)), pp. 297–304, IEEE, Aug. 2011. 337
- [30] A. Liapis, G. N. Yannakakis, and J. Togelius, “Computational Game Creativity,” in *Proceedings of the fifth International Conference on Computational Creativity (ICCC 2014)* (H. Toivonen, S. Colton, M. Cook, and D. Ventura, eds.), 2014. 337
- [31] A. Hoover, W. Cachia, A. Liapis, and G. Yannakakis, “AudioInSpace: Exploring the Creative Fusion of Generative Audio, Visuals and Gameplay,” in *Proceedings of the EvoMusArt conference*, vol. 9027, pp. 101–112, Springer, Apr. 2015. 337

- [32] N. I. Holtar, M. J. Nelson, and J. Togelius, “Audioverdrive: Exploring bidirectional communication between music and gameplay,” in *Proceedings of the International Computer Music Conference*, pp. 124–131, 2013. 337
- [33] D. Karavolos, A. Liapis, and G. N. Yannakakis, “A multi-faceted surrogate model for search-based procedural content generation,” *IEEE Transactions on Games*, 2019. Publisher: IEEE. 337
- [34] M. Cook and S. Colton, “A Rogue Dream: Automatically Generating Meaningful Content For Games,” in *Joint Proceedings of the AIIDE 2014 Workshops co-located with 10th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2014)*, 2014. 337
- [35] M. Treanor, B. Blackford, M. Mateas, and I. Bogost, “Game-O-Matic: Generating Videogames that Represent Ideas,” in *Proceedings of the Procedural Content Generation Workshop at the Foundations of Digital Games Conference*, 2012. 337
- [36] J. Dormans and S. Bakkes, “Generating Missions and Spaces for Adaptable Play Experiences,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, pp. 216–228, Sept. 2011. 337
- [37] A. M. Abuzurairq, A. Ferguson, and P. Pasquier, “Taksim: A Constrained Graph Partitioning Framework for Procedural Content Generation,” in *2019 IEEE Conference on Games (CoG)*, (London, United Kingdom), pp. 1–8, IEEE, Aug. 2019. 337
- [38] G. Smith, J. Whitehead, and M. Mateas, “Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 337
- [39] A. Liapis, G. N. Yannakakis, and J. Togelius, “Sentient sketchbook: Computer-aided game level authoring,” in *In Proceedings of ACM Conference on Foundations of Digital Games, 2013. In Print*, 2013. 337
- [40] M. Kreminski, M. Dickinson, J. C. Osborn, A. Summerville, M. Mateas, and N. Wardrip-Fruin, “Germinate: A Mixed-Initiative Casual Creator for Rhetorical Games,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2020. 337
- [41] M. Kreminski, M. Dickinson, M. Mateas, and N. Wardrip-Fruin, “Why Are We Like This?: The AI Architecture of a Co-Creative Storytelling Game,” in

International Conference on the Foundations of Digital Games, (Bugibba Malta), pp. 1–4, ACM, Sept. 2020. 337

- [42] J.-B. Mouret and J. Clune, “Illuminating search spaces by mapping elites,” *arXiv preprint arXiv:1504.04909*, 2015. 337
- [43] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, “Assessing the Effects of Interacting with MAP-Elites,” in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, pp. 124–131, AAAI, Oct. 2021. Section: Poster Papers. 337, 347
- [44] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering Quality Diversity in Dungeon Design with Interactive Constrained MAP-Elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 337, 338, 340, 351
- [45] M. Charity, A. Khalifa, and J. Togelius, “Baba is Y’all: Collaborative Mixed-Initiative Level Design,” in *2020 IEEE Conference on Games (CoG)*, pp. 542–549, 2020. 337
- [46] M. Charity, M. C. Green, A. Khalifa, and J. Togelius, “Mech-Elites: Illuminating the Mechanic Space of GVG-AI,” in *International Conference on the Foundations of Digital Games*, FDG ’20, (New York, NY, USA), Association for Computing Machinery, 2020. event-place: Bugibba, Malta. 337
- [47] A. Khalifa, S. Lee, A. Nealen, and J. Togelius, “Talakat: Bullet hell generation through constrained map-elites,” in *Proceedings of The Genetic and Evolutionary Computation Conference*, ACM, 2018. 337
- [48] C. Richmond and D. Schoentrup, “TV Tropes,” 2004. 338
- [49] J. Harris, “Periodic Table of Storytelling,” 2016. 338
- [50] Nintendo R&D1 and Intelligent Systems, “Super Mario Bros.,” Sept. 1985. Game [SNES]. Nintendo, Kyoto, Japan. Last played August 2020. 346
- [51] Nintendo R&D4, “The Legend of Zelda: A Link to the Past,” 1991. Game [SNES]. Nintendo, Kyoto, Japan. Last played January 2022. 346
- [52] Nintendo R&D4, “The Legend of Zelda: Ocarina of Time,” 1998. Game [N64]. Nintendo, Kyoto, Japan. Last played December 2016. 346
- [53] G. Smith and J. Whitehead, “Analyzing the Expressive Range of a Level Generator,” in *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, (New York, NY, USA), ACM, 2010. event-place: Monterey, California. 347

- [54] D. Gravina, A. Liapis, and G. N. Yannakakis, "Constrained Surprise Search for Content Generation," in *Proceedings of the IEEE Conference on Computational Intelligence and Games (CIG)*, 2016. 349
- [55] A. Liapis, G. N. Yannakakis, and J. Togelius, "Constrained Novelty Search: A Study on Game Content Generation," *Evol. Comput.*, vol. 23, pp. 101–129, Mar. 2015. Place: Cambridge, MA, USA Publisher: MIT Press. 349

PAPER XII - TOWARDS AI AS A CREATIVE COLLEAGUE IN GAME LEVEL DESIGN

Tinea Larsson, Jose Font, and Alberto Alvarez

ABSTRACT

In Mixed-Initiative Co-Creative tools, the human is mostly in control of what will and can be created, delegating the AI to a more suggestive role instead of a colleague in the co-creative process. Allowing more control and agency for the AI might be an interesting path in co-creative scenarios where AI could direct and take more initiative within the co-creative task. However, the relationship between AI and human designers in creative processes is delicate, as adjusting the initiative or agency of the AI can negatively affect the user experience. In this paper, different degrees of agency for the AI are explored within the Evolutionary Dungeon Designer (EDD) to further understand MI-CC tools. A user study was performed using EDD with three varying degrees of AI agency. The study highlighted elements of frustration that the human designer experiences when using the tool and the behavior in the AI that led to possible strains on the relationship. The paper concludes with the identified issues and possible solutions and suggested further research.

PUBLISHED IN

Proceedings of The 18th AAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE), AAI, 2022

TOWARDS AI AS A CREATIVE COLLEAGUE IN GAME LEVEL DESIGN

Introduction

Collaboration between AI and humans to co-design and co-create content is a significant challenge and the main focus of Mixed-Initiative Co-Creativity (MI-CC), which is the joint effort by a human user and AI to create content together [1, 2]. In an MI-CC environment, designers can unleash their creativity while the computer ensures playability, measures quality, and potentially inspires them towards more creative designs. These systems' objectives are to foster creativity and provide seamless proactive collaboration, ultimately enabling a mutually beneficial collaboration. The AI role has been categorized depending on the computer agency and initiative: nanny, pen-pal, coach, and colleague [3]. For an AI to be a colleague, it would have to intervene in the human process and take initiatives directly affecting the end product and creative process.

Morai Maker is an AI-driven Level Editor for Super Mario Bros-style games [4], which aims at having an AI as a colleague, with an equal role as the human designer, both adapting to each other. The Evolutionary Dungeon Designer (EDD) is a mixed-initiative design tool for rogue-like dungeon games [5]. EDD uses an evolutionary algorithm (MAP-Elites) to constantly generate finished rooms for the user to pick and replace their design based on the user's manual designs. The AI does not have any definitive control over the design decisions. Rather it suggests content adapted to the designer's current design, and the designer has the option not to incorporate the AI in their creations [6]. Nevertheless, it seems relevant to explore how other degrees of AI agency could affect the resulting co-creative process in terms of frustration, constraints, efficiency, or diversity, compared to when two humans create together. This comes with potential issues derived from altering the AI's agency; that human creativity can be dampened by restrictions in the creative process [1].

This paper explores how AI with varying degrees of agency affects the human

Version properties	AIv1	AIv2	AIv3
Can override AI-tiles	✓	✓	
Can override human-tiles	✓	✓	✓
Places directly		✓	✓

Table 1: Properties and differences between AI versions

users' design process in EDD. Three different versions of the tool are developed with varying degrees of the AI's control over the design process. These versions are then examined in a user study, and the results are analyzed to understand further the colleague relationship between humans and AI in MI-CC systems. The study also analyzes the degree of support these three AI companions have on lateral thinking, which is a vital part of the creative process. By assessing the three variants of agency, it is possible to compare the differences in the resulting creative relationships between the designer and AI, identifying factors that affect the designer's creative process in terms of frustrating elements, perceived limitations, and adaptation to their creative colleague.

Related Work

MI-CC focuses on tackling tasks between humans and AI with proactive initiative, where AI does not only assist humans but could also collaborate with them, leveraging on both their strengths [1,7]. *Initiative* is a multi-factor model combining: choosing the task, the agent in control and how the interaction is established, and the expected outcome [8]. In this work, both humans and AI have the same task, and the interaction is established as turn-based, each taking discrete control. The outcome is expected to vary as AI agency increases since larger constraints are added for the human that might need to adapt towards those.

Some MI-CC systems enable different collaborative approaches, which are considered in this paper. Tanagra [9] is a design tool for platform levels where the system takes as input and constraints the current user's design and creates content fulfilling gaps around it. Morai Maker [10] is an MI-CC tool where the human designer and AI take turns to design Super Mario Bros. levels. The AI adds content in its turn, which can be maintained or erased by the human designer, which the AI learns to adapt to through reinforcement learning. Furthermore, Lode Encoder [11] explores a creative collaboration where the human is constrained by only being able to use AI-generated content, which they need to choose to compose their design. This shows an unusual collaboration that users expressed as a playful, game-like creative process.

This paper uses EDD as the tool to explore AI agency and control. EDD is an MI-CC system where designers can create interconnected rooms composing a dungeon [6]. As designers create their content, the AI constantly suggests content adapted to the designer's design using the Interactive Constrained MAP-Elites (IC MAP-Elites). We make extensive use of IC MAP-Elites to generate rooms that are adapted to the target room. In [12], the authors show that IC MAP-Elites can generate high-performing and diverse rooms from different targets and using different dimension combinations. Its adaptiveness and stability, two necessary properties, were assessed with continuously edited rooms in [13], showing that the designer has a positive effect and can steer the algorithm with their design.

AI Roles and Adaptability

Lubart discusses four different roles a computer might take to promote creativity; *computer as nanny*: management of creative work; *computer as pen-pal*: communication service between collaborators; *computer as coach*: Using creative enhancement techniques; and *computer as colleague*: partnership between computer and humans [3]. This is further explored by Guzdial et al. where designers perceived the AI collaborator with more or less value depending on their desired role for the AI, varying between: *friend*, *collaborator*, *student*, or *manager* [4].

Establishing different roles such as colleague and collaborator might require some user model within the system. Designer modeling, as defined by Liapis et al. [14], is a way to classify and predict a designer's style, goals, preferences, and processes. Preference models [15, 16] have been built based on designers' choices and used as surrogate models to evaluate further generated content. Similarly, using the designers' creation, the designers' processes and styles could be modeled to inform other systems and adapt the generated content [17–19].

Altering Human-AI collaboration dynamics

The standard version of EDD presents a unidirectional relationship between the human designer and the AI, where the AI can only suggest content adapted to the designer's room, called the target room. As concluded in *Morai Maker* [10], allowing the human designer and the computer-controlled agent to take turns in the creative process enables the possibility of an even influence between the co-creators; thus, the human user and the AI take turns placing down tiles. Here we present three modified versions of EDD that implement three different dynamics for the human-AI co-creative process.

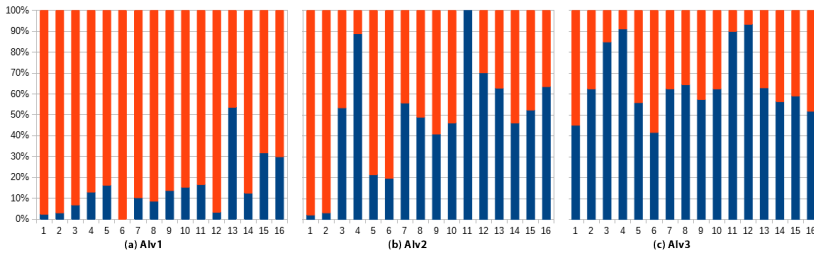


Figure 1: Percentage of tiles per co-creator in the rooms. Blue and Red bars relate to AI and Human placed tiles, respectively.

AI Version 1 (Alv1) - Low degree of agency

In this version the AI takes a suggestion colleague role. As the human manually designs a target room, the AI suggests tiles directly on top of the design. The human has the option to make use of the suggested tiles at will, placing them on the target room by clicking on them in the user interface. Both the human and the AI can override each other tiles.

AI Version 2 (Alv2) - Medium degree of agency

In this version, the AI places directly its recommended tiles rather than suggesting. Like in Alv1, both the human and the AI can override each other tiles.

AI Version 3 (Alv3) - High degree of agency

Unlike in the other versions, in Alv3, the human designer cannot be the sole contributor to the room designs. The AI places the tiles on their turn rather than suggesting, and the human cannot overwrite them. However, the AI can overwrite human tiles in their turn. This allows the exploration of how human designers react to being in a co-creative relationship where the AI has more control than them and add constraints to their design and goals.

The Design of the AI

The goal is for the AI to be perceived as dynamic, responsive, and helpful but not totally predictable. These qualities were selected to try to create a co-creator that supports the design choices of the human but also introduces unexpected elements to stimulate lateral thinking. By making the AI dynamic and responsive, the aim is to minimize the risk of unsatisfactory asymmetric design between the AI and the human, as reported by [10], where some of the critiques mentioned that the AI was designing its own parts of the level instead of creating consistency with the human

designer's contributions.

For all three versions of the AI, the AI component creates and regularly updates a list of generated tiles. IC MAP-Elites constantly runs in the background, maintaining a list of elites across its seven dimensions. When the human ends their turn, the elites are processed by a KNN algorithm ($K=20$) that picks the set of tiles that will be used by the AI contributor comparing the elites to the target room on the seven dimensions. The resulting list of elites is further processed tile by tile, creating a final list that contains the most reoccurring tile types per position in the contribution area, which constitutes the list of generated tiles for the computer to use. From this list, and to further the perception of the AI being dynamic, it selects a random amount of tiles between 50% and 100% of the amount of human-placed tiles. However, through this process it is likely that the agent will not be human-like. For example, humans are likely to favor symmetry in a room, but the AI in this tool will consider all dimensions equal. The AI will also consider all types of tiles equally when calculating the most common tile in a position in the generated rooms.

Furthermore, depending on the level of agency, these tiles are just displayed as suggestions on the UI or directly placed when the computer takes control of the creative process. Both creators have a maximum of 12 tiles that they can contribute per turn. This was determined through experimentation during development since it is large enough to allow the designer to create a small subsection during their turn. If the human does not contribute one round before pressing "End Turn," the previous turn's amount of tiles are used for the calculation to enable the human to press "End Turn" repeatedly to let the AI keep contributing if that is desired. The available locations for the AI to contribute each turn are limited to a rectangular area surrounding the tiles the human designer recently placed, including a margin of 1 tile.

Experiment Setup

We conducted a user study to explore the user experience of using different levels of AI agency, the different design characteristics, and the relationship between the human designer and the AI. We collected both quantitative data on the AI's impact on the co-designed end product and qualitative data through think-a-loud and semi-structured interviews regarding the users' experience when interacting with the AI. The interview structure is inspired by the pyramid model, meaning the interviews will begin with specific questions, and gradually have more open questions, which naturally allows for a discussion towards the end. This model is chosen to support the variation of subjects the interview is desired to cover, as

	AIv1	AIv2	AIv3
Leniency	0.56±0.07	0.62±0.09	0.57±0.08
Linearity	0.91±0.02	0.92±0.02	0.91±0.01
MesoPat	0.15±0.05	0.13±0.07	0.12±0.05
SpatialPat	0.35±0.1	0.41±0.11	0.34±0.09
Symmetry	0.43±0.11	0.35±0.18	0.35±0.12
W_{dens}	0.27±0.09	0.26±0.08	0.21±0.05
W_{spar}	0.21±0.05	0.19±0.03	0.15±0.01
E_{dens}	0.24±0.07	0.27±0.06	0.3±0.06
E_{spar}	0.22±0.05	0.32±0.05	0.35±0.06
T_{dens}	0.37±0.13	0.28±0.09	0.34±0.07
T_{spar}	0.36±0.11	0.3±0.1	0.37±0.05
Steps	39.25±6.38	84.31±14.85	76.75±17.02

Table 2: Summary of the created rooms filtered by the AI version used. All values are the average of all the created rooms using the specific AI version. The first five values relates to the MAP-Elites dimensions, then the fitness of the rooms, the density and sparsity values for wall (W), enemies (E), and treasures (T), and finally the avg. steps taken to design a room.

well as support natural transitions between the questions and their openness. The questions and user study procedure can be found in Appendix A.

Eight participants tested our tool with game design and level design experience. One participant was a professional game designer with eight years of professional experience (first participant), and seven participants were third-year Game Development students. They all had an individual digital session, where we shared our screen, and they took remote control to conduct the study. Participants accepted to participate, signed consent forms, and then received a short introduction describing the experiment and its steps. The participants were then asked to design two contiguous rooms in a dungeon, repeating this process for each of the AI variants and expressing their design decisions verbally whenever they felt like it. After using the tool, the participants were interviewed, focusing on and covering an overarching understanding of the user experience, particularly in terms of creativity and interaction with the AI.

For all the sessions, human designers could place up to 12 tiles, and the AI could place as many tiles as the human placed. The AI could contribute only in a rectangular area surrounding the tiles the human designer recently contributed with, including a margin of 1 tile. This choice is made to support a responsive and collaborative behavior of the AI that builds on the human designer's contribution.

	AI replaced	Human replaced	Interactions
AIv1	3.31±2.1	1.63±1.47	3.88±0.59
AIv2	24.94±6.67	12.06±5.29	8.56±3.28
AIv3	19.25±5.13	0	9.13±3.64

Table 3: Summary of the interactions between AI and human. The first two columns relate to the avg. amount of tiles that the AI or human replaced of each other. Interactions refer to the avg. number of times the human pressed “End Turn.”

Results

We present the results from the user study regarding the designed rooms, the room design process with the AI, and the participant’s interview responses. Figure 1 and 2 show the tile contribution for both human and AI per room and tested version, and the final tile distribution across all versions, respectively. Figure 3 shows a sample of the designed rooms corresponding to the different AI versions, respectively.

The designed rooms, when interacting with the AIv1 (fig. 3.a), generally include a vast majority of human-placed tiles or unedited tiles (86%). Many of the rooms display treasures and enemies placed close to each other, often with an enemy blocking a treasure. As displayed in fig. 3.b (AIv2), rooms contain more AI edited tiles (51%), and long continuous walls are less common compared to when using AIv1. Fig. 3.c shows a sample of the designs using AIv3. The resulting rooms are less symmetric, contain fewer continuous walls, and are, in general, somewhat less organized.

The relation between the total amount of tiles placed by the human designer and the AI for each version provides insight into how much the human designers incorporated the AI in their design process. When using AIv1, the human designers generally contributed with a majority of tiles to the resulting room (86% in general). When using AIv2, the results vary more between rooms and/or designers, such as fig. 3.b 1 and 9 with a general 49% of tiles placed by humans. In AIv3, the AI generally had a majority of tiles in the resulting rooms (32% placed tiles by human designers in general). Figure 2 shows the tiles by each co-creator. Whereas human designers focused mostly on walls and structures, the AI focused on removing tiles and adding floors.

Furthermore, table 2 and 3 summarizes the created rooms by AI version. Table 2 shows that there is not much variation regarding the IC MAP-Elites dimensions, but Symmetry and Meso patterns reduce as the AI gains more agency, and there are fewer spatial patterns in the high and low agency. It also shows that wall and enemy density and sparsity go lower and higher, respectively. This is expected,

given that as the AI is more dominant in the design (fig. 1), walls are diminished (fig. 2), which can also be observed in the examples in fig. 3. Finally, designers spent less time on their design for AIv1 and AIv3. In AIv1, designers can edit rooms towards their goal and the AI does not modify their design (unless wanted), which means that the designer does not need to modify the room much. This is expected to increase for AIv2 and AIv3 because the AI now adds its editions, which count towards the steps. However, since the AI can add as many tiles as the designer, the designer is still doing roughly half of the steps. When using AIv3, rooms take slightly fewer steps than AIv2, probably due to the AI taking over areas where to design and designers trying to work around constraints further discussed in the next section.

Table 3 shows the interaction between human and AI. As expected, the human designers interacted less with the AI in AIv1, and in general, they overwrote less since they could decide which elements from the AI to include. Five out of the 16 rooms included AI-placed tiles where the human designer did not replace any of the AI tiles. Interactions and human and AI tile replacements increase for AIv2 and AIv3, which is also expected. The AI overwrote an avg. of 24.94 and 19.25 tiles in each subsequent version, and human designers overwrote an avg. of 12.06 in AIv2. This shows that the AI had more involvement and changed the design more often. The human designer could either use their turn to replace the tiles or continue using those tiles. However, in AIv2, the designers did not replace all of the AI tiles; instead, they also chose to work around them and replaced them when needed, allowing the AI to participate more. In 13 out of the 16 rooms created the designer replaced fewer tiles than the AI did, and for 9 of those it was less than 50%.

Perceptions of the different AI versions

Four participants preferred to use AIv1 due to higher controllability over the final design. Two participants preferred AIv2 since they liked the efficiency of the AI placing down tiles, but they still remained in control over the design process. One participant said that they preferred both AIv1 and AIv3, as they felt that they fulfilled different purposes. AIv1 contributed with inspiration to get out of a writers block kind-of situation in level design, while AIv3 offered a more unusual creative experience, and created unique levels. One participant preferred AIv3 categorizing it as more efficient. However, AIv3 was the most disliked due to most participants feeling constrained by the AI's decisions, which forced them to work around the AI's design; too invasive on their design process, and in general, feeling the AI dismissed their contributions. Two participants disliked AIv2 the most due to

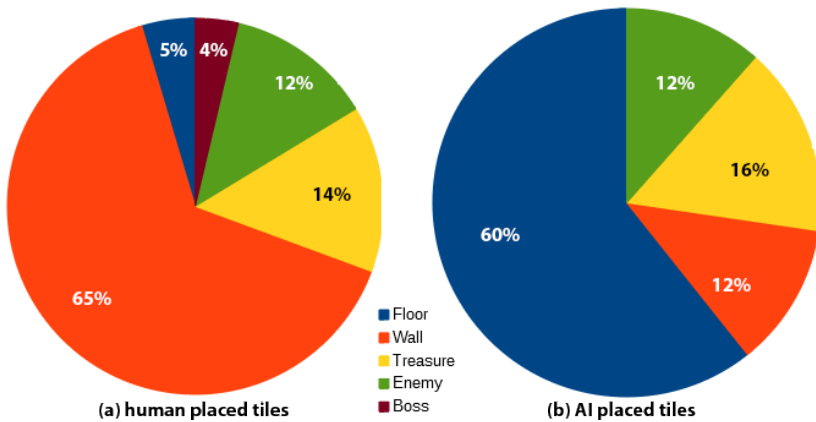


Figure 2: The types of tiles placed and their percentage of occurrence by human designers (a) and the AI (b)

frustration and slowing their design process because of the repetitive work of recreating their original idea as the AI would overwrite their tiles. One participant disliked AIV1 as it was the slowest to work with.

AI's behavior

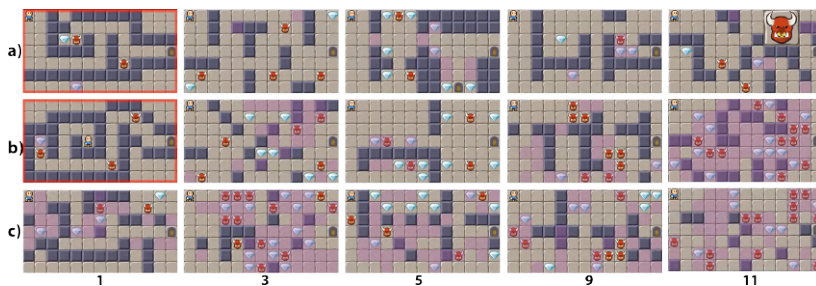


Figure 3: Sample resulting rooms from the study. a, b, and c are samples from the same users using the different versions. Red borders indicate that the room contains unreachable tiles. Purple tinted tiles are the ones added by the AI.

Five participants described the AI's behavior as random and unpredictable. Many mentioned that the AI often placed floor tiles over the human-placed walls, treasures, enemies, and bosses. Additionally, two participants mentioned that the AI often broke down sub-rooms or long walls with floor tiles. The more positive descriptions of the AI's behavior were that the AI repaired their level if it had unreachable tiles by placing floor tiles on the positions wall tiles were, making all tiles reachable. One participant expressed that they thought the AI created unique-looking high-quality

levels.

The Creative Relationship

Five participants expressed that the AI contributed with ideas they liked and either kept or considered incorporating. Many also expressed that they found it frustrating that the AI overwrote the human placed walls, treasures, enemies, or boss tiles with floor tiles, as they perceived this as the AI removing their contributions without adding anything new. All participants answered that they did adapt to AIv2 and AIv3 during their design process. When using AIv2, the participants adapted to the AI by either being inspired by the AI's contribution or by getting frustrated with the AI repeatedly placing tiles the human designer did not like. When using AIv3, they felt they had to adjust to whatever the AI contributed and felt forced to adapt. Additionally, six participants answered that they did not feel that the AI adapted to them. One answered that it did feel that the AI adapted to them, and one participant had no opinion.

Four participants described the relationship as working together with someone who only says no to your ideas without contributing to new ideas. Three participants described it as frustrating and/or a fight for control between the AI and the human designer. Two participants described it as an iterative collaboration and compared it to two people working on the same product but in different steps. One participant described the relationship as a brainstorming session where you work in a "Yes, and..." fashion, meaning you work iteratively and only adds to the idea and never decline the other collaborator's contribution.

The Creative Process

Four participants mentioned that the AI negatively affected their creative process, especially when the AI placed floor tiles over their placed walls, treasures, enemies, and bosses. This was described by multiple participants as the AI "destroying their creations", and many felt forced to let the AI "take control over the room." Two participants described the creative process as the AI and the human designer working against each other. Two other participants described it as a process where the AI brought forward ideas that they found interesting. One participant described it as letting the AI form an idea that the human designer finally polished.

Constraints and Design Goals

Five participants answered they felt progressively more constrained as the AI gained more control, although one participant answered that they did not feel constrained at all with any of the versions. Six participants answered they had a

design goal of creating a boss room. However, the AI placed floor tiles over the boss tiles, making this impossible in AIv2 and AIv3, forcing them to change their design goals. Two participants answered that they did not have set design goals and solely created content each turn with no specific concept for the whole room in mind.

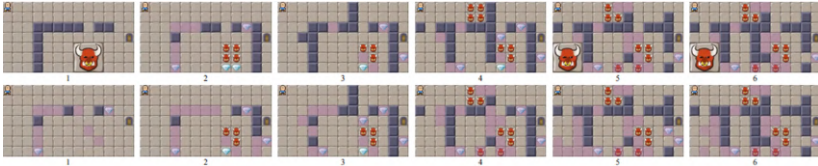


Figure 4: Sample step by step process when using AIv2. Human’s turn is top row, while the AI’s turn is the bottom row.

Discussion

Willingness to include the AI in the design process

All participants expressed an interest and willingness to see what the AI could come up with to design the rooms, emphasizing that they either considered or incorporated the ideas brought forward by the AI, which further supports other MI-CC research conclusions [4, 11]. However, many participants expressed multiple frustrating factors, and based on figure 1.a, when given the opportunity, the human designers did not include much of the AI’s contribution but might have provided some ideas that either influenced or were part of the final design. Figure 1 also displays that as the AI got more agency, the AI tiles at the end design increased. This can be due to frustration expressed by participants where they didn’t agree with the AI design and ended up handing over the control to the AI and giving up, to some extent, their aspiration to create. The lower meso- and spatial-patterns combined with the lower symmetry in AIv3 are also part of the issue. The fewer patterns that exist mean that the rooms are less structured, which, combined with no way to correct these, could result in designers feeling that the levels are more “random”.

Variants of the users

Most participants used the tool similarly except for participants 1 and 6. Participant 1 didn’t want to incorporate the AI’s contributions, as it can be seen in fig. 1, explicitly stating that “... I don’t think level design is a good place for an AI that has more control than the human... The little details that I love in level design would never be created by an AI. Nice little references, or easter-eggs, or how

humans get inspired by simple things..." On the other hand, participant 6 recurrently incorporated many of the AI's tiles. When using AIV2 and AIV3, they pressed "End Turn" repeatedly to find out what the AI would be able to create, commenting "I want to see if it can create something cool." However, while their approach was completely different, they both agreed that they would prefer the AI to create a complete room, and they could polish it from there.

Frustrating factors and Constraints

The participants expressed multiple frustrating factors within the tool. The main point was the repetitive behavior of overwriting the human tiles with floor tiles, removing their ideas without contributing with anything of value, and the human designer feeling forced to move on from those positions and contribute somewhere else in the room. This was exacerbated when using AIV2 as, unlike AIV1, the AI placed down the tiles rather than suggesting, and unlike AIV3, the human designer still had the option to overwrite the AI-placed tiles. The human designers assign value to each tile type as they provide different aspects in level design; for instance, participants often placed enemies and treasures close to each other, possibly to create a risk and reward in the level. Figure 4 shows one example of a participant creating a room using AIV2, step by step. The human designer's first contribution includes a long continuous wall and a boss. When the AI has its first turn, it contributes with floor tiles overwriting the designer's tiles. Towards the end of the design, the human designer places a boss tile in the bottom left corner that the AI overwrites with floor tiles twice before the human designer gives up and finishes the room without a boss. This sample creation process shows that the AI tried to steer the room towards more leniency and open areas, which contradicted the human's goal.

Another main frustrating factor was the loss of control experienced by human designers when co-creating with the AI, especially AIV3. Participants expressed that the AI's decisions limited them and were forced to work around what the AI designed. As the AI gained control, they felt their creative process got increasingly constrained. This aligns with the Lode Encoder study [11], where the participants expressed frustrations with completing a playable level, as they were forced to rely on the AI to generate the option they wanted in the final stages of the level creation. Further, when using AIV3, the number of positions that the human has available decreases with every turn, while the AI can continue to place tiles on any position, which unavoidably limits the human's control over the final design.

Most of the participants felt frustrated and constrained as the AI gained more

control over the design process. Additionally, all participants suggested removing the turns and constraints of the number of tiles per turn to improve the tool. Three of the eight participants expressed that adjusting the AI's role to one of an assistant to the human designer would improve their creative experience.

The concept of a well performing, high agency co-creator

Most participants showed a willingness to incorporate AI into the creative process, contributing with new ideas or performing services such as ensuring feasibility. Yet they were reluctant to incorporate higher agency AI, which suggests that the AI needs to be aligned with their goals, intentions, and procedures, i.e., have an accurate designer model [14]. Within EDD and level design tools, multiple practical improvements are to be made. Five participants described the AI's behavior as random and unpredictable, especially when overwriting human-made structures and contributions. The AI currently calculates the most common tiles in the positions of the contribution area and contributes with the tiles of highest occurrence among a set of generated elites. This contradicts how human designers perceive the design importance of tiles, valuing higher usable tiles rather than floors. The AI could then weigh higher those and the combined structures they create. Additionally, the AI could favor unedited areas before overriding human-placed tiles to support rather than override.

Another important point is that all AI versions are static in the design process, which means that the AI follows the same procedure regardless of the agency level. The collaboration do change in the design process (e.g., suggest or directly placing tiles, or if AI tiles can be removed) but other aspects and parameters do not change or adapt to designers. These parameters are connected to the overarching design of the AI rather than the AI's agency, which might have affected the designers' perception. For instance, given that the AIv3 tiles could not be replaced, changing the amount of tiles, rectangular area, or its adaptability in regards to what the designer had created thus far could be beneficial.

Furthermore, the AI seemed to break apart walls and open up sub-rooms. This is possibly a result of the Linearity and Meso-Pattern dimensions in the MAP-Elites algorithm. The resulting elites of the generated rooms with the highest linearity will have the highest amount of traceable paths. Many participants seemed to want to create rooms with long walls, sub-rooms, paths that required the player to encounter enemies, and common aesthetical features such as symmetry. Analyzing the path designers are taking in these dimensions could better inform the search for content and the generation of elites, so the content is adapted to those preferences. However,

adapting these dimensions might be counterproductive for the other dimensions, as symmetric rooms might not create balanced rooms regarding the Leniency dimension as it might not be considered as important. Another approach could be to incorporate designer modeling. By identifying possible design goals or design styles of the human co-creator, the AI can adjust its decisions and behavior to offer different levels of support depending on the human designer's behavior [14, 18].

Conclusion

This study explored the limitations and possibilities of an MI-CC-tool with an AI with a varied agency. We aimed at doing an initial exploratory study with static parameters, resulting in baselines to analyze what can be done and how designers experienced the system. This, in turn, opens up and continues the discussion towards AI collaborating as a colleague and enabling alternative ways to foster creativity (e.g., constraining the design space such as in [11]). Our study showed that AI gaining control over the design results in frustration and feeling constrained. Constraints are not bad per se, as they can be a way to foster creativity [20, 21], but they need to be placed in a way that the human designer might feel inspired, motivated, or supported to continue the design. Human designers had to adapt towards those imposed goals instead of the other way around, which creates an unwanted dynamic when human designers perceive the AI's behavior as erratic, random, and without clear objectives.

Many of the results pointed to a general preference for an AI with a more supportive role in collaborative tools. One approach could be to have a hybrid model between what is presented in this paper and other typical MI-CC systems that focus more on suggesting final designs. The AI could take parameters from the human designer, such as an area in a room, amount of tiles, or an attribute that the human designer would like to increase in the room, but still maintain their design, effectively constraining the AI to find creative ways to achieve its goals. In EDD, designers can lock tiles to not be changed by the AI, which is something to be experimented with. Although this would give the human designer a slightly higher degree of influence on the end product compared to the AI, the constraints of how many tiles can be locked, or possibly what types of tiles can be locked, can be experimented with to adjust the relationship between AI and human designer. Currently, the search is steered, to some extent, by the designers' design, but in future work, we could bias the search even more towards interesting areas based on the creation process and the trajectories they are taking in behavior dimension space.

Additionally, using designer models is a feasible approach. By predicting design goals, adapting to phases of the design process, or identifying certain design styles

and adapting to the human designer, a responsive and adaptive, intelligent, and human-like artificial co-creator could be developed. This could allow for an AI that adapts to the human designer and performs well enough that the frustrations and feelings of constraints are minimal or perceived as less prevalent as the designs turn out more similar to what the human desired.

Appendix A

Interview procedure and questions

Setting up a user study session

1. The conductor starts a meeting in Zoom.
2. The conductor explains the steps to the participant.
3. When consent of recording the session is acquired by the conductor, the conductor starts recording.
4. The conductor starts the tool and shares the screen.
5. The conductor allows the participant to control the conductors machine via Zoom remote control.
6. The conductor instructs the participant to perform the test.

Instructions

The task is to design at least two rooms in a dungeon world, with each variant of the AI.

- Step 1: Choose the LOW level of the AI and click “Create World”.
- Step 2: You are now in the World Editing View. Edit as you please. To enter the room editing view, double click the room you wish to edit.
- Step 3: You are now in the Room Editing View. Use the brushes on the left to edit the room. Click “End Turn” to end your turn, and let the AI contribute.
 - AI-placed tiles are tinted purple. AI-suggested tiles are tinted green.
 - If you are in the LOW variation of AI, click the green suggestions you’d like to place. Click continue when you want to have your turn again.
 - To go back to the World Editing View, click “Go To World View”.

- Step 4: When you feel satisfied with your creation, tell me so. Restart the program. Start over at step 1 for the next AI version until all three are used once.

Interview Questions

- Q1: Which of the three versions of AI did you prefer? Why?
- Q2: Which of the three versions of AI did you find least appealing? Why?
- Q3: How would you describe the creative experience?
- Q4: What is your perception of the AI's behaviour?
- Q5: Did you feel your creativity was constrained when using any of the three AIs?
- Q6: Did you adapt to the different AI versions? In what ways?
- Q7: Did you perceive that the AI adapted to you? In what ways?
- Q8: How would you describe the relationship between designer and the AI?
- Q9: How did the AI's decisions affect your creative process?
- Q10: How did the different versions affect your design goals?
- Q11: What do you think is missing or needs to be improved for an AI as the one of the HIGH-version (with high initiative) to be used in collaborative tools?

References

- [1] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, “Mixed-Initiative Co-Creativity,” in *Proceedings of the 9th Conference on the Foundations of Digital Games*, 2014. 363, 364
- [2] A. Liapis, G. N. Yannakakis, C. Alexopoulos, and P. Lopes, “Can Computers Foster Human Users’ Creativity? Theory and Praxis of Mixed-Initiative Co-Creativity,” *Digital Culture & Education*, vol. 8, no. 2, pp. 136–153, 2016. 363
- [3] T. Lubart, “How can computers be partners in the creative process: Classification and commentary on the Special Issue,” *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 365 – 369, 2005. 363, 365
- [4] M. Guzdial, N. Liao, J. Chen, S.-Y. Chen, S. Shah, V. Shah, J. Reno, G. Smith, and M. O. Riedl, “Friend, Collaborator, Student, Manager: How Design of an AI-Driven Game Level Editor Affects Creators,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, (New York, NY, USA), Association for Computing Machinery, 2019. event-place: Glasgow, Scotland Uk. 363, 365, 373
- [5] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering Creativity in the Mixed-initiative Evolutionary Dungeon Designer,” in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, FDG ’18, 2018. 363
- [6] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering Quality Diversity in Dungeon Design with Interactive Constrained MAP-Elites,” in *2019 IEEE Conference on Games (CoG)*, 2019. 363, 365
- [7] J. F. Allen, “Mixed-Initiative Interaction,” in *Intelligent Systems, Trends and Controversies section*, pp. 14–16, 1999. 364

- [8] D. Novick and S. Sutton, "What is Mixed-Initiative Interaction?," in *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, Jan. 1997. 364
- [9] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, Sept. 2011. 364
- [10] M. Guzdial, N. Liao, and M. Riedl, "Co-Creative Level Design via Machine Learning," in *Joint Proceedings of the AIIDE 2018 Workshops co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2018)*, Edmonton, Canada, November 13-14, 2018., 2018. 364, 365, 366
- [11] D. Bhaumik, A. Khalifa, and J. Togelius, "Lode Encoder: AI-constrained co-creativity," in *2021 IEEE Conference on Games (CoG)*, pp. 01–08, IEEE, Aug. 2021. 364, 373, 374, 376
- [12] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, "Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions," *IEEE Transactions on Games*, pp. 1–1, 2020. 365
- [13] A. Alvarez, J. Font, S. Dahlskog, and J. Togelius, "Assessing the Effects of Interacting with MAP-Elites," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 17, pp. 124–131, AAAI, Oct. 2021. Section: Poster Papers. 365
- [14] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for personalized game content creation tools," in *Artificial Intelligence and Game Aesthetics - Papers from the 2013 AIIDE Workshop, Technical Report*, vol. WS-13-19, pp. 11–16, AI Access Foundation, 2013. 365, 375, 376
- [15] A. Alvarez and J. Font, "Learning the Designer's Preferences to Drive Evolution," in *Applications of Evolutionary Computation* (P. A. Castillo, J. L. Jiménez Laredo, and F. Fernández de Vega, eds.), vol. 12104 of *Lecture Notes in Computer Science*, (Cham), pp. 431–445, Springer International Publishing, 2020. 365
- [16] A. Liapis, G. N. Yannakakis, and J. Togelius, "Adapting Models of Visual Aesthetics for Personalized Content Creation," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, pp. 213–228, 2012. 365

- [17] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for Sentient Sketchbook," in *Proc. IEEE Conf. Computational Intelligence and Games*, pp. 1–8, Aug. 2014. ISSN: 2325-4289. 365
- [18] A. Alvarez, J. Font, and J. Togelius, "Designer Modeling through Design Style Clustering," *IEEE Transactions on Games*, pp. 1–1, 2022. 365, 376
- [19] E. Halina and M. Guzdial, "Threshold Designer Adaptation: Improved Adaptation for Designers in Co-creative Systems," 2022. 365
- [20] M. Boden, *The Creative Mind: Myths and Mechanisms*. Routledge, 2004. 376
- [21] O. A. Acar, M. Tarakci, and D. v. Knippenberg, "Creativity and Innovation Under Constraints: A Cross-Disciplinary Integrative Review," *Journal of Management*, vol. 45, no. 1, pp. 96–121, 2019. _eprint: <https://doi.org/10.1177/0149206318805832>. 376