LICENTIATE THESIS

STUDIES IN COMPUTER SCIENCE NO 11, LICENTIATE THESIS

ALBERTO ALVAREZ

EXPLORING THE DYNAMIC PROPERTIES OF INTERACTION IN MIXED-INITIATIVE PROCEDURAL CONTENT GENERATION

MALMÖ UNIVERSITY

EXPLORING THE DYNAMIC PROPERTIES OF INTERACTION IN MIXED-INITIATIVE PROCEDURAL CONTENT GENERATION

Malmö University,
Studies in Computer Science No 11,
Licentiate Thesis

ISBN 978-91-7877-139-4 (print)
ISBN 978-91-7877-140-0 (pdf)
Holmbergs, Malmö 2020

ALBERTO ALVAREZ

EXPLORING THE DYNAMIC PROPERTIES OF INTERACTION IN MIXED-INITIATIVE PROCEDURAL CONTENT GENERATION

Malmö University, 2020
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.

The publication is also available at:
http://doi.org/10.24834/isbn.9789178771400

To the brave Venezuelans; the sun will shine again.

ABSTRACT

As AI develops, grows, and expands, the more benefits we can have from it. AI is used in multiple fields to assist humans, such as object recognition, self-driving cars, or design tools. However, AI could be used for more than assisting humans in their tasks. It could be employed to collaborate with humans as colleagues in shared tasks, which is usually described as Mixed-Initiative paradigm. This paradigm creates an interactive scenario that leverage on AI and human strengths with an alternating and proactive initiative to approach a task. However, this paradigm introduces several challenges. For instance, there must be an understanding between humans and AI, where autonomy and initiative become negotiation tokens. In addition, control and expressiveness need to be taken into account to reach some goals. Moreover, although this paradigm has a broader application, it is especially interesting for creative tasks such as games, which are mainly created in collaboration. Creating games and their content is a hard and complex task, since games are content-intensive, multi-faceted, and interacted by external users.

Therefore, this thesis explores MI collaboration between human game designers and AI for the co-creation of games, where the AI's role is that of a colleague with the designer. The main hypothesis is that AI can be incorporated in systems as a collaborator, enhancing design tools, fostering human creativity, reducing their workload, and creating adaptive experiences. Furthermore, This collaboration arises several dynamic properties such as control, expressiveness, and initiative, which are all central to this thesis. Quality Diversity algorithms combined with control mechanisms and interactions for the designer are proposed to investigate this collaboration and properties. Designer and Player modeling is also explored, and several approaches are proposed to create a better workflow, establish adaptive experiences, and enhance the interaction. Through this, it is demonstrated the potential and benefits of these algorithms and models in

the MI paradigm.

Keywords. Mixed-Initiative, Procedural Content Generation, Quality Diversity, Computer Games, Evolutionary Algorithms

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, ACM, 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, ACM, 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, IEEE, 2019.

[4] A. Alvarez and M. Vozaru, "Perceived behaviors of personality-driven agents," *Violence | Perception | Video Games: New Directions in Game Research*, pp. 171–184, 2019.

[5] 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.

[6] A. Alvarez, J. Font, and J. Togelius, "Designer modeling through design style clustering," *Submitted to IEEE Transactions on Games*, 2020. Preprint available: https://arxiv.org/abs/2004.01697.

Related papers but not included in the thesis

[7] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions,” *Submitted to IEEE Transactions on Games*, 2020. Preprint available: https://arxiv.org/abs/2003.03377.

Personal Contribution and Clarification

Publication [7] is an extended version of [3], adding new features to the proposed algorithm, experiments, and evaluations. It is currently under review in IEEE Transactions on Games.

For all publications above except for [1], [2], and [4], the first author was the main contributor with regards to inception, planning, execution and writing of the research. For [4], both authors contributed equally.

For both [1] and [2], 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. However, both [1] and [2] research originated from the following Master theses for which I was the 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.
  Available: http://muep.mau.se/handle/2043/25889
- S. Johansson, "A Study on Fitness Functions and Their Impact in PCG," M.S. Thesis, DVMT, TS, Malmö University, Malmö, 2018.
  Available: http://muep.mau.se/handle/2043/25548

ACKNOWLEDGEMENT

I want to thank all my supervisors, Jose Font, Steve Dahlskog, Julian Togelius, and Nancy L. Russo; your advice, uncountable conversations, mentorship, and friendship is paramount for 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!

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 throughout the process of completing my licentiate. Your suggestions and discussions on my study plan helped shaped several of the decisions on this thesis. I would also like to thank Associate Professor Sebastian Risi for acting as the opponent in the licentiate defense.

Furthermore, I would like to thank all my PhD colleagues for uncountable conversations, discussions, fikas, and random encounters! This goes especially to Majid Ashouri, Sergei Dytckov, Lars Holmberg, Johan Salo, and Jens Sjöberg (PhD at Jonköping University); your support and ears to discuss much more than research have enriched my journey so far. I would like to thank the faculty (and visiting faculty) for your support, friendship, and experience. Especially Cecilia Ovesdotter Alm, Solveig-Karin Erdal, Johan Holmberg, Åse Jevinger, Annabella Loconsole, Susanne Lundborg, Petra Mullerova, and Thomas Pederson; this little space is not enough to write all your help and support throughout this period.

Everyone that knows me (or has heard me talk for more than 5 minutes) knows how much I speak about my experience at the IT University of Copenhagen. Therefore, I cannot miss this opportunity to thank everyone I met there, from professors to former colleagues and fellow students,

especially Pablo Sarmiento and Miruna Vozaru, great friends and collaborators.

I would like to thank my partner, Petronella Ottosson. In many ways, the path that brings me here presenting this work would have never been followed if it wasn't for you. Through eight wonderful years, your love, intelligence, and kindness have made me thrive and made the path endurable and achievable even in times when everything seemed impossible. I love you.

Last but not least, I would like to thank my parents, Armidas Alvarez and Beatriz Uribe, for your life teachings, sacrifices, and encouragement to follow my dreams even if it meant to be at a very different time zone. I would like to extend this gratefulness to my brothers, Carlos Alvarez and Orlando Alvarez, for the encouragement over the years and for distracting our parents while I went farther and farther away. Of course, I could not end this acknowledgment without thanking my Swedish and second family, Weronica Bernhardsson, Kajsa-Lena Ottosson, Matilda Ottosson, and Mats Ottosson. Thank you so much for all you have done for me and all we have lived together; the family and closeness feeling is the secret ingredient for a wholesome and endurable path.

Finally, thanks to all the persons that I have met and talked to thus far, which these pages are not enough to list; we might have met for a brief moment or be long time friends, but know for sure that you have in one way or another influence my path.

CONTENT

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

During the first three millennia, the Earthmen complained a lot.

John McCarthy

John McCarthy was spot on, as we humans are a major source of complaint. Nevertheless, such complaints make us strive and search for solutions and better approaches to cope with our needs and objectives. Ironically, we would end up complaining about it, restarting the loop.

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 and helpful in many ways; this raises an essential question that holistically motivates and drives 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 exploring multiple approaches for Human-AI collaboration. The goal is to develop systems and algorithms representing a computational designer to collaborate in the creation of content with a human designer. Tasks that 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 [1]. *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 [2].

Moreover, Horvitz discussed such a question in terms of Intelligent User Interfaces [3], 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 [4]. Horvitz presented several principles of mixed-initiative interaction and its challenges, many of which still exist [5], mainly describing this interaction as conversation systems between AI and humans [6]. 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 [7]. 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 [8–10].

Paramount is the role of the computer agent in this interaction, as it would help establish the boundaries of the interaction, what is expected, and how creativity could be fostered. Lubart analyzed this interaction and examined the different ways computers could be involved in creative work to promote creativity. In his work, he proposed four roles: *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 [11]. Recently, this was 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* [12].

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 [9]. Usually, games are developed by more than a person (although many exceptions exist [13–15]), 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 [16, Chapter 14]. However, no matter the team's size and talent, the fact remains that developing games is a hard challenge [17]. 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 [18].

Procedural Content Generation (PCG) is a field within computational

intelligence in games, that focuses on the use of algorithms to create game content [19]. PCG algorithms have been used to aid in the creation of a plethora of games such as No Man Sky [20], Spelunky [21], or Minecraft [13], to the extent that PCG and AI have enabled experiences and interactions that were not possible before [22–24]. Moreover, as one of the properties of PCG is to increase replayability by creating an abundance of well-made content [25], 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 [26], or a step towards open-endedness Evolutionary Computation (EC) [27].

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 [28]. This is especially relevant when the aim is a creative work such as games, where the creative expression needs to be fostered [10]. 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. [29]". 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 [30–32]. 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* [33, 34]. 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 for rogue-like and adventure games such as Zelda [35] or The Binding of Isaac [36]. In EDD, the human designer can 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. The human designer can interact in several ways with the computational designer so that this adapts its output to whatever goal the human designer has, while still providing a diverse amount of alternatives and different experiences to the human designer.

Research Questions

As motivated thus far, this thesis focuses on exploring different approaches for procedurally generating content for games or other creative content, 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 [11], 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 quality-diversity algorithms into a mixed-initiative approach to help designers produce high-quality con-

tent and foster their creativity while allowing them to control, to a certain extent, the generated content?

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 [37]. 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 [8]. 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 some 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 differently, producing data on how they both interact. Research areas such as Experience-driven PCG [38], player modeling [39, 40] or designer modeling [33], explore the use of such data to understand particular users [33, 41, 42] and to improve and enhance the experiences of players and designer. Especially focusing on enabling adaptive experiences [43] and more accurate heuristics [44–46]. 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 [34, 47]. 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 its 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 [2, 5]. 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 creative artifacts, which in this thesis focuses on games, 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.

Furthermore, RQ2 and RQ3 drive the research on how to gather and use different types of data, i.e., player and designer data. Whereas, designer modeling could be used in the MI-CC feedback loop to create adaptive experiences. Through RQ3.1 and RQ3.2, the thesis focuses on exploring the trade-off of using designer modeling. Specifically, the interest lies in the 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 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?

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 (or group of both), 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 [1].

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 [19]. Content can be divided into game facets: audio, visuals, narrative, levels, rules, and gameplay [48], and have been categorized within the PCG field as *Game Bits*, *Game Space*, *Game Systems*, *Game Scenarios*, *Game Design*, and *Derived Content* [49].

There are plentiful of commercial games that utilize one way or another PCG such as The Binding of Isaac [36] or Civilization [50], to the point that some games rely critically on these algorithms, providing experiences otherwise not possible such as Rogue [23], Dwarf Fortress [51] or AI Dungeon [22]. However, there has been an increasing interest in PCG during the past decade in academia [52]. Furthermore, there exist multiple approaches addressing different challenges in the creation of content, resulting in algorithms that can autonomously create game rule's [53, 54], narratives [55, 56], levels [57–59], graphics [60, 61], and audio [62, 63]. Other approaches have focus on creating content in multiple facets aiming at creating intertwined content [64–68], and others on creating complete games [53, 69, 70].

Broadly, 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 [71]. 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 constructed [72–74]. 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 [75, 76]. 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 [43, 77].

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 any of the

aforementioned processes [78, 79]. Other examples are techniques within the PCG via ML approach [80] such as approaches to repair unplayable generated content [81] or generating content using learned probabilities from sample content [82]. 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 [83]. 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 as well used 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 [84], rules [54] or weapons [85]. 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 [1]. However, evaluation functions come in all shapes and sizes, and they are all valid with their own set of ups and dows. For instance, they might come from game design concepts such as design patterns [86] or game level metrics [45], or from aesthetic indicators such as symmetry [44], or subjective evaluation from users [87], or even continuously adapting the

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

evaluation based on gameplay [43] or to the designer's preferences [47].

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 [37, 88]. Through this, QD algorithms seek to generate a collection of high-performing solutions that are as diverse as possible[2]. 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 [89, 90]. From that moment onward, several divergent search algorithms have been proposed, such as surprise search [91], as well as variations to novelty search such as constrained novelty search [92] or Novelty Search Local Competition (NSLC) [93].

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 convergence search for preserving the high-performing individuals within the novel niches [93]. 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 [94, 95] and robotics [96]. 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 [97].

One major challenge with MAP-Elites is the *curse of dimensionality*,

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

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 [98] or Cluster-Elites [99]. 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 [100]. 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 [101].

Moreover, within the field of games and PCG, QD algorithms have started to be used extensively, especially MAP-Elites [83]. MAP-Elites has been used to create and find levels with just the right difficulty for a set of agents [102], to balance and create decks in hearthstone [95], or create levels for puzzle games through crowdsourcing [103]. Constrained MAP-Elites introduced by Khalifa et al. [104], combines MAP-Elites with the Feasible-infeasible Two-Population (FI-2Pop) algorithm [105], 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 [94, 106], and was combined with interactive evolution resulting in the Interactive Constrained MAP-Elites [107].

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 [108–111]. 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.

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 [112]. 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 [2]. 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 [1]. 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 [113]. 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 [4, 5].

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 [7]. 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 [8, 114].

Germinate [115] is a MI-CC system to co-create rhetorical games using the constraint-based game generator Gemini [75] 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 [116] 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 [117]. Their approach is the first implementation of the work by Khalifa et al. that introduced a new approach to create content: PCG via RL [118]. 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 [119]. 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 [92]. 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 [120]. 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)

framework[3] [121] and Video Game Description Language (VGDL) [122], rather than individual game content [123]. 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 [124]. 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 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 [125].

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" [126]. 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 [127] (a Super Mario Bros. [128] clone for AI research[4]) through turn-taking phases between designer and computational designer [129]. 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

[3] http://www.gvgai.net/

[4] Ahmed Khalifa is the current mastermind behind the Mario AI Framework: `https://github.com/amidos2006/Mario-AI-Framework`

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 [130], a MI-CC system to create dungeons in the game Legend of Grimrock 2 [131], where the computational designer acts as a colleague working in lockstep. Like 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 [78], 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 [79].

Recently, MI was proposed to be used in the setting of teaching young children handwriting in a tool called Djehuty, which leverage the use of technology in developing countries to foment literacy. Djehuty continuously generates handwriting styles and suggest them as paths to the child [132]. Djehuty is another example of MI's strengths, and as described above, MI can be used virtually in any collaborative scenario where agents can leverage in their strengths to contribute to a solution proactively.

This thesis revolves around the Evolutionary Dungeon Designer (EDD), a MI-CC tool to co-create dungeon levels that uses design patterns to provide information to the designer and to drive the generation of sug-

gestions for the designer [107]. EDD uses the Interactive Constrained MAP-Elites QD algorithm to continuously suggest adaptive, diverse, and high-performing solutions [133]. 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 being adopted by the game industry yet. 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 [86, 134]; 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 [4], 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 [135, 136]. 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 [40, 137], identifying player types (using Bartle's taxonomy [138]) based on their playstyle [41], to understand and model in-game player's motivations [42], or for market purposes, to understand how players play and are engage in free-to-play games [139–141].

Using player data from *Iconoscope*, a freeform creation game for visually depicting semantic concepts, Liapis et al. trained and compared several ML algorithms by their ability to predict the appeal of an icon from its visual appearance [142]. Furthermore, Alvarez and Vozaru explored personality-driven agents based on individuals' personalities using the *cibernetic big five model*, evaluating how observers judged and perceived agents using data from their personality test when encountering multiple situations [143].

Moreover, Yannakakis and Togelius discussed how the player experience could be modeled and used to drive the generation of new game content, and in this way, create content that is adapted to the experience and expectations of the player [38]. 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 [42], 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 [41]. Melhart et al. take these approaches one step further by modeling a user's *Theory of Mind* in a human-game agent scenario [144], 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 [145]. In the same line, goal thirteen in the guidelines for Human-AI interaction [146] 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. [33, 34] 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 fostering their creativity.

As part of this thesis work, two approaches to model different designer's processes have been proposed, the designer's preference model [147], and design style cluster together with designer personas [148]. The work presented in [147] 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 [148] 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) [149]". 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 [29, 150–152]. Moreover, Computational Creativity is a multidisciplinary field that studies computational systems that demonstrate human-like creative behaviors [153]. 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 [29].

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* [9]. As described above, game content relates to the main facets that represent any game: audio, visuals, narrative, levels, rules, and gameplay [48]. 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 [154]. 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 [69]. 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 EAs, they can generate levels satisfying some metric that drives the generation process [155].

Moreover, Mikkulainen discuses 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 [156]. 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 [157]. 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 [158] and Darwinian principles of natural selection and evolution of population over generations to primary 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* [159].

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 producing and encountering high-performing individuals and cre-

```
[.,.,.,.,.,.,.,.,.,.,5,#,#,#,
 .,.,4,4,4,.,.,.,.,.,.,#,#,
 .,.,4,4,4,.,.,#,.,.,3,3,#,
 5,.,4,4,4,.,#,#,#,3,3,.,.,
 .,.,.,.,.,#,#,#,#,#,.,.,.,
 .,.,.,.,.,.,#,#,#,.,.,.,2,
 .,.,.,.,5,.,.,#,.,.,.,2,2]
```

(a) Genotype

(b) Phenotype

Figure 1: Example of direct encoding

ating 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 [159].

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 [100]. 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 [160, 161]) 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 [162–164].

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 [165]. More work and discussion on this is presented in section 2.2.

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).

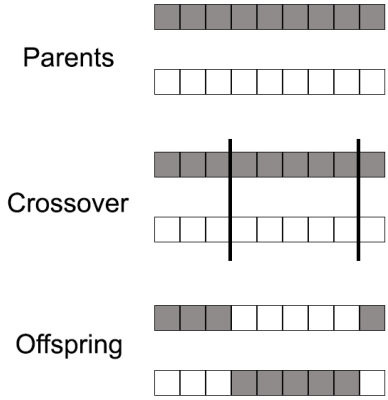


Figure 2: Example of the crossover variation operator

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 3.1.2, not all EA work the same. Some have local competition with individuals with similar behavioral novelty [93]; other approaches have competition among phenotypical similars to preserve innovations encountered in the search space [166].

MAP-Elites

Quality Diversity algorithms are a relatively new family of algorithms that leverage the strengths of divergence and convergence search [37]. 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 [96]. 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 [97].

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 selection[5] 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 need to compete if occupied with the current occupant or if unoccupied,

[5]Gravina et al. have experimented on using other selection approaches with beneficial and exciting results [167].

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 [97].

```
procedure MAP-Elites Algorithm (simple, default version)
    (𝒫 ← ∅, 𝒳 ← ∅)
    for iter = 1 → I do
        if iter < G then
            x′ ← random_solution()
        else
            x ← random_selection(𝒳)
            x′ ← random_variation(x)
        b′ ←feature_descriptor(x′)
        p′ ←performance(x′)
        if 𝒫(b′) = ∅ or 𝒫(b′) < p′ then
            𝒫(b′) ← p′
            𝒳(b′) ← x′
    return feature-performance map (𝒫 and 𝒳)
```

Furthermore, MAP-Elites have become popular and attractive due to all the abovementioned benefits and characteristics, which have not only spread it's use in many fields and many experiments but also sparked many variations. The Constrained MAP-Elites by Khalifa et al. [104] 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 [168]. Formally (and generally), learning in ML was operationally defined by Mitchell [169] 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 to complex to be solver by fixed programs such as the creation of games [80] or playing games [170, 171]. Tasks in ML might be a *classification* task: what category k an input belongs [172]; 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 [173]; or *machine translation* task: translate an input from one language to another [174].

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 [175] or for machine translation the **BLEU** is used to measure the quality of the translated text [176]. 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 [168].

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 [177]. Self-Supervised Learning (SSL) has also been gaining popu-

larity as an approach to move away from traditional supervised training by not using human-annotated dataset to learn representations of data [178].

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 [80]. On the other hand, ML can take advantage of PCG approaches that continuously create content to increase generality, as discussed by Risi and Togelius [26]. 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 [179].

EVOLUTIONARY DUNGEON DESIGNER

This section presents the Evolutionary Dungeon Designer (EDD) research tool, all its features, and the multiple interactions between the human designer and the AI. 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 generation process and multiple designer interactions. Finally, it describes the tool's workflow.

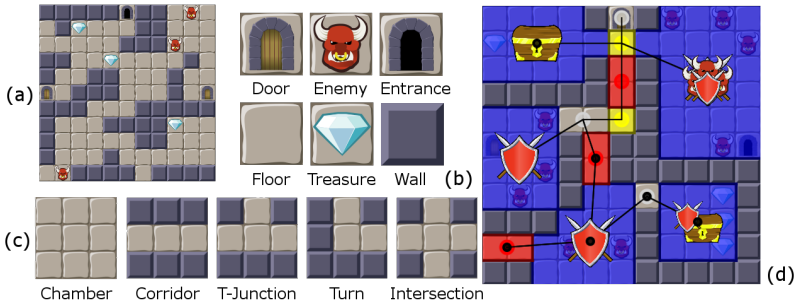


Figure 3: Main components in EDD. (a) Basic example room, (b) different placeable tiles (the boss tile looks the same as the enemy tile but 3 times bigger), (c) spatial-patterns and (d) meso-patterns

The EDD is an MI-CC system to create dungeons for rogue-like and adventure games, similar to the ones found in Zelda [35] or the Binding of Isaac [36]. The main feature of EDD is the collaboration between designer and computational designer to create rooms. The designer focuses on editing the room to design their goals. In parallel, they are continuously offered a set of suggestions adapted to their current design, which they can use. 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. This informa-

tion is given to the designer at different time steps to help them develop their design.

The first iteration of EDD was described and presented by Baldwin et al. [28, 134]. 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 being suggested four generated rooms by request. Furthermore, 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 [10, 114]. Finally, the latest version of EDD is the one presented on this thesis, which incorporates the IC MAP-Elites and allows the designer to create dungeons in whatever layout preferred [107, 133]. In addition, EDD now includes prototype implementations of different designer models: the designer preference model [147] and designer personas [148].

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

Design Patterns

An essential part of EDD is its use of design patterns to hierarchically divide a room into micro-patterns (inventorial and spatial patterns) and their combination into meso-patterns. The design patterns in EDD are based on the work by Bjork and Holopainen [180], Dahlskog et al. [181], and Dahlskog and Togelius [86]. While EDD is 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, design patterns 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.

To perform this evaluation, design patterns in EDD calculate a *quality* metric. For instance, the enemy pattern's quality is a combination of the number of enemies placed in the room and how they are distributed. Thus, if the enemies are a certain amount and are not simply added next to each other but distributed with other patterns, the "enemy quality" improves.

Micro-patterns

Micro-patterns are the minimum bits that compose an artifact, in this case, a room. Micro-patterns in EDD are divided into inventorial patterns and spatial patterns. Inventorial patterns relate to each individual passable tile; thus, each passable tile is represented by an inventorial pattern, and its aggregated quality can be measured. Spatial patterns are composite patterns based on the used space and are a combination of passable and unpassable tiles (i.e., walls).

Inventorial patterns relate to passable tiles. Yet, rather than calculating each pattern's quality based on individual tiles, they are calculated as a whole. For instance, two enemies placed in different parts of the level will have a shared quality. Inventorial patterns' quality is a trivial calculation based on a user-defined target proportion and the number of tiles of the same type in the room. Doors are a special inventorial pattern, where their quality is based on a linear combination of four measures: *door safety*, *door greed*, *average treasure safety*, and *treasure safety variance*.

Spatial patterns focus on each room's spatial characteristics and are categorized as a combination of passable tiles and walls. A combination of spatial patterns might give an indication of the type of gameplay the designer wants to create. For instance, multiple interconnected chambers in a room could indicate more small goals within a room and where combat could take place with more maneuvers.

The spatial patterns are (shown in figure 3.c): *chamber*, *corridor*, *t-junction*, *turn*, and *intersection*. Like inventorial patterns, they are also driven by user-defined targets for creating "high-quality" areas and user-defined minimums for areas to be identified as a spatial pattern. For in-

stance, a chamber is identified if there exists at least a 3×3 open area in a room. Quality is also a trivial calculation based on the expected user-defined targets and the spatial pattern's actual proportion.

Meso-patterns

Meso-patterns are the next level in the design pattern hierarchy. They are identified as pattern compositions and are defined as how micro- and meso-patterns relate. For instance, a chamber (spatial pattern) containing some enemies (inventorial patterns) would result in a "guarded chamber" meso-pattern.

The meso-patterns that are currently implemented are (shown in figure 3.d): *dead-end*, *ambush*, *guard chamber*, *treasure chamber*, *guarded treasure*. *Dead-end* is the only meso-pattern that acts as a pattern modifier rather than a composition of patterns. They are calculated by traversing the pattern graph from all the spatial patterns containing a door to all the other spatial patterns. If any spatial pattern can only be reached by one way, the spatial pattern and all the steps towards it that are not connected to other patterns are classified as dead-ends. For the rest of the meso-patterns, the crucial aspect is that their main component is a chamber (spatial pattern). Thus, these meso-patterns can be seen as a specialization of the chamber.

Ambush: Relates to a chamber containing at least one enemy and a door (inventorial pattern). Similar to others, the amount of enemies is also a user-defined minimum.

Guard chamber: Relates to a chamber containing at least two enemies (inventorial pattern) and nothing else. Similar to others, the amount of enemies is also a user-defined minimum.

Treasure chamber: Relates to a chamber containing at least two treasures (inventorial pattern) and nothing else. Similar to others, the amount of treasures is also a user-defined minimum.

Guarded treasure: Relates to a chamber, which is a dead-end, containing at least two treasures (inventorial pattern) and nothing else, but preceded by a guarded chamber. Similar to others, the amount of treasures is also a user-defined minimum.

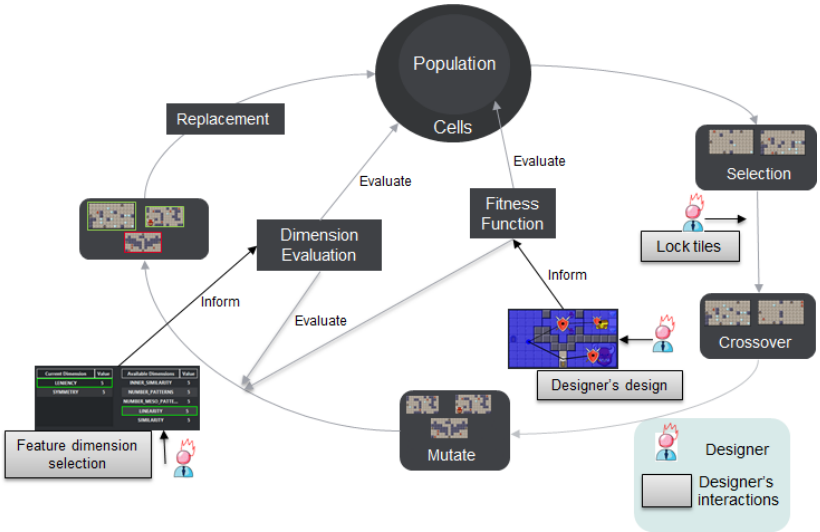


Figure 4: IC MAP-Elites evolutionary loop showing the possible designer interactions and what part of the loop these interaction affect

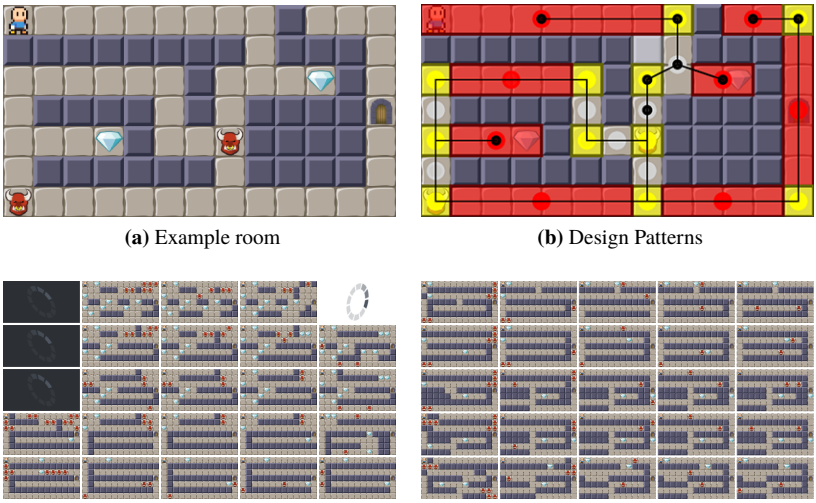
Room Generation

The main feature of EDD is to suggest variations of the designer's work adapted to their design, are 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 [11]. The suggestions have the main characteristic: they 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 section 5.1) to generate and suggest rooms to the designer. Its main features are the use of divergent and convergent searches, the use of behavior feature dimensions, and the designer's ability to interact with them. 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.



(a) Example room

(b) Design Patterns

(c) Meso-pattern and Leniency

(d) Symmetry and Leniency

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.

Evolutionary Components

IC MAP-Elites is an extension of the Constrained MAP-Elites, which uses a FI-2Pop and MAP-Elites at its core. In our implementation of IC MAP-Elites, a solution is deemed infeasible if the playability constraints are not satisfied, i.e., there is a passable tile that is not reachable from at least one door. If this happens, the solution is placed in the respective infeasible population to evolve into a feasible one encouraging diversity in the population.

Moreover, EDD implements seven behavior feature dimensions (shown

in table 1). The designer can select one at a time, and only two dimensions can be active at any given time, in order to be able to present the suggestions intuitively to the designer and to focus the search on the pair of dimensions. By giving the designer this interaction, they can effectively reshape the search space of the IC MAP-Elites.

Furthermore, besides using feature dimensions to encounter diverse solutions, EDD uses a single-objective fitness function to calculate the generated rooms' fitness. The 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 refers to the distribution between corridors and rooms, and the meso-patterns that those encompass. The fitness adapts to the user's current design, automatically informing target ratios and distributions to be achieved by the EA.

The fitness function is shown in Eq. 1. $f_{inventorial}$ is the evaluation of the aggregated and normalized quality of treasures, enemies, and doors (inventorial patterns). Quality refers to positioning, safety, and the relation between inventorial patterns. $f_{spatial}$ refers to the quality and distribution of chambers, i.e. open areas in the room and corridors, and the meso-patterns created within chambers.

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

Designer Interaction

One of the objectives of this thesis, and a significant feature in EDD, is the control and interaction designers have with non-intuitive components of the algorithms. In EDD, the designer can interact with the IC MAP-Elites in three different ways: 1) *Locking tiles*: A special tile that locks tiles in the room to be unchangeable. 2) *Feature dimensions*: The designer can change the feature dimensions of the IC MAP-Elites at any time, reshaping the search space. 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.

Locking tiles: Besides the set of tiles provided to the designer to edit the room, they are given a modifier that allows them to lock tiles in their design, which establishes what tiles or structures must be preserved in

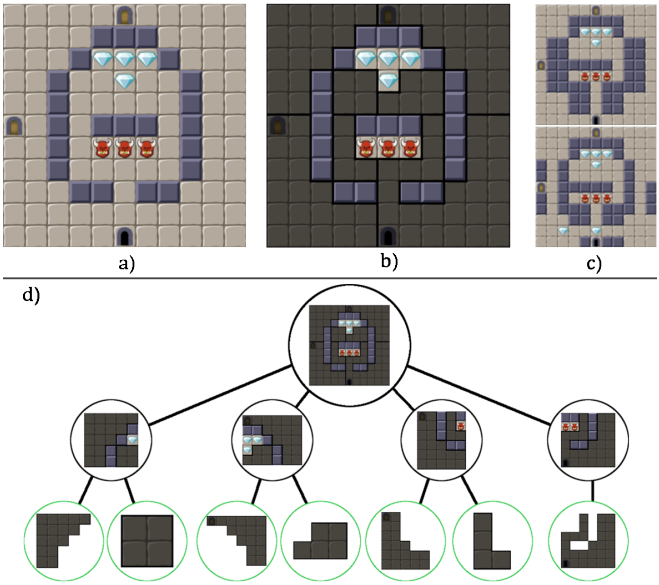


Figure 6: Locking Tiles interaction: (a) A sample edited room with (b) its division into zones based on the tiles locked by the user.(c) Suggestions preserve these locked tiles. (d) The room and its zones are internally represented with a tree structure, where the leaf nodes (green) are the valid genes to apply variation operators.

the suggestions. By locking tiles, the designer is effectively constraining the building blocks IC MAP-Elites have for applying variation operators, as mutation and crossover will only happen on non-locked tiles. However, this also allows IC MAP-Elites to focus on the areas that the designer is not interested in, segmenting the room's design and promoting the colleague role. Figure 6 shows an example of this interaction and how the genes are segmented in the EA.

Feature Dimensions: A major feature in IC MAP-Elites that differentiate the approach to other MAP-Elites variations is the interaction designers have with the algorithm. Feature dimensions are an essential component of MAP-Elites, used to discretizes the behavior space as a grid of cells, illuminating such space. Through this, MAP-Elites can explore the space more vastly by retaining high-performing solutions in cells informed by the feature dimensions encountered in different parts of the behavior space. Furthermore, The designer can interact with these feature dimensions, choosing those of most interest to them. By doing this, the search and solutions encountered are conditioned to the designer's interest. Changing the dimensions reshapes the behavior characteristics where encountered

individuals in the search space will be retained.

Current Design: EDD uses an adaptive fitness function based on the designer's design by analyzing the room configuration based on the design patterns previously described. This allows the designer to interact indirectly with how IC MAP-Elites estimates the quality of the encountered solutions. Therefore, to achieve high-fitness, the generated content will need to have certain similar characteristics than the current design, in order for the content to do not be too disparate from the designer's goal. However, while this might limit the tool's expressivity, IC MAP-Elites use of feature dimensions and leverage on convergent and divergent searches still yields diverse solutions.

Finally, in figure 4, it is presented an overview of the IC MAP-Elites steps and the multiple areas designers can interact.

Workflow

EDD has three views that enable the MI-CC wokflow depicted in figure 7.

World View

In figure 7(a), it is shown the world view. This view provides the designer with an overview of their dungeon while allowing them to create new rooms and edit their connections. The dungeon is a cyclical graph in code, which allows the needed flexibility for the designer to create whatever layout they want. The designer can create or delete any number of rooms and can connect them as needed. By editing these connections, they edit the rooms' entrance and exit doors, effectively changing the layout and gameplay flow. For instance, the designer can create linear gameplay with rooms one after the other, or gameplay that rewards exploration through dead-ends with some side-objective.

To edit a room, the designer can: 1) double-click or scroll to zoom-in any room to change the view to the *room view* or 2) select a room and click the "Suggestions" button, which will take them to the *suggestion view*

Suggestion View

In this view (shown in figure 7(b)), the designer is proposed six variations to their design based on six different preset room configuration: Favoring open areas, favoring corridors, a mix of open areas and corridors, favoring dead-ends, favoring meso-patterns, and using the actual room configura-

tion. The designer can then go back to the world view or choose any of the suggestions and start editing their room using that as a base in the *room view*.

Room View

The room view (shown in figure 7(c)) is the main view in EDD, where both the designer and the computational designer interact to co-create rooms. As mentioned above, the designer has five different tiles at their disposal to create a room that fulfills their objective. Besides this, the designer can enable the "patterns visualization" that overlays on the designer's design what spatial- and meso-patterns are identified.

EDD incorporates a feasibility and playability check on the designer's design. This informs the designer if an area in the room is inaccessible from within the room or through another room by adding a red border in their design and an information box. This feasibility check is reused in the EA used by the computational designer to generate suggestions for the designer.

Moreover, in this view, the designer edits the room as they want, while on the right side of the view, they are suggested a collection of rooms. Whenever a suggestion is selected, a quantitative comparison between the suggestion and the current design is presented to the designer. If the designer decides to replace their design with the suggestion, they apply it and continue the design process. The system aims to give suggestions to the designer that are interesting for them, aligned with their current design, but at the same time different enough to foster their creativity. Designers might not apply any suggestion, but a designer might be interested in doing something similar by observing them.

Once the designer finishes editing their room, they can go back to the world view and start another room's design process until they finish their dungeon.

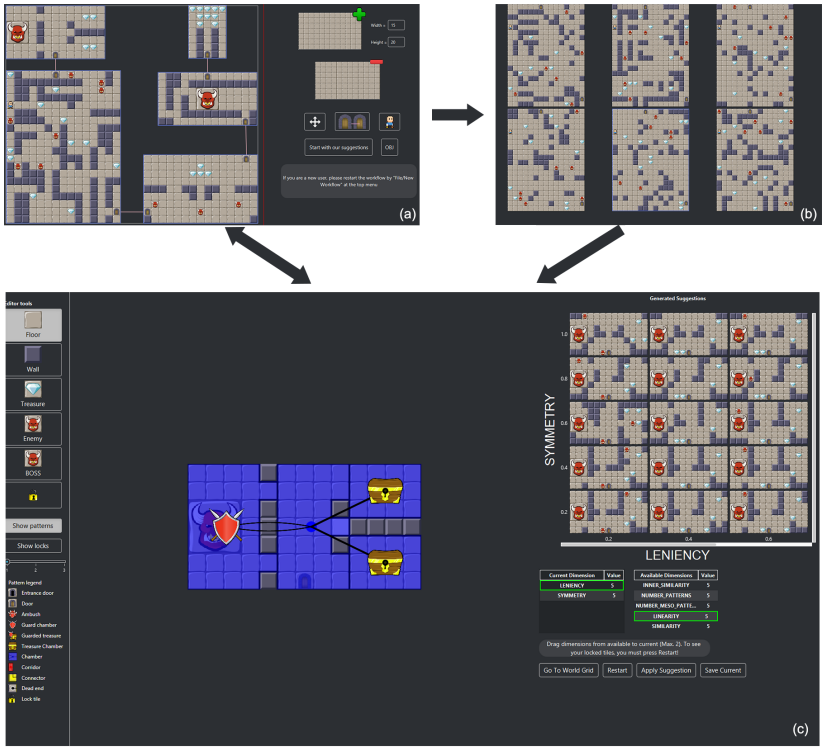


Figure 7: Workflow of the Evolutionary Dungeon Designer. (a) Shows the world view with a prototype dungeon created. (b) Shows the suggestion view where designers can receive instantly six suggestions to start their design with, rather than from scratch. (c) Shows the room view, the main interactive view of EDD. In this view the designer can edit their room while the computational designer is offering suggestions for their design.

ALGORITHMS

Interactive Constrained MAP-Elites

The Interactive Constrained MAP-Elites (IC MAP-Elites) [107, 133] is the main algorithm developed in this thesis, which focuses on adapting the Constrained MAP-Elites to be used in an MI-CC system allowing designers to interact with it, and change non-intuitive aspects of MAP-Elites. The main novel characteristics of IC MAP-Elites are:

- MAP-Elites adapted to be used in a MI-CC system to collaborate and interact with the designer.
- Use of MAP-Elites to create dungeons and adventure levels.
- Continuously adapting the fitness landscape and search space based on the designer's creation, which in turn, makes MAP-Elites continuously adapting to a changing space.
- Customizable behavioral feature dimensions as part of the designer's ability to interact with the system. By changing the feature dimensions, pressure is imposed in the algorithm to adapt the search as the behavior dimensions are no longer the same (i.e., the niches changed).

In terms of the EA components, our implementation of IC MAP-Elites in EDD uses direct encoding similar to the one depicted in figure 1, selection is through competition of individuals in each cell population, and selected individuals are applied crossover and mutation. Finally, the replacement strategy is elitist; if any offspring is better than individuals in their cell, these are replaced.

Moreover, there are three ways designers can interact with IC MAP-Elites: 1) By editing their design, which automatically updates and adapt

the fitness function. 2) By changing behavior feature dimensions and the size of cells, which is reflected in the algorithm's search space. And 3) By locking areas of their design to be preserved in the evolutionary run, which limits the building blocks IC MAP-Elites has to apply variation operators to, but preserves the designer's structures. This last interaction was developed earlier than IC MAP-Elites but is still available for the designer, is documented in [114], and presented in section 4.2.2.

Algorithm 2 Interactive Constrained MAP-Elites

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

In terms of execution, IC MAP-Elites is very similar to the vanilla MAP-Elites and Constrained MAP-Elites with three main differences: 1) The evolutionary run never ends and continuously update and adapts the fitness function based on the designer's current design, and through this, populations within cells might change their order, putting new pressure in

individuals. 2) After *n* amount of generations, the designer is presented a collection of suggestions in the same MAP-Elites grid used in the search. When this happens, all individuals in all cells are selected, cloned, and minimally mutated to promote diversity. 3) At any moment, the designer might decide to replace their design with a suggested one, which in turn drastically change the algorithm, since rather than adapting the fitness gradually, the change might directly make high-performing individuals low-performing. However, one of the benefits of MAP-Elites is it's fast convergence; thus, the algorithm adapts fast to the changes. And 4) the designer can change the feature dimensions (a pair at a time) and the amounts of cells per dimension to create bigger or smaller niches. Through this, the designer is effectively reshaping the behavior characteristics where encountered individuals in the search space will be retained, thus changing non-intuitive parameters in intuitive ways. The algorithm is depicted in listing 2, and some examples of the MAP-Elites grid presented to the designer are shown in figure 8.

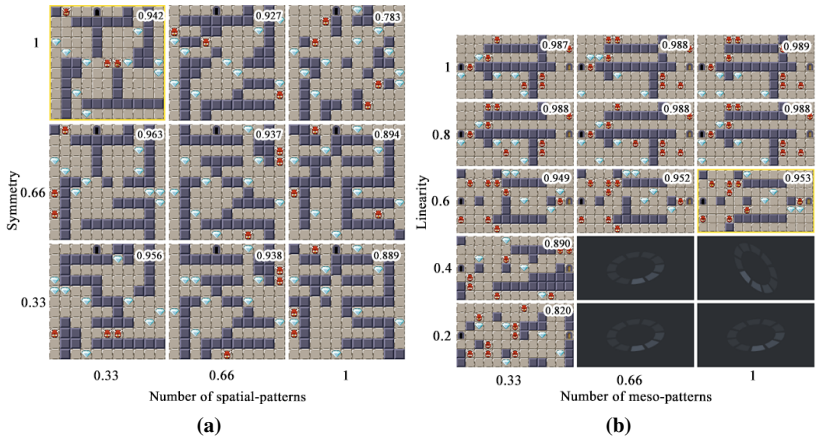


Figure 8: Examples of suggestion grids presented to the designer as they design different rooms.

As highlighted, behavioral feature dimensions are one of the main components of MAP-Elites and all of its variations. They are context-dependent similar to fitness functions, with the caveat that they are only used to categorize and place individuals in cells. The features currently available in our implementation of IC MAP-Elites in EDD are presented in table 1 along with a description of each dimension, and an example of each dimension and multiple cells in each dimension is shown in figure 9.

Finally, Mouret and Clune discuss that a traditional single-objective

Table 1: Developed game based features used as dimensions in the Interactive Constrained MAP-Elites

Feature	Definition
Similarity	Refers to the aesthetic (tile-by-tile) similarity between a room and the current designer's design.
Inner Similarity	Refers to the similarity of the sparsity and density of the different tile types of a room designer's current design.
Symmetry	Refers to the aesthetic symmetry of a room.
Leniency	Refers to how challenging rooms are; calculated based on the position of enemies and balance between enemies and treasures.
Linearity	Refers to the amount of paths connecting doors within a room; calculated based on how many spatial patterns are traversed.
#Meso-Patterns	Refers to the number of meso-patterns that exist within a room, normalized by an estimated maximum number based on the room's size and the minimum chamber size.
#Spatial-Patterns	Refers to the number of spatial-patterns that exist within a room, which can be chambers, corridor, turns, junctions, and intersections.

approach could reach as good fitness in an individual for some of the tasks they experimented [97]. Nevertheless, 1) this would only focus on obtaining one high-performing individual, 2) the search would focus in fewer places of the generative space, and as a consequence, and 3) the diversity of the generated individuals would be scarce. The nature of our mixed-initiative tool not only requires but promotes the identification of multiple solutions that would satisfy similar constraints. For instance, a linear room could be one with narrow corridors connecting doors or a simple open chamber containing all doors. Therefore, a rich, diverse, and high-performing set of levels to be suggested to the designer that could be generated in a short period, and that can adapt to changes over time is a key necessity for EDD.

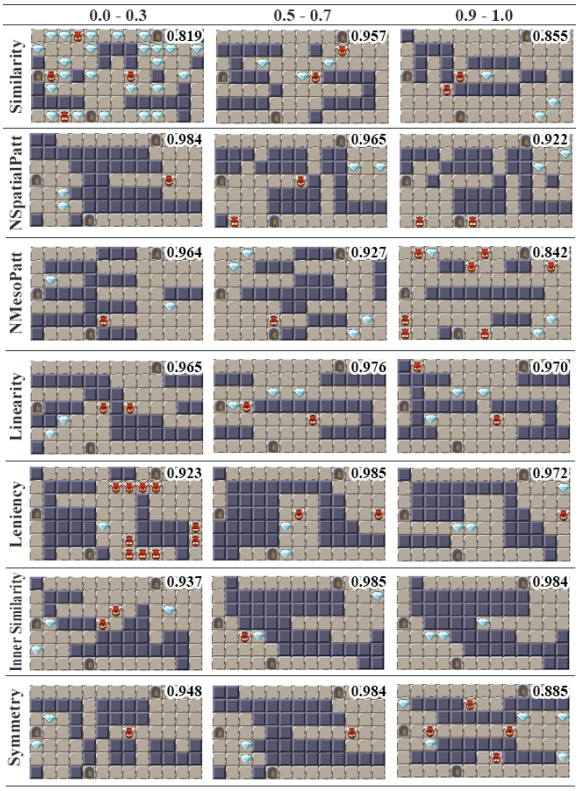


Figure 9: Example of Elites generated using IC MAP-Elites. Each row represents an independent run of the algorithm using the dimension specified to the left. Each column splits the dimension score into three intervals: 0.0-0.3 (low), 0.5-0.7 (medium), and 0.9-1.0 (high). Each cell displays (top-right) the fitness of the optimal individual in its related interval

Designer Preference

To explore designer modeling and its benefits, the designer's preference model was proposed to drive the content generation in IC MAP-Elites [147]. In this work, the suggested MAP-Elites grid is used to compose a dataset used as an estimator of the designer's preferences. The goal and motivation of modeling the designer's preferences were to use a surrogate model that could capture the designer's subjective evaluation and an objective evaluation from the fitness function. The model, the training, and the usage is depicted in figure 10.

Following a proactive learning approach [182], whenever the designer chooses some suggestion, the system trained a learning model M (i.e., a neural network) with a new set of preferred content (dataset S) using the

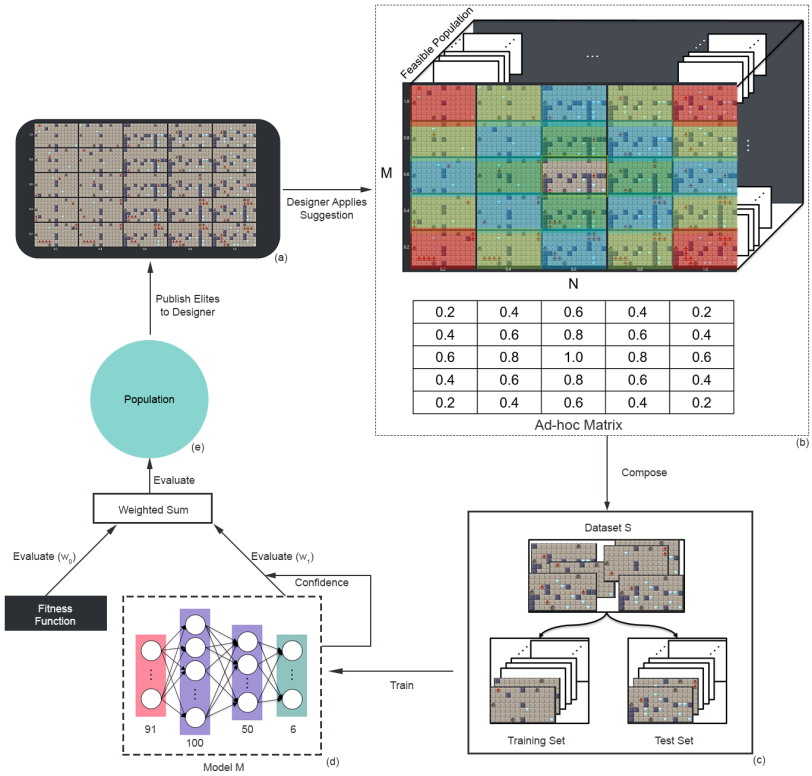


Figure 10: Overview of the Designer Preference Model integrated into the fitness function of EDD. Elites are published and shown to the designer in a grid fashion (a). Once the designer chooses and applies one of the suggestions, an ad-hoc matrix is created based on the selected suggestion's position 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. The weight of the model M conditioned by the confidence of the network (e).

current suggested cells and their population. The result was an adapted model similar to the work by Liapis et al. [47], which learned overtime the designer's preferences in relation to their choices. Once the model is trained, it is incorporated in the evolutionary loop to drive evolution by means of evaluating the content in a weighted sum with the fitness function. The model slowly fits towards the designer's preference, and as it's predictions become more confident, the more weight W_1 it has in the final evaluation. Confidence is calculated based on the softmax layer's output, which can be interpreted as probabilities for each class. The evaluation resulted in the weights (Eq. 2) and the final weighted sum (Eq. 3).

$$\begin{aligned} w_1 &= \min(M_{conf} \cdot M_{TestAcc}, 0.5), \\ w_0 &= 1.0 - w_1 \end{aligned} \tag{2}$$

$$weightedSum = (w_0 \cdot objective) + (w_1 \cdot predicted_{pref}) \tag{3}$$

We evaluated the model and its usability in a user study with fifteen game design students. While not significant as there was not enough data to demonstrate the advantage of the model, the results allowed the analysis of the system's challenges. These challenges were presented as three open areas for active research: *Dataset*: the type and amount of data collected; *Preference modality*: what represents preference in the design and creative process; and *Dynamic-Dynamic System vs. Dynamic-Static System*: the competing properties of a dynamic learning environment, e.g., a designer that traverse the design space, with a dynamically adapting model, e.g., a learning model that tries to adapt as the designer traverse the design space, and it's trade-offs.

Style Clustering and Designer Personas

To further explore designer modeling, an alternative approach was proposed by clustering designers' design styles [148]. The basic premise was that by identifying a set of styles that most designers follow when creating content, and model how the designer traverse such a style space, a model that captured the designer's intentions, goals, and style could be created. For instance, rooms for a game like the binding of Isaac [36] could be classified based on multiple characteristics such as the room's objectives

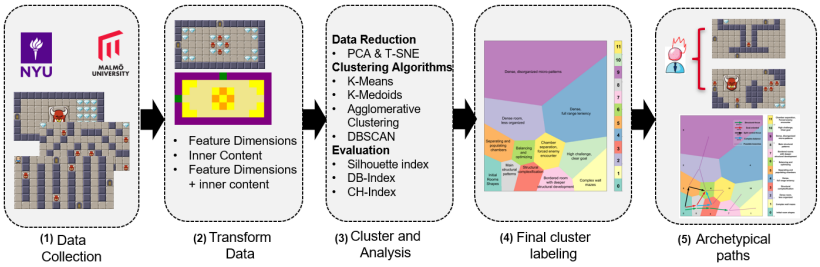


Figure 11: The stages of the design style clustering development: (1) Data was first collected through two user studies. (2) Then, using the design sequences, the data was processed into five different datasets, one using the room images, a second using the tiles information, and three using tabular 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.

regarding enemies and treasures, access to different areas, or hidden challenges 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 topology before anything else, whereas others would focus first on the objectives a player must achieve.

With such as motivation and objective, it was compiled data of designers' designs and their creation process in the EDD to create these models. The data was collected from two user studies, the one documented in the Designer's Preference Model previously described [147] with game design students, and another with practitioners and academic researchers within the computational intelligence in games area. The data was used to create five different datasets that were used in the experiment to analyze multiple variations and possibilities. The datasets were analyzed, experimented on, and compared to obtain the final clusters that effectively divided the design style space. The process is shown in figure 11. Figure 12 shows the final design style clusters with the setup that performed best among all the different experiments (K-Means (K=12), using the **Tiles** Dataset).

Moreover, with the design style cluster, we could then analyze the design process in function of the traversed clusters rather than evaluating each individual step. To do this, we analyzed the traversed clusters as unique trajectories. We gathered the common patterns from the trajectories by

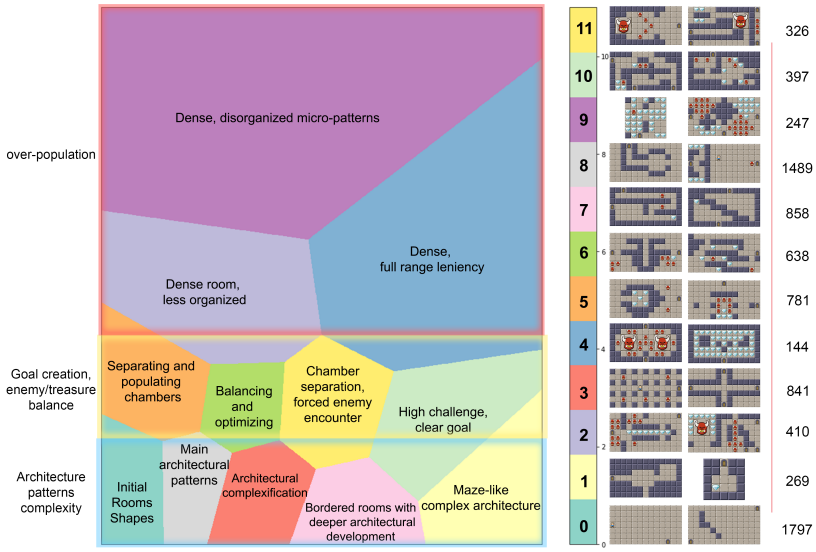


Figure 12: Best resulting cluster set. K-Means (K=12), using the **Tiles** Dataset. While it scores slightly less in the internal indices that other setups, a qualitative analysis successfully gives us more granularity by subdividing the main bottom clusters to label and cluster designers' design process. Sample rooms belonging to each cluster are displayed on the right, next to the total number of rooms in the cluster.

applying the Generalized Sequential Pattern (GSP) algorithm, which locates frequent subsequences in the analyzed trajectories. Through this, four *designer personas* were identified. *Designer personas* are defined as archetypical paths through style space that are commonly taken by designers when creating their content. In figure 13 it is shown the identified *designer personas*.

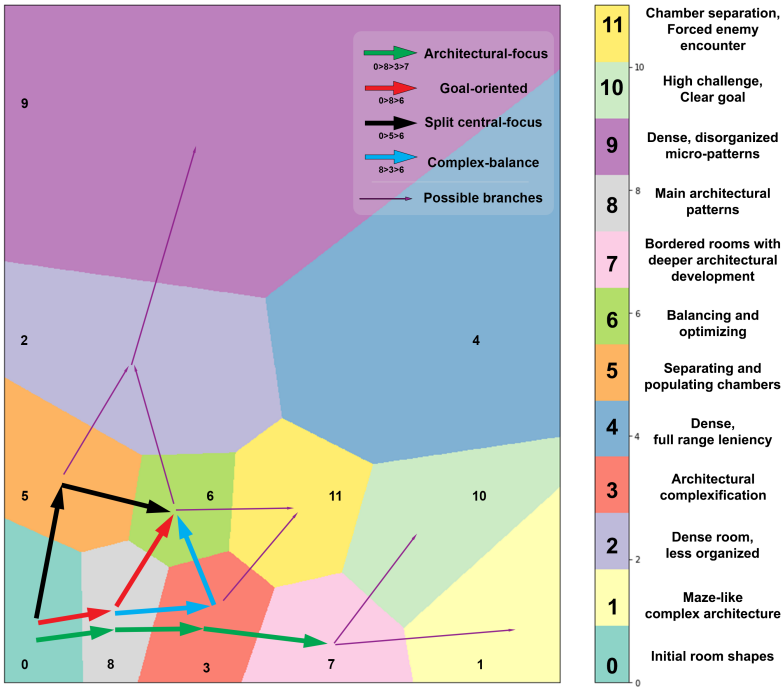


Figure 13: Final and common designer trajectories. With thick arrows it is presented the archetypical paths, calculated using the frequencies of subsequences from 180 diverse rooms. Each color represent a unique trajectory; with green the Architectural-focus, with red the Goal-oriented, with black the Split central-focus, and with blue the Complex-balance. Finally, thinner purple arrows extending from clusters traversed by the archetypical paths show the multiple possible branches that an archetypical path can deviate or extend to.

RESEARCH METHODOLOGY

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

To explore the multiple research questions presented in section 1.2, the various MI-CC processes and the properties that arise in them, this thesis follows a Design Science Research Methodology (DSRM) [183]. 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 [183].

Through the thesis, two artifacts have been under parallel development. The first one, more related to the definition of **instantiation**, is the Evolutionary Dungeon Designer, the main research tool where the interaction between human designer and computational designer is investigated. 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 test systems that aim at improving and exploring the co-creative, co-creation, and co-design capabilities of MI-CC. As well as setting the environment that is required to support the development and creative expression of human designers.

The second artifact is 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 a specific problem such as aesthetic consideration of the designer (PAPER II), while others addresses several RQs such as developing designer models (PAPER VI).

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 [183]. Thus far, the focus has been on doing analytical, experimental, and descriptive methods, with a mix of quantitative and qualitative data collection.

Both EDD and the computational designer 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 and reactiveness of the algorithms. The *dynamic analysis* focused on performing expressive range analysis (PAPER III), which is an evaluation tool to examine the variety of the generated content and how different setups have an impact on the algorithm [184]. Expressive Range Analysis is a popular method for content generators, and it is used to investigate the expressiveness of the artifact and to use it as a comparison method with other algorithms and similar tools [25, 185, 186]. The *simulation* has focused on evaluating how the tool responds to different inputs and the robustness of algorithms with noisy and unpredictable data. For instance, in PAPER III and PAPER VI, the conclusions, and evaluations

were mostly based on simulations.

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 PAPER I and PAPER VI.

Finally, Hevner et al. [183] 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, in PAPER VI, it was developed a model informed on data from user studies that resulted in the partitioning of the design space into design style clusters. We further used this to explore archetypical design paths, namely 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 [187].

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) [188] since, after all, the iterative design of multiple artifacts and the investigation of how this open 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 [189].

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 [190]. 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 [190].

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 PAPER V, 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 2: Relationship between the different research questions and the publications

RQ	Papers
RQ I	I, II, III
RQ II	I, IV, V, VI
RQ III	V, VI
RQ III.I	V, VI
RQ III.II	V

RQ1: How can we use and integrate quality-diversity algorithms 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. As well as 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 a substantial part of the research into 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 [28], 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 expressive power 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.

Moreover, built on top of the Constrained MAP-Elites by Khalifa et al. [104], in PAPER III, we introduced the Interactive Constrained MAP-Elites, the first use of MAP-Elites in a mixed-initiative setup (see section 5.1). 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 (presented in table 1). This resulted in increasing diversity in the exploration of the search space while retaining high-performing solutions. Furthermore, it was established a control mechanisms of non-intuitive aspects of the EA for the designer through controlling the feature dimensions that discretize the search space.

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 [191] or their experience [38] 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 [33].

Therefore, we have delved into collecting, analyzing, and using player and designer data, reported in PAPER I, IV, V, VI. Thus far, we have conducted three user studies: 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 V 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. Finally, the third user-study is reported in PAPER VI, which consisted primarily of increasing the scope of the study in PAPER V with data from a more diverse and wider group.

In PAPER IV, we collected personality scores from several players using the *cybernetic big five personality test*. We used the scores to model AI agents that then had to engage in particular situations such as jumping a gap or going around it. Through this, we focused on agents that could not only resemble the decision-making of their human counterpart, but that could have complementing characteristics. Moreover, PAPER V and PAPER VI focused on collecting and using designer data, specifically their actions within EDD with the aim of modeling designer processes.

With the data collected and the techniques employed, it was possible to analyze certain players' and designers' actions and characteristics. In the case of PAPER IV, the similarities presented between agents and humans helped us identify characteristics that could be valuable to model for creating adapted content for the end-user. While in PAPER V and PAPER VI, the data collected corresponding to the designer's actions, not only allowed us to create models representing certain designers' procedures but also highlighted interesting design processes.

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 V and PAPER VI explore such a paradigm, where we proposed multiple approaches to model different but related processes.

The approach presented in PAPER V 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 V, drove the approach presented in PAPER VI, 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 it's final version. By clustering this design process, we were able to identify twelve representative clusters (shown in figure 12). 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 V was presented, implemented, and tested. While the designer personas and design style clusters in PAPER VI 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.

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

PAPER V and PAPER VI 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 V 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 andor 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 V. 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 VI.

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 V, 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 VI 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 V, 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 VI. 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 V, or to indicate possible steps the designer

might take as in PAPER VI. However, having these constraints also limits, to some extent, the generative space and expressive range of the AI. In PAPER V, 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.

CONCLUSIONS AND FUTURE WORK

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

This thesis explored MI collaboration between human designers and AI for the co-creation of games in EDD, a MI-CC system. 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. Furthermore, 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 constraint the output of other agents based on some criteria. *Expressivity* relates to the diversity of solutions that can be created by either agent.

One of this thesis's objectives was to develop algorithms that could collaborate with designers, giving them a varying degree of control through control mechanisms while still being expressive (RQ1). Three control mechanisms were introduced where the designer had direct and indirect control over non-intuitive parameters of the EA. These were: *Locking tiles*,

designer's design, and *feature dimensions*. In PAPER I, an explorative study was carried out to evaluate EDD and it's current functionalities with game developers to gather and analyze game designer's requirements and impressions. PAPER II focused on introducing aesthetic criteria to evaluate the content, and as a way for the designer to preserve their content. The locking tiles feature was introduced, which allowed designers to designate their design areas to be preserved by the EA. This allowed the computational designer to be more supportive by focusing on parts that the designer was not currently working on.

Moreover, 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 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 3.1.2). 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 are further supported by preliminary results, where the behavior and generative space of IC MAP-Elites was 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 show that the algorithm adapts adequately, and the designer has, to some large extent, an impact in 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. This result can be seen in figure 14.

In short, controllability, in many cases, is a competing property with expressivity. Since the control and constraints imposed usually limit the expressive range for any of the constrained agents. However, in this thesis, it is demonstrated that QD algorithms, specifically, IC MAP-Elites, can cope with this, showing robustness, adaptability, and stability when interacted. It was provided meaningful control for the designer, yet IC

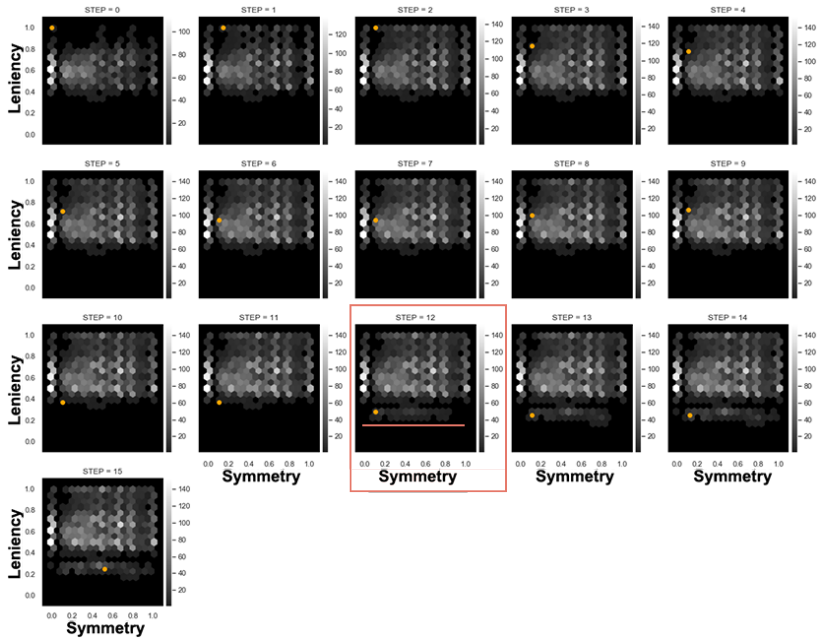


Figure 14: Aggregated Expressive Range Analysis using Symmetry and Leniency as feature dimensions. The orange dot represents the designer's design that each step is edited and moving in the generative space. In step 12, it is highlighted the step when the designer's design enters a new generative area, which helps IC MAP-Elites explore the new area.

MAP-Elites adapted and kept exploring vast amounts of the generative space encountering high-performing solutions.

Moreover, the work in PAPER I, II, II and the interest on seeking alternative approaches to foster creativity, to create adaptive experiences, and to enable more autonomy and initiative for the AI directed the research towards player and designer modeling. The former was explored in PAPER IV, where personality-driven player models were created to investigate their usability as a representative surrogate model and possible complement value in gameplay.

PAPER V and PAPER VI presented examples of designer modeling by modeling different designers' 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 V 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 VI 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 an abstract level rather than specific steps, akin to procedural personas or game design patterns.

These approaches towards 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.

Future Work

This thesis dived into the essential question of how to create tools where human designers can have a *colleague* partnership and collaboration with the AI, the properties that arise from this, and ways to enhance the collaboration, several if not more areas remain open.

For instance, RQ1 was partially addressed by introducing and implementing IC MAP-Elites in EDD. However, it is necessary to examine the behavior of IC MAP-Elites when used for this interaction. These behaviors could be: how the generative space is affected by the design sessions, the combination with designer modeling, or how dynamic feature dimensions (e.g., similarity and inner similarity) affect the expressiveness of the algorithm. PCG through QD was proposed recently [83], pointing towards the interesting avenues that are approaching, and as discussed in this thesis, the many possibilities they have when introduced in human-AI interaction

and collaboration.

Furthermore, 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 VI 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.

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 [31, 32].

Zhu et al. [30] 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 [192], where they explored visualization techniques through an interactive level designer tool called *QUBE* to explain and introduce machine learning principles to game designers.

Holistic Procedural Content Generation

An interesting and exciting path would be to explore the concept of holistic PCG and orchestration of the different game facets [48] in connection with the MI paradigm. Holistic PCG is the generation of multiple contents (in different game facets) fitting each other in harmony as a collaborative process akin to how games are developed, with a limited amount of examples, but exhibiting exciting results [64, 65, 84, 193, 194]. However, to what extent the generation is created in such a harmony that the facets interact and affect each other, and to what degree the user can interact with it is an open area for active research.

Furthermore, there exists an essential link and relation between space

(e.g., level or objects within) and narrative (e.g., the story tried to be told). Thus, choosing and associating level design and narrative as two facets to explore within the holistic PCG approach is appropriated. This was presented and described by Kybartas and Bidarra's survey [195], and explored and supported by related work [193, 196–200]. Within the narrative facet and in most games, quests are an essential component. Therefore, exploring and generating quests is paramount by exploring multiple quest concepts [201], analyzing quest patterns [194, 202], and using surrounding ideas such as kernels and satellites for event division [203].

Thus far, some exploratory work has been done combining both facets in the generative process of EDD. Firstly, through a simple but effective way of analyzing patterns and objectives created by designers when designing their levels [204]. Secondly, by introducing MI creation of quests using grammars based on the quest analysis by Doran and Parberry [205], to compose a series of subsequent objectives together with the designer [206].

Exploring Agency and Initiative in Mixed-Initiative

As paths towards adaptive experiences that recognize several designer procedures are explore, 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. However, this thesis barely scratched the surface of the relationship, the possibilities that emerge with them, and the AI's initiative and role. 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. 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. F. Allen, "Mixed-initiative interaction," in *Intelligent Systems, Trends and Controversies section*, pp. 14–16, 1999.

[2] D. Novick and S. Sutton, "What is mixed-initiative interaction?," in *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, 01 1997.

[3] 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), p. 173–175, Association for Computing Machinery, 1997.

[4] E. Horvitz, "Uncertainty, action, and interaction: In pursuit of mixed-initiative computing," in *Intelligent Systems, Trends and Controversies section*, pp. 17–20, 1999.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Mojang Studios, "*Minecraft*." Game [PC], May 2009. Microsoft Studios, Washington, US. Last played October 2019.

[14] T. Fox, "*Undertale*." Game [PC], 2015. Fox Toby, Boston, US, Last played December 2018.

[15] E. Barone and ConcernedApe, "*Stardew Valley*." Game [PC], 2016. Chucklefish, London, UK. Last played January 2018.

[16] T. Fullerton, C. Swain, and S. Hoffman, *Game Design Workshop: Designing, Prototyping, & Playtesting Games*. Gama Network, Taylor & Francis, 2004.

[17] J. Blow, "Game development: Harder than you think," *Queue*, vol. 1, pp. 28–37, 02 2004.

[18] M. Washburn Jr., P. Sathiyanarayanan, 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.

[19] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018.

[20] Hello Games, "*No Man's Sky*." Game [PC], August 2016. Sony Interactive Entertainment, Tokyo, Japan. Last played March 2017.

[21] D. Yu and Mossmouth, LLC, "*Spelunky*." Game [PC], December 2008. Mossmouth, California, US. Last played December 2019.

[22] N. Walton, "*AI Dungeon 2*." Game [PC], 2019. Latitude Inc.Last played April 2020.

[23] M. Toy, "*Rogue*." Game [PC], 1980. Epyx, San Francisco, US. Last played November 2017.

[24] D. Braben and I. Bell, "*Elite*." Game [BBC Micro and Acorn Electron], 1984. Acornsoft, Cambridge, UK. Last played February 2012.

[25] N. Shaker, J. Togelius, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[26] S. Risi and J. Togelius, "Increasing generality in machine learning through procedural content generation," *Nature Machine Intelligence*, vol. 2, pp. 428–436, Aug 2020.

[27] J. Clune, "Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence," 2019.

[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), ACM, 2017.

[29] M. Boden, *The Creative Mind: Myths and Mechanisms*. Routledge, 2004.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Nintendo R&D4, "*The Legend of Zelda*." Game [NES], 1986. Nintendo, Kyoto, Japan. Last played December 2017.

[36] E. McMillen and F. Himsl, "*The Binding of Isaac*." Game [PC], 2011. McMillen, Edmund and Himsl, Florian, California, US. Last played March 2020.

[37] 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.

[38] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] M. Hendrikx, 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.

[50] Firaxis Games and S. Meier, "*Civilization IV*." Game [PC], October 2005. 2K Games, California, US. Last played November 2009.

[51] T. Adams and Z. Adams, "*Dwarf Fortress*." Game [PC], August 2006. Last played December 2013.

[52] A. Liapis, "10 years of the pcg workshop: Past and future trends," in *Proceedings of the 11th Workshop on Procedural Content Generation, FDG*, 2020.

[53] 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.

[54] 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, 01 2013.

[55] A. Calvin and N. Michael, "The quest in a generated world," in *DiGRA &#3907 - Proceedings of the 2007 DiGRA International Conference: Situated Play*, The University of Tokyo, September 2007.

[56] 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.

[57] 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.

[58] 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.

[59] M. C. Green, L. Mugrai, A. Khalifa, and J. Togelius, "Mario level generation from mechanics using scene stitching," 2020.

[60] 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.

[61] 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.

[62] 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.

[63] A. K. Hoover, "Functional scaffolding for musical composition: A new approach in computer-assisted music composition." PhD Dissertation, University of Central Florida, 2014.

[64] 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, 04 2015.

[65] 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.

[66] 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.

[67] 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.

[68] D. Karavolos, A. Liapis, and G. N. Yannakakis, "A multi-faceted surrogate model for search-based procedural content generation," *IEEE Transactions on Games*, 2019.

[69] M. Guzdial and M. Riedl, "Conceptual game expansion," 2020.

[70] M. Cook, S. Colton, and J. Gow, "The angelina videogame design system, part ii," *IEEE Transactions on Computational Intelligence and AI in Games*, 2016.

[71] 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.

[72] N. Shaker, A. Liapis, J. Togelius, R. Lopes, and R. Bidarra, *Constructive generation methods for dungeons and levels*, pp. 31–55. Cham: Springer International Publishing, 2016.

[73] 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.

[74] 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.

[75] 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.

[76] 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), p. 221–228, Association for Computing Machinery, 2018.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] S. Dahlskog and J. Togelius, "A multi-level level generator," in *2014 IEEE Conference on Computational Intelligence and Games*, Aug. 2014.

[83] 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.

[84] 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.

[85] D. Gravina, A. Liapis, and G. N. Yannakakis, "Constrained surprise search for content generation," in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pp. 1–8, 2016.

[86] 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.

[87] 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), p. 148–156, Association for Computing Machinery, 2020.

[88] 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), p. 115–116, Association for Computing Machinery, 2019.

[89] 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.

[90] 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), p. 103–110, Association for Computing Machinery, 2010.

[91] 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.

[92] 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.

[93] 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), p. 211–218, Association for Computing Machinery, 2011.

[94] M. Charity, M. C. Green, A. Khalifa, and J. Togelius, "Mech-elites: Illuminating the mechanic space of gvgai," *arXiv preprint arXiv:2002.04733*, 2020.

[95] 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*, Jul 2019.

[96] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, pp. 503–507, May 2015.

[97] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.

[98] V. Vassiliades, K. I. Chatzilygeroudis, and J. Mouret, "Scaling up map-elites using centroidal voronoi tessellations," *CoRR*, vol. abs/1610.05729, 2016.

[99] 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.

[100] M. C. Fontaine, J. Togelius, S. Nikolaidis, and A. K. Hoover, "Covariance matrix adaptation for the rapid illumination of behavior space," 2019.

[101] A. Cully, "Multi-emitter map-elites: Improving quality, diversity and convergence speed with heterogeneous sets of emitters," 2020.

[102] 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.

[103] M. Charity, A. Khalifa, and J. Togelius, "Baba is y'all: Collaborative mixed-initiative level design," 2020.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] J. Togelius and J. Schmidhuber, "An experiment in automatic game design," in *2008 IEEE Symposium On Computational Intelligence and Games*, pp. 111–118, 2008.

[109] 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.

[110] 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, no. 2, pp. 192–203, 2017.

[111] M. Cook, "Getting started in automated game design," 2020.

[112] A. Liapis, "Searching for sentient design tools for game development." PhD Dissertation, IT University of Copenhagen, 2014.

[113] G. Ferguson and J. Allen, "Mixed-initiative systems for collaborative problem solving," *AI Magazine*, vol. 28, p. 23, Jun. 2007.

[114] 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.

[115] 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.

[116] 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.

[117] O. Delarosa, H. Dong, M. Ruan, A. Khalifa, and J. Togelius, "Mixed-initiative level design with rl brush," 2020.

[118] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, "Pcgrl: Procedural content generation via reinforcement learning," 2020.

[119] 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.

[120] 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.

[121] 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. `https://gaigresearch.github.io/gvgaibook/`.

[122] T. Schaul, "A video game description language for model-based or interactive learning," in *2013 IEEE Conference on Computational Inteligence in Games (CIG)*, pp. 1–8, 2013.

[123] 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.

[124] 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.

[125] 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.

[126] 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.

[127] 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.

[128] Nintendo R&D1 and Intelligent Systems, "*Super Mario Bros.*." Game [SNES], September 1985. Nintendo, Kyoto, Japan. Last played August 2020.

[129] 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.

[130] 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.

[131] Almost Human, "*Civilization IV*." Game [PC], October 2014. Almost Human, Espoo, Finland.

[132] J. M. A. Sarr, G. N. Yannakakis, A. Liapis, A. Bah, and C. Cambier, "Djehuty: A mixed-initiative handwriting game for preschoolers," in *Proceedings of the 15th International Conference on the Foundations of Digital Games*, FDG '20, 2020.

[133] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, "Interactive constrained map-elites: Analysis and evaluation of the expressiveness of the feature dimensions," *Submitted to IEEE Transactions on Games*, 2020. Preprint available: https://arxiv.org/abs/2003.03377.

[134] 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.

[135] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. Andre, "Player modeling," 2013.

[136] R. Thawonmas, J. Togelius, and G. N. Yannakakis, "Artificial general intelligence in games: Where play meets design and user experience," *NII Shonan Meet. Rep.*, 2019.

[137] 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.

[138] R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit muds," *Journal of MUD Research*, 1996.

[139] A. Saas, A. Guitart, and Á. Periáñez, "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.

[140] A. F. del Río, A. Guitart, and África Periáñez, "A time series approach to player churn and conversion in videogames," 2020.

[141] A. Guitart, P. P. Chen, P. Bertens, and Á. Periáñez, "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.

[142] A. Liapis, D. Gravina, E. Kastbjerg, and G. N. Yannakakis, "Modelling the quality of visual creations in iconoscope," in *Games and Learning Alliance* (A. Liapis, G. N. Yannakakis, M. Gentile, and M. Ninaus, eds.), (Cham), pp. 129–138, Springer International Publishing, 2019.

[143] A. Alvarez and M. Vozaru, "Perceived behaviors of personality-driven agents," *Violence | Perception | Video Games: New Directions in Game Research*, pp. 171–184, 2019.

[144] D. Melhart, G. N. Yannakakis, and A. Liapis, "I feel i feel you: A theory of mind experiment in games," *KI-Künstliche Intelligenz*, pp. 1–11, 2020.

[145] K. Compton, "Casual creators: Defining a genre of autotelic creativity support systems." PhD Dissertation, University of California Santa Cruz, 2019.

[146] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, *et al.*, "Guidelines for human-ai interaction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019.

[147] 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.

[148] A. Alvarez, J. Font, and J. Togelius, "Designer modeling through design style clustering," *Submitted to IEEE Transactions on Games*, 2020. Preprint available: https://arxiv.org/abs/2004.01697.

[149] R. Sternberg and T. I. Lubart, "The concept of creativity: prospects and paradigms," in *Handbook of Creativity*, ch. 1, Cambridge University Press, 1999.

[150] R. Sternberg and C. U. Press, *Handbook of Creativity*. Cambridge University Press, 1999.

[151] R. J. Sternberg, "Creativity or creativities?," *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 370 – 382, 2005. Computer support for creativity.

[152] 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.

[153] S. Colton and G. A. Wiggins, "Computational creativity: The final frontier?," in *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, (NLD), p. 21–26, IOS Press, 2012.

[154] M. Guzdial and M. O. Riedl, "Combinets: Creativity via recombination of neural networks," in *Proceedings of the Ninth International Conference on Computational Creativity*, 2018.

[155] A. Sarkar, Z. Yang, and S. Cooper, "Controllable level blending between games using variational autoencoders," in *Proceedings of the EXAG Workshop at AIIDE*, 2019.

[156] R. Miikkulainen, "Creative ai through evolutionary computation: Principles and examples," 2020.

[157] A. Sarkar, "Game design using creative ai," in *NeurIPS 2019 Workshop on Machine Learning for Creativity and Design*, 2019.

[158] C. Darwin, *On the Origin of Species by Means of Natural Selection*. London: Murray, 1859. or the Preservation of Favored Races in the Struggle for Life.

[159] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Natural Computing Series, Springer Berlin Heidelberg, 2007.

[160] P. Prusinkiewicz and A. Lindenmayer, *The Algorithmic Beauty of Plants*. Berlin, Heidelberg: Springer-Verlag, 1996.

[161] 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.

[162] D. Ahslock, S. Risi, and J. Togelius, "Representations for search-based methods," in *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, ch. 9, Springer, 2016.

[163] 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.

[164] K. O. Stanley, "Compositional pattern producing networks: A novel abstraction of development," *Genetic Programming and Evolvable Machines*, vol. 8, p. 131–162, June 2007.

[165] K. O. Stanley and J. Lehman, *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer Publishing Company, Incorporated, 2015.

[166] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, p. 99–127, June 2002.

[167] 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.

[168] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[169] T. M. Mitchell, *Machine Learning*. USA: McGraw-Hill, Inc., 1 ed., 1997.

[170] 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.

[171] 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.

[172] 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, 9 2019.

[173] 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.

[174] M. Hartmann, Y. Kementchedjhieva, and A. Sø gaard, "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.

[175] 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.

[176] 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), p. 311–318, Association for Computational Linguistics, 2002.

[177] A. Juliani, A. Khalifa, V. 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. 28th International Joint Conference on Artificial Intelligence, IJCAI 2019 ; Conference date: 10-08-2019 Through 16-08-2019.

[178] 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.

[179] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius, "Deep learning for procedural content generation," *Neural Computing and Applications*, Oct 2020.

[180] S. Bjork and J. Holopainen, *Patterns in Game Design (Game Development Series)*. USA: Charles River Media, Inc., 2004.

[181] 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.

[182] 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.

[183] 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.

[184] 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.

[185] 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.

[186] 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.

[187] 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.

[188] 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.

[189] 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.

[190] C. Spinuzzi, "The methodology of participatory design," *Technical Communication*, vol. 52, pp. 163–174, May 2005. Article.

[191] 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.

[192] 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.

[193] 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.

[194] 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.

[195] 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.

[196] E. Aarseth, "From hunt the wumpus to everquest: introduction to quest theory," in *International Conference on Entertainment Computing*, pp. 496–506, Springer, 2005.

[197] 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), p. 16–18, Morgan Kaufmann Publishers Inc., 1981.

[198] 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), p. 63–65, Morgan Kaufmann Publishers Inc., 1983.

[199] D. Karavolos, A. Bouwer, and R. Bidarra, "Mixed-initiative design of game levels: integrating mission and space into level generation," in *Proceedings of FDG 2015 - Tenth International Conference on the Foundations of Digital Games*, jun 2015.

[200] A. M. Abuzuraiq, 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.

[201] K. Yu, N. R. Sturtevant, and M. Guzdial, "What is a quest?," in *Proceedings of the Intelligent Narrative Technolgies Workshop at AIIDE*, 2020.

[202] 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.

[203] 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.

[204] 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.

[205] 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.

[206] E. Grevillius and E. Olsson, "Mixed-initiative quest generation: A mixed-initiative quest tool for the evolutionary dungeon designer." B.S. Thesis, Malmö University, 2020.

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-initia-tive 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 9.2 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 9.3 describes the contributions of this paper together with a description of the last release of the tool. Section 9.4 presents the results from the user study conducted with game developers, followed by the conclusions and future work discussed in Section 9.5.

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 content generation [9]. These factors are mostly decided by the human designer,

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

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 user to choose from. The user can freely alternate between views during the

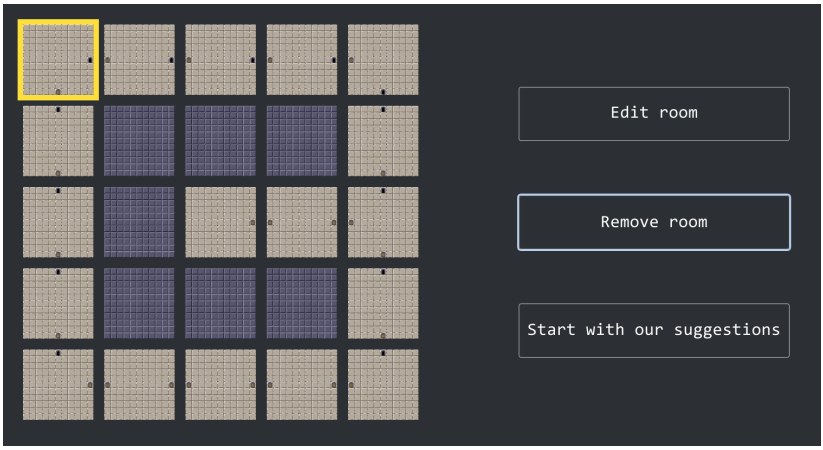


Figure 2: Sample world view showing a 5x5 dungeon with 7 disabled rooms. User actions are displayed in the rightmost buttons.

design process. The current dungeon layout can be saved at any moment from either of the views.

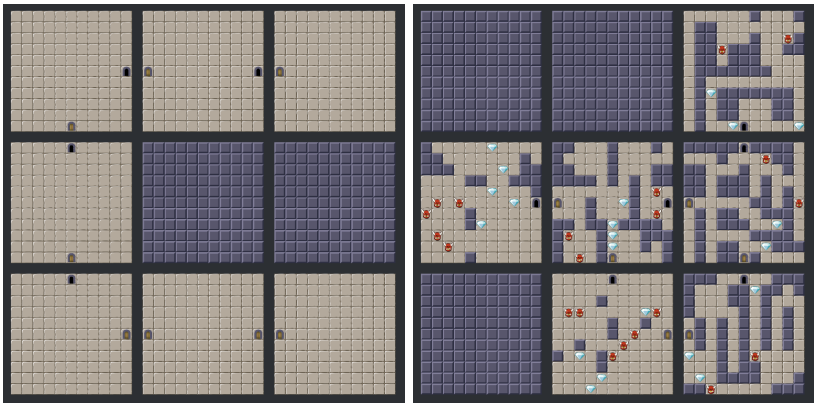
The world view (fig. 2) opens up right after the start screen, displaying a grid of the selected size composed by a fully connected set of empty rooms (all rooms are connected to their neighbors). The users can load a previously saved dungeon design, skipping the start screen and resume work from the 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 view (fig. 4). When clicking any of the suggestions, it will replace the previously



(a) (b)

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

selected room in the dungeon.

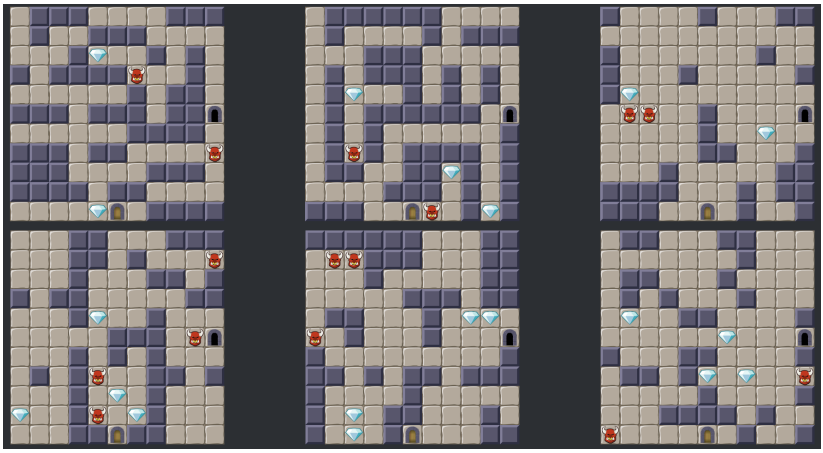


Figure 4: Six procedurally generated rooms presented to the user in the suggestions view.

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.

Suggestions preserve the door layout from the room that was selected in the

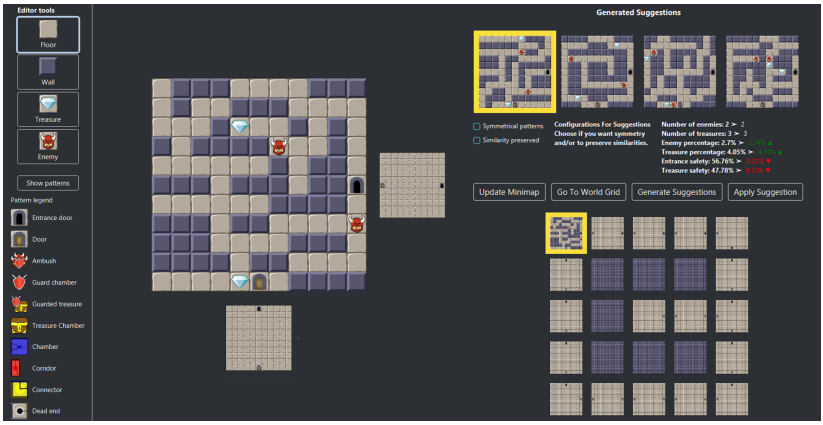


Figure 5: The room view while editing the top-left corner in a 5x5 dungeon with seven disabled rooms.

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 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 as 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 consider-

ation 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.

[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.

[3] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Springer, 2018. `http://gameaibook.org`.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[12] Nintendo Research & Development 4, "The Legend of Zelda," 1986.

[13] Blizzard North, "Diablo," 1996.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[24] J. Allen, C. I. Guinn, and E. Horvtz, "Mixed-initiative interaction," *IEEE Intelligent Systems and their Applications*, vol. 14, no. 5, pp. 14–23, 1999.

[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.

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 as 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].

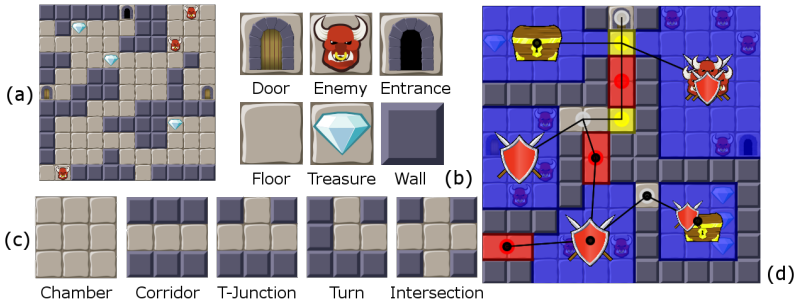


Figure 1: Current version of EDD and its different components. (a) Basic room, (b) different placeable tiles, (c) micro-patterns and (d) meso-patterns.

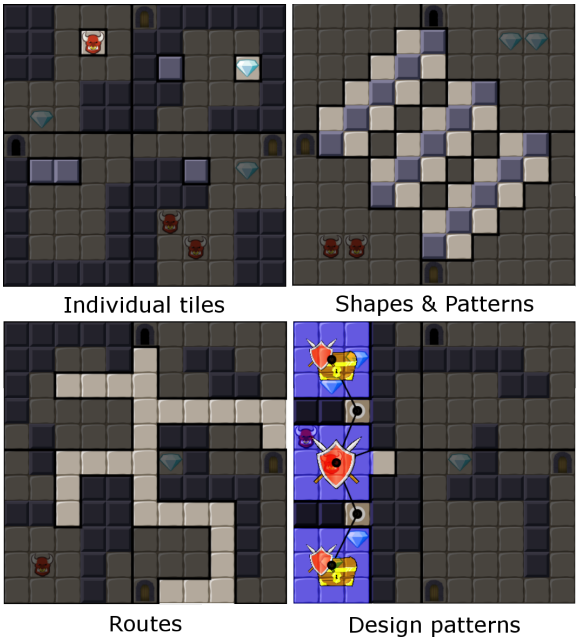


Figure 2: Different uses and possibilities that the designers can have for locking the tiles in the Room, in order to, preserve their manual changes and diverse objectives

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 10.2 presents previous and related works in mixed-initiative design. Section 10.3 describes in detail the contributions of this paper and presents the results from the laboratory experiments used for validating them. Finally, Section 10.4 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 [13–15].

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 [16] 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, [17] 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 [18] 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 [19] 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 sym-

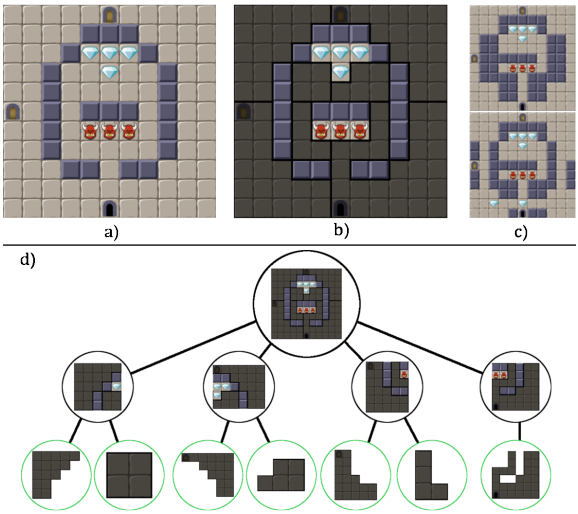


Figure 3: A sample edited room (a) with its division into zones (b) based on the tiles locked by the user. Suggestions preserve these locked tiles (c). The room and its zones are internally represented with a tree structure (d), where the leaf nodes (green) are the valid candidates to operate within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

metry and similarity for the EA. On the other side, the designer was given control over 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

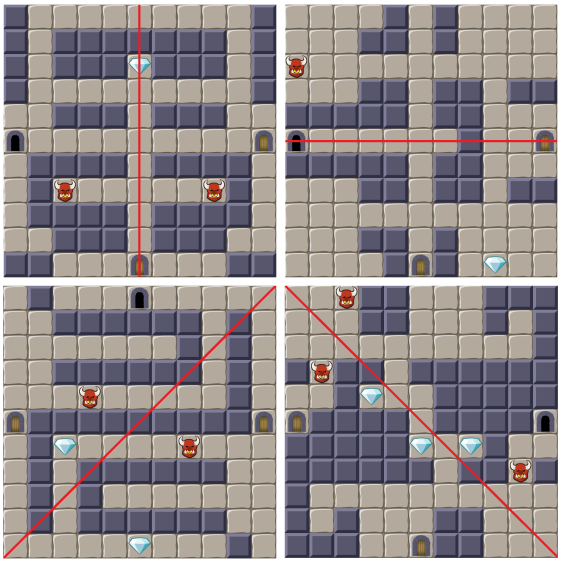


Figure 4: Different types of symmetry evaluated

for crossing and mutation. Figure 3 shows the room, it's division into zones and 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

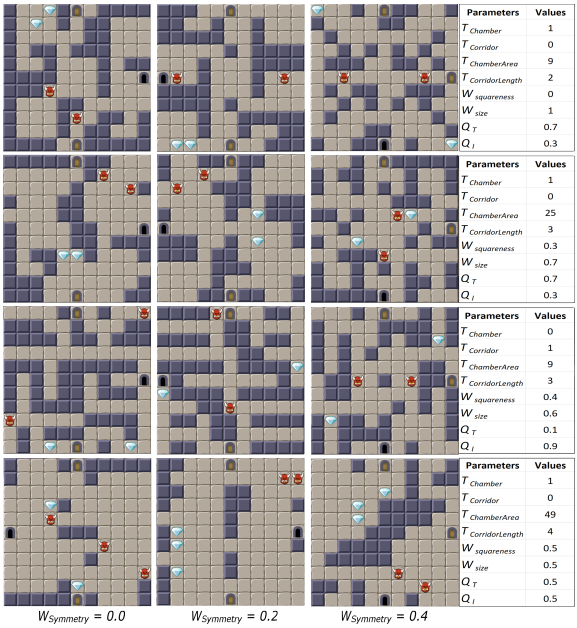


Figure 5: Each row shows three results ($W_{symmetry} = 0$, $W_{symmetry} = 0.2$, $W_{symmetry} = 0.4$) produced under the settings displayed on the rightmost column. Metrics adapted from [19].

symmetry along the X and Y axes, backslash and front slash diagonal as 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 4 to calculate the fitness.

$$f_{symmetry} = \frac{highestSymmetricValue}{totalWalls} \tag{4}$$

Equation 4 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.

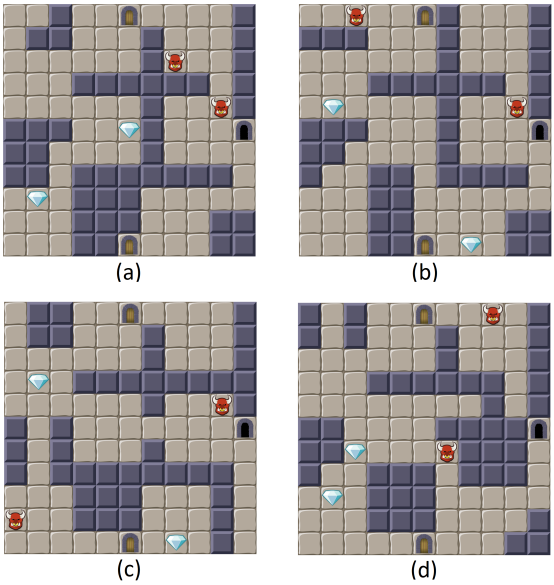


Figure 6: (a) Sample original room and the evolved solutions with different $idealSimilarity$ values in order: (b) 0.95, (c) 0.90 and (d) 0.85.

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 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 5.

$$similarityPercentage = \frac{totalTiles - notSimilarTiles}{totalTiles} \tag{5}$$

We introduced a second parameter called $idealSimilarity$, which represents how similar we want the individuals to be. Following equation 6 we measured the error between both similarities and used it as the similarity fitness.

$$f_{similarity} = 1 - |idealSimilarity - SimilarityPercentage| \tag{6}$$

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 7. $f_{inventorial}$ 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].

$$\begin{aligned} f_{fitness}(r) = (\frac{a}{10} f_{inventorial}(r) + \frac{b}{10} f_{spacial}(r) \\ + \frac{c}{10} f_{symmetry}(r)) * f_{similarity}(r) \end{aligned} \tag{7}$$

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 throughly 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 [20], 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 [20] 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.

[2] M. Toy, G. Wichman, K. Arnold, and J. Lane, "Rogue," 1980.

[3] Maxis, "Spore," 2008.

[4] Hello Games, "No Man's Sky," 2016.

[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.

[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.

[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.

[8] M. Scirea, P. Eklund, J. Togelius, and S. Risi, "P R I M A L - I M P R O V : Towards Co-Evolutionary Musical Improvisation," in *Proceedings of the Computer Science & Electronic Engineering Conference (CEEC)*, (Colchester), pp. 172–177, IEEE, 2017.

[9] C. Browne, "Yavalath," 2007.

[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.

[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.

[12] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, "Towards pattern-based mixed-initiative dungeon generation," in *Proceedings of the 12th Inter-*

national Conference on the Foundations of Digital Games, vol. 74, pp. 1–10, ACM, 2017.

[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.

[14] E. J. Hastings, R. K. Guha, L. Member, 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.

[15] a. Machwe and I. Parmee, "Integrating aesthetic criteria with evolutionary processes in complex, free-form design – an initial investigation," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 165–172, 2006.

[16] 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.

[17] J. R. H. Mariño and L. Lelis, "A computational model based on symmetry for generating visually pleasing maps of platform games," in *AIIDE*, 2016.

[18] 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.

[19] 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.

[20] 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.

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

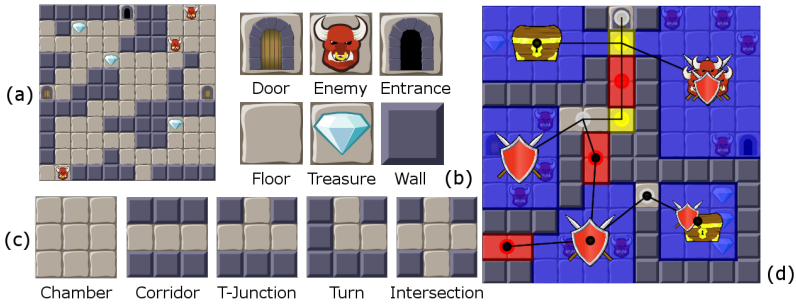


Figure 1: Main components in EDD. (a) Basic room, (b) different placeable tiles, (c) micro-patterns and (d) meso-patterns [21].

(loot), boss fights, locked door and key (you need to find the key to open the 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

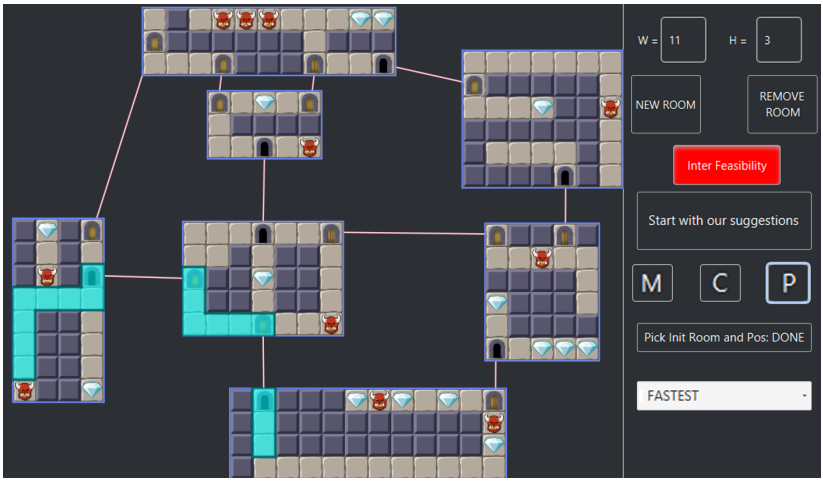


Figure 2: Screenshot of the dungeon editor screen in EDD, displaying a sample dungeon composed by seven rooms. The shortest path between two given tiles is highlighted in blue. The right pane contains all options for editing the dungeon. "M", "C", and "P" stand for "Move rooms", "Connect rooms", and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

suggestions, and is driven through the use of game design micro- and meso-patterns (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 EDD[1], 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

[1]available for download at `https://drive.google.com/file/d/1lCUfc4OF7lY3vUlPzAqf7i7OUfaKQoem/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.

removed at any time, and their associated doors removed with them.

- calculating paths between any pair of passable tiles located in any connected room. Paths are automatically calculated following one 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 3). 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 3 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 (Section 11.4). 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 (8) 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 3x3, and results in $Max_{chambers} = \lfloor Cols/3 \rfloor * \lfloor Rows/3 \rfloor$.

$$D_{mesoPat} = \min\left\{\frac{\#MesoPat}{Max_{chambers}}, 1.0\right\} \tag{8}$$

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 (9) 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.

$$D_{spatialPat} = \min\left\{\frac{\#SpatialPat}{\max\{Cols, Rows\} * K}, 1.0\right\} \tag{9}$$

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 (10) 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.

$$D_{lin} = 1 - \frac{AllPathsBetweenDoors}{\#spatialPat + \#NeighborsPerDoor} \tag{10}$$

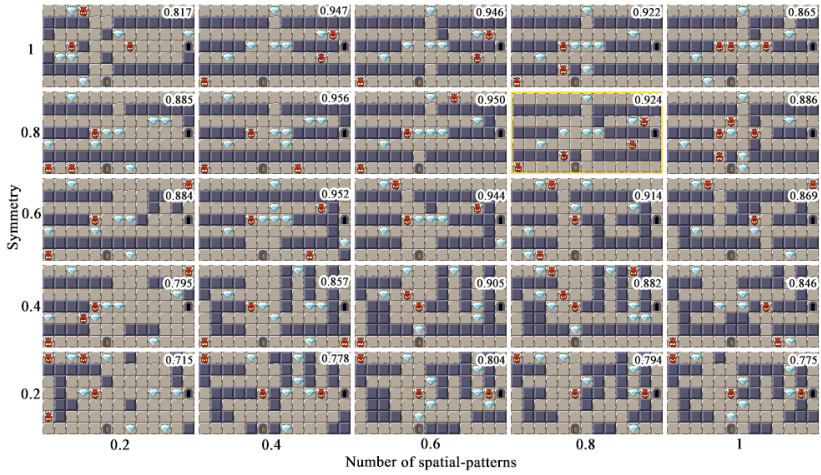


Figure 4: Rooms at generation 2090 targeting Number of spatial-patterns (X) and Symmetry (Y). Each cell displays (top-right) the fitness of the optimal individual in its related feasible population.

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

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 changed, and if so, recreate the cells and populate them with the previous individuals, and proceed through the evolutionary strategies. Selection is through

Algorithm 3 Interactive Constrained MAP-Elites

```
procedure IC-MAP-ELITES([{d_1, v_1}, ..., {d_n, v_n}])
    target ← curEditRoom                                   ▷ Always in background
    createCells([{d_1, v_1}, ..., {d_n, v_n}])
    for i ← 1 to PopSize do
        add mutate(target) to population
    CheckAndAssignToCell(population)
    while true do                                          ▷ start continouous evo
        for generation ← 1 to publishGen do
            if dimensionsChanged then
                previousPop ← cells_pop
                createCells(newDimensions)
                checkAndAssignToCell(previousPop)
            repeat [for feasible & infeasible pop.]
                for i ← 1 to ParentIteration do
                    curCell ← rndCell(cells)
                    add tournament(curCell) to parent
                offspring ← crossover(Parent)
                checkAndAssignToCell(offspring)
            sortAndTrim(cells)
        broadcastElites()                                  ▷ render elites
        pop' ← cells_population
        add mutate(cells_pop) to pop'
        add target to pop'
        checkAndAssignToCell (pop')
        sortAndTrim(cells)
procedure CREATECELLS(DIMENSIONS)
    for each dim ∈ dimensions do
        add newCell(dim_d, dim_v) to cells
procedure CHECK&ASSIGNTOCELL(curPopulation)
    for each individual ∈ curPopulation do
        individual_f ← evaluate(individual)
        individual_d ← dim(individual)
        add individual to cell_pop(individual_d)
```

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.

Section 11.6 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 4 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.

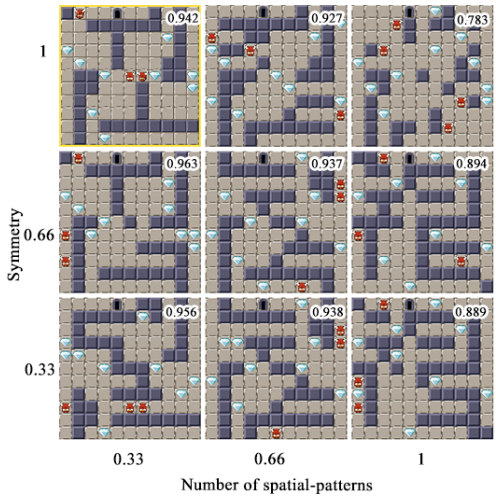


Figure 5: Rooms at generation 5303 targeting the same dimensions as in Figure 4, but with the size 11×11 instead.

Results in Figure 4 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 4 shows difficulties producing symmetric

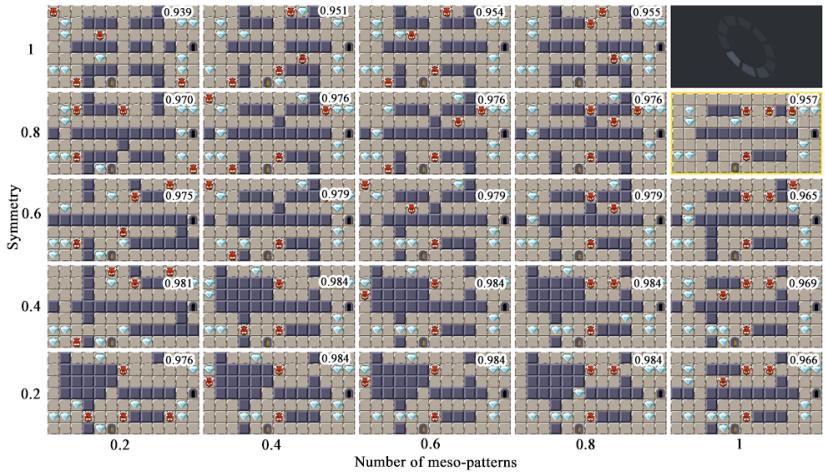


Figure 6: Rooms at generation 7088 targeting Number of meso-patterns at the X-axis and Symmetry at the Y-axis. The top-right cell shows that no optimal room could be generated under dimension values $[1, 1]$.

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. 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

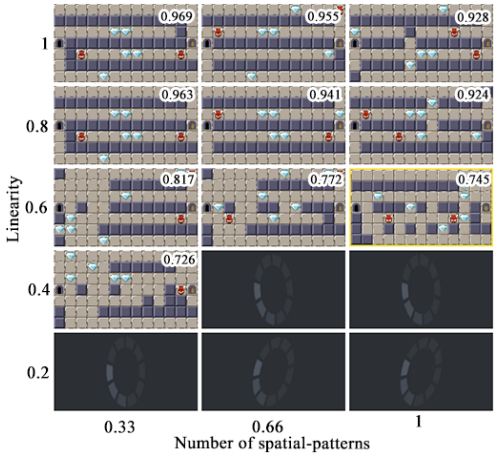


Figure 7: Rooms at generation 12545 targeting Number of spatial-patterns at the X-axis and Linearity at the Y-axis.

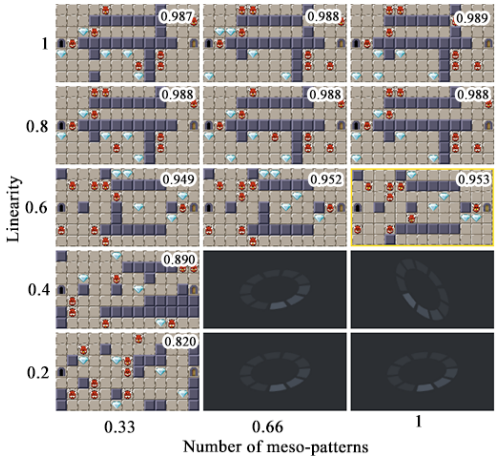


Figure 8: Rooms at generation 20348 targeting Number of meso-patterns at the X-axis and Linearity at the Y-axis.

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 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.

[2] M. Toy, "Rogue," 1980.

[3] D. Braben and I. Bell, "Elite," 1984.

[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.

[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.

[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.

[7] J. Dormans, "Unexplored," 2017.

[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.

[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.

[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.

[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.

[12] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.," in *AIIDE*, 2013.

[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.

[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.

[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.

[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.

[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.

[18] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.

[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.

[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.

[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.

[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.

[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.

[24] C. W. Totten, *Level Design: Processes and Experiences*. CRC Press, 2017.

[25] M. Barton, *Dungeons and Desktops: The History of Computer Role-playing Games*. A K Peters Ltd, 2008.

[26] N. Brewer, "Going Rogue: A Brief History of the Computerized Dungeon Crawl." accessed 2019, March 20.

[27] R. Rutherford, "pedit5, alternately called The Dungeon," 1975.

[28] D. Worth, "Beneath Apple Manor," 1978.

[29] C. Bateman and R. Boon, *21st Century Game Design*. Charles River Media, 2006.

[30] A. Rollings and E. Adams, *Andrew Rollings and Ernest Adams on Game Design*. New Riders Publishing, 2003.

[31] E. McMillen and F. Himsl, "The Binding of Isaac," 2011.

[32] V. Vassiliades, K. I. Chatzilygeroudis, and J. Mouret, "Scaling up map-elites using centroidal voronoi tessellations," *CoRR*, vol. abs/1610.05729, 2016.

[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.

PAPER IV - PERCEIVED BEHAVIORS OF PERSONALITY-DRIVEN AGENTS

Alberto Alvarez and Miruna Vozaru

PUBLISHED IN

Violence | Perception | Video Games: New Directions in Game Research, [transcript], 2019

PERCEIVED BEHAVIORS OF PERSONALITY-DRIVEN AGENTS

Introduction

The discussion regarding the believability of video game characters in the fields of game analysis and artificial intelligence research has taken many forms over the years, generally focusing on appearance and behavior [1–3]. In the paper at hand, we will present a study in which we chose to focus on behavioral believability.

The pervading notions related to the degree of character believability seem to be their awareness, reaction capabilities, and adaptability to the events taking place around them [4]. These factors seem to be connected to the mental schemas activated by the visual depictions of characters and the game world [5]. This made us question how the believability of an agent would be affected by the absence of anchoring references. Stripping away referential visual depictions, narrative, and the relevance of affordances to the traversal of the game world, we sought to understand the narratives that observers create around an ambiguous entity acting within an abstract environment.

In this paper, we will first present our reasoning and the theoretical background of the research design, the technical aspects of the AI agent that served as our character, and the responses that participants provided following the viewing of several films depicting the actions of the agent. Finally, we will discuss our conclusions and implications for future research and game development.

Theoretical Background

The purpose of this research is to analyze the means through which the viewer makes sense of ambiguous behavior in the absence of corresponding mental schemas. To do this, we needed to understand the means by which information is perceived and integrated, and what the observer uses to fill in the blanks when the stimulus is too ambiguous to fit into pre-existing information. New information, such as that presented by the behavior of an observed game character, is integrated

within the pre-existing mental schemas of the observer, which are used to shape the meaning of the new information and make predictions about future developments. For instance, in the video game PORTAL [6], the portal gun activates the mental schemas corresponding to previously encountered guns in video games. Namely that it can shoot, it is a tool for progressing in the game, and it damages enemies. When the portal gun is used, instead of damaging an enemy, it creates a gateway that the player can use to traverse the game. This result does not match pre-existing knowledge, which will force a schema modifi- cation concerning the video game gun functionality. The visual representation affords the integration of the portal gun within the player's previously acquired knowledge; it is referential, descriptive, and concrete. The differences become apparent once the observed functionality does not match expected performance.

Affordance theory, popularized in design by Donald Norman, describes the action and use possibilities that an entity, object, or environment possesses [7]. This theory has been widely appropriated by game design, due to the designer's needs to communicate briefly, clearly, and coherently the means through which a player can traverse a game. Affordances can be tied to previously acquired knowledge, but also be assigned meaning derived strictly from their application within the game world. They are used to telegraph the ways in which the players can use the different elements at their disposal to navigate the game world and to constrain the situational role of the elements.

That being said, the perception of use and role is not a necessary factor in the perception of agency and attribution of specific behaviors. In a study conducted by Heider and Simmel, participants were shown a brief video in which the actors were a circle, a large triangle, and a small triangle [8]. The shapes were depicted in various types of motion, seemingly interacting with each other and the environment. The participants were then prompted to describe the events taking place in the video. All of the participants, with the exception of one, described the events of the video as part of a narrative, whether it was as two parents fighting in front of their child, or two people finding themselves in a romantic situation and then being interrupted by a third. This led us to conclude that the perception of self-directed motion transforms the interpretation of a pattern into one involving an agential entity [9].

So far, we can conclude that the believability of the agent hinges on its recognizable visual representations, as well as the affordances displayed within the game world. By stripping these factors and endowing a visually ambiguous object with self-directed motion, it will be interpreted as an entity with perceived agency.

Personality traits have generally been viewed as probabilistic determinants for the predictability of a certain type of behavior mediated and moderated by the current situation [10–13]. The Cybernetic Big 5 model, henceforth CB5T, treats personality-endowed agents as goal-based entities perpetually engaged in goal attainment loops [14]. The goal loops are divided into stages, with personality traits exerting their influence on each stage of the loop. Traits are manifested through characteristic adaptations, which, unlike personality, are constructed based on individual life experiences and are thus not universal. For instance, the manifestation of the trait compassion can take different characteristic adaptations, such as volunteering or monetary donations, which are dependent on the individual's socio-cultural environment.

While our intentions steer clear of transforming the research into a projective test[1], we decided to use personality factors as behavior determinants for our AI agents. Our hypothesis was that in the absence of other anchoring visual primers, the observers would integrate the perceived behavior within familiar characteristic adaptations. While we used Heider and Simmel's experiment as a starting point, our research was also informed by the similarity-attraction hypothesis, which states that individuals grant more positive appraisals to responses that match their own personality traits [15]. The agents were given the same traits as the corresponding observer. We used an AI agent instead of a pre-rendered movie, due to the options it offered in terms of personality customization. We distinguish our purposes from the creation of narratives surrounding ambiguous agents. To clarify, our purpose was not the exploration of the participants' creation of narratives surrounding the ambiguous behavior of an agent, but to explore the participants' propensity to recognize behaviors with which they are most familiar – the ones they have observed in themselves.

Agent Behavior and Design

The following section will cover the psychological and visual design of the AI agent, its personality, emotions, and affective behavior, as well as the reasoning behind the aesthetic choices we made.

As mentioned above, the CB5T model moves away from previous models and classifies traits as global influencers of the goal attainment loop, which is broken down into five stages: goal activation, action selection, action, outcome interpretation, and goal comparison. As a result of the ongoing process of receiving,

[1] A projective test is a psychological assessment during which participants are asked to interpret ambiguous stimuli with the assumption that the interpretation will reveal insights regarding their personality traits

perceiving, and filtering environmental stimuli, multiple goals can be active at the same time. The first and second phases are internal and generally com- prised of parallel processes. The third phase presents a bottleneck to the goal loop, due to the fact that, while a person can hold multiple goals in their memory at the same time, actions are generally performed in sequence. In the fourth phase, the result of the action is measured against the intended results, which will generate a match or a mismatch. This will in turn inform the next goal and the subsequent cycle. Personality traits exert their influence in concert at every stage of the goal loop, becoming moderators to the goal attainment stages. For a better understanding of the goal loop, we can consider the following situation: The AI agent feels hungry, and in the process of trying to obtain food it realizes that it must jump over an obstacle. While both goals exist in its memory, the fact that it can perform only one action at a time determines that it must first perform the jump. This is the action selection phase. The agent jumps and fails to over- come the obstacle. The actual result and the intended result do not match, and its personality score will influence its reflection on this failure.

The CB5T model was coupled with a simplified version of the Ortony, Clore, and Collins model of emotions (henceforth OCC), "The OCC model revisited. [16]" The reasoning behind the combination of personality and emotion models was derived from the need to visually depict the results of the internal processes taking place during the goal attainment phases. The influence exerted by personality traits on the goal attainment loop materializes in affective behavior, where the emotional valence and intensity is dependent on the personality score.

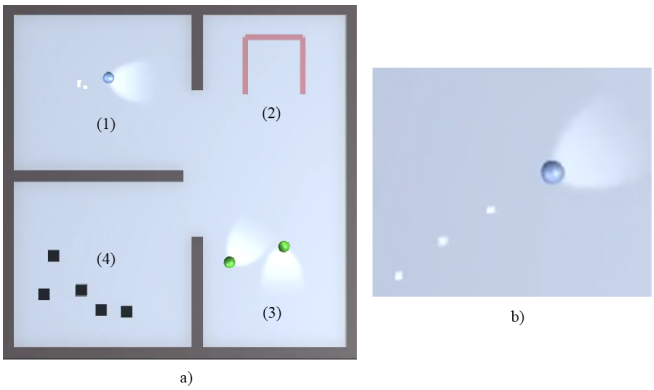


Figure 1: Simulation environment.(a) Example environment that was presented to users, containing the personality-driven agent (1), and three different situations that the agent will encounter (2,3,4). (b) Agent trail and view.

Intentionality and attention presented a large part of the concretization of

personality-derived behavior. The agent's attention and intentions were depicted by a cone of light in front of it, maintaining the ambiguity of the stimulus but strengthening the perception of agency. Similarly, the trail the agent leaves behind signifies its movement speed which is dependent on its emotional arousal. The affective behavior exhibited by the agent had to be contextualized in specific situations, in order to be granted environmental referentiality. We created several situations, including but not limited to: positive and negative social situations, environmentally challenging situations, and situations that could produce distractions.

Agent Architecture and Simulation

Human-like behavior is a complex subject and one that cannot be approached by using only one model or technique. Rather, different approaches use a compendium of specialized modules. For instance, emotional, personality, memory, or social modules that have various responsibilities in order to simplify the decision-making process.

The TOK architecture represents an agent as a set of different modules that handle the perception, reactivity, goals, emotions, and social behaviors. TOK is divided into three main components: (1) HAP, the goal-based reactive engine, (2) EM, the emotional model, and (3) Glinda, the natural language system [17].

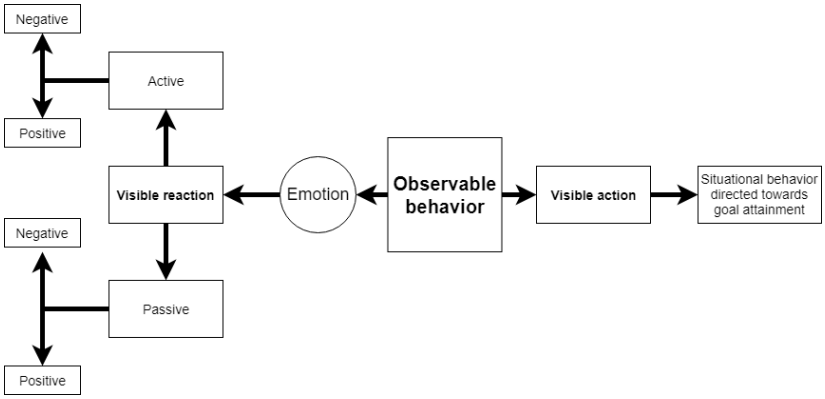


Figure 2: Observable Behavior. The observable behavior by the users is based on the set of actions provided by the encountered situation, which results in an emotional reaction. The reaction is based on the outcome interpretation phase, where the agent will choose the respective emotion (active/passive and negative/positive), intensity and reactive behavior.

Our model assimilates to HAP and EM, in that HAP selects an action based primarily on the agent's goals, emotions and perception by using the CB5T model, and with EM it calculates the emotional valence of the agent by comparing envi-

ronmental stimuli with goals, possible actions with standards, and environmental objects with attitudes.

The agent's goals have a dynamic weight and a priority. The weight is determined by the agent's moment-to-moment actions and reflect its progression and perception of environmental cues. The priorities indicate the goal type, and their influence on survival and self-actualization. To exemplify, "hunger" is a priority 1 goal, and its weight will increase according to the presence of food in the agent's proximity and the time they spend roaming around the environment.

Subsequently, according to the outcome of the situation, the agent will feel pleased or displeased in accordance with the match or mismatch between the desired outcome and the actual one. The combination of the outcome difference and the personality traits trigger different emotional reactions. As presented in figure 2, the agent can perform several actions in specific situations and the consequence of the selected action entails not only different emotions but also different levels of arousal.

Table 1: Reaction table used to choose the respective emotional reaction. First, we choose the agent's emotion by choosing if active/passive and negative/positive as presented in figure 2, with the addendum that if more than one option is viable, the choice will be based on the situation's target. Finally, the intensity is calculated.

Situation type	Event type	Agent type
Negative/Positive	Displeased/pleased	Standards
Passive/Active	Neuroticism	Neuroticism
LOW/MID/ HIGH	Goal weight	Neuroticism level
Inner choice	Situation target	Situation target

The way in which different emotional reactions occur was modelled in a table which is live queried to extract the agent's reaction. Table 1 illustrates how a reaction is chosen based on the two types of situations presented in our simulation.

Simulation

In order to control the agent's decisions, we built a decision tree that uses personality traits, perceived situations, and the current goal weight as inputs. The decision tree allows the agent to acquire its next goal, perform the action, and exhibit the corresponding emotional reaction. Therefore, the decision tree has five steps: (1) sense the environment, (2) reason about the possible actions based on their goals, (3) engage with the situation, (4) reflect on the outcome, and (5) produce an emotional reaction.

The agent constantly senses its surroundings to gather encountered entities, it inventories the situations they generate and compares them to its inner goals, and then chooses the situation that would satisfy the primary goal. To simulate the attraction towards novelty, every situation the agent engages in becomes less novel every time it is chosen. Once a specific type of situation becomes ordinary (i.e. not novel), the agent will give more weight to other types of situations.

Each situation has its own set of afforded behaviors. For instance, in a situation in which the agent can encounter other agents, it can approach them, give objects to them, or perform other context-specific actions. This allowed us to simplify the agent's decision-making step by allocating complexity to each situation.

Situations are divided into two main categories: environmental challenges and social challenges. For instance, the agent might **encounter an obstacle**, and finding out what is on the other side of it will satisfy its curiosity goal. Or it might engage in a situation where other agents are **having fun**, which in turn would satisfy the **social acceptance goal**.

Although the agent is given a set of behaviors to perform, there are cases where the agent will simply not perform the action, due to not having the necessary resources, being physically unable to do it, or due to a conflict with its personality traits. For instance, the agent might not be able to give any object to collecting agents, as the agent has not found anything yet, or to jump a gap due to low levels of assertiveness and high withdrawal.

Once the situation has been resolved, the agent compares the actual outcome to the expected one and the respective weight is modified accordingly. This final step triggers the agent's emotional reaction, defined by the Figure 2 process and reached by the queried information from the reaction table (Table 1). Finally, the agent goes back to its ordinary state and repeats the goal attainment loop.

Participant Assessments and Responses

Participants were selected using a snowball method. Both the selection method and the low number of participants (n=6) preclude us from generalizing results. Participants were informed that they would take part in a study focusing on behavior perception in a virtual environment. In the first stage of the research, the participants were asked to fill out a personality evaluation questionnaire. The questionnaire was comprised of 50 items, taken from the International Item Pool, and tailored to the CB5T [18].

The scores were subsequently calculated and assigned to an individual agent.

The agents were placed in the same environment and recorded while acting within the given environment. Consequently, the recordings were shown to the participants, who were asked to describe the behavior of the agent and what they thought the actions represented, without being aware of the specific traits given to the agent.

While the small number of participants does not allow us to draw generalizable conclusions, the responses indicate the recognition of the agent as an entity with directed behavior and agency, to which they attributed familiar characteristic adaptations. One of the participants described their agent's behavior as follows:

> "(. . .) the agent is similar to a person that works in an office. He is engaging in conversation with his co-workers/superiors and tries to do his day-to-day tasks. As I see it, the agent has a lot of work to do, as he is in a continuous movement. I think he should take a break from time to time."

We can see that the ambiguous movement of the agent is identified with a specific characteristic adaptation, and the environment is given characteristics that e replace abstract representations with imagery from an everyday office space. The participant also evaluates the agent in a sympathetic manner, which we can consider a byproduct of the contextualization within a familiar characteristic adaptation.

A different participant, viewing the behavior of their own agent in the same environment, described the behavior as follows:

> "He's pretty anxious, he isn't sure of himself, of what he's going to do (with the box). At first he seemed a bit shy, he basically wanted to avoid the other two, and I think he didn't even say hello to them (...)"

We can see that the agent's movements of approach and withdrawal are described here through the lens of common human social behavior of greeting and social avoidance. The agent is also given specific, human-like traits, which signal recognition and contextualization of behavior.

The responses also reflect drawbacks in our aesthetic choices, but strengthen the hypothesis that, when confronted with ambiguous cues, the participants will appeal to their most readily available mental schema. One participant wrote:

> "This agent looks like it's sweeping the ground for something with a metal detector. It checks both sides of the box but looks like it doesn't find anything."

While the cone of light was intended to be a signifier of attention and intention,

it was interpreted in this case as a concrete object, a metal detector. However, the participant's interpretation that grounds the agent's behavior as exploratory is consistent with our hypothesis of the need to concretize ambiguous behaviors.

We can see the ways in which the participants interpret, and ascribe meaning to, the ambiguous behaviors of the agent. While remaining in the realm of the abstract, the motions and actors could be described as "the dot got closer to the other dots and then got further away." However, the participants attributed emotion and reasoning to the entity.

Conclusion

This pilot experiment explored the ways in which people attribute known and familiar behaviors to an AI agent in the absence of other anchoring visual cues. Participants who, unknowingly at the time, contributed to the creation of the AI by providing data regarding their personality traits, largely interpreted ambiguous behavior by association with their own characteristic adaptations. Future research into this area could explore the interpretation of behavior exhibited by AI agents that have different personality traits than those of the observers. While stripping visual cues from the environment and characters is not a valid aesthetic choice for most video games, the central take-away of this experiment should be the importance of missing information, whether deliberate or not.

The behavior descriptions reflect the participant's propensity for filling in blanks with their own familiar characteristic adaptations. When presented with merely a few rectangles and spheres, one participant saw an office, while another saw a social situation that the protagonist was trying to avoid. These results point to an important value that should be considered in the design, critique, and analysis of digital games: the ambiguity variable.

One of the key missing pieces of this research was the capability of the participants to execute actions within the environment. This would have given the agent in-world affordances, allowing the players to integrate their own intentionality and project their characteristic adaptations onto the performed actions. However, at this stage we did not want to assess the participants' projection of personal actions, but rather their perception of ambiguous events and characters.

Our agent was a capsule, a dot on a two-dimensional plain. However, motion granted it the status of an entity and its ambiguous actions afforded it reasoning, motives, and personality (in the eyes of the participants). The participants were not aware of the personality traits that the agent had been given, but they were able to recognize the narrative around them. The results of the research underline that

when ambiguity is present, the space will be filled by the viewer's characteristic adaptations. This research is just a pilot, and drawbacks such as the limited number of participants and lack of interactivity should be addressed in future iterations.

Acknowledgement

Miruna Vozaru ackowledges the financial support received from the European Research Council (ERC) under the European Union's H2020 ERC-ADG program (grant agreement No 695528)

References

[1] L. Petri and B. Staffan, "Gameplay design patterns for believable non-player characters," in *DiGRA &#3907 - Proceedings of the 2007 DiGRA International Conference: Situated Play*, The University of Tokyo, September 2007.

[2] I. Umarov and M. Mozgovoy, "Believable and effective ai agents in virtual worlds: Current state and future perspectives," *Int. J. Gaming Comput. Mediat. Simulations*, vol. 4, pp. 37–59, 2012.

[3] M. Lee and C. Heeter, "What do you mean by believable characters?: The effect of character rating and hostility on the perception of character believability," *Journal of Gaming & Virtual Worlds*, vol. 4, pp. 81–97, 03 2012.

[4] W. Henrik, J. Magnus, and V. Harko, "Analyzing the believability of game character behavior using the game agent matrix," in *DiGRA &#3913 - Proceedings of the 2013 DiGRA International Conference: DeFragging Game Studies*, August 2014.

[5] D. J. Stein, "Schemas in the cognitive and clinical sciences: An integrative construct.," *Journal of Psychotherapy Integration*, vol. 2, no. 1, pp. 45–63, 1992.

[6] V. Corporation, "*Portal*, year = 2007, day = 10, month = October, howpublished = Game [PC], note = Valve Corporation, Washington, US.Last played August 2019., publisher = Valve Corporation."

[7] D. A. Norman, *The Design of Everyday Things*. USA: Basic Books, Inc., 2002.

[8] F. Heider and M. Simmel, "An experimental study of apparent behavior," *The American Journal of Psychology*, vol. 57, no. 2, pp. 243–259, 1944.

[9] J. Harris and E. Sharlin, "Exploring the affect of abstract motion in social human-robot interaction," in *2011 RO-MAN*, pp. 441–448, 2011.

[10] R. R. McCrae and O. P. John, "An introduction to the five-factor model and its applications.," *Journal of Personality*, vol. 60, no. 2, pp. 175–215, 1992.

[11] W. Mischel and Y. Shoda, "A cognitive-affective system theory of personality: Reconceptualizing situations, dispositions, dynamics, and invariance in personality structure.," *Psychological Review*, vol. 102, no. 2, pp. 246–268, 1995.

[12] R. Tett and H. Guterman, "Situation trait relevance, trait expression, and cross-situational consistency: Testing a principle of trait activation," *Journal of Research in Personality*, vol. 34, pp. 397–423, 12 2000.

[13] P. T. Costa Jr. and R. R. McCrae, *Trait theories of personality.*, pp. 103–121. The Plenum series in social/clinical psychology., New York, NY, US: Plenum Press, 1998.

[14] C. G. DeYoung, "Cybernetic big five theory.," *Journal of Research in Personality*, vol. 56, pp. 33–58, 2015.

[15] D. Byrne, W. Griffitt, and D. Stefaniak, "Attraction and similarity of personality characteristics.," *Journal of Personality and Social Psychology*, vol. 5, no. 1, pp. 82–90, 1967.

[16] B. Steunebrink, M. Dastani, J.-J. Ch, and J.-j. Meyer, "The occ model revisited," in *Proceedings of the 4th Workshop on Emotion and Computing*, AAAI, 01 2009.

[17] J. Bates, A. B. Loyall, and W. S. Reilly, "An architecture for action, emotion, and social behavior," in *Artificial Social Systems* (C. Castelfranchi and E. Werner, eds.), (Berlin, Heidelberg), pp. 55–68, Springer Berlin Heidelberg, 1994.

[18] L. R. Goldberg, J. A. Johnson, H. W. Eber, R. Hogan, M. C. Ashton, C. R. Cloninger, and H. G. Gough, "The international personality item pool and the future of public-domain personality measures," *Journal of Research in Personality*, vol. 40, no. 1, pp. 84 – 96, 2006. Proceedings of the 2005 Meeting of the Association of Research in Personality.

PAPER V - 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-

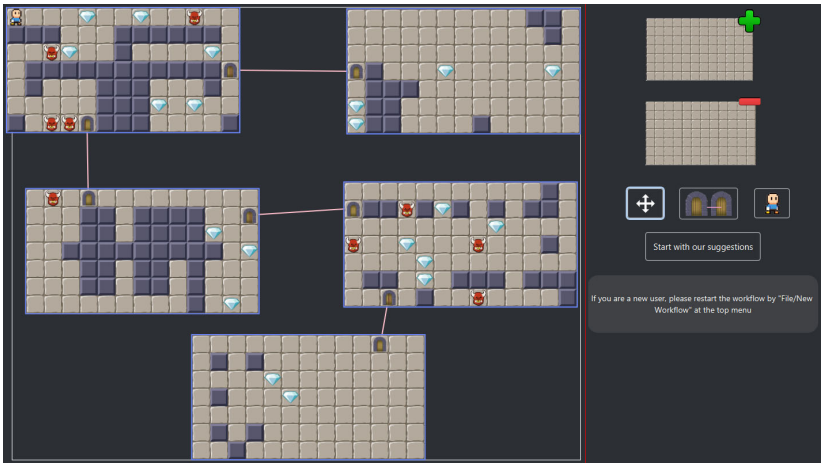


Figure 1: Screenshot of the dungeon editor screen in EDD, displaying a sample dungeon composed by five rooms.

initiative 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 capa-

bility 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

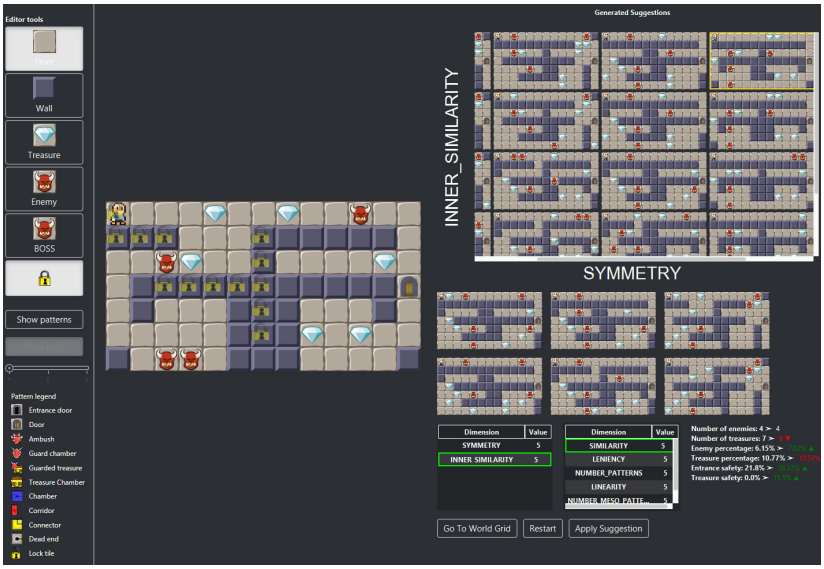


Figure 2: The room editor screen in EDD. The top-right pane shows the suggestions provided by the IC MAP-Elites algorithm. Below are the six top-raked suggestions by the Designer Preference Model. The left pane contains the manual edition features.

assessment to EDD's original fitness function, which evaluates individuals first 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

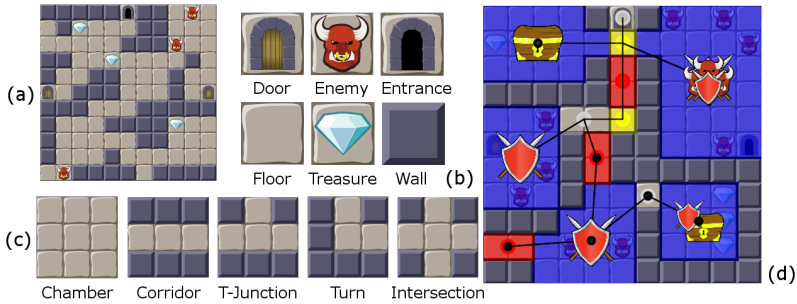


Figure 3: A sample room in EDD (a) compose by tiles (b), spatial patterns (c) and meso-patterns (d). Detailed descriptions for these components can be found in [27].

designer's chosen suggestion. The loop, depicted in figure 4, can be described in the following two steps:

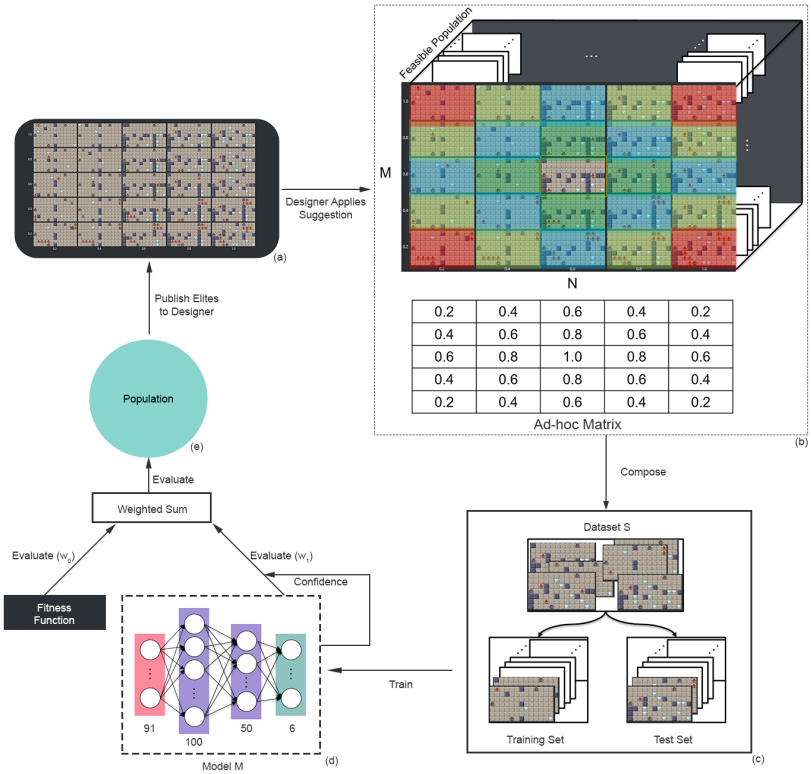
Dataset creation:

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



0.2	0.4	0.6	0.4	0.2
0.4	0.6	0.8	0.6	0.4
0.6	0.8	1.0	0.8	0.6
0.4	0.6	0.8	0.6	0.4
0.2	0.4	0.6	0.4	0.2

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).

softmax layer, which limits the output of all the neurons into the range 0 to 1, as 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. 11) and weighted sum (Eq. 12) to evaluate each of the individuals in the EA were the following:

$$\begin{aligned} w_1 &= \min(M_{conf} \cdot M_{TestAcc}, 0.5), \\ w_0 &= 1.0 - w_1 \end{aligned} \tag{11}$$

$$weightedSum = (w_0 \cdot objective) + (w_0 \cdot predicted_{pref}) \tag{12}$$

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 13.3 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 13.3, 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 previ-

ously 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.

[2] G. N. Yannakakis and J. Togelius, *Artificial intelligence and games*, vol. 2. Springer, 2018.

[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.

[4] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.," in *AIIDE*, 2013.

[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.

[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.

[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.

[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.

[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 Con-*

ference on Computational Creativity, Atlanta, Georgia, USA, June 19-23, 2017. (A. K. Goel, A. Jordanous, and A. Pease, eds.), pp. 205–212, 2017.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[17] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

[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.

PAPER VI - 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 roguelike 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. We further discuss how the results here can be used to create style-sensitive suggestions. Such suggestions would allow the system to be one step ahead of the designer, offering suggestions for the next cluster, assuming that the designer will follow one of the archetypical design traces.

PUBLISHED IN

Submitted to IEEE Transactions on Games, IEEE, 2020.

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 in order to create useful and novel game content? Various systems have been proposed that allow for humans and algorithms to share authorship by both editing and critiquing the content being created, in what is called the mixed-initiative paradigm [1, 2]. However, for such collaboration to reach its true potential, there needs to be an understanding between the human designer and the software 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 background. When one of the collaborators is a computer program, this task is perhaps AI-complete. But 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 session 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 different 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 different models, predicting the next step for different designer "personas" (akin to procedural personas in game-playing [3]). However, for such a procedure to be effective, we need to have a sufficient amount of 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 [4], 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 or other procedural generation methods to suggest ways of getting to the next design style from the current one.

In this paper we provide a prototype implementation of designer personas as archetypical paths through style space. Through this, we take a step further into modeling designers For this we use the Evolutionary Dungeon Designer (EDD), a research platform for exploring mixed-initiative creation of dungeon crawler content [5, 28]. Data from several dozen users designing game levels with the tool have been used to train 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 [6], 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 out of several sorts of user-generated gameplay data [3, 7–10].

Using player data from *Iconoscope*, a freeform creation game for visually depicting semantic concepts, Liapis et al. trained and compared several ML algorithms by their ability to predict the appeal of an icon from its visual appearance [7]. Furthermore, Alvarez and Vozaru explored personality-driven agents based on individuals' personalities using the *cibernetic big five model*, evaluating how observers judged and perceived agents using data from their personality test when encountering multiple situations [11].

Moreover, training models on gameplay data from *Tom Clancy's The Division* has also been used to model, and therefore find predictors of player motivation [10], 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 [9]. Melhart et al. take these approaches one step further by modeling a user's *Theory of Mind* in a human-game agent scenario [8], finding that players' perception of an agent's frustration is more a cognitive process than an affective response.

The Player is the Designer

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 [12–16]. 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 player is the designer in MI-CC, 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 [17].

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. [18, 19] 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. Through these models, designer's and their design process could be understood in-depth, enabling adaptive experiences, further reducing their workload and fostering their creativity.

Furthermore, lack of transparency is a key impediment for the advancement of human-AI systems, being eXplainable AI (XAI) an emergent research field that holds substantial promise for improving model explainability while maintaining high-performance levels [20, 31]. However, explanations should be aligned with the users' understanding to don't hinder the usability of systems, as demonstrated

by Nourani et al. [21], who discuss the effects of meaningful and meaningless explanations to users of an AI interactive systems.

Zhu et al. [22] 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. In the same line, Xie et al. [23] explored visualization techniques through an interactive level designer tool called *QUBE* to explain and introduce machine learning principles to game designers.

Moreover, Guzdial et al. [24] 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 [25] 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 generated suggestions to the designer [26,28]. EDD uses the *Interactive Constrained MAP-Elites* (IC-MAP-Elites) [5], an evolutionary algorithm that combines Constrained MAP-Elites [27] with interactive and continuous evolution.

The work presented in [28] introduced the Designer Preference Model, a data-driven solution that learns from user-generated data in the MI-CC Evolutionary Dungeon Designer. This preference model uses an Artificial Neural Network to model the designer based on the choices she makes while using EDD. Both systems

constantly interact and depend on each other, so that the Designer Preference Model learns from the generated and selected elites, and IC-MAP-Elites uses the Designer Preference Model as a surrogate model of the designer to complement the fitness evaluation of new individuals.

This approach's main goal is modeling the user's design style to better assess the tool's procedurally generated content, increasing the user's agency over the generated content without stalling the MI-CC loop [17] or increasing user fatigue with periodical suggestion handpicking [2, 29]. 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 many of the ideas and concepts introduced by Liapis et al. [18], in relation to style, goals, preferences and design processes of designers. Nevertheless, given the interdisciplinary scope of this system, and the multiple concepts discuss throughout the paper, it is essential to have operational definitions on the different terms used.

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* can be analyzed as overarching goals that different designers have when creating a dungeon. For instance, dungeons in games such as Zelda [4] or The Binding of Isaac [30], 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 understand the designers' holistic design process and the expected player experience, *Design Style* can also be discussed from 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 architecture of the room 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 amount of enemies, or create a concentrated high-challenge in the center of the room, or divide the room into smaller choke areas. Therefore, in this paper, we treat *Design Style* as the style designers follow to create a room, informed by the individual steps each has taken connected to their preferences and goals.

Designer's Goals

The designer's goal is defined as the current state of rooms and the set of interactions done in the tool or sequence of steps taken thus far, to reach such a state. Goals by the designer are linked to the addition and strategic placement of enemies and treasures, giving some goal for the player, e.g., forcing the fight with an enemy or allowing the player to avoid the conflict through side paths.

System Goals

The system goals are defined as the system's approach to support and foster the work of the designer by providing suggestions aligned with her current design or giving assistance, information, visualization, and measurements when needed. In general, when providing suggestions, the system aims at generating rooms among multiple areas of the generative space, simultaneously providing rooms adapted to the designer's goal and different from it.

Shared Goals

The shared goals between the system and the designer are defined as the goals the designer has when creating the dungeon and the individual rooms. Thus, in this paper, the shared goal is set and defined by the designer with her design, and as she develops, adapts, and changes, the system seeks to adjust its goals to support the designer's work. Furthermore, the aim of this paper is to propose a system that is able to identify the designer's current goal and style to adapt further the system's goals to provide a personalized experience.

Room Style Clustering

This paper presents an approach and fundamental steps towards the implementation of designer personas: an analysis of designer style clustering to isolate archetypical paths that can be later be used to build ML surrogate models of archetypal designers. Such models would adapt to the dynamic designer during the mixed-initiative creative process by being placed in the solution space, allowing the designer to traverse such space of models as she drifts through the many

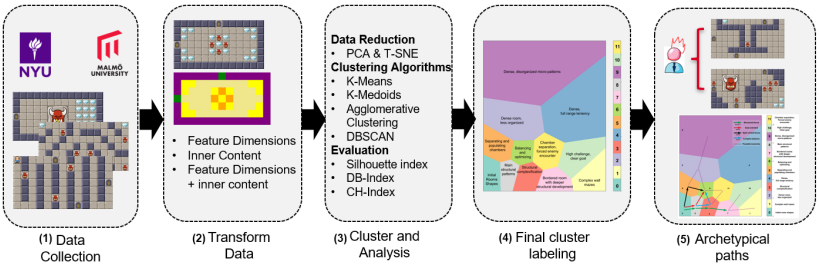


Figure 1: The stages of the design style clustering development: (1) Data was first collected through two user studies. (2) Then, using the design sequences, the data was processed into five different datasets, one using the room images, a second using the tiles information, and three using tabular information. (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 cluster 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.

dimensions of her creative process.

The proposed system builds on top of EDD's Designer Preference Model and preliminary results [28], expanding it to classify the designers' designs based 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 14.5 shows the experimental results.

Data Collection

We conducted two user studies where participants were tasked with freely designing a dungeon in EDD and the rooms that compose it with no further restrictions, using all 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.

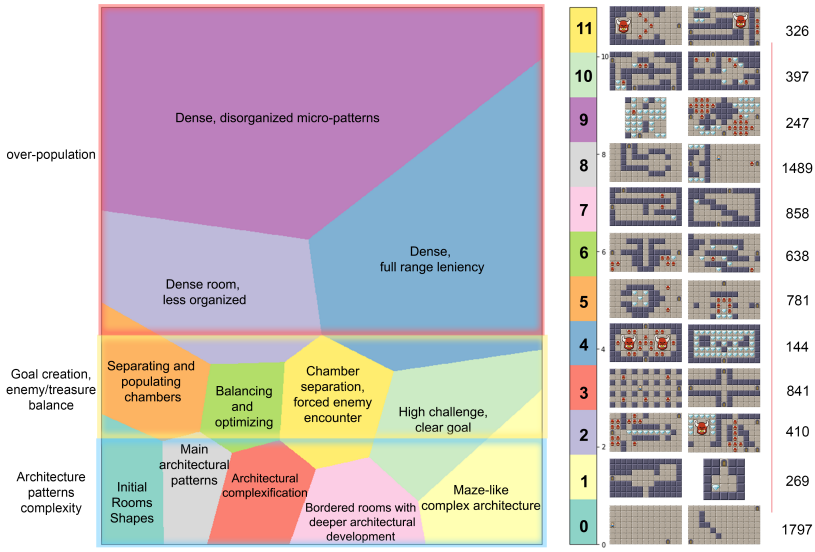


Figure 2: Best resulting cluster set. K-Means (K=12), using the **Tiles** Dataset. While it scores slightly less in the internal indices that other setups, a qualitative analysis successfully gives us more granularity by subdividing the main bottom clusters, to label and cluster the design process of designers. Sample rooms belonging to each cluster are displayed on the right, next to the total number of rooms in the cluster.

Dataset pre-processing

From the 180 unique rooms, we extracted and used the edition sequence of each of the rooms, 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. Through this, we ended up using 8196 data points in our dataset. 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, #MesoPat-

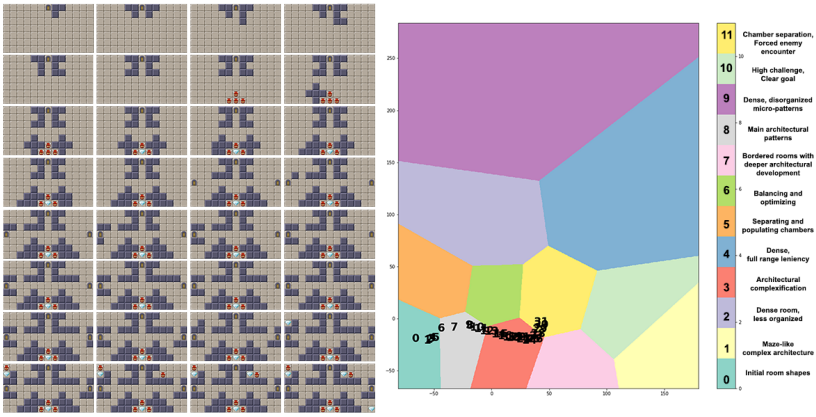


Figure 3: Example of a step by step edition sequence of a design session and it's clustering. At the left, we present the actual sequence and steps of one of the rooms in the dataset, in a 4×7 grid, starting at the top left with the first edition. At the bottom, it is the actual trajectory of the design in the cluster space. Numbered and in black, it is shown how each step of the design process is clustered by our approach

terns, #SpatialPatterns, and Symmetry. A complete description of these features can be found in [31].

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

To run all setups, data reduction algorithms, clustering algorithms, and do the internal evaluation of the clusters, we used scikit-learn machine learning toolset [32]. 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 data 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 initializing algorithm, and transforming the data into a new dataset *pca_dataset* and *tsne_dataset* per dataset. Each two-dimensional point in the new datasets represents a step in the sequences described above.

Moreover, 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++ implemented in scikit-learn, which initialize 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 cluster 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, as well as manually inspecting the rooms composing the resulting clusters.

- **Silhouette Score:** The Silhouette Score shows how similar a data point is to the cluster it is associated with, through calculating the difference between the $\overline{distance}$ from the point to the points in the nearest cluster and the $\overline{distance}$ 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:** 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, with 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:** 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.

Cluster Labelling

Table 1 shows the best performing setups according to their internal indices scores. The clusters in these setups were manually inspected in order to detect

Table 1: Best performing setups based on their internal validation and visualization of clustered data points.

Algorithm	Data	K	◊	□	△
K-Means	Tiles-PCA	9	0.43	0.73	9438.233
K-Means	Tiles-PCA	12	0.41	0.77	9436.928
K-Means	Dimensions-PCA	12	0.43	0.73	7738.343
Agglomerative single	Combined-PCA	6	0.51	0.43	38.833
Agglomerative avg.	Dimensions-PCA	6	0.44	0.67	3463.567

◊ Silhouette Score □ Davies Bouldin Index △ Calinski-Harabasz Index

the qualitative features that better define them.

When using the **Dimensions** and **Combined** datasets, the clusters do perform good, if not better, in certain indices than when using the **Tiles** dataset. However, when analysing the resulting setups, they were missing a clearer relation between the clustered rooms, which was exacerbated when analysing sequences and paths on these setups, where they missed continuity between clusters.

Conversely, given that we are creating tile-based rooms and dungeons, the features were more representative for the **Tiles** dataset, which when used, generally performed well in the evaluated internal indices, and the produced clusters meaningfully separate the data. Further, as it will be presented in Section 14.5, when clustering sequences and analyzing the cluster path of the designs, there exist a continuity between designs that supports its usability. Figure 2 shows the best-resulting cluster set found among all the experiments run.

In the figure, we have plotted on top of the clusters the labels describing in general, the content that is within them. The following is a description of the clusters and the rooms that were 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. Maze-like complex architecture: This cluster to the extreme of the architectural patterns complexity layer, relates to more highly-linear, confined and maze-like rooms.

2. 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.

3. architectural complexification: This cluster relates mostly to the complexification of wall structures by having dense wall chunks, representative architectural patterns, or symmetrical patterns.

4. 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. Dense, disorganized micro-patterns: This cluster clusters the extreme rooms that contain a high density of tiles, other than floor-tiles, without a clear structure or objective for the player.

10. High challenge, clear goal: This cluster relates to well-shaped rooms with clear wall structures and goals, towards more challenge.

11. 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.

Furthermore, 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 done with walls, from empty rooms to mazes. (b) Goal creation, enemy/treasure balance, with clusters comprehending 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 enemy and treasure addition do not necessarily need to follow any clear objective.

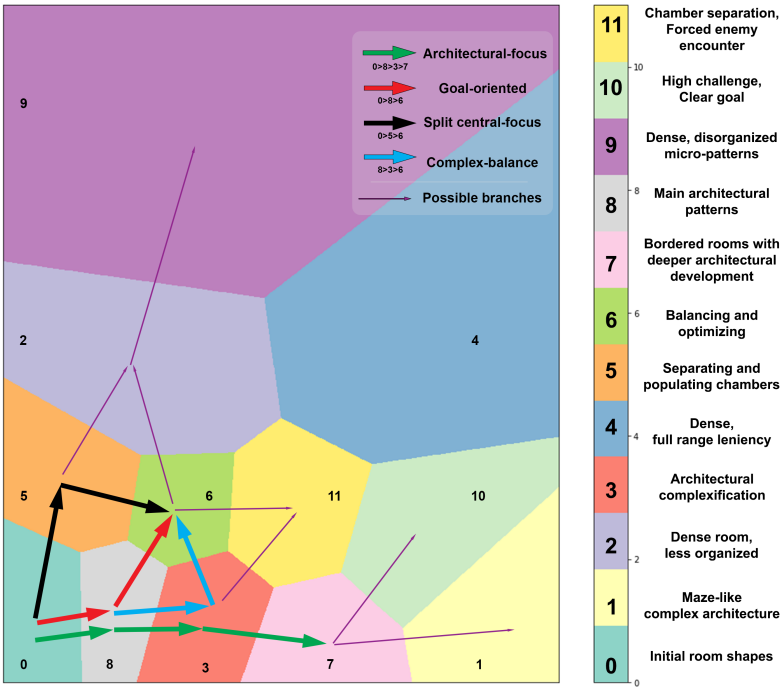
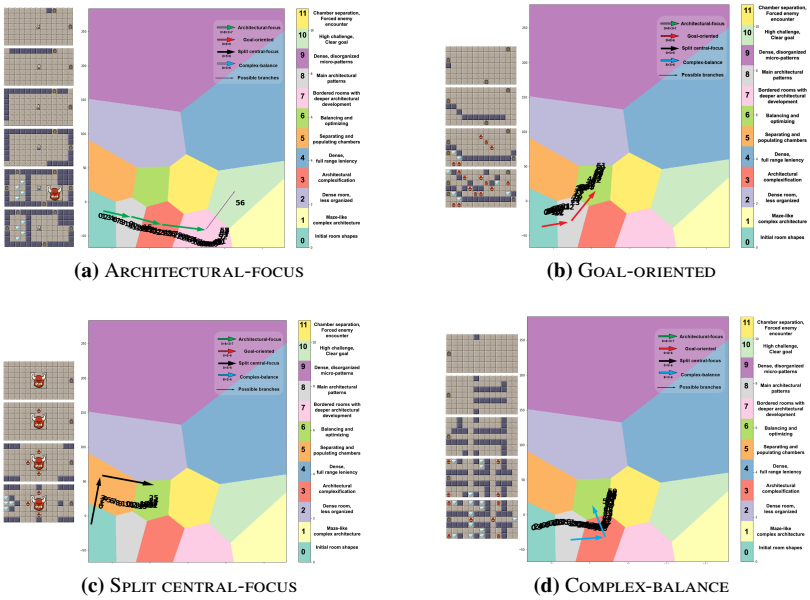


Figure 4: Final and common designer trajectories. With thick arrows it is presented the archetypical paths, calculated using the frequencies of subsequences from 180 diverse rooms. Each color represent a unique trajectory; with green the Architectural-focus, with red the Goal-oriented, with black the Split central-focus, and with blue the Complex-balance. Finally, thinner purple arrows extending from clusters traversed by the archetypical paths show the multiple possible branches that an archetypical path can deviate or extend to.

Identifying the designer in such 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, in what phase of the design process she is at the moment i.e. trying the tool or observing how the tool reacts or scraping her current goal towards a new goal within the room.

Designer Personas

Once we created, evaluated, and labeled the 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 clusters, even capturing when a designer probably applied one of the procedural suggestions due to bigger steps in the design style clusters. Further, through this process, we could understand



(a) ARCHITECTURAL-FOCUS **(b)** GOAL-ORIENTED

(c) SPLIT CENTRAL-FOCUS **(d)** COMPLEX-BALANCE

Figure 5: Examples of each of the archetypical paths from one of the 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 cluster. (a) presents the ARCHITECTURAL-FOCUS archetypical path where the focus is firstly on creating the structural design of the rooms; the design process jumps back and forth suddenly to cluster 10 (one of the possible branches) due to the designer adding a boss, and removing it immediately. (b) presents the GOAL-ORIENTED archetypical path where the design focus on a minimal structure complexity and mix between adding structural changes and enemies/treasures. (c) shows the SPLIT CENTRAL-FOCUS archetypical path where, intentionally, the designer creates a center obstacle with a boss and build around it. Finally, (d) presents the COMPLEX-BALANCE archetypical path; the design focuses on building complex, uncommon structures first and then add some goal to it with enemies and treasures, taking advantage of the spaces.

the progress of designers in their design process and represent their trajectory in relation to the traversed clusters rather than individual editions.

Unique Trajectories

Using the clusters in Figure 2, we clustered the design session of all the 180 designs and collected the unique trajectories that arose from traversing the various clusters. These unique trajectories varied in the starting point, length, and end-point, however, when analyzing the trajectories we identified common patterns among them. They had a similar shape as the following $Unique = \{0>8>4>7>10\}$, where the first and last element of the sequence are respectively, the starting- and end-points, with all the unique intermediate steps in between.

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) {5>1>3>11>9}, (b) {5>1>3>11>4 and (c) {0>1>3>11}, none of these is a perfect match in its entirety, but GSP can spot that subsequences {1>3>11}, {1>3}, {3>11}, among others, appear with frequency = 3.

Furthermore, after doing a preliminary analysis, we identified some steps that we classified as "border designs": steps that are borderline between two clusters. These *border designs* disrupted the sequence pattern mining by creating noise in the unique trajectories, specifically when these *border designs* entered a different cluster for just a few steps. Therefore, we filtered them out by applying a threshold $\theta = 3$, so that all subsequences inside one cluster with less than θ steps are removed from the main sequence. I.e, the sample trajectory {0>0>0>0>8>8>8>6>8} turns into {0>8} instead of {0>8>6>8}. Through this, we were able to reduce the noise and the search space, obtaining more 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 cluster for easier visualization and referencing.

Moreover, in the figure, it can also be observed thinner purple arrows pointing to different clusters from several of the 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>8>3>7}.

Goal-oriented

Design processes following this archetypical path, create the rooms in a more standard way, combining simpler symmetric wall structures with distributed 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-in-mind. The sequence is denoted with a red arrow in Figure 4, and following the sequence {0>8>6}.

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>5>6}.

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 {8>3>6}.

Furthermore, using these archetypical paths, we can then categorize certain clusters as key clusters or being more relevant than others based on their contribution to the paths, their frequency, and their usage. Most of the paths go through or end in cluster 6 (“Balancing and optimizing”) and cluster 8 (“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 6, 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 any of these three clusters.

Nevertheless, it is the steps in-between what 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 cluster as three other

paths, and tentatively ends in the same cluster as three other. However, the designs following Split central-focus are more different to the other trajectories, since it enters a 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 how 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.

In (a), it is shown 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. In (b), it is shown the Goal-Oriented, where the designer sketched the main shape of the room followed by alternating between enemies, treasures, and walls to design the goal of the player within the room. 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.

In (c), it is shown 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. In (d), it is shown 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.

Conclusions

This paper presents a novel approach 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 design style using the edition sequences of the collected 180 unique rooms, ending in 8196 data points, and resulting in a set of twelve cohesive, coherent, and meaningful 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 design clusters.

Designer modeling was proposed as an approach to capture multiple designer's processes to create a better workflow by Liapis et al. [18], and our work draws on many of their ideas, concepts, and goals. Furthermore, a prototype of such was implemented in the sentient sketchbook [19], where it is proposed different approaches to model style, process, and goals based on choice-based evolution and the designer's current design to adapt the provided suggestions accordingly. We propose an alternative route to designer modeling through clustering the design space and the room style based on the collected data. Moreover, we differ in the type of level design, being the sentient sketchbook a tool for strategy games [16], while EDD is a tool for adventure and rogue-like games [31]. These differences strengthen the importance and usefulness of designer modeling and highlight the holistic and generic properties of this designer-centric perspective and its possibilities.

These contributions allow us to better understand, cluster, categorize and isolate designer behavior. This is very valuable for mixed-initiative approaches, where a clear virtual model of the designer's style allows us to better drive the search process for procedurally generating content that is valuable for the designer. 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 [33].

Moreover, recognizing the designers' current style and the path taken so far, which would indicate a possible designer persona, could 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 her current location on the map and offering content suggestions that lie in the most promising clusters to be visited next. Conversely, it could 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, it is also important to observe the nature of the previous and future rooms created by a designer. Observing the dungeon as a whole, as briefly introduced in section 14.3.1, to understand the designers' intentions and goals when they proceed to create a new room is a promising future step to take with the current system.

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.

[2] A. Liapis, G. Smith, and N. Shaker, "Mixed-initiative content creation," in *Procedural content generation in games*, pp. 195–214, Springer, 2016.

[3] 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.

[4] Nintendo R&D4, "The Legend of Zelda," 1986.

[5] 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.

[6] 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.

[7] A. Liapis, D. Gravina, E. Kastbjerg, and G. N. Yannakakis, "Modelling the quality of visual creations in iconoscope," in *Games and Learning Alliance* (A. Liapis, G. N. Yannakakis, M. Gentile, and M. Ninaus, eds.), (Cham), pp. 129–138, Springer International Publishing, 2019.

[8] D. Melhart, G. N. Yannakakis, and A. Liapis, "I feel i feel you: A theory of mind experiment in games," *KI-Künstliche Intelligenz*, pp. 1–11, 2020.

[9] 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.

[10] 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.

[11] A. Alvarez and M. Vozaru, "Perceived behaviors of personality-driven agents," *Violence | Perception | Video Games: New Directions in Game Research*, pp. 171–184, 2019.

[12] M. Charity, A. Khalifa, and J. Togelius, "Baba is y'all: Collaborative mixed-initiative level design," *arXiv: 2003.14294*, 2020.

[13] 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.

[14] N. Shaker, M. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels.," in *AIIDE*, 2013.

[15] 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.

[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.

[17] K. Compton, "Casual creators." PhD Dissertation, University of California Santa Cruz, 2019.

[18] 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.

[19] 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.

[20] 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.

[21] S. M. E. D. R. Mahsan Nourani, Samia Kabir, "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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[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.

[27] 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.

[28] 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.

[29] 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.

[30] E. McMillen and F. Himsl, “The Binding of Isaac,” 2011.

[31] 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.

[32] 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.

[33] A. Khalifa, P. Bontrager, S. Earle, and J. Togelius, “Pcgrl: Procedural content generation via reinforcement learning,” 2020.

ISBN 978-7877-139-4 (print)
ISBN 978-7877-140-0 (pdf)

MALMÖ UNIVERSITY
205 06 MALMÖ, SWEDEN
WWW.MAU.SE