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

Alberto Alvarez
Malmö University
Malmö, Sweden
alberto.alvarez@mau.se

Eric Grevillius
Malmö University
Malmö, Sweden
ericgrevillius@gmail.com

Elin Olsson
Malmö University
Malmö, Sweden
elinolssonq@gmail.com

Jose Font
Malmö University
Malmö, Sweden
jose.font@mau.se

ABSTRACT

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

CCS CONCEPTS

• **Applied computing** → **Computer games**; • **Theory of computation** → *Grammars and context-free languages.*

KEYWORDS

Quest Generation, Procedural Content Generation, Mixed-Initiative Co-Creativity, Grammars

ACM Reference Format:
Alberto Alvarez, Eric Grevillius, Elin Olsson, and Jose Font. 2021. Questgram [Qg]: Toward a Mixed-Initiative Quest Generation Tool. In *The 16th International Conference on the Foundations of Digital Games (FDG) 2021 (FDG'21), August 3–6, 2021, Montreal, QC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3472538.3472544

1 INTRODUCTION

Defining quests and related concepts have been the focus of considerable research, where quests have been related to tasks, challenges, rewards, or as a storytelling device adding nuances to what a quest is [12, 16, 50, 54]. Most games have some quest driving the game's plot and gameplay. Adventure games, action-adventure games, and role-playing games (RPG) are among the main genres using quests [24], where most of these genres take place or containing some type of dungeon such as *The Legend of Zelda*, *Skyrim*, or *The Binding of Isaac*. Dungeons as game content can be defined as a single level or set of levels containing enemies, treasures, hidden passages, puzzles, decorations, or Non-playable characters (NPC), thus creating space that allows the player to explore the unknown areas [15]. Dungeons are a popular level design, especially within PCG [29, 43, 52], where it has been present ever since the 1970s in games such as *Rogue*.

The increasing usage of Procedural Content Generation (PCG) in both research and industry [28, 29] has shown successful results regarding the efficiency of the game development process [38] but also to generate a big amount of variation in games, increasing their replayability [44]. PCG can generate game content quickly such as missions and levels [17], content adapted to players [23], or data-driven generation [20, 49]. Narrative and quest generation as objectives and goals has also been the focus of PCG [9, 18, 36], where the aim has been to capture and use quest concepts and patterns to approach the generation, such as the work by Trenton et al. [50], Kreminski and Wardrip-fruin [26], Smith et al. [45] or Doran and Parberry [16].

Nevertheless, much of the content is still best made by humans, especially when subjective evaluations are needed [43]. To cope with this, one could use a mixed-initiative approach. Mixed-initiative Co-creativity (MI-CC) was introduced by Yannakakis et al., where both human and AI co-create and -design some game facet with a proactive initiative [51]. MI-CC has been explored mainly for level design in tools such as the Sentient Sketchbook [34], Tanagra [47], Morai Maker [21], or the Evolutionary Dungeon Designer (EDD) [7]. EDD lets the user design interconnected room in a dungeon while receiving room suggestions adapted to their creation. Questgram is implemented in EDD, taking advantage of its level design capabilities and mixed-initiative approach.

This research takes the quest analysis, quest patterns, and quest grammars identified by Doran and Parberry [16], implements it

FDG'21, August 3–6, 2021, Montreal, QC, Canada

ACM ISBN 978-1-4503-8422-3/21/08…$15.00
https://doi.org/10.1145/3472538.3472544

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

2 RELATED WORK

Howard defined quests as "... a conceptual bridge that can help to join together many two-part or binary pairs [...] these include game and narrative, gaming and literature, technology and mythology and meaning and action [24]". Howard argues that quests unify both meaning and action, where meaning hails from strategic actions with thematic, narrative, and personal implications; and actions being those that are meaningful for the player on the level of ideas, personal ambitions, benefits to society, and spiritual authenticity [24]. Aarseth describes quests as concrete and attainable goals, and such can be hierarchic, concurrent, serial, or a combination of those. Further, Aarseth describes three basic quest types *Time-*, *Place-*, and *Objective-oriented*, which can also be combined to form seven different quest types [1]. Questgram is based on the quest analysis and proposed grammar by Doran and Parberry, constructed and extracted from analyzing over 750 quests from four RPGs and where they defined quests as a task given to the player that challenges them to complete some goals in exchange for some reward [16]. While helpful to understand quests as a whole, these definitions create a sense of ambiguity over different concepts surrounding quests. Yu et al. [54] proposed a generic quest definition in games that aims at unifying related concepts that appear in most of other's work, clearing ambiguity and easing it's use in PCG quest generation tools. Formally, they define a quest as $Q = \langle T, \leq, R \rangle$, where a quest Q is a partially ordered set $\leq$ of tasks T to be done to receive one or more rewards from a set R, which usually are in-game items.

2.1 Story and Quest Generation

Quests are fundamental elements in most games, driving the plot and player actions and providing goals and tasks to engage players with the game and the narrative. Doran and Parberry analyzed quests in four RPGs and found nine different "motivations" from NPC's, which resulted together with a specific strategy in a "verb-noun" pair, for example, "steal supplies" or "attack enemy". They used this grammar to generate quests while the user chose between nine identified NPC motivations [16]. Based on Doran and Parberry's action classification, Breault et al. developed an engine capable of creating quests similar to human-made ones, and since the engine generates quests based on the world state at the time of generation, the creation of possible quests increases as the game progresses [12].

One key characteristic in games is that they are interactive, and as such, can present choices to players. However, quests do not tend to provide such choice, especially in RPGs; rather, it is common that they are limited as a series of steps to follow. An interesting approach is Questbrowser [48], a quest design brainstorming tool where the designer can query the system for ideas, alternatives, and possibilities on elements or concepts that foster designers' creativity and help make quests playable (i.e., adding choice for players). However, presenting choices to players could create competing objectives for designers as they want to impose their narrative but, at the same time, want to create adaptable experiences for players.

Planning algorithms are a common technique to compose stories and quests meaningfully and with some partial-ordering [39, 53], focusing and optimizing character believability together with multi-agent systems [41, 42], replicating common quests and quest patterns [11, 16, 50], or identifying fundamental units and assembling them based on various pre-conditions [19, 26]. Kreminski and Wardrip-Fruin [26] mapped and compared multiple storylets-based systems and proposed the use of storylets, which are discrete, atomic, and recombinable narrative chunks, to assemble narratives based on a set of preconditions to create different narrative structures. Storylets were used by Garbe et al. in the StoryAssembler to generate dynamic narratives, which attempts to create a valid story with a planner using a set of provided storylets and storytelling goals that the planner uses as objective [19].

Moreover, Questgram functions within EDD's level design tool, which means both would function in relation to each other. Kybartas and Bidarra discussed the relation between plot and space, focusing on the degree of automation for story elements. This resulted in six categories: *automated space*, *constrained space*, *space simulation*, *space modification*, and *manual space that builds a gradient between automatic and manual generation* [27]. Ashmore and Nitsche investigated a player-centric quest generation, where the progression through level generation is achieved with a "key and lock" structure, which results in a bridge between the generated space and the quests [13].

Dormans and Bakkes used two grammars to generate both missions and game space, where the latter was informed by the first. Missions are generated using graph grammars, creating a non-linear structure suited for exploration, while extended shape grammar generates the corresponding space required [17]. Further, Flodhag et al. use the information from levels co-created in *EDD* and categorize them based on the meso-patterns within them to present a set of main and side objectives to designers in their dungeon [18]. Hartsook et al. explore the creation of complete RPGs from a story created by either a computational system or human-authored and a set of player preferences. Their approach creates and represents game worlds as transition graphs based on a story composed of plot points, player's playstyle preferences, and designer constraints [22].

2.2 Mixed-Initiative Co-Creativity

MI-CC is a paradigm where both humans and AI have a proactive initiative in the collaboration to co-create some creative content [30, 51]. Both human and AI leverage on each other's strengths to achieve the task and continuously negotiate to determine roles; thus, collaborating as a team [3]. One critical aspect of MI-CC systems is the link between these systems and theories of computational and human creativity, where a main focus of MI-CC is on fostering human's creativity while reducing their workload [5, 32].

The *Sentient Sketchbook* is an MI-CC tool for the co-creation of strategy games where the designer focused on creating low-resolution sketches, and the computational designer suggested variations generated with different evolutionary algorithms [34]. Cicero is an MI-CC system that helps designers create complete games using a recommender system and A-Priori to suggest what content might be added next regarding sprites, mechanics, rules, or interactions [35]. Another interesting MI-CC system is *Why Are We Like This? (WAWLT)* where two players can develop a story transcript while supported by an AI system with tools to inspect the story world and with suggestions to direct the plot [25].

EDD is an MI-CC system where the designer can create interconnected rooms that compose a dungeon while receiving a set of diverse suggestions using the IC MAP-Elites evolutionary algorithm driven by level design patterns and considering the designer's current design. The designer can interact with the suggestion system by locking tiles, editing their design, and selecting and interacting with hyper-parameters of IC MAP-Elites [4, 7]

3 QUEST GENERATION

Questgram is a quest generation tool that lets the designer compose one long sequence of quest actions to create an overarching objective for the dungeon they are creating. These quest actions are based on the quest analysis and classification and produced grammar by Doran and Parberry [16]. Questgram builds on top of EDD extending its level design and generation capabilities with a mixed-initiative quest editor and takes advantage of its mixed-initiative perspective and the level design system.

EDD was extended with some key elements such as a new quest editor view depicted in figure 1a, and two new generic tiles; an NPC acting as quest giver and target, and a quest item, which is the subject of many quests. These two new tiles were kept as generic as possible for future systems to have the responsibility of handling what type of NPC and object should replace those, similarly as with the other tiles in EDD such as the generic enemy, boss, and treasure. These tiles, together with the pre-existing enemy and boss tiles, have been intertwined with the actions, resulting in the "unlocking" mechanism of different quests, which can be observed in table 1. It must be noted that while Questgram integrates and utilizes the different features and tiles of EDD's level generation facet, there is no integration of the new tiles and quests with EDD's evolutionary algorithm IC MAP-Elites [7]. This is left for future work.

While games can be either linear, semi-open, or open, with branching narratives and the design structured by the types of quests featured in a game [1], with concepts such as kernels and satelites [2]; our approach only allows for the creation of a single overarching quest.

3.1 Quest Actions

Doran and Parberry identified 19 different actions to be used as quest actions [16], which we implemented and where each has its contextual prerequisites to be able to add them. In some cases, we have decided regarding if the actor represented in the action is friendly, e.g., NPC, or hostile, based on the tool's nature and the levels created. These actions are available for both the grammar and the designer to create as many steps as wanted in the overarching quest. The actions, their original prerequisite, and the domain-specific prerequisite are depicted in table 1.

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

3.2 Quest Grammar

The system employs a generative grammar, specifically Lindenmayer Systems (L-Systems) [40] to generate the different set of quests using the production rules depicted in table 2. The production rules are divided into two categories: 1) motivations for NPCs to start a quest such as *knowledge* where the focus would be to create quests with more passive actions or *reputation* where the focus would be to kill some enemy to gain reputation with some NPC. 2) Non-motivation rules related to the development of quests (i.e., non-terminal symbols) such as "go_to" or "get". In table 2 non-terminal symbols are represented with "<" and ">", and terminal symbols simply list the action.

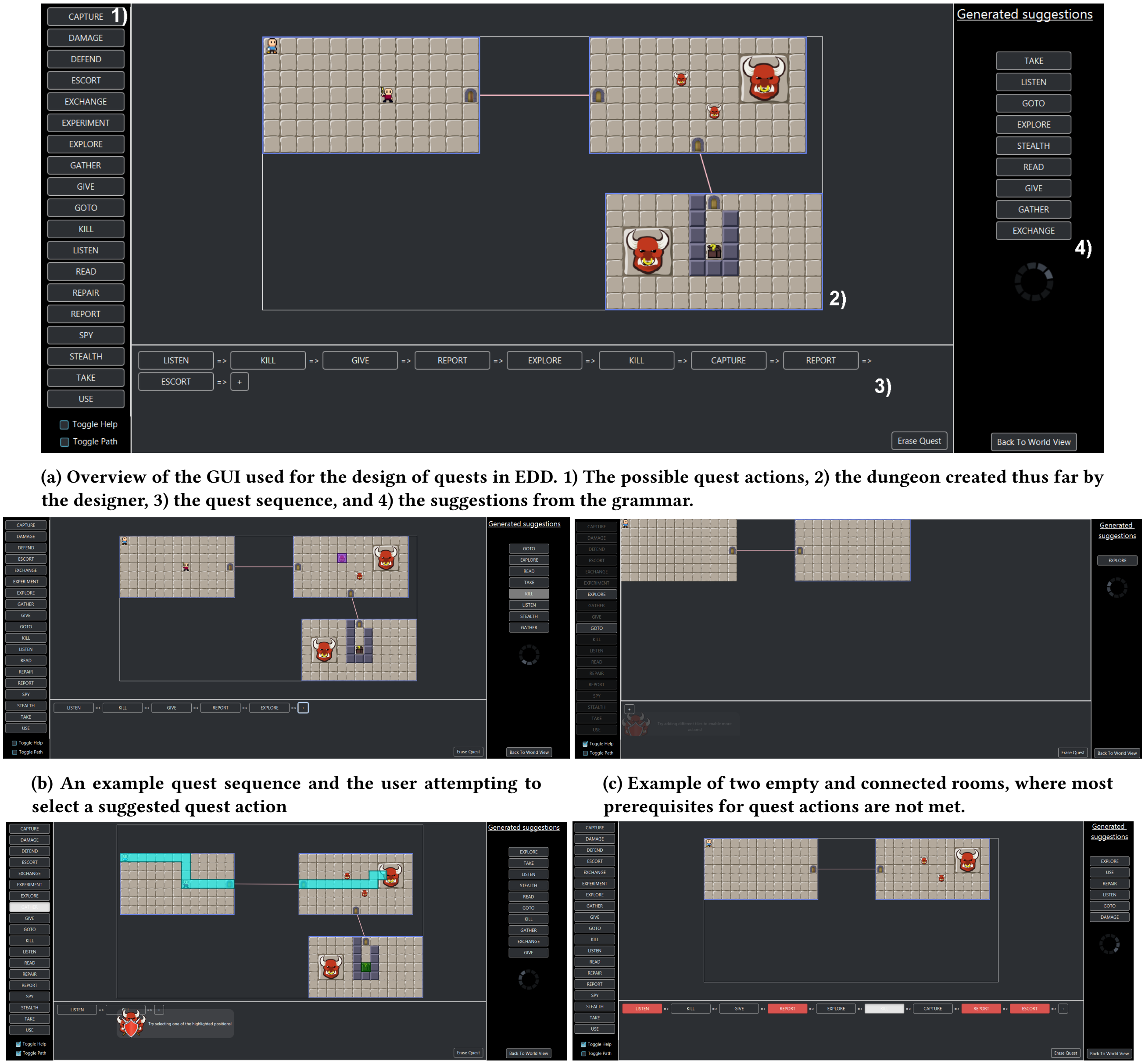
The system can be used for generating complete quests on its own, which select one of the NPC motivations as an axiom or with the designer in a mixed-initiative approach. As a mixed-initiative approach, the designer can manually create a quest sequence while the system uses the current quest sequence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

Quest actions are suggested to continue the current sequence and append a new action at the end, or they can be used to replace an action in any position of the current sequence. For both, the system continuously produce quests using the grammar and filters out those that do not match the designer's sequence up to the position where they wish to change a quest action. In this way, the designer can choose suggestions for either continuing and finishing the quest or replacing existing parts for other valid actions.

3.3 Workflow

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

The designer cannot add any quest action, which prerequisite has not been fulfilled yet as described in table 1 and exemplified



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

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

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

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

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

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

in figure 1c. Quest actions need to be linked to some actual tile representation in the dungeon. For instance, a "KILL" action requires selecting an enemy, while the "GO TO" requires any floor tile. Therefore, once the designer adds a new quest action, they must choose which tile is linked to this, presented to the designer in green. Similarly, when a quest action is suggested, the system randomly picks an available tile shown in purple to the designer as shown in figure 1b.

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

Table 1: displaying the actions together with Doran and Parberry's [16] prerequisites and how the actions and the previously mentioned prerequisites have been implemented in EDD. This indirectly explains the "unlocking" - describing what tiles that must be placed for an action to be available. Note that "Goto" & "Explore" do not have any special tile prerequisites besides available floor.

Action	Prerequisites in [16]	Prerequisites in EDD
Capture	"Somebody is there"	A NPC or boss/enemy must be placed.
Damage	"Somebody or something is there"	An item or NPC must be placed.
Defend	"Somebody or something is there"	An item or NPC must be placed.
Escort	"Somebody is there"	A NPC must be placed.
Exchange	"Somebody is there, they and you have something"	A NPC and an item must be placed (requires two positions).
Experiment	"Something is there"	An item must be placed.
Explore	"none"	An available floor tile.
Gather	"Something is there."	An item must be placed.
Give	"Somebody is there, you have something.“	A NPC and an item must be placed (requires two positions).
Goto	"You know where to go and how to get there.“	An available floor tile.
Kill	"Somebody is there.“	A boss/enemy must be placed.
Listen	"Somebody is there.“	A NPC must be placed.
Read	"Somebody is there.“	A NPC must be placed.
Repair	"Somebody is there.“	A NPC must be placed.
Report	"Somebody is there.“	A NPC must be placed.
Spy	"Somebody or something is there.“	A NPC or boss/enemy must be placed.
Stealth	"Somebody is there.“	A NPC or boss/enemy must be placed.
Take	"Somebody is there, they have something.“	A NPC and an item must be placed (requires two positions).
Use	"There is something there.“	An item must be placed.

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

4 EVALUATION

Questgram has undergone a two-fold evaluation, top down (expressivity analysis) and bottom up (user study), as suggested by Shaker et al [43].

4.1 Expressive Range Analysis

Expressive Range Analysis visualizes the expressivity and diversity of the generator and measures variations in the generated content according to specific metrics [46]. In our case, these metrics are quest length and actions. With them, we visualize each action's probability to be included in a quest of any given length and the existing dependencies between the actions and the grammar productions.

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

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

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

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

4.2 Experiment Discussion

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

Table 2: displaying the grammatical rules. The columns marked with asterisks are identified as "motivations" by Doran and Parberry [16], but are used as a starting point for the quests. The "<>" indicates the next production rule to be taken, and actions without "<>" is the terminating action.

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

available floor tile. While most of the other actions require NPCs, items, or both, the existence of available floor tiles is several times higher than any of those elements. "Go_To" is the other action with such a simple prerequisite, and its chance to appear is also high. As opposed to "Explore", it raises from 0% to 28% in the later quest steps. Though both actions imply space exploration, "Explore" is more commonly used in the early stages of a quest, when the map remains uncharted, whereas "Go_To" gets used more in the later stages, where some map locations have been already visited. It is also remarkable that the first action in 87% of the quests is "Explore", while the only other actions that appear in the first step (in a much lower degree) are "Use", "Damage", and "Repair". No other actions are used as quest starters. This "Explore" and "Go_To" dominance can also be observed in table 2, where "Go_To" appears in 77% of the production rules, sometimes more than once per production, and "Explore" has a 50% chance to appear per "Go_To".

The appearance rate of the combat-related actions, "Damage" and "Kill", is relatively low, though their peak rates are located in shorter quests (Figure 2a). "Damage" has a 21% chance to appear in quests of length 3, whereas "Kill" has its peak at 7.8% in 5-step quests. This can be extended to other actions such as "Use", "Give", "Repair", "Gather", and "Exchange", suggesting that this subset of actions is much more likely to appear in short, quickly solvable quests. Nevertheless, all of them still appear in longer quests at stable rates, though movement actions are much more predominant in the long run.

Some actions are very underrepresented regardless of quest length or step number, as is the case for "Defend", "Report", "Experiment", "Escort", "Capture", and "Spy". This implies that these actions have very little chance to be suggested at any quest step, so it is more likely to end up in a quest if manually added by the designer. A future evaluation of the utility of these actions seems interesting in light of these results.

Finally, Figure 3 indicates a clear bias in the grammar towards exploration ("Explore" and "Go_To"), as one or both appear, at least once, in any of the most commonly generated subsequences. The actual relevance of these dominant actions should also be evaluated for the grammar's future development.

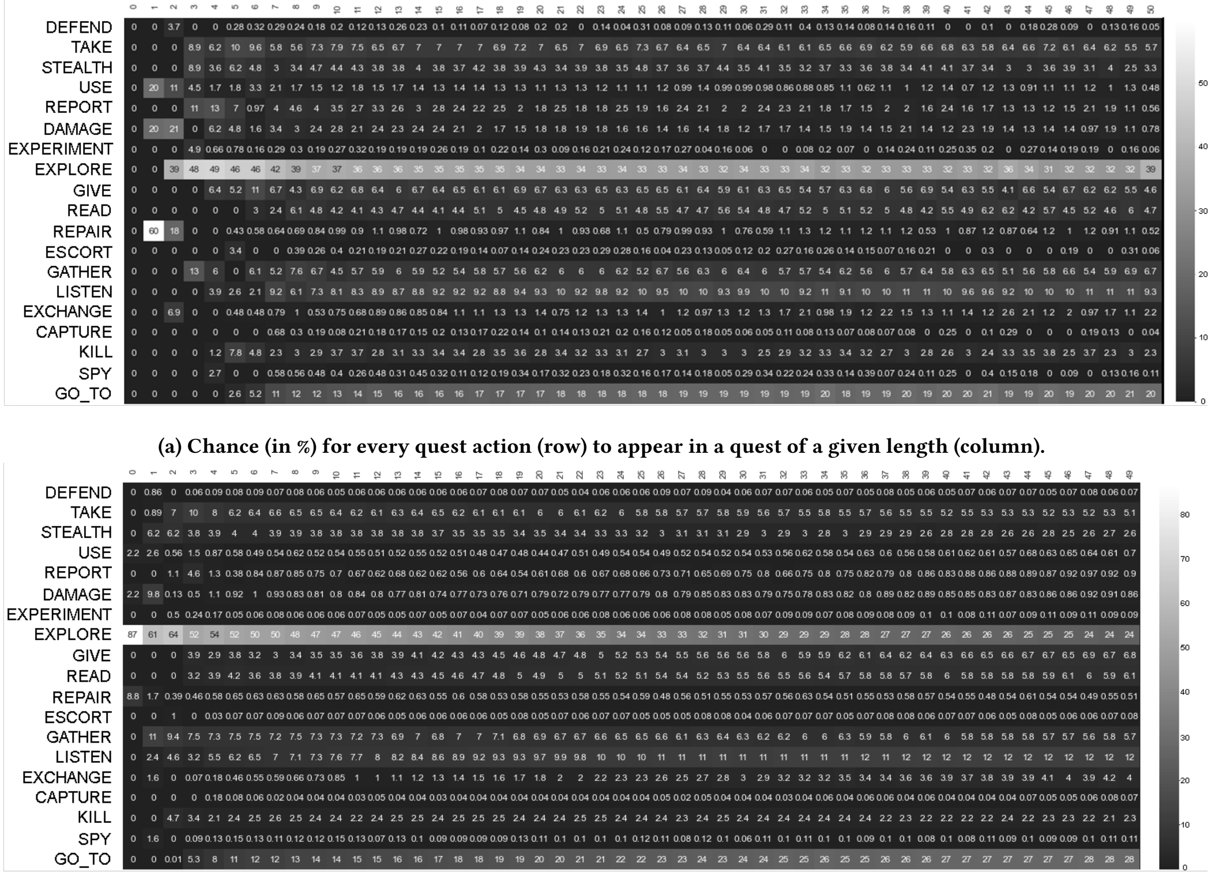
4.3 User Study

Six participants tested our tool following three pre-designed tasks and questionnaires to evaluate Questgram's usability, functionality, and usability. They were all given a document describing the study's purpose and aim, a brief introduction to EDD, and the interview overview. The users were then asked to complete three tasks that covered the tool's functionality and different approaches to creating quests. The tasks were to 1) manually create a quest, 2) automatically create a quest, and 3) create a quest through mixed-initiative. They were also asked to create a dungeon that suited their preferences and objectives before creating quests. The questionnaire consisted of 17 closed-ended questions, and the rest were open-ended. The interview began with a questionnaire with six questions about the users' background and experience within game development and finish with questions about their experience and opinions on the tool. Both the questionnaire and interview followed guidelines described by Oates [37].

The participants were selected through convenience sampling and were game developers working in game and level design (2) and game development alumni (4), without any experience with EDD or mixed-initiative tools. Five out of six have played dungeon/adventure games and have developed some game with quests and missions, while only two out of six have developed dungeon style games.

4.3.1 *Manual Quest Creation.*

Participants reported that the tool was easy to use, clear, intuitive, and while simple and basic, it had enough building blocks for them to create their objective quest. Positive feedback was also given regarding the UI, integration with the rest of EDD functionalities, clarity of the quest action concerning the quest, and making the tool overall more interesting. However, some participants expressed confusion when quest sequences became too large as they would have preferred to separate the quests into sections or subquests. Another concern expressed was the inability to change the order of already created quests without redoing the whole quest sequence.



(a) Chance (in %) for every quest action (row) to appear in a quest of a given length (column).

(b) Chance (in %) for every action (row) to appear at the Nth step of a quest (column), regardless its length.

Figure 2: Results from the Expressive Range Analysis

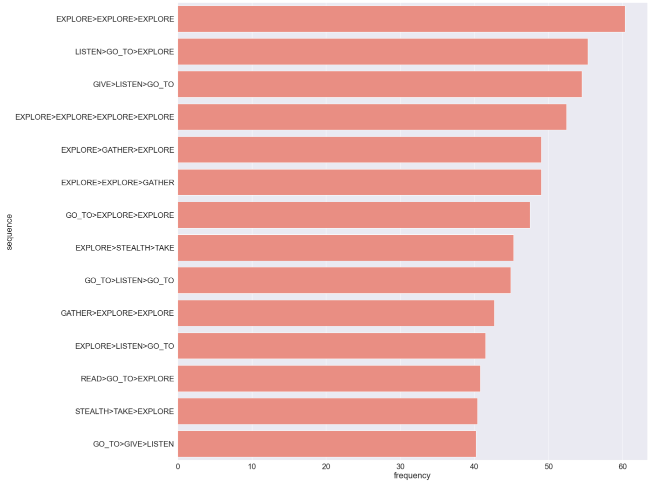


Figure 3: Most commonly generated subsequences, with a minimum size of 3.

4.3.2 *Automatic Quest Creation.* Some participants reported that the system showed potential and a good addition to the manual creation, especially for the creation of side quests such as how similar systems work in *Skyrim*, and for learning how to use the tool as some kind of tutorial. Nevertheless, most participants remarked the system as random and illogical regarding random tile picks connected with quest actions as the system picked farther away NPC and targets with no purpose. In addition, participants felt that a system like this complicated the creation and took the freedom of creating their world and ideas.

4.3.3 *Mixed-Initiative Quest Creation.* The participants described the system as helpful and useful; pointing out that the main advantages and potentials were related to when they reached an inspiration blockage as the designer could get inspired by the suggestions; to allow designers to focus on key parts of a quest sequence, and the speed gained to create quests "in just a matter of moments." While most feedback was positive, there were still concerns among participants regarding the system's cohesiveness as it could feel hard to make the suggestions cohesive with what the participant had in mind. Nevertheless, a majority of the participants experienced that both the manual and automatic complemented each other.

4.3.4 Automatic Suggestions. Participants generally described the automatic suggestions as useful to gain inspiration, keep the quest creation diverse, and learn what could be created, rather than useful to replace or add to their work. For instance, one participant said that "[the system] suggested to capture a monster which had thought about killing. The "capture" option might be more interesting and might have been an option I had otherwise overlooked." Similarly, another participant pointed out that the automatic suggestions "... were useful in getting inspiration for quests, and to learn the program and what kinds of quests I was actually able to make." Usually, designers have a predetermined idea on how they would like the narrative to unfold. However, based on the responses we received, the system gave a different perspective to the users on what they could create or how they could continue a quest, which makes the tool useful for brainstorming quest design similar to tools such as Questbrowser [48], albeit constrained to the possible quest actions.

Still, participants preferred to use the system as inspiration rather than effectively incorporating changes to the quests. This was mainly due to the suggestion not feeling cohesive enough with what participants created until then, and the random tiles picked for the quest action. For instance, receiving a "GO TO" suggestion to a random tile on the opposite side of the level and not near the player or the previous quest action.

4.3.5 Quest Actions. Quest actions and their goals were perceived as easy to understand and suitable for the type of game they were creating, except for "experiment," "stealth," and "spy," since they felt ambiguous and not clear for some participants. Fulfilling some quest actions' prerequisites was somewhat obscure, and some participants needed to go through trial and error to gain access to the quest action. However, once they fulfilled the prerequisites, it was clear and made sense in the context.

4.3.6 Usability. All participants described the tool as a useful addition for game developers when developing dungeon games but with different arguments and situations on when it would be useful. For instance, to create mundane quests in games with a grinding flow such as *Diablo*, to fast-prototype ideas and systems to give insights into what it might be possible and what might work, and to complement the creation of content and quest-design. Some participants expressed that making the game playable is a must to get the full benefit from the tool.

4.3.7 Creativity. Most of the participants reported that they experienced increased creativity when using mixed-initiative creation. The participants described that it helped when they got stuck, and it showed different alternatives and routes they did not previously consider. Further, one participant explained that they could make more creative decisions and not "staying safe" and adding "extra steps" without any effort. For instance, using a spy before a kill, thus prolonging the sequence by an extra action. Two participants highlighted that while they did not experience increased creativity, they saw the tool as useful when no new ideas are there given the possible "out of the box" suggestions.

4.3.8 Overall Experience and Missing Features. Some participants expressed that the tool was useful for someone in the gaming community, but it would be hard to grasp for someone unfamiliar with the concept. Further responses were that it was simple to work with, felt scalable, and the software's recommendation felt useful when designing a quest. Additional response from one participant, who, despite having no prior experience in either creating or playing dungeon games, said it was a fun introduction to the genre and that it was easy to learn and understand. Further, they explained that the simple workflow inspired them to continue creating. In addition, further responses were that the software was quick, feature-rich, simple to use, and got their creativity flowing.

In general, the participants expressed that more interaction with the quest sequence itself, such as changing the order of subsequences, adding quest actions in arbitrary parts, having separate quests, or knowing which quests were manually and automatically, would improve the system considerably.

4.4 User Study Discussion

Since we use the work by Doran and Parberry [16] as a base for the quest generation, this research indirectly tests their quest patterns and their applicability into a mixed-initiative tool. We leveraged on these quest patterns similar as others have on quest patterns [45, 50], and how EDD leverages on level design patterns for the level generation [6, 10]. The use of quest patterns greatly improves the communication with the designers as they can use concepts they feel comfortable with and relate better to the content they create. All of our participants confirmed the previous statement by pointing out how the tool was straightforward, easy to understand with quests actions to be found in any other game type, and easy to use even if they had never used a mixed-initiative tool.

Furthermore, while relevant, the suggestions by the system felt impractical mainly and according to participants because the system randomly assigned tiles for the suggested quest action, which limited the tool's perceived usability. Another reason is the use of abstract quest actions. On the one hand, this allows us to disconnect the system from specific implementations and gameplay functionalities of the quests, creating and representing a more generic system. On the other hand, this resulted in a lack of thematic and concrete elements such as NPC roles or defined plot lines and plot elements to follow. These make it harder for designers to contextualize their creations and why the system recommends specific quest actions.

Nevertheless, the system was helpful for creativity support and design aid as the suggestions were used as an inspiration to what was possible and what to do next rather than using the actual suggestion. Participants only applied suggestions when the next step felt mundane, and the system suggested a logical position. Leveraging on the human designer for deciding location, while the system provides the quest actions that would follow a more typical quest based on the quest patterns would probably compose a better collaboration.

Some feature suggestions such as separating quests or reordering the quest sequence would have improved the user experience considerably, as after manually placing just a few quest actions, the quest became harder to approach. Even if the system was tested to generate 50 quest actions as explained in section 4.1, this might be impractical for human designers. An interesting suggestion was to use color tags or similar to understand which agent (Human or machine) created the quest. Then, designers could also use this as a way to understand decisions made by the system and for the system to create a model of the current designer's quests. This could also be used for a collaborative tool where several designers interact with each other and a centralized system; for a more crowdsourced approach such as the one proposed by Charity et al. [14].

5 CONCLUSIONS AND FUTURE WORK

This paper presents Questgram [Qg], a quest generation tool with a mixed-initiative approach integrated into the Evolutionary Dungeon Designer. Questgram lets a human designer co-create an overarching quest that fits in a dungeon level, as the dungeon is being developed in the designer. Both map and quest are designed in parallel and with the suggestions provided by EDD. Quests make use of Doran and Parberry's quest structure as production rules in a grammar so that all quests are well-formed with respect to the grammar and the level landscape. We show results from a two-fold evaluation, an expressive range analysis, and a user study.

The expressive range analysis shows several dominant quest actions and structures, though all types of actions could be generated at a wide range of quest lengths. The mixed-initiative approach was positively met by the user study participants, along with the manual creation. However, automatic creation and automatic suggestions received a mixed response, mainly because of a random placement of position on the actions and the use of abstract quest actions. The tool's overall response was positive, and a majority of the participants reported increased creativity while using the tool. Many participants expressed its usability to gain inspiration, as a solution to inspiration blockages, and as a resource-efficient tool for game developers to use. None of the testers noticed the dominance of some actions detected in the expressive range analysis.

This inspirational use points towards the need to explore other fundamental and more useful ways to establish effective MI-CC workflows where systems can adapt and be effectively employed and used. For instance, some interesting future paths would be to explore the creation of more adaptive collaboration that considers the designer's style or to give more autonomy to the AI to have more participation in the creative process and its effects. Within this, one interesting area is the one of eXplainable AI for Designers [55] where the goal is to achieve system explainability to improve the collaboration and interaction between human and AIs.

This research sets the first step toward intertwined story and level mixed-initiative generation on EDD, and future work could be to incorporate quest elements as input in the level generation process so that quests and levels reciprocally influence their generative processes [33]. Furthermore, adding semantic evaluation on the generated suggestions would allow Questgram to generate interconnected quests that make sense as subsequent parts of an overarching story plot involving game elements, as well as adding a natural-language generation layer to enhance quests with the automatic generation of detailed descriptions and narratives. Another interesting future research would be to create designer models to adapt quest suggestions to the designer's particular style [8, 31]. Finally, more extensive user studies will be conducted to analyze further the tool's usability and intuitiveness.

REFERENCES

[1] Espen Aarseth. 2005. From hunt the wumpus to everquest: introduction to quest theory. In *International Conference on Entertainment Computing*. Springer, 496–506.

[2] Espen Aarseth. 2012. A Narrative Theory of Games. In *Proceedings of the International Conference on the Foundations of Digital Games* (Raleigh, North Carolina) *(FDG '12)*. ACM, New York, NY, USA, 129–133. https://doi.org/10.1145/2282338.2282365

[3] James F. Allen. 1999. Mixed-Initiative Interaction. In *Intelligent Systems, Trends and Controversies section*. 14–16.

[4] Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, and Simon Johansson. 2018. Assessing Aesthetic Criteria in the Evolutionary Dungeon Designer. In *Proceedings of the 13th International Conference on the Foundations of Digital Games (FDG '18)*.

[5] Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, Chelsi Nolasco, and Axel Österman. 2018. Fostering Creativity in the Mixed-initiative Evolutionary Dungeon Designer. In *Proceedings of the 13th International Conference on the Foundations of Digital Games (FDG '18)*.

[6] Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius. 2019. Empowering Quality Diversity in Dungeon Design with Interactive Constrained MAP-Elites. In *2019 IEEE Conference on Games (CoG)*.

[7] Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius. 2020. Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions. *IEEE Transactions on Games* (2020), 1–1. https://doi.org/10.1109/TG.2020.3046133 Preprint available: https://arxiv.org/abs/2003.03377.

[8] Alberto Alvarez, Jose Font, and Julian Togelius. 2020. Designer Modeling through Design Style Clustering. *Arxiv* (2020). Preprint available: https://arxiv.org/abs/2004.01697.

[9] Prithviraj Ammanabrolu, William Broniec, Alex Mueller, Jeremy Paul, and Mark Riedl. 2019. Toward Automated Quest Generation in Text-Adventure Games. In *Proceedings of the 4th Workshop on Computational Creativity in Language Generation*. Association for Computational Linguistics, Tokyo, Japan, 1–12. https://www.aclweb.org/anthology/2019.ccnlg-1.1

[10] Alexander Baldwin, Steve Dahlskog, Jose M. Font, and Johan Holmberg. 2017. Towards Pattern-based Mixed-initiative Dungeon Generation. In *Proceedings of the 12th International Conference on the Foundations of Digital Games (FDG '17)*. ACM, New York, NY, USA, Article 74.

[11] Vincent Breault, Sebastien Ouellet, and Jim Davies. 2018. Let CONAN tell you a story: Procedural quest generation. arXiv:1808.06217 [cs.AI]

[12] Vincent Breault, Sébastien Ouellet, and Jim Davies. 2021. Let CONAN tell you a story: Procedural quest generation. *Entertainment Computing* 38 (2021), 100422. https://doi.org/10.1016/j.entcom.2021.100422

[13] Ashmore Calvin and Nitsche Michael. 2007. The Quest in a Generated World. In *DiGRA &#3907 - Proceedings of the 2007 DiGRA International Conference: Situated Play*. The University of Tokyo. http://www.digra.org/wp-content/uploads/digital-library/07311.20228.pdf

[14] Megan Charity, Ahmed Khalifa, and Julian Togelius. 2020. Baba is Y'all: Collaborative Mixed-Initiative Level Design. arXiv:2003.14294 [cs.HC]

[15] Steve Dahlskog, Julian Togelius, and Staffan Björk. 2015. 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*, José Pablo Zagal, Esther MacCallum-Stewart, and Julian Togelius (Eds.). Society for the Advancement of the Science of Digital Games. http://www.fdg2015.org/papers/fdg2015_paper_30.pdf

[16] Jonathon Doran and Ian Parberry. 2011. 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* (Bordeaux, France) *(PCGames '11)*. ACM, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/2000919.2000920

[17] Joris Dormans and Sander Bakkes. 2011. Generating missions and spaces for adaptable play experiences. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (2011), 216–228.

[18] Alexander Flodhag, Simon Tolinsson, Alberto Alvarez, and Jose Font. 2020. 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)*. ACM, New York, NY, USA, 4 pages.

[19] Jacob Garbe, Max Kreminski, Ben Samuel, Noah Wardrip-Fruin, and Michael Mateas. 2019. StoryAssembler: An Engine for Generating Dynamic Choice-Driven Narratives. In *Proceedings of the 14th International Conference on the Foundations of Digital Games* (San Luis Obispo, California, USA) *(FDG '19)*. Association for Computing Machinery, New York, NY, USA, Article 24, 10 pages. https://doi.org/10.1145/3337722.3337732

[20] Michael Cerny Green, Gabriella A. B. Barros, Antonios Liapis, and Julian Togelius. 2018. DATA Agent. In *Proceedings of the 13th International Conference on the Foundations of Digital Games* (Malm&ouml;, Sweden) *(FDG '18)*. ACM, New York, NY, USA, Article 19, 10 pages. https://doi.org/10.1145/3235765.3235792

[21] Matthew Guzdial, Nicholas Liao, and Mark Riedl. 2018. 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.*

[22] Ken Hartsook, Alexander Zook, Sauvik Das, and Mark O. Riedl. 2011. 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*, Sung-Bae Cho, Simon M. Lucas, and Philip Hingston (Eds.). IEEE, 297–304. https://doi.org/10.1109/CIG.2011.6032020

[23] E. J. Hastings, R. K. Guha, and K. O. Stanley. 2009. Evolving content in the Galactic Arms Race video game. In *2009 IEEE Symposium on Computational Intelligence and Games.* https://doi.org/10.1109/CIG.2009.5286468

[24] J. Howard. 2008. *Quests: Design, Theory, and History in Games and Narratives.* CRC Press. https://books.google.se/books?id=LFvNBQAAQBAJ

[25] Max Kreminski, Melanie Dickinson, Michael Mateas, and Noah Wardrip-Fruin. 2020. Why Are We Like This?: The AI Architecture of a Co-Creative Storytelling Game. In *International Conference on the Foundations of Digital Games* (Bugibba, Malta) *(FDG '20)*. Association for Computing Machinery, New York, NY, USA, Article 13, 4 pages. https://doi.org/10.1145/3402942.3402953

[26] Max Kreminski and Noah Wardrip-Fruin. 2018. Sketching a Map of the Storylets Design Space. In *Interactive Storytelling*, Rebecca Rouse, Hartmut Koenitz, and Mads Haahr (Eds.). Springer International Publishing, Cham, 160–164.

[27] Quinn Kybartas and Rafael Bidarra. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games* 9, 3 (2016), 239–253.

[28] R. Lavender. 2016. The Zelda Dungeon Generator: Adopting Generative Grammars to Create Levels for Action-Adventure Games.

[29] Antonios Liapis. 2020. 10 Years of the PCG workshop: Past and Future Trends. In *Proceedings of the 11th Workshop on Procedural Content Generation, FDG.*

[30] Antonios Liapis, Gillian Smith, and Noor Shaker. 2016. Mixed-initiative content creation. In *Procedural content generation in games*. Springer, 195–214.

[31] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. 2013. 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. AI Access Foundation, 11–16.

[32] Antonios Liapis, Georgios N Yannakakis, Constantine Alexopoulos, and Phil Lopes. 2016. Can Computers Foster Human Users' Creativity? Theory and Praxis of Mixed-Initiative Co-Creativity. *Digital Culture & Education* 8, 2 (2016), 136–153.

[33] A. Liapis, G. N. Yannakakis, M. J. Nelson, M. Preuss, and R. Bidarra. 2019. Orchestrating Game Generation. *IEEE Transactions on Games* 11, 1 (2019), 48–68.

[34] Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. 2013. Sentient sketchbook: Computer-aided game level authoring. In *In Proceedings of ACM Conference on Foundations of Digital Games, 2013. In Print.*

[35] Tiago Machado, Dan Gopstein, Andy Nealen, and Julian Togelius. 2019. Pitako-recommending game design elements in cicero. In *2019 IEEE Conference on Games (CoG)*. IEEE.

[36] Stacey Mason, Ceri Stagg, and Noah Wardrip-Fruin. 2019. Lume: A System for Procedural Story Generation. In *Proceedings of the 14th International Conference on the Foundations of Digital Games.* Association for Computing Machinery, New York, NY, USA, Article 15, 9 pages. https://doi.org/10.1145/3337722.3337759

[37] Juan Oates. 2013. Researching Information Systems and Computing. In *SAGE.*

[38] C. Pedersen, J. Togelius, and G. N. Yannakakis. 2010. Modeling Player Experience for Content Creation. *IEEE Transactions on Computational Intelligence and AI in Games* 2, 1 (2010), 54–67.

[39] Julie Porteous. 2017. *Planning Technologies for Interactive Storytelling.* Springer Singapore, Singapore, 393–413. https://doi.org/10.1007/978-981-4560-50-4_71

[40] Przemyslaw Prusinkiewicz and Aristid Lindenmayer. 1996. *The Algorithmic Beauty of Plants.* Springer-Verlag, Berlin, Heidelberg.

[41] Mark O. Riedl and R. Michael Young. 2005. An Objective Character Believability Evaluation Procedure for Multi-agent Story Generation Systems. In *Intelligent Virtual Agents*, Themis Panayiotopoulos, Jonathan Gratch, Ruth Aylett, Daniel Ballin, Patrick Olivier, and Thomas Rist (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 278–291.

[42] Mark O. Riedl and R. Michael Young. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing* 24, 3 (01 Sep 2006), 303–323. https://doi.org/10.1007/BF03037337

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

[44] Gillian Smith. 2014. Understanding Procedural Content Generation: A Design-Centric Analysis of the Role of PCG in Games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 917–926. https://doi.org/10.1145/2556288.2557341

[45] Gillian Smith, Ryan Anderson, Brian Kopleck, Zach Lindblad, Lauren Scott, Adam Wardell, Jim Whitehead, and Michael Mateas. 2011. Situating Quests: Design Patterns for Quest and Level Design in Role-Playing Games. In *Interactive Storytelling*, Mei Si, David Thue, Elisabeth André, James C. Lester, Joshua Tanenbaum, and Veronica Zammitto (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 326–329.

[46] Gillian Smith and Jim Whitehead. 2010. Analyzing the Expressive Range of a Level Generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games* (Monterey, California). ACM, New York, NY, USA, Article 4, 7 pages. https://doi.org/10.1145/1814256.1814260

[47] G. Smith, J. Whitehead, and M. Mateas. 2011. Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design. *IEEE Transactions on Computational Intelligence and AI in Games* 3, 3 (Sept. 2011).

[48] Anne Sullivan, Michael Mateas, and Noah Wardrip-Fruin. 2009. QuestBrowser: Making Quests Playable with Computer-Assisted Design. In *Proceedings of Digital Arts and Culture* (Irvine, California) *(DAC '09).*

[49] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. 2018. Procedural Content Generation via Machine Learning (PCGML). *IEEE Transactions on Games* 10, 3 (2018), 257–270.

[50] Marcus Trenton, Duane Szafron, Josh Friesen, and Curtis Onuczko. 2010. Quest Patterns for Story-Based Computer Games. In *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Stanford, California, USA) *(AIIDE'10)*. AAAI Press, 204–209.

[51] Georgios N. Yannakakis, Antonios Liapis, and Constantine Alexopoulos. 2014. Mixed-Initiative Co-Creativity. In *Proceedings of the 9th Conference on the Foundations of Digital Games.*

[52] Georgios N Yannakakis and Julian Togelius. 2018. *Artificial intelligence and games.* Vol. 2. Springer.

[53] R. Michael Young, Stephen G. Ware, Bradly A. Cassell, and Justus Robertson. 2013. Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives. *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative* 37, 1-2 (2013), 41–64.

[54] Kristen Yu, Nathan R. Sturtevant, and Matthew Guzdial. 2020. What is a Quest?. In *Proceedings of the Intelligent Narrative Technolgies Workshop at AIIDE.* http://www.cs.ualberta.ca/~nathanst/papers/yu2020quest.pdf

[55] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood. 2018. Explainable AI for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. 1–8. https://doi.org/10.1109/CIG.2018.8490433