Quest to Dungeon (QtD): Towards a Tool that Supports Collaboration between Narrative and Level Designers

Oscar Boutani[1,†], Sam Shariati[1,†] and Alberto Alvarez[1,2,*]

[1]Game Lab, Malmö University, Sweden

[2]Research Centre for Imagining and Co-Creating Futures, Malmö University, Sweden

Abstract

Quest to Dungeon (QtD) is a tool designed to bridge the gap between narrative design and procedural level generation in games, two processes that are typically developed in isolation from each other. QtD connects narrative design with level design by enabling designers to create quests by combining tasks in a grid interface, where each narrative task automatically generates corresponding dungeon rooms using task-specific algorithms. We evaluated QtD through a user study (N=8) where all participants created first a quest with two endings, and then a quest without constraints. Our results showed that participants found the tool intuitive and effective in visualizing how narrative objectives translate into level design. QtD helped enable faster iteration cycles for the participants, who speculated about its possible benefits for collaboration and communication between narrative and level designers.

Keywords

Procedural Content Generation, Quest Design, Challenges in Game Development, Quest-Driven Generation

1. Introduction

Collaboration and communication challenges permeate game development at different levels and development stages [1, 2, 3]. One reason for this is how heterogeneous teams are, with developers from distinct disciplines working together towards a shared goal. Yet, they might not have a shared understanding, language, and terminology. In this paper we present Quest to Dungeon (QtD), a tool developed for quest-constrained generation focusing on how manually placed quest-task actions can influence level generation and vice-versa in early prototyping. We also focus on a significant real-world problem that persists in game development, PCG systems and narrative (and narrative elements such as quests) design are typically developed in isolation, resulting in disconnected deployment. In QtD, designers use a node grid to make quests that then are automatically turned into levels. Designers can compose a main quest and subquests with alternative routes and optional tasks. These tasks are then used to generate a proposed level layout relevant to the tasks and that interconnects them. The spatial layout of the quests and levels is the same in a 6x6 grid, which forces the level layout to track 1-to-1 the layout of the quests. The level layout, while not modifiable, is created with the possibility to be used as-is, or as a way for designers to communicate their ideas to different teams on what they imagine the level design or narrative to be. QtD's long-term goal is to help the communication and collaboration between narrative and level designers at early production stages.

We focus on quest design and its implication within the overall game design, particularly the impact on level design. This is because narrative and level (or space) facets are linked and complement each other [4, 5]; thus, the design of either and their generation benefits from a more holistic design. Within research there exist many examples of these being interconnected for Procedural Content Generation (PCG) [5, 6, 7]. Similarly, the game industry has worked with similar objectives, mainly trying to either fit generated quests in the gameworld or human-authored quests in generated levels. For instance,

Joint AIIDE Workshop on Experimental Artificial Intelligence in Games and Intelligent Narrative Technologies, November 10-11, 2025, Edmonton, Canada.

*Corresponding author.

†These authors contributed equally.

oscar.boutani@gmail.com (O. Boutani); shariati.sam@gmail.com (S. Shariati); alberto.alvarez@mau.se (A. Alvarez)

© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2025-11-12

CEUR-WS.org/Vol-4090/short6.pdf

The Elder Scrolls V: Skyrim[8] infamously uses radiant quests (procedural objectives) but deploys them in generic, repetitive dungeons, creating a disconnect between narrative goals and level design [9]. This approach leads to inefficiencies: level designers constantly need to align environments with quest requirements, while narrative designers lack tools to ensure their objectives are supported by the generated content. The result is increased iteration time, inconsistent player experiences, and missed opportunities for fun replayability.

We evaluated the effectiveness of QtD and how it could be used through a user study (N=8) where we collected qualitative feedback and tool logs. Our results indicate that QtD was easy-to-use, helped visualise how quests and tasks could be placed in level design with faster iteration cycles. Participants also speculated about how it could be used in practice and for collaboration between departments and designers, emphasizing its role in brainstorming, setting a clearer plan, and communication.

2. Related Work

Narrative is a key facet in games and has a strong relation with the gameworld, levels, and in general spaces where the game will take place [10]. Within the narrative facet and PCG, quests have been the target of a large body of research for generating them such as Questbrowser [11], a quest design tool where users can query the system for ideas and alternatives and the work by Grinblat and Bucklew [12] in "Caves of Qud", dynamically generating quests constrained by the world and authored templates. Combining quests and level generation has also been explored with interesting results. Questgram [13] enable designers to create quests and receive recommendations on what task to add next based on their current quest and the dungeon composed of both human-authored and generated rooms. Taksim [14] focuses on combining generated mission graphs with game spaces using answer-set programming. Dormans and Bakkes [5] generated missions and spaces with grammars have also been used for Unexplored and Unexplored 2. While these systems show the possibilities of connecting the two facets, how content limits and interact with each other is unclear as well as how human-authored content could fit with generated content, which is one of the goals with QtD. Yet, significant challenges remain when trying to connect quests and levels automatically as well as the communication and collaboration challenges that exist when designing these. Mixed-initiative PCG systems could help mitigate some of these challenges by giving designers control leveraging automatic generation.

The interest and challenge of aligning quests and levels, as well as human-authored content (either quest or levels) with automatic generation has also being the object of discussion within the game industry. Levine [15] described the process required to align BioShock's narrative objectives with level design, while Smith and Colantonio [16] discussed Arkane Studios' "narrative ecology" approach that places story elements within environmental design for Dishonored[17]. Games such as Hitman (2016) [18] uses "opportunity" structures that place narrative moments within level design, allowing players to discover and trigger quests through environmental interaction. The Legend of Zelda: Breath of the Wild [19] uses "chemistry engine" principles where quest objectives interact with environmental systems, creating narrative possibilities within designed spaces.

3. Quest to Dungeon Tool

QtD takes a narrative-first approach where quest structures drive level generation, in contrast to most prior systems that combine the two facets, which integrate narrative within the level generation. QtD follows a modular architecture that separates the quest structure definition from dungeon generation, giving users full control over the quest structure, while the AI system procedurally generates the dungeons. QtD was developed in Unity using version 6000.0.36f1.

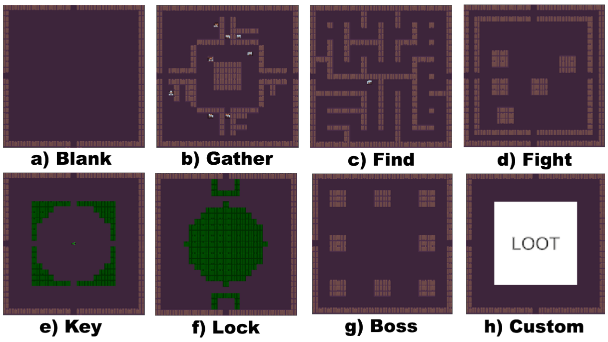

Figure 1: Sample rooms generated using each room generation pattern

3.1. Quest Structure Module

The quest assignment interface uses Unity's UI system to create an interactive 6×6 grid shown in figure 3 top row. Each grid node consists of a drop-down menu containing multiple task types users can choose from. For this iteration, users can choose between: **Blank** - used as placeholder and no task needs to be done, **Gather** - tasks related to gathering resources, **Find** - tasks related to finding resources or quest items, **Fight** - tasks related to engaging in combat, **Key** and **Lock** - enable a key and lock task sequence represented as separated tasks that condition each other, **Boss** - task associated to final encounters and **Custom** - placeholder task for users to add their own quest task idea. Once a task type is selected for a node, a subsequent drop-down appears that allows users to further specialize that task type and that is related to the task itself (e.g., prototypical quest items appear for a **Find** task). After creating a grid node, users can add adjacent grid nodes to the previous one, linking these two nodes together. Through this graphical process, users can shape diverse sequence of tasks with optional tasks, secondary objectives, and multiple paths to complete a quest. When a user finalizes a quest structure by pressing *Generate*, the system then takes all the tasks and their relative position to generate the rooms and interconnect them.

3.2. Dungeon Generation Module

The level layout is the same as the task graph as shown in figure 3 bottom row. The dungeon generation utilizes Unity's tile-map system to transform all stored data into 2D tile-based rooms, using the same dungeon layout as the quest interface. Each task has an analogous level design algorithm for the creation of the room. The room generation process follows a template-based approach with controlled randomization to create visual variety while preserving functional requirements. This approach ensures that, while rooms of the same task type share common characteristics, they maintain visual distinctiveness. Each room is 25x25 tiles consisting of wall, floor, and treasure tiles as well as special key tiles for key-lock mechanisms. Patterns with these tiles such as "corridors", "cages" and "pillars" are also generated depending on which room to add visual representation to the dungeon layout. The level cannot be edited as it is meant as the automatic projection of the set of tasks designed by the user, but can be fully regenerated at any time.

For each task type, we define a set of room templates that satisfy the fundamental requirements of that task, **Core Structural Elements** such as walls and pathways between connected rooms, and **Task-specific Features** such as number of enemies and item spawn locations.

3.3. Room Generation

Rooms are generated based on the tasks that users manually add. Each room pattern includes specific constraints to ensure that every generated room is feasible. For rooms that include corridors, these are always fixed in size for the algorithm to guarantee navigability. For rooms that can generate pillars, additional constraints are applied to ensure that pillars are never placed adjacent to each other or to walls. The QtD tool contains eight different room types and they are exemplified in figure 1.

Blank Pattern Room (fig. 1.a): The Blank pattern is the default task for every room. It consists only of an empty layout surrounded by walls and acts more as a room without a task.

Gather Pattern Room (fig. 1.b): The Gather Pattern Room creates centralized room layouts centered around a core area with radiating branches. The design features a central hub with a wall structure in its middle, connected to 3-4 main corridors that extend outward. These main corridors may branch into smaller side passages, creating exploration opportunities. Items are then placed throughout the room.

Find Pattern Room (fig. 1.c): The Find Pattern Room generates a maze based on a modified recursive backtracking approach. The algorithm carves paths through an initially solid block, ensuring all areas remain connected while maintaining sufficient wall thickness between passages.

Fight Pattern Room (fig. 1.d): The Fight Pattern Room generates layouts with an outer square boundary and an inner "cage" structure. This creates a room-within-a-room design. The inner area contains dynamically placed pillars whose number and size adapt based on the number of enemies the user chose to add to the task.

Key Pattern Room (fig. 1.e): The Key pattern Room creates a layout with color-coded shapes in the center of the room. The room features a colored square in the center that is hollowed out by a specific shape chosen by the user (triangle (red), square (green), circle (blue), and hexagon (yellow)).

Lock Pattern Room (fig. 1.f): The Lock Pattern Room generates rooms that restrict access to other areas. The layout features a specific color-coded shape in the center of the room chosen by the user. Shapes and colors are the same as for the key pattern rooms.

Boss Pattern Room (fig. 1.g): The Boss Pattern Room creates a spacious, symmetrical layout designed for boss encounters. The algorithm produces a square room with an outer perimeter of walls and an open central area. Within this space, eight pillars are positioned in a symmetrical pattern.

Custom Pattern Room (fig. 1.h): The Custom Pattern Room consists of a blank room with a white square in the center of the room, where users can write whatever they want inside this white square. It is primarily used as placeholder of not implemented tasks.

4. Experiment

We conducted a user evaluation and collected both data from surveys and data logs from the tool as participants used the tool. Quantitative metrics (total nodes, branch depth, grid usage percentage, regenerate events) were logged. Qualitative survey responses were manually analyzed. Quotes were collected from qualitative questions during the survey. Our study setup consist of first, an online pilot study with two participants following the same procedure as in-person participants to understand if the tool required any revision or something in the survey was unclear, and an in-person user study with eight participants that used the tool for up to 13 minutes. Our goal was to assess the usability of QtD, its effectiveness in bridging the gap between quest-based narrative approaches and level design, and understand how it could be used as a communication device by using its automatic level generation.

Participants were given an explanation of the project and informed consent was collected. They then viewed an on-screen tutorial pop-up (up to 5 minutes), explaining tool features and task types. During the evaluation they had first 3 minutes to create a quest with two possible endings to get used to the tool followed by a 5 minutes free-exploration phase to create a quest without constraints.

5. Result

All participants studied or were students of a game development program, had between 2 and 3 years of game development experience and 6 out of 8 self assessed as programmers while the other 2 as game designers. 6 out of 8 had worked within narrative design or level design to some extent, while two others have not worked on that area. Finally, 5 out of 8 (all self-assessed as programmers and with experience in narrative or level design) have previously worked with PCG tools and algorithms.

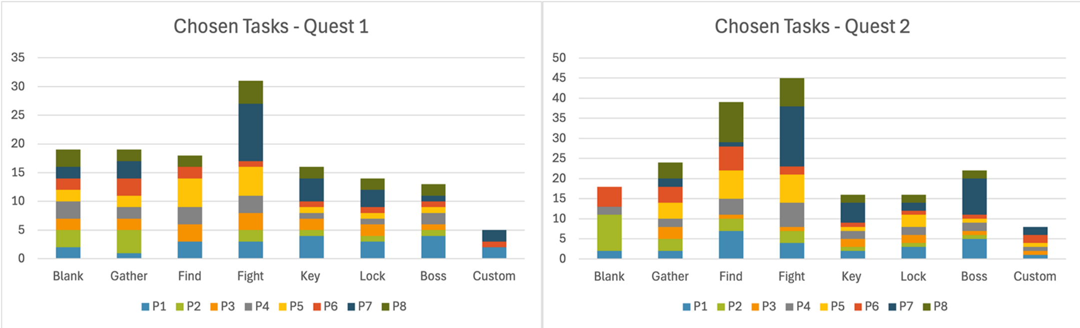

Figure 2: Total amount of utilized task-type per quest (experiment tasks). Color indicates different participants and the amount of utilized task-types they used.

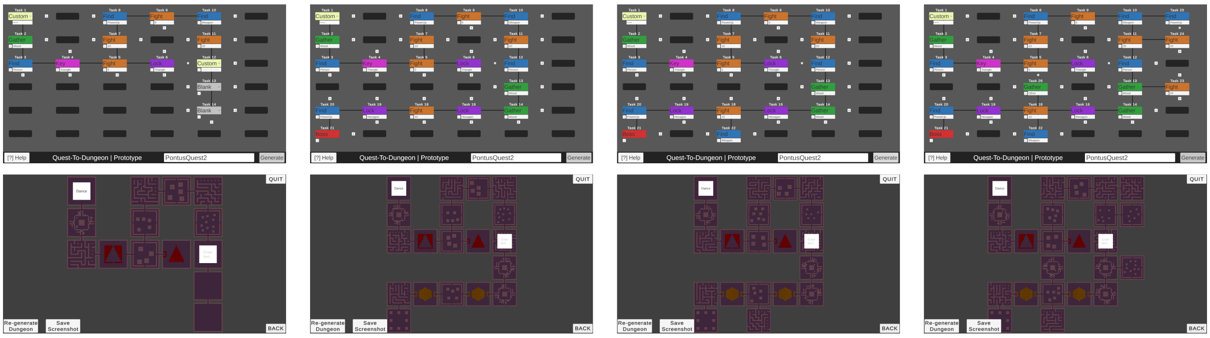

Figure 3: Development process of quest 2 for participant 5. On the top row, it can be seen the quest and tasks system and the corresponding dungeon at the bottom row.

5.1. Data from the QtD tool

Data collected from the tool logs shows the total distribution of task types across the two quest creation tasks (Quest 1 - 3 minutes, and Quest 2 - 5 minutes). For Quest 1, which consisted of users designing a dungeon with two separate endings, the **Fight task** was the most frequently utilized (31 instances), followed by nearly equal usage of Blank (19), Gather (19), and Find (18) tasks. Key (16), Lock (14), and Boss (13) tasks were moderately used, while the Custom task was used sparingly (5 instances). This distribution suggests that combat-focused narrative elements were prioritized in the first quest design, which is expected as the tasks are biased towards this type of quests. Quest 2, which consisted of users being able to freely explore and design a dungeon, showed a slight difference in design approach. **Fight** tasks still dominated but with less frequency (45 instances), a significant rise in Find tasks however occurred (39 instances compared to 18 in Quest 1). Gather tasks also increased from 19 to 26 instances. This shift could indicate that as designers gained familiarity with the tool, they incorporated more exploration and collection-based narrative elements, creating more complex quest structures.

The average dungeon size also increased from 17.2 rooms in Quest 1 to 23.5 rooms in Quest 2, resulting in more elaborated quest structures. The dimensional consistency across sessions, despite the increased room count, indicates that designers focused on creating denser and more interconnected quest structures. This suggests that creating more dense and eventful quests are more prioritized than creating linear quest lines. Lastly, since the tool allows users to switch back and forth between the quest and dungeon modules, multiple iterations of the same quests were saved each time users moved back and forth to analyse the different ways of using QtD. Figure 3 shows an example of a participant development process for Quest 2, consisting of four iterations.

5.2. Survey Results

Tool usability, effectiveness, and interface: The task assignment interface was rated highly intuitive by participants, with the average score being 7.88/10 (SD=1.25). Most participants highlighted its ease to use and beneficial visualization, emphasizing that QtD was "easy to understand and [had a] very good visualization"(p1) and "easy to use, I would probably use it in an early stage of my game"(p2). This strong usability score indicates that the 6x6 grid system and the node-based design provided an accessible framework for quest design without requiring extensive technical knowledge. Tasks included variables that affected generated rooms and gave more control to users. 7 out of 8 participants found them to have a significant impact on their design process, with the average score being 8.13/10 (SD=1.73). This suggests that the additional task-specific parameters were successful at enhancing the tool's flexibility while maintaining its accessibility.

Participants felt that generated rooms were representative of the specific task (avg. score of 7.63/10 (SD=1.6)) and most of the participants felt that the tool helped them visualize how narrative objectives translate to physical spaces (avg. score of 8/10 (SD=2.27)). This suggest that the translation from narrative task to spatial design was clear and coherent with participants highlighting "the visualization gave the feeling of how the game could look like/time it takes to play,"(p1) and "I was thinking about patterns, what room should come after another, trying not to have repetivness."(p2), which show how their process was influenced by the QtD workflow and that the system successfully conveyed gameplay pacing and narrative progression.

Participants also highlighted how tools like QtD "... can make it faster to make levels"(p8) and "would give a clear plan of what to do and how to do it. leaving no holes in the proccess"(p1) as well as the tool "give ideas of a level where you can have smaller quest together with a broader quest"(p4) and how "it could be effective as a "brain storming" tool to help narrative designers easily explain their vision and ideas to the development team and improving the communication between the two.". This shows that having a tool that allows for quick iterative quest development and how that could be seen within level design is appreciate and could be an important tool to understand design process and collaboration. However, participants felt mixed about the control over the generation from task to room (avg. score of 5.63/10 (SD=1.51)). This was expected as the room generation takes follow a specific pattern that is not adapted or controllable by designers, which will be explored in future work.

Participants highly rated the tool's potential to reduce dungeon creation iteration time, giving it an average score of 7.63/10 (SD=1.69). This suggests the tool could effectively reduce iteration time between quest narration and map design during planning phases.

Design Process and Workflow Patterns: The qualitative responses revealed consistent patterns in how participants approached quest design using the QtD tool. In the pre-defined task (Quest 1), where designers were asked to create a quest with two different endings, most participants adopted a branching structure approach. One participant described their process as creating "a level layout that had two different types of ending, one being focused on fighting, and one being focused on finding the key and the exit."(p6) Another mentioned, "Two separate paths, each one with different approaches or types of rooms before reaching the last rooms." (p5).

For the free-roam task (Quest 2), participants demonstrated more diverse and complex design approaches. Many described creating more interconnected pathways and utilizing a broader range of task types. One participant explained, "I wanted to create a 'World of Warcraft' type quest, that followed a linear path, and taking the player on a journey through the different tasks that were available to me."(p6) Another mentioned focusing on "a puzzle focused dungeon with a lot of locked doors and keys."(p3) This evolution in design complexity from Quest 1 to Quest 2 aligns with the quantitative data showing larger, more diverse dungeons in the second session.

Several participants described an iterative workflow where they would first establish a layout using basic tasks and then refine it. One participant explained: "Create a bunch of empty rooms, find a structure that I liked and then swapped those empty rooms into either fights, gathers, or finds, wherever I felt like it would fit."(p2). This suggests that QtD supported a layered design process from a more draft version towards something participants envisioned in a quick iterative manner.

6. Discussion

6.1. Integration of Narrative and Level Design

Our results show the value of coupling game facets such as narrative and level, and the potential of these to mediate design and collaborative decision-making. Users could quickly see how their quest choices affected the physical layout, allowing for immediate adjustments. The high rating for effectiveness of generated rooms that represents the tasks indicates that the translation from narrative design to level design was largely successful. QtD was able to show how participants process was influenced by the QtD workflow and that the system successfully conveyed gameplay pacing and narrative progression. QtD and similar tools could help address challenges with isolated facet development, particularly when affected by procedural systems such as side quests in *Skyrim* to create better gameplay.

The varying effectiveness of different task pattern rooms provided insight into what designers value in procedurally generated content. The Find and Gather algorithms for example rated significantly higher than the Fight algorithm in terms of how interesting the room layout were, suggesting that designers prefer room layouts with more complex structural elements. The tool could therefore help designers to create more complex patterns automatically as a starting point that would be time-consuming to design manually, rather than simpler and more open layouts. The effectiveness of the specific room generation algorithms could also suggest how procedural techniques can enhance gameplay in the context of narrative generated rooms. For example, the Gather pattern algorithm with strategically placed collection points effectively communicates a scavenger hunt experience, while the Find pattern's maze communicates exploration and discovery.

6.2. Collaborative Design Workflows

Collaboration and communication challenges permeate game development [1] and tools like QtD could enable a better workflow. While QtD was evaluated with a single designer to create quests and see how generated rooms affected their design and expressed their ideas, our final goal is to explore how QtD and similar tools might transform collaborative workflows between different design specialties. First steps were taken in that direction as participants expressed that the tool could be used effectively as a communication medium, for early development phases, and planning work in teams.

Further, our usability results also suggests that QtD simplifies the design process, lowering the technical barrier to expressing design intent, potentially enabling narrative designers that could have limited technical skills to communicate layout requirements effectively. This direct translation of task-types to room structures could also create a vocabulary that can be shared between narrative and level designers, potentially reducing miscommunication and iteration cycles typically required to align these two aspects of game design.

6.3. User Development Process

The data logs showed that most users had an iterative approach to quest development. Users frequently moved back and forth between the quest and dungeon modules, making adjustments to their quest structures after seeing the generated layouts (seen in figure 3). This iterative process suggests that the visual feedback from the dungeon generation could have influenced quest design decisions. Participants approached the quest development using both a layout-first and narrative-first approach. Some participants began by establishing a physical dungeon layout of empty rooms that was later populated with appropriate tasks. One participant described it as: "I just clicked out rooms in a random pattern then I connected the different branches into a flow that would seem nice. After that I began choosing different rooms that would fit together to create a nice game experience." (p8). However, some users took a narrative-first approach instead, conceptualizing the quest story before considering dungeon layout. A participant described: "My idea for the quest was already in my mind when I started to build the level, so the quest did influence the level but not as much the other way around." (p4). This shows that the QtD tool can be flexible enough to be utilized in different design approaches.

6.4. Limitations and Future Work

Our evaluation showed the possibilities of using QtD to bridge quest design and level design. Participants emphasized its possible use to help communication and collaboration between designers as they speculated about it, and gave us a good insight into how the tool is used, its effectiveness and the design process. Yet since we only tested with single users and not with a team, we cannot conclude its usability as a communication and collaboration tool, which would be the next step for this project.

The implementation has several structural and expressive limitations. The fixed 6×6 grid, while being easy to understand, could constraint the scope and complexity of dungeons that designers can create. Participants created increasingly dense quest structures between quest 1 and 2, suggesting that the grid size may become a bottleneck for more complex narrative designs. Further, users can only select eight task types and the room generation follows similar templates for these tasks which could hinder designers' expressivity. While participants responded positively to the Find and Gather tasks/rooms, the scope of narrative expression is still constrained compared to the full spectrum of quest possibilities in modern game design. The rigid, tile-based room generation also produces geometric layouts that, while functional, lack the organic feel that many games have.

Therefore, in future iterations of QtD we aim at increasing task-types to include more nuanced narrative elements such as npc rooms, dialogue, important events etc. as well as common quest task types that could be extracted from previous research [20, 11] as well as improving the interface for easier development such as being able to delete entire tree of task nodes or quickly show key-lock combinations. We also aim at giving more control to designers with the generated rooms such as providing more granular control over room generation parameters while maintaining the tool's accessibility. We aim at using color-coding and quest structure terms such as *satellites* and *kernels* [4] to help structure linear quests and alternative paths.

7. Conclusions

This paper presented Quest to Dungeon, a tool that connects human-authored quest design with procedural generation of levels constrained by the designed quests. Our overarching goal is to address collaboration challenges at design-phase (e.g., communication between departments and design specialties, brainstorming and pre-production development) and deployment-phase (e.g., misaligned content, combination of human-authored content with PCG tools). Our tool enables designers to define quest structures through a node grid that directly impacts the procedural generation of a dungeon layout and through this, we have demonstrated how narrative-driven level design can be achieved by translating quest tasks into interconnected rooms with task-specific properties. Our evaluation emphasized the easy-to-use and effectiveness of QtD to represent quests and tasks, as well as their level and room representation. By coupling quest logic with procedural generation, the tool demonstrates how quest design can guide PCG to create dynamic, task-aligned dungeons while also making it easier for developers to iterate rapidly. While limitations exist in the current implementation, the positive reception from participants suggests that tools such as QtD approach could meaningfully contribute to more efficient and cohesive game development workflows.

Acknowledgments

The research and project was conducted and supported with funding from FORTE, grant Reg. No. 2023-01394, "I Get by with a Little Help from AI: Artificial Intelligence Supported Human Collaboration".

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 and grammarly in order to: Grammar and spelling check, converting plain text into LaTeXformat, suggestions on better structure of the sections, and to identify inconsistencies in terminology and notation. All AI-generated output was reviewed and edited by the authors as needed and take full responsibility for the publication's content.

References

[1] F. Blomberg, L. Lind, LET'S LEVEL UP OUR COMMUNICATION A Study of Communication Challenges in Game Development Teams, B.S. Thesis, Göteborgs Universitet, 2022.

[2] C. M. Kanode, H. M. Haddad, Software Engineering Challenges in Game Development, in: 2009 Sixth International Conference on Information Technology: New Generations, 2009, pp. 260–265. doi:10.1109/ITNG.2009.74, journal Abbreviation: 2009 Sixth International Conference on Information Technology: New Generations.

[3] C. Politowski, F. Petrillo, G. C. Ullmann, Y.-G. Guéhéneuc, Game industry problems: An extensive analysis of the gray literature, Information and Software Technology 134 (2021) 106538.

[4] E. Aarseth, A Narrative Theory of Games, in: Proceedings of the International Conference on the Foundations of Digital Games, FDG '12, ACM, 2012, pp. 129–133. URL: https://doi.org/10.1145/2282338.2282365. doi:10.1145/2282338.2282365.

[5] J. Dormans, S. Bakkes, Generating Missions and Spaces for Adaptable Play Experiences, IEEE Transactions on Computational Intelligence and AI in Games 3 (2011) 216–228. URL: http://ieeexplore.ieee.org/document/5763766/. doi:10.1109/TCIAIG.2011.2149523.

[6] C. Ashmore, M. Nitsche, The Quest in a Generated World, in: Proceedings of the 2007 DiGRA International Conference: Situated Play, The University of Tokyo, 2007. URL: http://www.digra.org/wp-content/uploads/digital-library/07311.20228.pdf.

[7] A. Alvarez, J. Font, J. Togelius, Story Designer: Towards a Mixed-Initiative Tool to Create Narrative Structures, in: Proceedings of the 17th International Conference on the Foundations of Digital Games, 2022.

[8] Bethesda Game Studios, The Elder Scrolls V: Skyrim, 2011. Published: Multiple platforms.

[9] V. Breault, S. Ouellet, J. Davies, Let CONAN tell you a story: Procedural quest generation, Entertainment Computing 38 (2021) 100422. doi:10.1016/j.entcom.2021.100422.

[10] Q. Kybartas, R. Bidarra, A Survey on Story Generation Techniques for Authoring Computational Narratives, IEEE Transactions on Computational Intelligence and AI in Games 9 (2017) 239–253.

[11] A. Sullivan, M. Mateas, N. Wardrip-Fruin, QuestBrowser: Making Quests Playable with Computer–Assisted Design, in: Proceedings of Digital Arts and Culture, 2009, p. 9.

[12] J. Grinblat, C. B. Bucklew, Warm rocks for cold lizards: Generating meaningful quests in caves of qud, in: Proceedings of the AIIDE Workshop on Experimental AI in Games, 2020.

[13] A. Alvarez, E. Grevillius, E. Olsson, J. Font, Questgram [Qg]: Toward a Mixed-Initiative Quest Generation Tool, in: Proceedings of the 16th International Conference on the Foundations of Digital Games, FDG'21, Association for Computing Machinery, New York, NY, USA, 2021.

[14] A. M. Abuzuraiq, A. Ferguson, P. Pasquier, Taksim: A Constrained Graph Partitioning Framework for Procedural Content Generation, in: 2019 IEEE Conference on Games (CoG), IEEE, London, United Kingdom, 2019, pp. 1–8. doi:10.1109/CIG.2019.8848065.

[15] K. Levine, Storytelling in BioShock: Empowering Players to Care About Your Stupid Story, 2008. Place: San Francisco, CA Published: Presentation at the Game Developers Conference.

[16] H. Smith, R. Colantonio, Dishonored: Designing a Game for Multiple Play Styles, 2012. Place: San Francisco, CA Published: Presentation at the Game Developers Conference.

[17] Arkane Studios, Dishonored, 2012. Published: Multiple platforms.

[18] IO Interactive, Hitman, 2016. Published: Multiple platforms.

[19] Nintendo, The Legend of Zelda Breath of the Wild, 2017. Published: Nintendo Switch.

[20] J. Doran, 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, ACM, New York, NY, USA, 2011. doi:10.1145/2000919.2000920.