To Make Sense of Procedurally Generated Dungeons

Simon Tolinsson
Alexander Flodhag
simontolinsson@gmail.com
alexander_flodhag@hotmail.com
Malmö University

Alberto Alvarez
Jose Font
alberto.alvarez@mau.se
jose.font@mau.se
Department of Computer Science and Media Technology,
Malmö University

ABSTRACT

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

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**; • **Applied computing → Computer games**; • **Software and its engineering → Interactive games**.

KEYWORDS

Procedural Content Generation, Mixed-Initiative Co-Creativity

ACM Reference Format:
Simon Tolinsson, Alexander Flodhag, Alberto Alvarez, and Jose Font. 2020. To Make Sense of Procedurally Generated Dungeons. In *Extended Abstracts of the 2020 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '20 EA), November 2–4, 2020, Virtual Event, Canada.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3383668.3419890

1 INTRODUCTION

Procedural content generation (PCG) has found itself in the spotlight within game development with games such as Minecraft [17], No Man's Sky [13], and Spelunky [9], improving replayability, reducing the developers' workload, and fostering the designers' creativity [4, 12, 18]. However, some type of narrative or context is required to make sense of the PCG content when implemented into the game space [8]. An example of narrative is objectives. If there is an objective within the content, for example, finding a sacred gem in a dangerous dungeon, then that creates interaction between the user and the content, which creates the needed context.

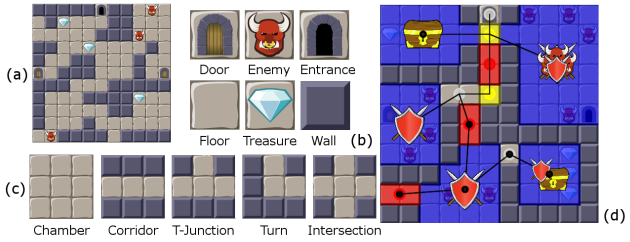

Figure 1: The main components in EDD. (a) A basic room, (b) different placeable tiles, (c) micro patterns and (d) meso patterns [3].

The Evolutionary Dungeon Designer (EDD) is a mixed-initiative design tool used for creating and generating dungeons [5]. This paper gathers the first step towards implementing a sub-system for EDD, which procedurally generates objectives for dungeons. The current sub-system gathers and continuously adapts to the designer's dungeon to place different objectives based on the rooms' content. Through this, the designer focuses on creating the dungeon while seamlessly, they are provided with the different generated objectives. We evaluate the artifact's utility, quality, and efficacy based on how well the objectives represent the layout of the dungeon with experimental scenarios.

2 RELATED WORK

2.1 The Evolutionary Dungeon Designer

EDD is a mixed-initiative tool for designers to create dungeons as a set of interconnected tile-based rooms [5]. Each tile in a room can be modified to represent different types of paths, obstacles or rewards, and are used to form inventorial (Fig. 1.b) or spatial (Fig. 1.c) micro patterns. These micro patterns can be further combined to form meso patterns (Fig. 1.d) such as treasure or guarded rooms. Furthermore, as the designer creates rooms, EDD dynamically offers procedurally generated room suggestions through the Interactive Constrained MAP-Elites, using such patterns as evaluation and continuously adapting to the designer's design [6].

2.2 Procedural Generation of Game Narrative

Interactivity and narrative have conflicting demands [14]. With narrative, the author decides the direction of the flow, while interactivity turns to the player for motive power. Straying from the

CHI PLAY '20 EA, November 2–4, 2020, Virtual Event, Canada

ACM ISBN 978-1-4503-7587-0/20/11…$15.00
https://doi.org/10.1145/3383668.3419890

author's path may make for a less satisfying story, but restricting the player's freedom of actions will have the same effect on the game. But game designers are not only storytellers, but they also sculpt and design game worlds and spaces.

Generated content needs context in the game space. A lack of context may negatively affect user experience, with the content being perceived as empty or meaningless [8]. The limitations of a story are related to the quest combinations available. By understanding the structure of quests, we can also understand the limits and potential of these kinds of games and how to create rich, open game worlds and tell interesting stories within them [1].

The common factor with objectives in games is to provide the player a reason to further progress through the game [8]. When generating content for game space, narrative, or context, needs to be generated as well. In action-adventure games, the level design is essential, and when procedurally generating levels for these games, it is best to break down the generation process in two steps, one for generating game space and one for generating missions [10].

Charbitat bases narrative generation on sets of tiles, which partitions the game space and creates a graph that keeps track of the player's position. Through this, the system evaluates new possible objectives to generate that would suit better. This evaluation takes in mind previous objectives and actions done by the player, resulting in a more adaptive experience while also increasing the replayability [8].

Procedural narrative generation is often approached split into two tasks, plot and space, either automatically or manually generated [2, 10, 11, 15]. The plot is defined as a set of events with an overall structure that represents both the temporal ordering and the causal relations between the events. Space includes the characters, settings, props, and anything which is present either physically or abstractly in the space of the narrative. By generating space, they also generate context for it, thus creating a unique narrative for each possible outcome of the generative process.

3 GENERATING OBJECTIVES FOR DUNGEONS

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

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

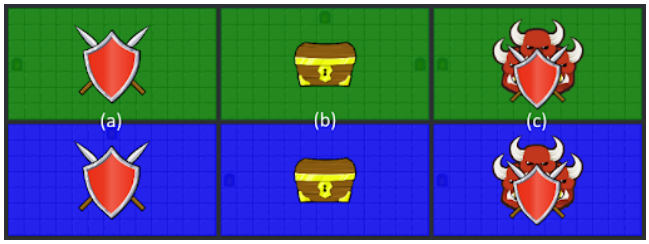

Figure 2: Every type of dungeon objective as both main objective (green) and side objective (blue). The icons represent the different types of objectives (a) "Defeat the enemies", (b) "Find the treasure" and (c) "Defeat the boss".

Dead end. Dead ends have a higher chance of hosting interesting content for the player [8]. Therefore, to not make the outskirts of the dungeon feel meaningless, dead ends are prioritized locations for objective placement.

Distance to the player. Placing all objectives close to the start position, preventing full space exploration, would break the players' immersion [14]. Therefore, large distances (measured in rooms) from the player are encouraged when placing objectives.

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

Quality. The objectives are based on the existing meso patterns in the room. Each meso pattern receives a quality score [7], that when combined, create the quality of the room.

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

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

4 RESULTS AND DISCUSSION

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

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

Figure 4 shows two different small sequential scenarios. In simulation (a), the initial room is the leftmost one, and the two rightmost rooms become the side objective and the main objective. This specific order makes use of all the available game space. However, in

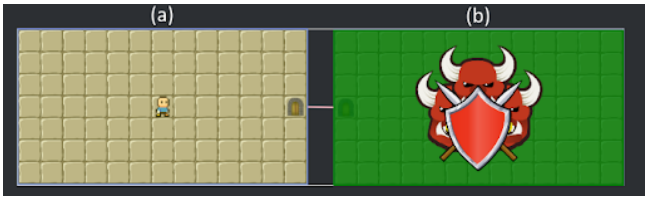

Figure 3: The most simplistic layout of a dungeon with a main objective (green).

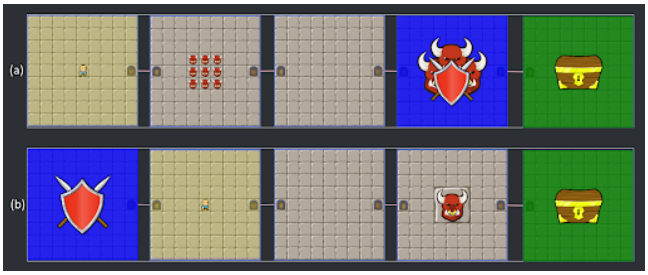

Figure 4: Small single-path dungeon layouts.

simulation (b), the initial room is the second to the left. The system adapts to this change by placing the side objective to the left of the start position to use both dead ends and maximizing the game space usage.

Figure 5: Small circular dungeons.

Figure 5 shows two small scenarios without any dead ends. Only one objective (the main one) is placed under these configurations. In both cases, to utilize most of the dungeon layout, the room on the opposing side of the initial room gets assigned with the main objective.

Figure 6: Large dungeon layouts with several dead ends.

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

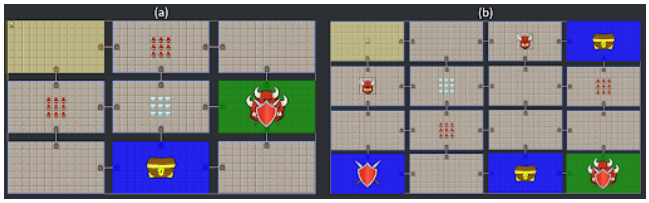

Figure 7: Two large circular dungeon layouts without dead ends.

Figure 7 generates objectives for larger circular dungeons with no dead ends and (a) no content in any of the corners, and (b) corner rooms with meso patterns. In (a), the lack of content in the corner rooms makes the system choose objectives in the neighboring rooms to the furthest corner. Being both equally distant from the initial room, the "Defeat the boss" has a higher quality and is then marked as the main goal. In (b), the corners of the dungeon layout contain content, and the system makes use of this to generate objectives in every corner of the dungeon to maximize the usage of game space. The remaining non-corner "Find the treasure" side objective is prioritized over the other rooms without an objective based on its distance from the initial room.

Figure 8: Large circular dungeons layout with dead ends.

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

In (b), the system adapts to the placement of the initial room in the dead end that hosted the main objective in (a). The main objective is relocated to another dead end, being one side objective now placed inside the inner circular structure of the layout. "Defeat the boss" is still a side objective since it is connected to fewer rooms than the remaining non-objective rooms, exhibiting the relevance order introduced in section 3.

The dungeon layout in Figure 9 is a variation of the layout in Figure 8, now showing how the system reacts to different types and sizes of dungeon layouts with minor changes to the dungeon content. In simulation (a), we changed the different rooms' sizes to show that the system does not consider the size as a factor when

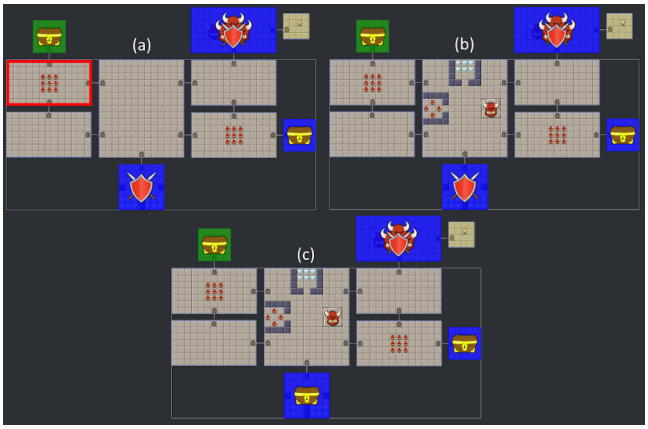

Figure 9: A large circular dungeon layout with dead ends, rooms of different sizes and varying content.

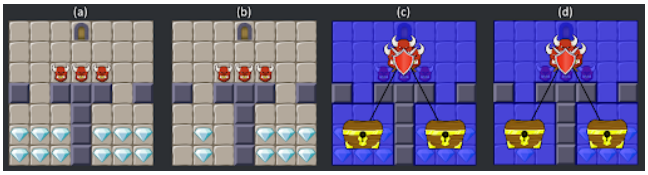

Figure 10: Two different room designs shown with and without meso patterns toggled. (a) and (c) as well as (b) and (d) are the same design. The two room designs represent the bottom room from Figure 9.b) and c), respectively.

evaluating and assigning objectives. Therefore, the objectives in 9.a) are nearly identical to 8.b). The only difference is that the "Defeat the enemies" objective in the circular structure is removed (see the red-bordered rooms in 8.b) and 9.a)). The reason is that the two middle rooms in the circular structure are now combined into one big room, thus reducing N_{obj} from 5 to 4. In (b), the big room in the middle is now filled with several meso patterns to show that the system does not prioritize the amount of content in a room when evaluating the dungeon objectives.

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

5 CONCLUSIONS AND FUTURE WORK

We have developed a sub-system integrated into EDD that generates suitable objectives based on dungeon layouts created in a mixed-initiative environment. This integration has been carried out in a harmonic way [16] with EDD's already existing functionalities. The developed artifact also enhances the mixed-initiative creative loop in EDD, and helps the designer to visually validate their creation in terms of narrative.

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

REFERENCES

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

[2] Ahmed M. Abuzuraiq, Arron Ferguson, and Philippe Pasquier. 2019. Taksim: A Constrained Graph Partitioning Framework for Procedural Content Generation. In *2019 IEEE Conference on Games (CoG)*. 1–8.

[3] 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* (Malmö, Sweden) *(FDG '18)*. ACM, New York, NY, USA, Article 44, 4 pages. https://doi.org/10.1145/3235765.3235810

[4] 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* (Malmö, Sweden) *(FDG '18)*. ACM, New York, NY, USA, Article 50, 8 pages. https://doi.org/10.1145/3235765.3235815

[5] 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)*. 1–8.

[6] Alberto Alvarez, Steve Dahlskog, Jose Font, and Julian Togelius. 2020. Interactive Constrained MAP-Elites: Analysis and Evaluation of the Expressiveness of the Feature Dimensions. *arXiv: 2003.03377* (2020).

[7] 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* (Hyannis, Massachusetts) *(FDG '17)*. ACM, New York, NY, USA, Article 74, 10 pages. https://doi.org/10.1145/3102071.3110572

[8] 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

[9] Derek Yu and Mossmouth, LLC. 2008. *Spelunky*. Game [PC]. Mossmouth, California, US. Last played December 2019.

[10] 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.

[11] Ken Hartsook, Alexander Zook, Sauvik Das, and Mark O. Riedl. 2011. Toward supporting stories with procedurally generated game worlds. In *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2011*. 297–304. https://doi.org/10.1109/CIG.2011.6032020

[12] Erin J. Hastings, Ratan K. Guha, and Kenneth 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

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

[14] Henry Jenkins. 2004. Game design as narrative. *Computer* 44, 53 (2004), 118–130.

[15] Ben 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.

[16] Antonios Liapis, Georgios N Yannakakis, Mark J Nelson, Mike Preuss, and Rafael Bidarra. 2018. Orchestrating game generation. *IEEE Transactions on Games* 11, 1 (2018), 48–68.

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

[18] Noor Shaker, Julian Togelius, and Mark J Nelson. 2016. *Procedural content generation in games*. Springer.