Assessing Aesthetic Criteria in the Evolutionary Dungeon Designer

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

Steve Dahlskog
Malmö University
Malmö, Sweden
steve.dahlskog@mau.se

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

Johan Holmberg
Malmö University
Malmö, Sweden
johan.holmberg@mau.se

Simon Johansson
Malmö University
Malmö, Sweden
simon.johansson@fjelie.com

ABSTRACT

The Evolutionary Dungeon Designer (EDD) [1] is as a mixed-initiative tool for creating dungeons for adventure games. Results from a user study with game developers positively evaluated EDD as a suitable framework for collaboration between human designers and PCG suggestions, highlighting these as time-saving and inspiring for creating dungeons [2].

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

CCS CONCEPTS

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

KEYWORDS

Procedural Content Generation, Evolutionary Algorithms, Mixed-Initiative Design.

ACM Reference Format:
Alberto Alvarez, Steve Dahlskog, Jose Font, Johan Holmberg, and Simon Johansson. 2018. Assessing Aesthetic Criteria in the Evolutionary Dungeon Designer. In *Foundations of Digital Games 2018 (FDG18), August 7–10, 2018, Malmö, Sweden.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3235765.3235810

1 INTRODUCTION

Procedural content generation (PCG) has been widely used to generate content in games for different reasons, due to constraints in memory [3], create new experiences for the user [18], animations [14] or more recently, to create most of the assets [7]. Moreover, interest in PCG has increased as researchers have explored ways to automate, reduce cost and, produce novel and interesting content, for instance, weapons [6], levels [16], music and sound [9, 15], and even complete commercial games [4].

Search-based procedural content generation (SBPCG) is a popular PCG approach that uses evolutionary algorithms (EA) for guiding the content generation process by means of evaluation functions [17]. Mixed-initiative SBPCG involves human users in the evolutionary process so that promotes the co-creation of human and machine-made designs [11].

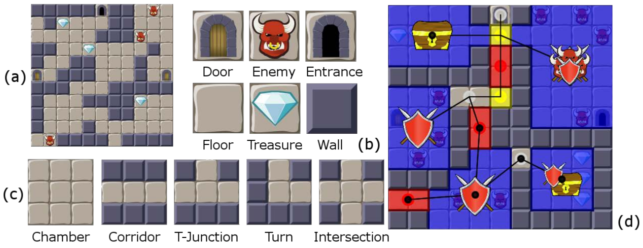


Figure 1: Current version of EDD and its different components. (a) Basic room, (b) different placeable tiles, (c) micro-patterns and (d) meso-patterns.

As discussed by Baldwin et al. [2], it is important for a mixed-initiative SBPCG approach to evaluate the degree to which generated designs are aesthetically pleasing and interesting to the human designer. This is stressed by the designer's will to imprint and preserve their custom designs on the generated content offered by the PCG system. It is a non-trivial task to know which parts the designer wants to preserve, as well as correctly balancing human and procedurally designed content in the generated solutions. This motivates the work presented here, in which we address the need for assessing aesthetic criteria by improving both: the solution encoding mechanism and the fitness evaluation function in EDD's evolutionary algorithm.

This paper is organized as follows: Section 2 presents previous and related works in mixed-initiative design. Section 3 describes

FDG18, August 7–10, 2018, Malmö, Sweden
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6571-0/18/08.
https://doi.org/10.1145/3235765.3235810

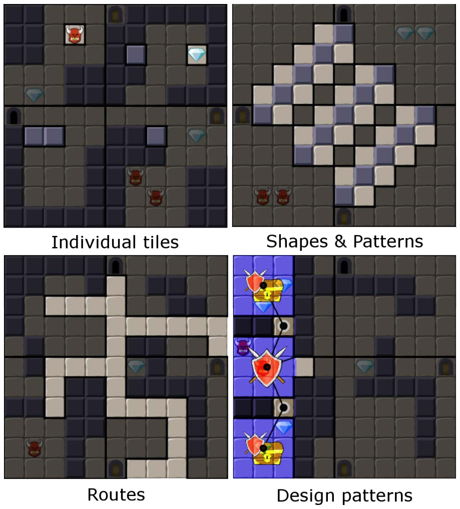


Figure 2: Different uses and possibilities that the designers can have for locking the tiles in the Room, in order to, preserve their manual changes and diverse objectives

in detail the contributions of this paper and presents the results from the laboratory experiments used for validating them. Finally, Section 4 summarizes and discusses these results, as well as sets future questions to be addressed by further research in the area of aesthetic criteria and EDD.

2 RELATED WORK

Aesthetic criteria was specified by previous research as a key feature while evaluating content, as it leads to the generation of more customized content in the eyes of the human designer, whose aesthetic vision on the content is preserved [6, 10, 12].

Interactive evolutionary approaches incorporate human evaluation by allowing the user to select, either implicitly or explicitly, the parents of the next generation of procedurally generated individuals. In Zhang et al. [19] the system allows users to draw simple primitive shapes to seed an evolutionary algorithm and train a neural network with their aesthetic vision. In Galactic Arms Race [6] players preferences on the evolved weapons is implicitly deducted from the amount time they actively select those weapons during the gameplay.

Liapis et al. [10], incorporated visual aesthetics as an evaluation of their generated spaceships by calculating different aesthetic concepts: symmetry along axes, weight distribution or design simplicity. Moreover, Mariño and Lelis [13] generated levels for Mario using symmetry as objective function, which based on their user study, were as visually pleasing as the ones created by human designers and even more than other similar approaches.

2.1 The Evolutionary Dungeon Designer

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

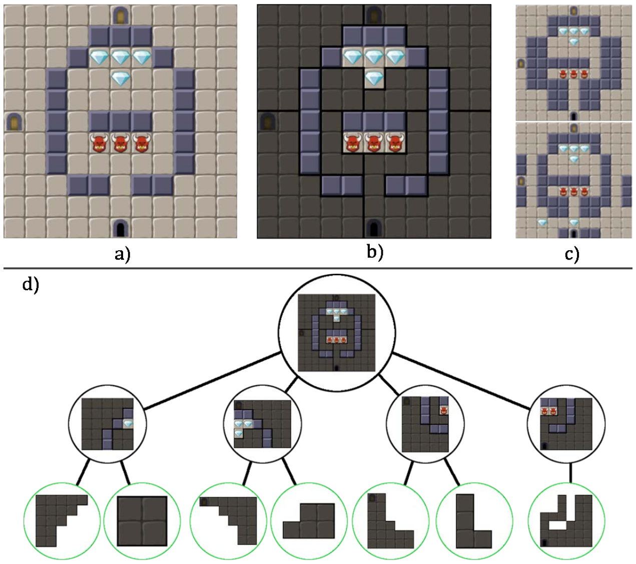


Figure 3: A sample edited room (a) with its division into zones (b) based on the tiles locked by the user. Suggestions preserve these locked tiles (c). The room and its zones are internally represented with a tree structure (d), where the leaf nodes (green) are the valid candidates to operate within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

EDD takes the work presented in The Evolutionary World Designer [5] one step further, by procedurally generating rooms and their specific content. EDD's EA follows the approach of Liapis et al. [10] using the evaluation of the user to change the internal evaluation and configuration of the system. Its fitness evaluation is driven by the use of game design micro- and meso- patterns, as shown in Figure 1 c) and d). A detailed description of EDD's pattern-based fitness, genetic algorithm and mixed-initiative approach can be found in [1] and [2].

3 ASSESSING AESTHETIC CRITERIA

Our approach is divided in two; on one side, the algorithm implicitly has control over different aesthetic criteria using the edited room as a base to measure symmetry and similarity for the EA. On the other side, the designer was given control over what they wanted to preserve by being able to select tiles in the room to be immutable (i.e. not changeable in following generations).

3.1 Preserving Custom Aesthetic Structures

To preserve the aesthetic criteria of a designer's edited room, we give the users the ability to manually lock custom structures in it, preserving these in the upcoming suggestions. This is possible by incorporating a new brush which is used as a complementary modifier when editing the room. The designer can now lock any

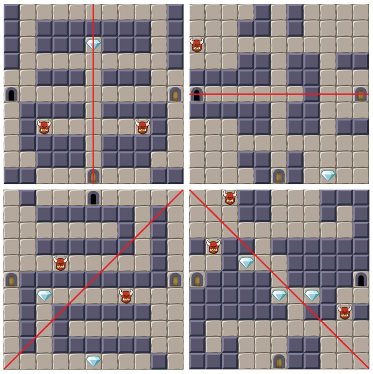


Figure 4: Different types of symmetry evaluated

range of tiles, making it possible to preserve individual tiles, shapes, patterns, routes and even design patterns as shown in Figure 2.

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

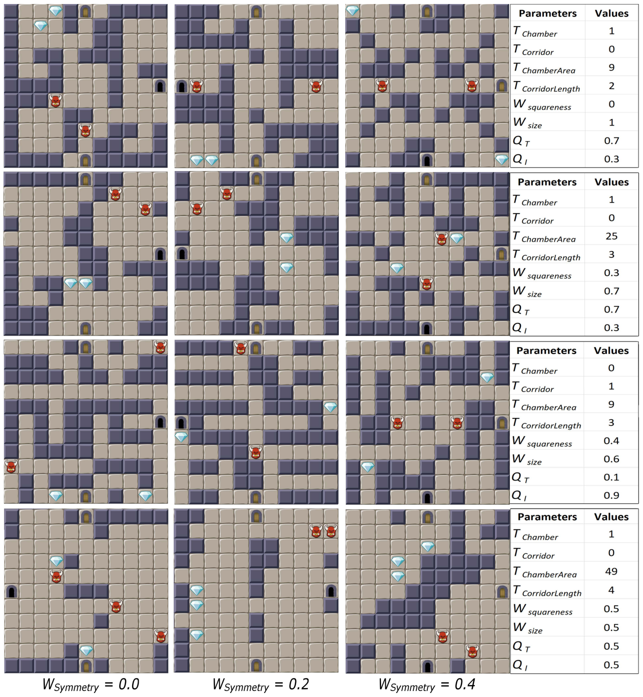
An individual's genotype is now changed from a direct encoding (each tile is a gene) to a semi-direct encoding using a tree structure, with the nodes of the tree as different zones of the room, constructed from the mutable and immutable tiles, and the leaf nodes, only containing sets of mutable tiles,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

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

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

3.2 Evaluating Symmetry and Similarity

While the pattern-based fitness function worked well for functionality purposes, it did not consider nor capture any aesthetic aspects into it. Therefore, in order to consider and preserve visual aesthetic criteria, we evaluate the rooms for their symmetry along the X and Y axes, backslash and front slash diagonal as shown in Figure 4 and calculate the similarity that subsequent individuals had in comparison with the original edited room. For simplicity, we differentiate the room by impassable (i.e. walls) and passable (i.e. floor, treasure and enemy) tiles.



Parameters	Values
$T_{Chamber}$	1
$T_{Corridor}$	0
$T_{ChamberArea}$	9
$T_{CorridorLength}$	2
$W_{squareness}$	0
W_{size}	1
Q_T	0.7
Q_I	0.3

Parameters	Values
$T_{Chamber}$	1
$T_{Corridor}$	0
$T_{ChamberArea}$	25
$T_{CorridorLength}$	3
$W_{squareness}$	0.3
W_{size}	0.7
Q_T	0.7
Q_I	0.3

Parameters	Values
$T_{Chamber}$	0
$T_{Corridor}$	1
$T_{ChamberArea}$	9
$T_{CorridorLength}$	3
$W_{squareness}$	0.4
W_{size}	0.6
Q_T	0.1
Q_I	0.9

Parameters	Values
$T_{Chamber}$	1
$T_{Corridor}$	0
$T_{ChamberArea}$	49
$T_{CorridorLength}$	4
$W_{squareness}$	0.5
W_{size}	0.5
Q_T	0.5
Q_I	0.5

Figure 5: Each row shows three results ($W_{symmetry} = 0, W_{symmetry} = 0.2, W_{symmetry} = 0.4$) produced under the settings displayed on the rightmost column. Metrics adapted from [1].

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

$$f_{symmetry} = \frac{highestSymmetricValue}{totalWalls} \tag{1}$$

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

3.2.2 Similarity evaluation. The similarity value between an edited room and successive evolved rooms is calculated by comparing every tile in the original with the corresponding tile in subsequent individuals. Once the total amount of equal tiles is known, we calculate the similarity percentage based on the total amount of tiles, following equation 2.

$$similarityPercentage = \frac{totalTiles - notSimilarTiles}{totalTiles} \tag{2}$$

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

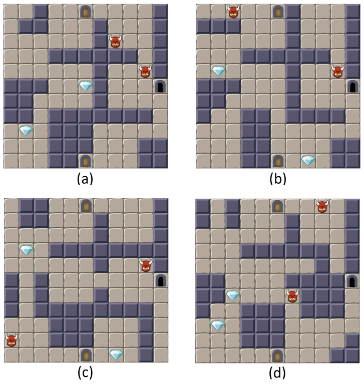


Figure 6: (a) Sample original room and the evolved solutions with different *idealSimilarity* values in order: (b) 0.95, (c) 0.90 and (d) 0.85.

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

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

Finally and expanding over the previous work on EDD [2], these calculations (i.e. $f_{symmetry}$ and $f_{similarity}$) are included into the existing fitness evaluation of an individual as shown in equation 4. $f_{inventorial}$ and $f_{spacial}$, evaluates the overall layout of the room, and the frequency and quality of the design patterns in the room, respectively. An in-depth explanation of both can be found in [2].

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

4 CONCLUSIONS AND FUTURE WORK

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

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

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

We aim to more throughly evaluate the system by incorporate the three techniques into a user study, similar to the one done by Baldwin et al. [2] to validate the tool's capacity on assessing the designer's criteria. It would be interesting to add more aesthetic concepts to evaluate the produced content, for instance, density, simplicity, sparseness and individuality.

The subdivision of the map could be extended to perform a parallel evolution on the custom aesthetic structures locked by the designers and propose interesting variations. Moreover, a zone analysis could be introduced to increase the dungeon's knowledge for the designer by suggesting changes to fulfill different player models, similar to Holmgård's approach [8], or paths and statistics. Finally, we would like to explore different types of representations towards more generative encodings to test, compare and measure the differences and advantages of the resulting maps.

ACKNOWLEDGMENTS

The project was supported by The Crafoord Foundation.

REFERENCES

[1] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg. 2017. Mixed-initiative procedural generation of dungeons using game design patterns. In *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2017*. 25–32.

[2] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg. 2017. Towards pattern-based mixed-initiative dungeon generation. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, Vol. 74. ACM, 1–10.

[3] D. Braben and I. Bell. 1984. Elite. (1984).

[4] C. Browne. 2007. Yavalath. (2007). http://www.cameronius.com/games/yavalath/

[5] J. M. Font, R. Izquierdo, D. Manrique, and J Togelius. 2016. Constrained level generation through grammar-based evolutionary algorithms. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9597 (2016), 558–573.

[6] E. J. Hastings, R. K Guha, and K. O Stanley. 2009. Galactic Arms Race Video Game. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG09).*, Vol. 1. 245–263. https://doi.org/10.1109/CIG.2009.5286468

[7] Hello Games. 2016. No Man's Sky. (2016).

[8] C. Holmgård, A. Liapis, J. Togelius, and G. N. Yannakakis. 2014. Evolving Personas for Player Decision Modeling. In *Proceedings of the Computational Intelligence and Games 2014 (CIG).*

[9] A. K. Hoover, P. A. Szerlip, and K. O. Stanley. 2011. Interactively evolving harmonies through functional scaffolding. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation - GECCO '11*. ACM, 387–394.

[10] A. Liapis, G. N. Yannakakis, and J. Togelius. 2012. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games* 4, 3 (2012), 213–228.

[11] A. Liapis, G. N. Yannakakis, and J. Togelius. 2013. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of the 8th International Conference on the Foundations of Digital Games (FDG 2013)*. 213–220.

[12] A. Machwe and I. C. Parmee. 2006. Integrating aesthetic criteria with evolutionary processes in complex, free-form design – an initial investigation. In *Proceedings of the IEEE International Conf. on Evolutionary Computation.* 165–172.

[13] Julian R. H. Mariño and Levi Lelis. 2016. A Computational Model Based on Symmetry for Generating Visually Pleasing Maps of Platform Games. In *AIIDE.*

[14] Maxis. 2008. Spore. (2008).

[15] M. Scirea, P. Eklund, J. Togelius, and S. Risi. 2017. PRIMAL - IMPROV: Towards Co-Evolutionary Musical Improvisation. In *Proceedings of the Computer Science & Electronic Engineering Conference (CEEC).* IEEE, 172–177.

[16] N. Shaker, M. Nicolau, G. N. Yannakakis, J. Togelius, and M. O'Neill. 2012. Evolving levels for Super Mario Bros using grammatical evolution. In *Proceedings of the IEEE Conference on Computational Intelligence and Games, CIG 2012.* IEEE, Granada, 304–311.

[17] N. Shaker, J. Togelius, and M. J. Nelson. 2016. The search-based approach. In *Procedural Content Generation in Games: A Textbook and an Overview of Current Research.* Springer, Chapter 2, 17–31.

[18] M. Toy, G. Wichman, K. Arnold, and J. Lane. 1980. Rogue. (1980).

[19] J. Zhang, R. Taarnby, A. Liapis, and S. Risi. 2015. DrawCompileEvolve: Sparking interactive evolutionary art with human creations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9027 (2015), 261–273.