



HAL
open science

Genetic WFC: Procedural Level Generation Maximizing Perceived Diversity

Raphaël Bailly, Guillaume Levieux

► **To cite this version:**

Raphaël Bailly, Guillaume Levieux. Genetic WFC: Procedural Level Generation Maximizing Perceived Diversity. IFIP International Conference on Entertainment Computing 2021, Nov 2021, Coimbra, Portugal. hal-03807178

HAL Id: hal-03807178

<https://hal.science/hal-03807178>

Submitted on 9 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Genetic WFC: Procedural Level Generation Maximizing Perceived Diversity

Raphaël Bailly^[0000–0003–4080–0902] and Guillaume Levieux^[0000–0001–8844–4447]

Conservatoire National des Arts et Metiers, CEDRIC, Paris, France
raphael.bailly@cnam.fr
guillaume.levieux@cnam.fr

Abstract. Our research proposes a step towards generating levels mainly focused on player’s experience, to help reach a high level of perceived diversity in procedural level generation. For a given gameplay, we want an algorithm that generates levels with enough structural quality, while optimizing mainly for a specific play experience. Then, diversity should be reached by searching for levels that are far apart in play experience space. To do so, we propose Genetic WFC, a way to mix genetic optimisation with the Wave Function Collapse algorithm, and describe the synthetic player used to evaluate each level’s fitness.

Keywords: Procedural Content Generation · Player Experience · Curiosity · Wave Function Collapse · Genetic Algorithm · Video Games.

1 Introduction

Generating games levels is both a matter of quality and diversity [7]. In this research, we focus on the diversity aspect of PCG. We think that diversity is fundamental to foster the player’s curiosity, one of the main aspect of their motivation to keep playing [6] [4].

To help reach a high level of diversity in procedural level generation, we propose to evaluate our levels with regard to gaming experience. We consider that game levels mainly exist for the gaming experience they provide. Any rule, constraint, or design step that we add to our level generators are here to ensure that the generated levels will provide the kind of experience we are aiming for. We thus want to evaluate our levels using a simple simulation of play experience, and have very few a priori constraints or design rules about the level’s structure. We will let the structure emerge from the search for a specific gaming experience. We thus propose to use a synthetic player, that will go through the level and rate it according to what it’s perception of the level’s structure and gaming events. If our generator is able to generate levels that provide a specific gaming experience, then it might be able to provide diversity by generating levels far from each other in game experience space, using Objective-based Distance as proposed for instance by Preuss [7].

This work was funded with the help of Nouvelle-Aquitaine, for the Kiwi Project, under grant agreement No 18001957.

But choosing to generate levels by evaluating game experience also has downsides. First, this approach tends to be computationally intensive. Indeed, as we put the least amount of knowledge in our generator with regard to the generation process, we are more inclined to rely on search-based PCG [9], using an optimization algorithm to iteratively search through the game level space. Such an iterative search demands a lot of computing time, and might not be able to efficiently drive us to the best solution. Then, we need to be able to simulate the gameplay. Many games rely on the analysis of complex 3D spaces, like shooters, or have a complex game state and decision space, like real-time strategy games. These games might be too complex to be fully simulated at every step of the search-based PCG algorithm. We thus might need to rely on a more abstract version of their gameplay that is easier and faster to simulate. Finally, even if gameplay can be simulated, proposing meaningful game experience metrics can also be challenging.

In order to explore the feasibility of generating levels mainly focused on gaming experience and experiment with a generator, we need to choose a specific gameplay. We mainly draw inspiration from first/third-person shooter and many action RPGs : our synthetic player will navigate through a 3D level, that it needs to explore to reach a certain goal at a certain location, and might be attacked by enemies. Many games share these basic design principles, and thus a large design space may benefit from our approach.

2 Related Works

Our work is mainly based on the Wave Function Collapse (WFC) algorithm, a procedural generation algorithm based on constraint propagation, initially introduced in order to reproduce textures [2]. This algorithm seems promising with regard to our goal. It has already been used to generate playable levels by only propagating simple local constraints, and may thus allow us to focus on evaluating player experience for levels without blatant asset placement errors. This algorithm has also already been extended to add more complex level design constraints [8], and we follow a similar approach by biasing the random selection of assets to influence the selection propagation mechanism. This way, we can use an iterative search algorithm on top of WFC.

Indeed, game levels can be generated by computing multiple iteration of the generation stage until the best possible solution according to a certain goal is reached. A way to perform this iterative search is to rely on an evolutionary algorithm to optimize, step by step, a population of candidate levels [11]. This approach has already been applied in 2D platform games [1], as well RTS games [10].

Iterative search relies on the evaluation of generated levels' fitness. In order to focus on gaming experience, we propose to use a synthetic player, as in [5, 3].

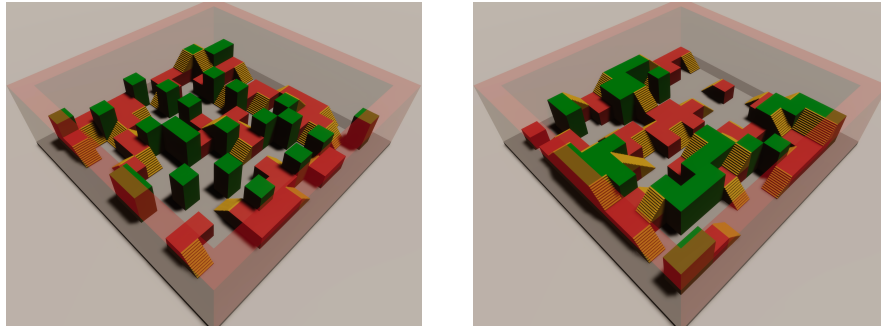
3 Genetic WFC

In order to use genetic optimisation, we need to introduce a way to control the WFC and encode it into a chromosome. To do so, we add *boost zones* that increase the selection probability of an asset in a specific area. This way, we can locally influence the asset selected by the WFC, without violating any constraints. The position and chosen asset of each boost zone are encoded in our chromosomes.

To evaluate each generated level, we use a synthetic player that evaluates its *safety* and *curiosity* feelings. When playing a shooter, the level geometry provides a way to hide from other players and their shots. We thus compute a safety score using the number of cells that the agent can see through in four directions. The farthest it can see, the less safe it will feel.

The curiosity feeling of the agent is related to the last time a cell was visited. If a cell was visited a long time ago, going there will provide more information and the agent will be more curious than if it was visited a few steps ago. We then use a weighted average of curiosity and safety, averaged for the whole play session of the agent, to rate a the gaming experience and thus calculate the fitness of each level. We show examples of levels generated with a positive (Safe) and negative (Unsafe) impact of safety on the total fitness in figure 1.

WFC computing time grows with the number of assets. As a result, we split level generation in two stages. Indeed, many graphical assets have the same level design function, and can be represented by a single greyblock asset. We thus firstly generate an abstract version of the level, using iterative search, with only greyblocks assets. Then, we use a final WFC pass to convert greyblock assets to final graphical assets.



(a) Safe Level

(b) Unsafe Level

Fig. 1: Example of results obtained with greyblocks assets

4 Discussion and Future Work

Mixing WFC with genetic search has a cost, as both algorithms can be computationally expensive. We thus need to combine Genetic WFC with more straightforward search based PCG approaches like pure genetic optimisation, without WFC, or brute force search using WFC only. Pure genetic search might reach higher fitness values, but create levels that violate asset placement constraints. Brute force with WFC only should provide a baseline to evaluate the gains provided by the genetic optimisation.

Also, our gameplay simulation is still very basic. We showed that with a simple walking agent with safety and curiosity metrics, we were able to influence the generation of game levels. By adding simple enemies and other play experience metrics, we might be able to explore a more diverse set of gaming experiences.

Then, we need to have real players play our levels to evaluate Genetic WFC's capacity to maximize perceived diversity, as well as the capacity of our metrics to describe meaningful dimensions of play experience.

References

1. Dahlskog, S., Togelius, J.: Procedural content generation using patterns as objectives. In: European Conference on the Applications of Evolutionary Computation. pp. 325–336. Springer (2014)
2. Gumin, M.: Wave function collapse.
URL: <https://github.com/mxgmn/WaveFunctionCollapse> (2016), last visited 3 Oct. 2019
3. Holmgård, C., Green, M.C., Liapis, A., Togelius, J.: Automated playtesting with procedural personas with evolved heuristics. *IEEE Transactions on Games* (2018)
4. Koster, R.: Theory of fun for game design. " O'Reilly Media, Inc." (2013)
5. Liapis, A., Holmgård, C., Yannakakis, G.N., Togelius, J.: Procedural personas as critics for dungeon generation. In: European Conference on the Applications of Evolutionary Computation. pp. 331–343. Springer (2015)
6. Malone, T.W.: Toward a theory of intrinsically motivating instruction. *Cognitive science* **5**(4), 333–369 (1981)
7. Preuss, M., Liapis, A., Togelius, J.: Searching for good and diverse game levels. In: 2014 IEEE Conference on Computational Intelligence and Games. pp. 1–8. IEEE (2014). <https://doi.org/10.1109/cig.2014.6932908>
8. Sandhu, A., Chen, Z., McCoy, J.: Enhancing wave function collapse with design-level constraints. In: Proceedings of the 14th International Conference on the Foundations of Digital Games. pp. 1–9 (2019)
9. Shaker, N., Togelius, J., Nelson, M.J.: Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Springer (2016), website: <http://pcgbook.com/>. Last visited 30 Jun. 2021
10. Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelbäck, J., Yannakakis, G.N.: Multiobjective exploration of the starcraft map space. In: Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games. pp. 265–272. IEEE (2010)
11. Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* **3**(3), 172–186 (2011)