Expanding Wave Function Collapse with Growing Grids for Procedural Map Generation

Tobias Møller*
toeb1992@live.dk
Aalborg University
Copenhagen, Denmark

Jonas Billeskov*
jonas.aksel@gmail.com
Aalborg University
Copenhagen, Denmark

George Palamas
gpa@create.aau.dk
Aalborg University
Copenhagen, Denmark

ABSTRACT

This paper proposes a method to augment the basic Wave Function Collapse algorithm with the Growing Grid neural network for procedural map generation. The system accept a higher level description of the map, in the form of an image, and the system returns a list of solutions based on learning parameters and module constraints. First, a description of the technical implementation of the system is given. Second, the augmented WFC variant is evaluated from a user experience perspective, in the context of a first person cognitive map formation.

KEYWORDS

Wave Fuction Collapse, Growing Grid, User Experience, Navigation, Spatial Mental Maps, Procedural Map Generation

ACM Reference Format:

Tobias Møller, Jonas Billeskov, and George Palamas. 2020. Expanding Wave Function Collapse with Growing Grids for Procedural Map Generation. In *International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3402942.3402987

1 INTRODUCTION

Procedural content generation (PCG) offers the advantage of ease of implementation by moving part of the artistic work-flow from the artist to the programmer and technical artist [16]. This approach has the drawback of the loss of control over the end result, which in turn might lower the graphical fidelity and the perceived realism of the content. Another aspect of PCG is the requirement of content evaluation, guaranteeing that the game is always playable [4].

Among the uses of PCG, procedural map generation poses an active role because it determines the architectural, aesthetic and utilitarian values of a virtual environment. This is important because the factors that most contribute to the perception of a user being present in a virtual world, lie in the surrounding visual context. Many game studios use PCG tools which can produce game assets that look different, but these changes are rather subtle and the player, eventually, will recognize a reused pattern [15].

A popular algorithm is the Wave Function Collapse (WFC), a constraint satisfaction algorithm inspired after the concept Wave Function Collapse from quantum physics, where an unobserved state allows for all states to be possible while observations constraints the possibilities. WFC became popular very quickly because it introduced design constraints into the generative process, and because of its modularity and expansibility.

In addition to this, models can be placed beforehand, as constraints, which makes it possible to handcraft parts of the world and let the WFC algorithm handle the rest. Despite the potential advantages of WFC, there are restrictions on the maps being produced because the rules have a global impact on the generated outcome and because it requires the developer to have a deep understanding of how these constraint based techniques work. PCG for game design need to ensure some important criteria; play-ability and an acceptable degree of aesthetic value. In the case of a map building algorithm, the way of spatial distribution and spatial coverage are the eligibility criteria that should be applied to ensure that all game purposes are being fulfilled. The proposed PCG system is an augmentation to the WFC algorithm based on the Growing Grid (GG) algorithm, which is a self-adaptive network structure that can be used to model a spatial distribution. The proposed approach accepts as an input a gray-scale image which is used to approximate the shape, but also the local densities, of a map from a sketch drawing (figure 2). The paper gives a detailed description of the system and demonstrates the applicability of the approach under a variety of conditions. The system can be utilized by users without any programming background.

2 BACKGROUND

Map generation has a long history going back to titles such as the Rogue (1980) with an entire sub-genre of video games named Roguelikes. Rogue was a fantasy game where the player had to explore a PCG dungeon, unique each time. Unlike other popular approaches, based on texture generation methods [6], WFC was mostly adopted in order to generate levels during game-play [7]. WFC is an appealing solution due to its probabilistic representation and because of the rule-based constraint reasoning. These two properties allow for solution generation but also for solution rejection, in case of invalid maps. WFC received attention due to its, quite enjoyable to watch, nature-like properties of procedural art generation, which pushed the popularity of the algorithm even further. The algorithm was later adapted for game map generation and very fast adapted to work in three dimensions.

A notable application of WFC was the game Proc Skater (2016) by Arcadia Clojure where the algorithm used for level generation. A variation of this approach used to create platforms in the game Bug with a Gun (2018) by SelfShame. Most constraint satisfaction algorithms are being used for offline content generation such as

*All authors contributed equally to this research.

FDG '20, September 15–18, 2020, Bugibba, Malta

ACM ISBN 978-1-4503-8807-8/20/09…$15.00
https://doi.org/10.1145/3402942.3402987

Tanagra [13] which is a tool that use a mixed initiative approach [8]. Tanagra empowers level designers with the ability to influence the level generation at run-time. This is a more design-led approach compared to automatically generated solutions where the designer can only refine the final solution [12]. Two notable rogue-like games making use of WFC with online constraints are Bad North [2], by Plausible Concept and Caves of Qud [5] by Freehold Games.

Almost all applications utilizing WFC make use of regular grids and evenly sized slots. This makes WFC a good candidate for alterations which could create the basis for a more robust design tool but without rejecting the qualities of the original WFC algorithm. Moreover, WFC only requires that the grid slots have an equal amount of edges, and does not necessarily need to have a regular shape, thus it seems quite promising to generate the grid by using a Growing Grid (GG) neural network that can learn topological relationships from spatial data represented by an image.

2.1 Wave Function Collapse

WFC is a constraint based procedural algorithm based in the idea that a quantum particle can exist in a multi-state when unobserved. As soon as the particle is observed, the possibilities disappear and the wave function collapses. WFC can be implemented with two different models, the tiled model and the overlapping model but for the purpose of this study the focus has been placed solely on the tiled model. The WFC algorithm initializes a grid, where each cell defines a slot. Each slot in the grid represents a module which contains information about the 3D model and the constraints for the module's neighbours. In the complete unobserved state, each slot has the same possibility to be filled by each module possible. The modules have lists of possible neighbours they allow to exist next to them. This means that if one slot is collapsed down to a single possible module, the neighbouring slots will also be restricted. This possibility constraint spreads to neighbouring slots, also constraining them in possible modules. WFC can be extended to many dimensions but the most common applications include two or three dimensions. The algorithm can be utilized with different grid shapes, although square is the most common shape. Hexagon grids with six neighbours are also popular but designing the set of modules that can fill out the grid can be a hard task, as they need to fit each other, which requires some flexibility for possible neighbours.

2.2 Augmenting the topology of WFC

WFC is organized in a regular grid which, depending on the module set, it may end up being very apparent in the final product. A certain module will always look the same given that all slots are the same size and shape, which limits the diversity of WFC. One way to combat this is to create an extensive module set but the modules will still be organized in a regular grid and will look thereafter. This is especially undesired when the objective is to create natural looking environments, such as terrains and caves, where the occurrence of regular, grid-based structures is very rare. A potential solution could be the use of *irregular quadrilateral grids*. This grid type consists of irregular polygons, with four edges and four vertices, that can be either convex or concave. The following section will describe a methodology that can augment the WFC with a quadrilateral grid.

2.3 Irregular Grids

The simplest method to create a structured irregular quadri-lateral grid is to distort the structure by introducing some noise to each vertex in the grid. This method can form irregularities but the base structure would still be visible. Another drawback is the maximum amount of noise that can be tolerated. If the noise exceeds a certain level then the vertices will overlap into neighbouring slots and the grid will be non-serviceable.

2.4 Growing Grid Network

The Growing Grid network (GG), proposed by Bernd Fritzke [3] is a self-organizing network which is generated by a growth process. The algorithm is used for the generation of topology-preserving mappings, for vector quantization and for dimensionality-reduction. The network has the form of a rectangular grid which increases its size during an adaptation phase. By inserting complete rows or columns the grid adapts its dimensions to a given input distribution 1.

The network grow until some performance criteria are met or until a desired network is reached. A final fine-tuning phase, with decaying adaptation strength, optimizes the topology of the network.

Assuming a probability distribution P(ξ) of n-dimensional vectors, the GG is initialized to a grid of size $m * k$ nodes. Each node position on the grid, associated with a n-dimensional vector, is adapted by a sequence of input data according to $P(\xi)$. For each input signal ξ the best matching unit s with most similar reference vector is determined according to:

$$\|w_s - \xi\| \leq \|w_c - \xi\| \quad (\forall c A) \tag{1}$$

The position of the unit s and its neighboring units in A are adapted towards ξ. The adaptation strength of each unit depends on the city-block distance of the grid to the best matching unit s. The neighboring units are also dragged towards ξ weighted by their distance in the grid to the best matching unit. This process keeps looping until an arbitrary epoch size is reached. After this, the algorithm will insert a new row or column in the grid. The row or column will be placed between the two units with the highest distance. Because of the use of the city-block distance, instead of the Euclidean distance, the grid will relax itself in the distribution.

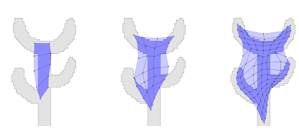

Figure 1: The Growing Grid expands in order to capture the 'cactus' shape.

3 IMPLEMENTATION

The intended purpose of GG is to act as a space filling algorithm where the input image determines the shape of the terrain (figure 2). By utilising a more complex image, such as a height-map, this method can also preserve the local densities of the image i.e. size of each slot. There are many parameters controlling the performance of the algorithm which requires a relative familiarity. Moreover, the same set of parameters does not ensure repeatability of the results.

3.1 Space Filling

Controlling the distribution of modules in WFC is challenging because the constraint system determines the neighboring modules locally. Common solutions to global control of module distribution are based on weighting the significance of modules. For an example, a module might have a higher chance of being selected than others but with the drawback of invalid distribution of models and thus incoherent maps.

3.1.1 Blend shapes in a WFC map. Combining WFC and GG required a way to skew the modules in order to fit into these irregular slots. Blend shapes, mostly used for character facial animations, is an attractive way to dynamically alter the shape of a model because the mesh vertices can be interpolated between two different poses. For this study, eight different poses assigned to each module's mesh, controlling the corner's position of the model in the slot (figure 4).

3.1.2 Chiseling. This is a technique used to ensure a path between two or more points [14], which could for example be the path between an entrance and an exit on a map. The algorithm works by first placing nodes in the map and then populating the rest of the nodes with traversable objects. The algorithm iteratively removes traversable objects at random while ensuring that a path between the points is still intact by using a depth first search, which allows removal of an object if all the placed nodes have been visited. The algorithm repeats until no more nodes can be removed, without breaking the path.

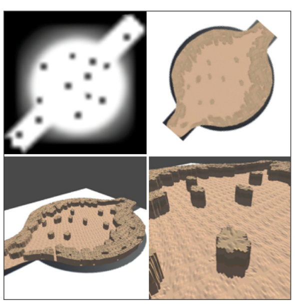

Figure 2: Starting with a gray-scale sketch, the system can generate a map based on a 3D 'cave' module set, modeling the overall shape and local densities of the pixel data.

4 OPTIMIZING GAME MAPS FOR EXPERIENCE

This evaluation focuses on the perceived differences between maps generated using WFC and WFC augmented with GG, from a first person perspective (figure 3).

4.1 Spatial Mental Maps

4.1.1 Spatial Mental Maps - Unsupervised Study. . Humans form mental maps in order to determine the spatial character of a game environment [11]. These maps are compact topological representations of the surrounding space [10] which contains a list of elements defined by Lynch [9] as a network of paths and landmarks. Game environments should provide the player with the necessary information for them to successfully reach the end destination. Therefore this evaluation attempts to explore how the formation of mental maps differ between WFC-GG and WFC. Fifteen participants were placed in the map and allowed to freely navigate for one minute. After time was up, participants were shown an image with six different maps, in perspective view and asked to point out the one corresponding to the environment they were placed in. The procedure repeated for a number of three times for each participant. Equal number of maps of the same size generated with WFC-GG and WFC, respectively (figure 4).

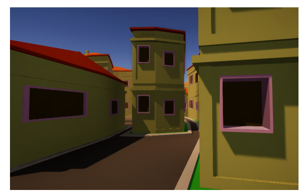

Figure 3: A generated city experienced from a first person perspective.

4.1.2 Spatial Mental Maps - Supervised Study. Fifteen Participants were also asked questions, for each map, about their confidence in their choice, their overview over the map and if they used any visual cues from the environment to match the maps in the image. They were also asked for general techniques they had used for navigation and what map they thought was the hardest to match in hindsight.

However, the patterns generated from WFC-GG contain unique versions of the modules that could act as distinct visual cues or local landmarks, which might enhance player's navigation and overall experience.

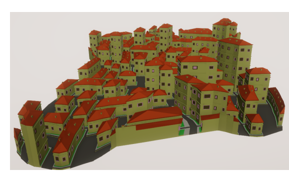

Figure 4: A perspective view of a city generated with WFC-GG.

5 RESULTS

The results from this evaluation consists of the participants' guesses matching the maps depicted in the image and their answers to a set of questions (tables 1 and 2).

The answers to the Question "Did you base your choice on a specific visual cue of the map?" showed that 10 out 15 participants navigating WFC-GG maps, described the shape of the building, using words such as "curved","thick" and "triangle". 19 answers of

Table 1: Evaluation of mental map formation

variable	Correct Guesses
WFC	60%
WFC-GG	80%

Table 2: Confidence on mental map formation

variable	Mean_C	Med_C	Mean_O	Med_O
WFC	7.66	8	6.4	6
WFC-GG	7.93	9	7.1	7

a total of 30 mentioned the height of the buildings signified with the words "tower" and "height" or mentioned the number of floors. For the question "What was the most distinct visual feature used for navigation?", eight participants mentioned the height of the buildings, three participants mentioned the shape of the buildings and three participants mentioned the corners of the maps. In the question "Which map would you rate the most difficult to match to the picture?" six participants mentioned a map based on WFC and four participants a map based on WFC-GG. Additionally, one participant mentioned a WFC map being the easiest and one participant mentioned a WFC-GG map being the easiest.

6 CONCLUSIONS & DISCUSSION

The results showed that the participants were better at matching the WFC-GG maps than the WFC maps. This suggests that generating maps of arbitrary size improve the players' ability to form mental maps of the environment. These maps also increased participants' self-asserted confidence and overview over the map. This seems in line with the work [1] suggesting that players would use patterns in procedural generated maps to orient themselves, even though it did not improve their navigation ability. Many of the evaluation participants mentioned that they used higher buildings as distinct landmarks, although the majority of the participants did mention the unique shape of the buildings. This together with the self assessment results and matching ability could indicate that when users experience the diverse maps created by WFC and GG they form more vivid mental representations of the surrounding environment.

Moreover, as a design tool, the GG can be adjusted using the predetermined variables and with different inputs can create interesting and unique results. By combining the WFC and the GG algorithms allows the designer to define the shape of the grid as well as the local densities of the slots, in a more intuitive way, directly reflecting the designer's intention, making the WFC more flexible and easy to use.

REFERENCES

[1] Michael Biggs, Ute Fischer, and Michael Nitsche. 2008. Supporting wayfinding through patterns within procedurally generated virtual environments. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*. 123–128.
[2] Plausible Concept. 2018. Bad North. https://www.badnorth.com/.
[3] Bernd Fritzke. 1995. Growing grid—a self-organizing network with constant neighborhood range and adaptation strength. *Neural processing letters* 2, 5 (1995), 9–13.
[4] Philip Galanter. 2012. Computational aesthetic evaluation: past and future. In *Computers and creativity*. Springer, 255–293.
[5] Freehold Games. 2015. Caves of Qud. http://www.cavesofqud.com.
[6] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9, 1 (2013), 1–22.
[7] Isaac Karth and Adam M Smith. 2017. WaveFunctionCollapse is constraint solving in the wild. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*. 1–10.
[8] Antonios Liapis, Gillian Smith, and Noor Shaker. 2016. Mixed-initiative content creation. In *Procedural content generation in games*. Springer, 195–214.
[9] Kevin Lynch. 1960. The image of the city (Vol. 11).
[10] George Palamas and J Andrew Ware. 2013. Sub-goal based Robot Visual Navigation through Sensorial Space Tesselation. *International Journal of Advanced Research in Artificial Intelligence* 2, 11 (2013).
[11] Amalie Rosenkvist, David Sebastian Eriksen, Jeppe Koehlert, Miicha Valimaa, Mikkel Brogaard Vittrup, Anastasia Andreasen, and George Palamas. 2019. Hearing with Eyes in Virtual Reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 1349–1350.
[12] Arunpreet Sandhu, Zeyuan Chen, and Joshua McCoy. 2019. Enhancing wave function collapse with design-level constraints. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*. 1–9.
[13] Gillian Smith, Jim Whitehead, and Michael Mateas. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on computational intelligence and AI in games* 3, 3 (2011), 201–215.
[14] Boris the Brave. 2018. Chiseled Random Paths. (2018). https://github.com/BorisTheBrave/chiseled-random-paths.
[15] Julian Togelius, Noor Shaker, and Joris Dormans. 2016. Grammars and L-systems with applications to vegetation and levels. In *Procedural Content Generation in Games*. Springer, 73–98.
[16] Xubo Yang, Milo Yip, and Xiaoyue Xu. 2009. Visual effects in computer games. *Computer* 42, 7 (2009), 48–56.