
Procedural Generation of Video Game Cities for Specific Video Game Genres Using WaveFunctionCollapse (WFC)

Werner Gaisbauer

University of Vienna
Währinger Straße 29, A-1090
Vienna, Austria
werner.gaisbauer@gmail.com

William L. Raffe

Jaime A. Garcia
University of Technology Sydney
PO Box 123 Broadway NSW
2007 Australia
william.raffe@uts.edu.au
jaime.garcia@uts.edu.au

Helmut Hlavacs

University of Vienna
Währinger Straße 29, A-1090
Vienna, Austria
helmut.hlavacs@univie.ac.at

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright held by the owner/author(s).
CHI PLAY EA '19, October 22–25, 2019, Barcelona, Spain
ACM 978-1-4503-6871-1/19/10.
<https://doi.org/10.1145/3341215.3356255>

Abstract

Virtual cities as background scenarios can be used for many 3D video game genres like action. However, the procedural generation of virtual cities for specific video game genres is an on-going research problem. In this paper, we seek to establish a grounding for future work into city generation for specific game genres by exploring how game designers approach existing generation tool-sets. Firstly, we look at the video game city Skara Brae from the party-based role-playing game *The Bard's Tale* and try to replicate it using the Wave Function Collapse (WFC) approach to procedural generation. We show in two experimental conditions which parameters for WFC are suitable for replicating the city. Secondly, a pilot user study with eight users shows how they approach creating different video game cities after they preselect a video game genre. The users' video game level ideas are then discussed, and different output levels are generated using WFC.

Author Keywords

Procedural content generation (PCG); Video game cities; Video game genres; WaveFunctionCollapse (WFC);

CCS Concepts

•Human-centered computing → Human computer interaction (HCI); User studies; *Graphical user interfaces*;
•Applied computing → Computer games; •Software and

its engineering → *Interactive games*;

Introduction

The topic of this paper is the procedural generation of video game cities based on specific video game genres. While the procedural generation of cities [3] and the procedural generation of game content [14] are both highly researched fields, we believe there is a gap in the generation of virtual cities for specific game genres. That is, rather than generating generic virtual cities [5] or cities for a real-world context [9], the cities should be generated to support the gameplay in question. This paper aims to provide an initial exploration of this gap by investigating how student game designers interact with an existing procedural generation technique for the purposes of creating city layouts for a specific game. This work is done to inform further investigation into this area of research.

Firstly, we picked the party-based role-playing video game *Tales of the Unknown, Vol.1: The Bard's Tale* [1] that has the virtual city Skara Brae as a starting background scenario. We studied the Skara Brae city map from this game and tried to learn what steps are necessary to take to make similar game levels (w.r.t. city layout, complexity, and size) using WaveFunctionCollapse (WFC). However, we did not try to copy the game look and feel or game art but used simple game objects (3D cubes) for the buildings instead and show in principle how to create similar cities as a proof of concept. We show in two experimental conditions which parameters for WFC are suitable for replicating the city form. The gained tasks can then practically be used to do similar things with WFC for other video game cities.



Figure 1: Start screen of the C64 version with the singing bard telling his story about Skara Brae.¹ Image © Electronic Arts.



Figure 2: Screenshot of the Apple II-GS version. The Guild of Adventurers is where the adventure starts and where the player can build a party of adventurers.² Image © Electronic Arts.

¹Image source from <https://bardstaleonline.com/images/shots/bt-c64-title.gif>

²Image source from <https://bardstaleonline.com/images/shots/bt-apples-guild.gif>

Secondly, a pilot user study with eight users was conducted that shows how human test users approach designing different video game cities after they preselect a video game genre from a given list [16]. The users' video game level ideas are then discussed, and different output levels are generated using WFC.

WaveFunctionCollapse

WaveFunctionCollapse (WFC) by Maxim Gumin is an algorithm for procedurally generating bitmap patterns given an input sample bitmap pattern [6]. The algorithm deconstructs the example input pattern and tries to replicate it with some procedural variations so the user can get high levels of variation given a single example pattern. An important parameter of WFC is N which defines the size of the $N \times N$ input chunks that are used to assemble the output (in other words, the algorithm generates bitmaps that are locally similar to the input bitmap).

Instead of pixels in a bitmap, it is also possible to use game objects in a video game engine so complete 3D scenarios can be generated this way. In the scope of this paper, we used the WFC adaptation by Joseph Parker for the Unity video game engine which has been used for generating skateparks in the *Proc Skater* game, plateaus in the game *Swapland*, and platform levels in the 2018 game *Bug with a Gun* [12]. An in-depth academic discussion of WFC is given in [7] and a good tutorial for how to setup WFC and some tips for procedural generation of game levels can be found here [11].

The Bard's Tale (1985 video game)

The original *Bard's Tale* (figures 1 and 2 show some screenshots from the game to give the reader an idea about the game look and feel), officially titled *Tales of the Unknown, Vol.1: The Bard's Tale* is a first-person party-based RPG [1].

In the game the player commands a stalwart party of player characters through cities and dungeons in a fantasy setting collecting treasures, fighting NPCs and monsters. This game uses a grid-based map system, and therefore it is an ideal candidate for usage with the WFC algorithm.



Figure 3: The city of Skara Brae (official map from Interplay).³ Image © Electronic Arts.

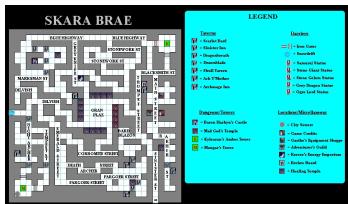


Figure 4: The city of Skara Brae on a grid map together with a legend.⁴ Image © Darendor (owner of the Bard's Tale Wiki).

Parameter Study

Capturing the Spirit of Skara Brae

How is it possible to procedurally replicate the spirit of a game city? What parameters for WFC are best when replicating a game city? If we knew the answer to these questions, then we could use that knowledge and learnings to create other video game cities for similar use cases (game genres) or even generalize the learnings to other game genres. To start answering the before mentioned questions, we analyzed the Skara Brae game map (see Figures 3 and 4) from the video game The Bard's Tale and manually copied a part of it as input for the Unity WFC input map. As a next step, we tried to replicate Skara Brae given that input pattern using different parameters and analyzing the output (comparing it with the original city layout).

The input grid area should be large enough for higher values of N [11]; our 20×20 cells input area seemed to be big enough for $N = 3$ and gave stable results, whereas higher values of N did not give any results, i.e., empty output areas. $N = 2$ usually works already with small input grids like 4×4 cells or pixels (e.g., see the 4×4 pixel Red Maze sample in [7]). As can be seen in the examples of the WFC GitHub repository README [6] the input area is usually smaller than the output area. This fact makes WFC excellent for rapid prototyping of game levels because the

user does not need to draw the whole game level but just a part of it to give the algorithm some idea of how the output should look like. It is also possible to generate endless content by setting *Periodic output = true*, which was successfully tested with both experimental conditions ($N = 2$ and $N = 3$). *Periodic input* was set to *false* because Skara Brae is a video game city with limits (a 30×30 grid which does not repeat).

In Figure 5, two different experimental cases are discussed in a parameter study (for $N = 2$ and $N = 3$). We used the same parameters for running both experiments and only changed the value of N (size of input chunks along each dimension) between the experiments (Experimental Case I: $N = 2$ and Experimental Case II: $N = 3$). The parameters shared by both experiments were: *Gridsize = 1*, *Width = 30*, *Depth = 30*, *Seed = 0*, *Periodicinput = false*, *Periodicoutput = false*, *Symmetry = 1*, *Foundation = 0*, *Iterations = 0*, *Incremental = true*. Experimental Case I: $N = 2$ seems to produce more “twisty little passages” with dead ends, the reason being probably that the input chunks are rather small (2×2 cells) and that seems to better capture the dead ends in the input map. Experimental Case II: $N = 3$ seems to better produce larger open spaces like the “Gran Plaz” on the input map (see Figures 3 and 4). This is most likely because it takes bigger input chunks (3×3 cells) and connects them in the output and a bigger input chunk can contain more empty cells. Procedural generation of video game cities with open areas as well as dead ends (or even dead areas for teleporting there) might be interesting for other game genres too (similar to the generated cities in Figure 5).

Steps Towards a Working Game

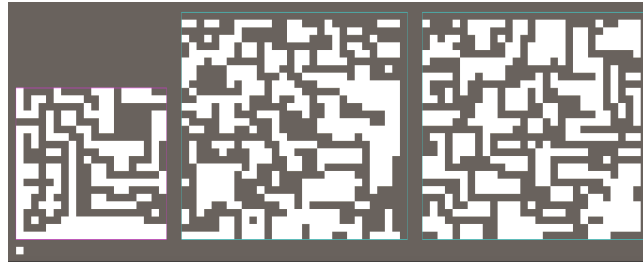
The standard implementation of WFC does not guarantee that the output is walkable, although most of it seems to be

³Image source from <https://bardstaleonline.com/files/lmaps/bt1-skara-brae.jpg>

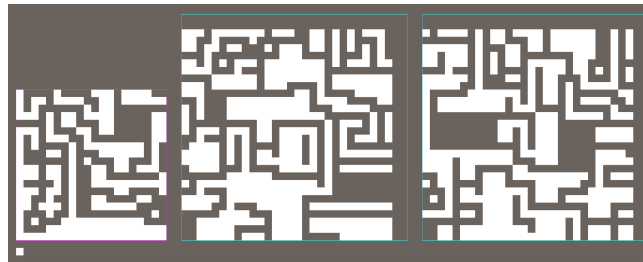
⁴Image source from <https://vignette.wikia.nocookie.net/bardstale/images/1/13/SkaraBrae.PNG/revision/latest?cb=20090203153715>



Figure 6: Tile set of Ultima I: The First Age of Darkness. This map format was reverse engineered by TheAlmightyGuru.⁵ Image © California Pacific Computer Company.



(a) Experimental Case I: $N = 2$.



(b) Experimental Case II: $N = 3$.

Figure 5: Possible outputs (the two larger 30×30 grid areas on the right hand side of each image) for running WFC on a sub area of the Skara Brae input map (the smaller 20×20 grid area on the left hand side of each image). The input map is a copy of the Skara Brae map starting with a window from the lower left corner and going 20 grid cells into each dimension (the original map of Skara Brae is 30×30 grid cells).

playable as is (out of the box without any modifications to the basic WFC algorithm). Unique buildings in Skara Brae (e.g., Adventurer's Guild, Roscoe's Energy Emporium, Review Board) can be added to the generated city in a second pass by randomly replacing existing buildings. A level designer would then need to be careful that each area of the map can be reached via a walkable area or utilize a placing algorithm that checks for that constraint. Another viable solution would be to modify the WFC algorithm to generate walkable areas by default. Also, monsters can be added in a second pass by randomly placing them on open areas that should be reachable by the player. Additionally, it is necessary to skin the buildings which are simple cubes with 3D game objects made by artists in a second pass.

Pilot User Study

We performed a pilot user study with eight test users from the internal games research team to test how human users would approach designing video game levels using our city construction toolkit. We based the WFC input tile set on the "Ultima I: The First Age of Darkness" [4] role-playing video game which is comprised of 52 different tiles including basic tiles (like Water, Grass, and Forest), and monster tiles (like Ness Monster, Giant Squid, and Dragon Turtle), and even science fiction oriented vehicle (like Aircar, Shuttle, and Time Machine) to give the users something to work with. In Figure 6 the set of map tiles from the video game Ultima I is displayed. The tested users were asked to choose a video game genre first from the Wikipedia entry [16] and then design a demo level with a city as a background scenario in it. After designing the level, they were shown sample outputs of the WFC algorithm based on the input level they created, and they were asked to explain their game level design and give some general feedback.

⁵Image source from http://www.shikadi.net/moddingwiki/Ultima_I_Tile_Graphic_Format

In Figure 7, we show four different cases of our pilot user study. It is interesting to see that not all concepts from the input level design translate well to the generated output levels. For example, single important game objects like the “Shuttle” or the “Time Machine” are often not present in the output so they would have to be placed manually. One test user tried to hide game objects from the player by placing them behind mountains at the far edges of the map. The WFC algorithm places the game objects behind the mountains but often not hidden from the player. Also, some generated levels are rather useless, i.e., just a level with water tiles and one “Frigate” or empty levels can happen. General feedback regarding the user interface was given by users too, including missing undo/redo, need for selecting a box for a faster drawing of tiles, and missing feedback on what tile has been selected. Also, one test user reported that he was not sure about sense of scale, e.g., does the town tile represent just one building or a bunch of buildings? The same is valid for enemy tiles.

When running the pilot user study, we chose a similar setup to the earlier section. A 20×20 cells input area and two 30×30 cells output areas next to it to show procedural variety. We chose $N = 2$ for all experiments in the pilot user study to stay consistent. $N = 3$ is possible too but gives different results in the procedural output. What value of N is better depends on the goal, i.e., bigger chunks can capture more detail at once, but this is not always an advantage depending on the input level. For some input levels, the procedural generation sometimes does not finish, and no output is generated. This fact is a peculiarity of the WFC algorithm and usually changing the input level a bit can remedy the situation [10].

Future Work and Conclusion

For future work, we want to extend the parameter study to more genres and games and also do a more extensive user study. Also, we want to investigate the interaction of human designer and generative algorithms (WFC) more closely as discussed in these studies on computational co-creativity [2, 8, 17]. Furthermore, we want to take a more thorough look at the demonstration of generators expressivity [15], evaluation of generation for gameplay, and expanding the generative algorithm beyond WFC's N. Generative Design in Minecraft (GDMC) [13] also investigates city generation within video games and provides interesting ideas on challenges for human-level settlement generation that could be used as a testbed for future experiments with WFC to move closer to human-level design.

As a conclusion, we performed a parameter study to show how to use WFC to replicate a specific video game city (Skara Brae). These findings might be of use to replicate similar video game cities of the same video game genre or even other genres. Furthermore, we did a pilot user study that shows how human test users approach the design of video game cities based on genre type and discuss their creations which are then used to generate different output levels using WFC.

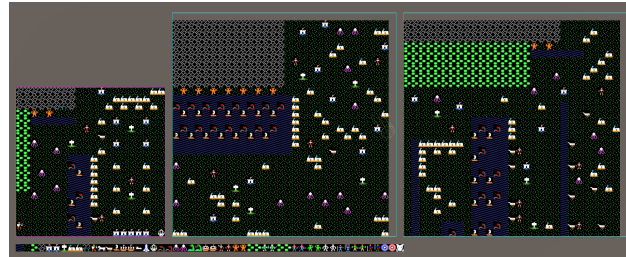
Acknowledgements

We thank the University of Vienna for supporting this work through a doctoral promotion stipend. The first author acknowledges a visiting appointment with UTS while working on this paper. The first author thanks his best friend Matthias Lachmann for most memorable The Bard's Tale playing sessions and Reiner Opföler for introducing him to the Ultima series.

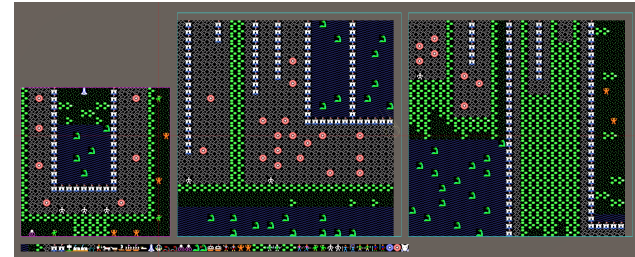
REFERENCES

1. Electronic Arts. 1985. *Tales of the Unknown, Vol.I: The Bard's Tale*. Game [Apple II]. (1985). Electronic Arts, Redwood City, California, U.S. Last played April 2019.
2. Nicholas Davis. 2013. Human-Computer Co-Creativity: Blending Human and Computational Creativity. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.
3. Werner Gaisbauer and Helmut Hlavacs. 2017. Procedural Attack! Procedural Generation for Populated Virtual Cities: A Survey. *International Journal of Serious Games* 4, 2 (June 2017), 19–29. DOI:<http://dx.doi.org/10.17083/ijsg.v4i2.161>
4. Richard Garriott. 1980. *Ultima I: The First Age of Darkness*. Game [Apple II]. (2 September 1980). California Pacific Computer Company, Davis, CA, United States. Last played April 2019.
5. Stefan Greuter, Jeremy Parker, Nigel Stewart, and Geoff Leach. 2003. Real-time Procedural Generation of 'Pseudo Infinite' Cities. In *Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*. ACM, New York, NY, USA, 87–ff. DOI : <http://dx.doi.org/10.1145/604471.604490>
6. Maxim Gumin. 2016. WaveFunctionCollapse. (2016). <https://github.com/mxgmn/WaveFunctionCollapse>
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*. ACM, New York, NY, USA, 68:1–68:10. DOI : <http://dx.doi.org/10.1145/3102071.3110566>
8. Antonios Liapis and Julian Togelius. 2013. Sentient Sketchbook: Computer-Aided Game Level Authoring. In *Proceedings of the 8th Conference on the Foundations of Digital Games*. 213–220.
9. Ahmed Mustafa, Xiao Wei Zhang, Daniel G Aliaga, Martin Bruwier, Gen Nishida, Benjamin Dewals, Sébastien Erpicum, Pierre Archambeau, Michel Pirotton, and Jacques Teller. 2018. Procedural generation of flood-sensitive urban layouts. *Environment and Planning B: Urban Analytics and City Science* 9, 4 (Nov. 2018). DOI : <http://dx.doi.org/10.1177/2399808318812458>
10. Joseph Parker. 2016. Unity Wave Function Collapse. Video. (29 October 2016). Retrieved June 24, 2019 from <https://www.youtube.com/watch?v=CTJJrC3BAGM>.
11. Joseph Parker. 2017. Generating Procedural Game Worlds with Wave Function Collapse. (2017). <https://gist.github.com/selfsame/e7ff11205c316888977f9cac04fe4035>
12. Joseph Parker. 2018. unity-wave-function-collapse. <https://github.com/selfsame/unity-wave-function-collapse/>. (2018).
13. Christoph Salge, Michael Cerny Green, Rodrigo Canaan, and Julian Togelius. 2018. Generative Design in Minecraft (GDMC): Settlement Generation Competition. In *Proceedings of the 13th International Conference on the Foundations of Digital Games (FDG '18)*. ACM, New York, NY, USA, Article 49, 10 pages. DOI:<http://dx.doi.org/10.1145/3235765.3235814>
14. Noor Shaker, Julian Togelius, and Mark J Nelson. 2015. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer.

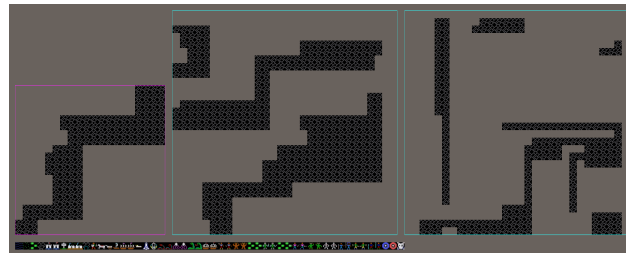
15. Gillian Smith and Jim Whitehead. 2010. Analyzing the Expressive Range of a Level Generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games (PCGames '10)*. ACM, New York, NY, USA, Article 4, 7 pages. DOI: <http://dx.doi.org/10.1145/1814256.1814260>
16. Wikipedia contributors. 2019. List of video game genres — Wikipedia, The Free Encyclopedia. (2019). https://en.wikipedia.org/w/index.php?title=List_of_video_game_genres&oldid=901735322
17. Georgios N. Yannakakis, Antonios Liapis, and Constantine Alexopoulos. 2014. Mixed-initiative co-creativity. In *FDG*.



(a) The selected game genre for this study session was “Role-playing/Action RPG.” Level idea: In this game, the player has the role of a medieval time traveler trying to get back home in a town invaded by monsters. The town is being in chaos, and the player has to figure out what is going on and needs to create his or her path using exploration, crafting, and also fighting monsters. There is an old war going on between creatures and humans. In the first level, the player character has no skills and has to fight with stick and stones and what she or he finds on the ground. The player finds gear which is not random, e.g., she or he finds sharp bones when defeating the octopus but can use exploration and crafting to get better items.



(b) The selected game genre for this study session was “Action-adventure/Survival horror.” Level idea: In this game, the player’s ultimate goal is to reach the rocket and to get it she or he has to go through obstacles (e.g., there are some hidden enemies placed on the grass). The player can either defeat or evade the enemies, and the level difficulty increases the closer she or he comes to the rocket. There are wandering warlocks that might surprise the player. The castle acts as a protection for the rocket, and when the player crosses it, she or he sees the water and some Ness monsters. The final surprise for the player are some monsters hidden in the woods before the rocket (the player might think she or he has already defeated the most difficult enemies, but that is not the case).



(c) The selected game genre for this study session was “Sports/Racing.” Level idea: The game Need for Speed inspired this game idea. The game works in a first-person view, and the flow is coming towards the player. It contains action on top like the game Grand Theft Auto (e.g., race and fight). The game can boost the player’s experience with rock music and nice sound effects.



(d) The selected game genre for this study session was “Role-playing/Fantasy.” Level idea: This game is based on a basic kingdom design, and the kingdom is surrounded by water. Each port (left and right) has different species. As the player travels upwards the protection becomes higher, and finally, the dog tile represents the kingdom.

Figure 7: The pilot user study had the purpose to show how human users would approach designing video game levels using our “City Construction Toolkit” for different game genres. On the left-hand side of each image, the input grid is visible (inside the magenta rectangle), and two different WFC outputs are displayed next to it (inside the cyan rectangles). On the bottom below the rectangles, the input tiles are visible. The “monster” tiles and other tiles used in each genre are comprised of the same tileset and are fully described in http://www.shikadi.net/moddingwiki/Ultima_I_Tile_Graphic_Format (see also Figure 6).