COMP3180 – Final Project Report

Name: Edgar Murga Garcia De Leon
Student ID: 47094133
GitHub Username: EJ-Corp

Table of Contents

1 Project Deliverables

My aim for this project was to learn and comprehend different procedural generation techniques for the purpose of creating a procedural level generation algorithm of my own. For this purpose, I constructed the following deliverables:

- **(1) A Unity Prototype** showcasing **Wave Function Collapse** – Technique 1
  https://github.com/COMP3180-23s2/comp3180-final-project-EJ-Corp/tree/main/UnityProjects/DungeonGenerator

  To Access Prototype:
  - *Open Unity Project*
  - *Open "2DWFC" scene*
  - *Press Play*
  - *While in play mode press space to generate a new level*

  This prototype was used to learn how **WFC** works and the special characteristics that set it apart. The prototype was inspired by existing showcases (Gumin 2023) and implemented using Unity 2D.

- **(2)** A **Unity Prototype** showcasing **Cellular Automata** – Technique 2
  https://github.com/COMP3180-23s2/comp3180-final-project-EJ-Corp/tree/main/UnityProjects/DungeonGenerator

  To Access Prototype:
  - *Open Unity Project*
  - *Open "CellularAutomaton" scene*
  - *Press Play*
  - *While in play mode press space to iterate through the generation*

  This prototype was used to learn how Cellular Automatons are created and how they can be used for procedural level generation. The prototype was inspired by a talk on known generation techniques (Wolverson 2020) and the implementation was derived from a YouTube video which outlined the workings using pseudo code and infographics (Kowalski 2020).

- **(3) Design plan** for a **Hybrid Approach** to procedural Level Generation - (Deliverable Substitution).
  https://github.com/COMP3180-23s2/comp3180-final-project-EJ-Corp/blob/main/Documentation/HybridPlan.pdf

  A document outlining the initial plan of how the hybrid algorithm would work utilising illustrations and pseudo code.

  This deliverable replaced the initial **Context Free Grammar** that was planned. This got replaced as I felt this would be a better way to plan the level generation.

- **(4)** A **Unity Project** demonstrating my **Hybrid Procedural Level Generation Algorithm** – Own Technique

https://github.com/COMP3180-23s2/comp3180-final-project-EJ-Corp/tree/main/UnityProjects/DungeonGenerator

To Access Prototype:
- *Open Unity Project*
- *Open "PersonalHybrid" scene*
- *Press Play*
- *While in play mode press space to generate a new level*

This project focuses on utilising the learnings form the previous 2 prototypes and combine their functionalities to create an algorithm which generates a level fit for a roguelike videogame.

- **(5)** A **Demonstration Video** showing the design and development process of the final algorithm.
  https://echo360.net.au/media/4f333e9b-f657-4a1e-a934-414398dce2bc/public

2 Milestones

2.1 Up to Week 7

Leading up to the submission of the research report, I began development on the first prototype focusing on Wave Function Collapse. I attempted to utilise my research to develop the algorithm, but quickly struggled to do it by myself and thus utilised a YouTube tutorial to guide me (Calice 2018). At this point, **Deliverable 1 - WFC** was done.

2.2 Mid-session break

During the break, I got excited and began development on my own generation algorithms. I began messing around with grid generation and room instantiation based on the grid. However, my lack of experience on the matter was a limiting factor. I was unable to stitch the rooms together. Nevertheless, this marked the beginning of **Deliverable 4**.

2.3 Week 8

After the meeting this week, and reviewing the feedback on my research report, it became clear that creating a context free grammar as a plan for the room generation wasn't the right way to go. Therefore **Deliverable 3** got changed form a grammar to a design plan.

2.4 Week 9

At this point, I got fixated on tyring to get my own technique to work and spent too much time without achieving anything. While unfortunate, I was able to self-reflect and notice that I was getting ahead of myself and thus decided to stop development and switch over to the development of **Deliverable 2 - CA**.

2.5 Week 10

I developed **Deliverable 2**. While developing the prototype translating the code form LUA to C# caused a major roadblock. Specially, when it came to recognising

neighbouring cells. After a lot of time, I was able to come up with a solution, however the algorithm has a long compile time.

2.6 Week 11

I recruited 10 developers to test both systems and collated the data for analysis. The data then got analysed and the results were used to guide the development needs for the hybrid algorithm.

2.7 Week 12

I began development of **Deliverable 4** based of the plan (Deliverable 3) and the information gathered from the evaluation. However, various bugs were stopping the algorithm from producing the desired outputs, thus creating the biggest roadblocks I encountered.

2.8 Week 13

This week I fixed all the bugs. These were incredibly painful to fix due to Unity's poor debugging support and the highly recursive nature of the algorithm.

Additionally, I wrote up the final report and recorded **Deliverable 5**.

3 Evaluation

Evaluation was a big part of the project, as I utilised the evaluation of both the prototype deliverables to guide and influence the design and development of my hybrid algorithm. By getting other developers to test both the prototypes I was able to determine whether the prototypes had been successful in fulfilling their purpose, as well as obtain data which heavily influenced the final project. Here are the results of the evaluation:

Prototype 1 – Wave Function Collapse

For this prototype, I mostly wanted to identify any issues with the algorithm, as well as the variety which it could produce. To do this, I ran some testing with 10 developers to gather data. From this data I was able to determine the prototype successful as all tests resulted in levels being generated. However, 2 major issues were found. Firstly, I often had some doors leading into the void, and secondly, I had some doors leading straight into a wall. Both the issues were a major problem as it meant there could be unreachable areas in a level. Therefore, I utilised this information to guide the design process of the final project in a way that these scenarios were never a possibility.

Prototype 2 – Cellular Automata

For this prototype, I wanted to test grid behaviours and environment simulations. To test this, I ran some tests with the same 10 people form before and gathered data on functionality and generation results. From the analysed data I was able to determine that while the algorithm is useful, the generation output often simulated a "cave-like" environment, which unfortunately doesn't align with the goal for the final project. Additionally, though the testing results I recognised that the initial algorithm steps is very slow and unoptimized, and therefore utilised this within the design considerations for the final project. Overall, the prototype was successful as it did everything it was needed to do; however, it did not seem as useful towards the final project.

Design Plan for Hybrid Approach

The creation of the design plan was mostly derived form the evaluation results of the two prototypes. The issues encountered within the WFC prototype heavily influenced the path generation, as well as the single task design. As for the CA prototype the performance was a big issue, and thus influenced the neighbouring room checking process within the design plan – cutting the process form 8 checks to 4 checks, reducing the time by half.

As far as evaluating the plan, I decided to simply evaluate it by myself by running through the plan in diagram form. This decision was mostly made because of my lack of time. However, I believe this form of evaluation was enough to deem the deliverable as successful. Success was measured as the plan worked both on paper after multiple iterations, as well as having the algorithm and working as planned.

Hybrid Procedural Level Generation Algorithm

Due to the time constraint, I was unable to conduct a second testing round with the original testers. If time allowed, I would've gotten the same group of testers to test the new system and evaluate its performance and outputs and compare them to the feedback they first provided.

Additionally, while the evaluation for the Cellular Automata prototype made it seem like it wasn't useful towards this algorithm, it turned out to differ. During the early stages of development, I realized that by placing the rooms within a grid would prove incredibly useful. Therefore, the hybrid algorithm ended up utilising a similar grid base, which massively optimised the neighbour checking process.

Finally, while there was no external testing on the final algorithm, the algorithm itself was a product of both the prototypes and their evaluation results. The algorithm utilizes the best characteristics of each algorithm and then accounts for their weaknesses discovered through their evaluation. Additionally, after thorough QA the output seems consistent with the system requirements, and thus I'd consider it a success.

4 Learning Goals Reflection

Demonstrate knowledge of procedural level generation.

I completed this goal, successfully creating multiple prototypes based on existing procedural generation techniques. The main thing I learnt in this area is the way procedural generation isn't exactly random, but instead uses randomness to create different outputs based on strict rules and guidelines. By looking at multiple techniques, I've been able to recognise various similarities between algorithms, which have led to me learning that randomness is a great way to deliver curated variety.

Demonstrate understanding of various known procedural generation techniques.

This goal was achieved both by the implementation of both the WFC prototype and the CA prototype, and the design of my hybrid approach to procedural level generation. While utilising the WFC tutorial was helpful, I believe that I learnt the most through the rough

guideline I followed while create the CA algorithm. As a result of having to translate the pseudo code and LUA snippets into a functioning prototype within unity, I was able to truly understand the steps and precautions needed to generate levels. Therefore, I believe this was one of the most beneficial goals as it allowed me to discover a valuable learning technique for future projects.

Apply procedural generation techniques to guide new projects.

Through the creation of my own algorithm, this goal was achieved. The journey from learning some techniques form the internet to going ahead and designing/developing my own technique was full of learning opportunities. However, the most significant outcome was being able to guide my own project as I saw fit with full accountability for my work. By utilising the learnings from the early prototypes, and then utilising them as building blocks for my own project I was able to learn new learning techniques, and now I recognise the value in utilising the work of others as a learning resource.

Utilise existing knowledge along with new findings to understand more complex algorithm creation techniques.

While this outcome was partly achieved through the creation of my level generation algorithm, I don't believe the outcome was fulfilled to its maximum potential. I would've liked to build upon the generation algorithm and create a way to dynamically populate the generated level based off different player conditions. However, due to the time restricting and my over-ambitious plans, this didn't come to be. Despite the situation, I believe I now have both the knowledge and confidence to make this algorithm a reality.

5 Industry Relevance and Future Work

When it comes to industry relevance, I believe the outcomes of this project are of great relevance. Throughout this project, I conducted valuable research, which guided my preparation, and then utilised my findings to create something of my own. This was the first time in which I have individually designed and developed a system without utilising outside information or strict instructions. By doing so, I can demonstrate my ability and drive to grow and independently learn new techniques. Both of which are incredibly valuable skills to have when working within the industry.

During the R&D process I was able to learn a lot through a variety of theoretical sources. However, when it came to applying the theory into the prototypes, I found that I got too fixated into making them perfect. This caused a few issues as it made me lose sight of the end goal, and I started spending unnecessary time on the prototypes. Since the prototype's purpose was only to help me understand the algorithms, there was no need to perfect them or optimise them. However, at times I realised that this was exactly what I was doing, I already understood the algorithm, but I was just trying to solve simple bugs that wouldn't even be relevant to the ultimate purpose of the project. Eventually, I was able to recognise this and moved on with development instead of sinking time trying to fix bugs that weren't significant. Ultimately, this taught me the valuable lesson of recognising importance of tasks and not loosing focus of the end goal.

As for the project itself, the algorithms created are valuable both as a portfolio piece which showcases my knowledge in procedural generation, as well as a great foundation for a game

I've been designing and planning to develop. Ultimately, the project served both as a stepping stone for a bigger picture game, as well to teach myself a different way to independently learn new techniques. In the end both outcomes have relevance in the industry, self-learning is a highly valued skilled within the industry, and the game can be a great introduction into the industry itself if published.

6 Bibliography

Calice, N. (2018). Random dungeon generator - easy unity tutorial - #1. Retrieved from https://www.youtube.com/watch?v=qAf9axsyijY&t=3s

Gumin, M. (2023). Retrieved from https://github.com/mxgmn/WaveFunctionCollapse Bitmap & tilemap generation using Wave Function Collapse

Kowalski, K. (2020). Retrieved from https://www.youtube.com/watch?v=slTEz6555Ts Cellular automata | procedural generation | game development tutorial

Wolverson, H. (2020). Retrieved from https://www.youtube.com/watch?v=TlLIOgWYVpI Herbert Wolverson - Procedural map generation techniques