COMP3180 – Final Project Report

Name: Edgar Murga Garcia De Leon

Student ID: 47094133

GitHub Username: EJ-Corp

The below suggested section word counts are rough estimates provided as a guide (totalling 2000 words); you should balance your report as appropriate (for the specifics of your project and to hit the overall required word count range of 1500-2500 words) and in consideration of the assessment criteria. You should integrate images (e.g. of your deliverables or playtesting results graphs) throughout your report to help illustrate your work. **Note: Remove this and all other instructions from the document prior to submission.** You should also right-click and ‘Update Field’ on the Table of Contents before submission to update the references to page numbers.

Table of Contents

[1 Project Deliverables 2](#_Toc143000163)

[2 Milestones 2](#_Toc143000164)

[2.1 Up to Week 7 2](#_Toc143000165)

[2.2 Mid-session break 2](#_Toc143000166)

[2.3 Week 8 2](#_Toc143000167)

[2.4 Week 9 2](#_Toc143000168)

[2.5 Week 10 2](#_Toc143000169)

[2.6 Week 11 2](#_Toc143000170)

[2.7 Week 12 2](#_Toc143000171)

[2.8 Week 13 2](#_Toc143000172)

[3 Evaluation 2](#_Toc143000173)

[4 Learning Goals Reflection 2](#_Toc143000174)

[5 Industry Relevance and Future Work 2](#_Toc143000175)

[6 Appendices 3](#_Toc143000176)

[7 Bibliography 3](#_Toc143000177)

# Project Deliverables

A (~250 word) summary of the Project Deliverables, with a description and link for each one. These links may be to your repo, or to content within your report/appendix, depending on the deliverable. Use the outline of deliverables you submitted in your Research Report as a starting point and note variations from your plan where present. Remember to identify and link to your industry portfolio piece (including the short demonstration video showcasing relevant prototype/s), since it is one of the required deliverables for the project.

* Unity prototype of proc gen technique 1 – Wave Function Collapse
* Unity prototype of proc gen technique 2 – Cellular Automaton
* ~~Context Free Grammar~~ – Hybrid approach plan / Pseudo Code (clarify that got replaced)
* Own proc gen hybrid approach implementation (Unity Project)
* Video explaining how the algorithm works.

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

*Put Link to It here – Instructions maybe.*

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

*Put Link to It here – Instructions maybe.*

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).

*Put Link to It here – Instructions maybe.*

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

*Put Link to It here – Instructions maybe.*

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.

*Put Link to It here – Instructions maybe.*

# Milestones

A (~350 word) list of weekly Milestones, starting with a single brief summary for work completed up to submission of your research report in week 7. Following that, you should briefly document milestone achievements and roadblocks for each week from weeks 8-13, and identify points at which deliverables were added, dropped or modified compared to your original plan. Your development journal from your repo will be a good reference/record to start from, but make sure you edit your weekly summaries to make them concise and about specific achievements/ roadblocks.

## Up to Week 7

Initial prototype done – tech 1

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.

## Mid-session break

Beginning of my own algorithm – grid generation and room instantiation based off grid.

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**.

## Week 8

Deliverable change due to reflection of feedback received for project plan.

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.

## Week 9

Ran into problems with room connection on own technique.

Got ahead of myself so switched to developing second prototype.

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**.

## Week 10

Prototype 2 got developed – ROADBLOCK was figuring put how to recognise neighbour cells. Figured it out by the end of the week but algorithm is slow.

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.

## Week 11

Evaluated both prototypes to get feedback which would guide the final project.

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.

## Week 12

Data analysis and hybrid technique got developed. Major roadblocks with bugs that were hard to debug due to the poor nature of debugging in unity and the recursive nature of the algorithm.

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.

## Week 13

Fixed all the issues with the algorithm, finished report and video.

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**.

# Evaluation

An (~700 word) Evaluation of utility, usability or UX (as appropriate) of the deliverables you made by their target audience (e.g. programmers, specific players, artists, designers, etc.). This section is not about you reflecting on how well you think you did, but rather reporting on a conducted evaluation/study of whether your deliverable/s achieves its goals with its intended audience (e.g. via playtesting or user testing). You should conduct this study on AT LEAST your major deliverable, summarising the results of this evaluation here and referring to the raw results from your study in an appendix or your repo. Make sure you justify the target audience of the deliverable by identifying the purpose of your deliverable in an industry context.

# Learning Goals Reflection

A (~350 word) Reflection on the advanced game development skills you learned with reference to the learning goals set in your research report. Again, start with the list of learning goals from your report, revised as necessary, and for each one you should reflect on the extent to which you have built new relevant skills, extended your capabilities or understanding, or conversely struggled to make tangible progress (and why).

**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.

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

# Industry Relevance and Future Work

A (~350 word) Future Work section, reflecting on how the skills you've learned might be applied in industry. You can also outline or hypothesise about specific plans to do more with your project/deliverables. You should include explicit reflection on your experience with the process of self-directed learning through research and experimentation, including goals for how you might improve your learning skills in future.

# Appendices

You can include any supporting data, tables or screenshots here that would otherwise break the flow of the document or blow out your word count. Material included in this section should be referred to in the text and contribute meaningfully to the document.

# Bibliography

References for any third party sources referred to in the text or relied on in your project.

\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