**Honours Project Feasibility Demo**

**Callum Myers**

**2102549**

**Research Question**

An investigation into using Procedural Level Generation to improve replayability and player engagement in 2D platformers.

**Aim**To investigate and analyze the potential benefits of procedural generation when creating levels for 2D platformers and consider how this could be used to improve the game’s replayability, then to use this information to develop and implement a level generator for a simple 2D platformer.

**Objectives**To research how procedural level generation could be used in future 2D platformers to give the player a unique experience each time they start a new game.

To explore the links between new, unique levels and player engagement levels.

To utilise programming techniques such as Perlin Noise and Markov Chains in order to implement a procedural level generator using the Unity game engine.

**Statement of Changes**

I no longer plan to use Wave Funtion Collapse in my implementation since it is made mostly obsolete by my use of Markov Chains.

**Risk Analysis**

**Time**

The biggest risk to my project is time. Since the implementation is a new project for me, there are some aspects which I need to learn whilst implementing the main level generator. For example, I have not used tilesets and tilemaps in Unity before however I feel these would be vital to prevent performance issues. To mitigate the risk of running out of time, I will be using my Gantt Chart to ensure I am meeting expected deadlines, as well as reviewing how the progress is going at the end of each week and making changes to the timeline and deadlines if needed. I have also set a priority for each task within the implementation which needs completed and have set some goals to be stretch goals which I will only attempt if I find myself ahead of schedule. The order of priority for primary goals is:

* The generator uses Perlin Noise to generate a new level layout every time the level is restarted.
* Every level layout generated can be complete by the player.
* The generator always completes the first pass (using the Perlin Noise to set the height of the ground at every point in the level) and the fourth pass (the level is tested to ensure it can be complete by the player, and if it is not then it will decide whether to rework certain sections of the level or discard the entire level and create a new one).
* The generator completes the second pass (which will be spawning a variety of enemies at random points through the level) and the third pass (which will be spawning a variety of collectibles throughout the level).

The following are stretch goals which I hope to complete however these could be skipped without any major impact to the base project if the primary goals cannot be completed in time for any reason. The priority for these is:

* The sprites for each section will be varied and show biomes or background variation.
* The player can use the menu to set parameters for the next level which will be generated, as well as being able to generate this new level at any point.
* The player will be able to zoom in and out to see more or less of the level at the same time.