**Honours Project Feasibility Demo**

**Callum Myers**

**2102549**

**Research Question**

How can procedural level generation be used to lengthen players’ playtime and improve player engagement in 2D platformer games?

**Aim**To investigate and analyse the benefits of procedural generation algorithms when creating levels for 2D platformers, then consider how this could be used to lengthen the time a player can stay engaged with the game.

**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 evaluate the level and length of player engagement in a classic linear platformer when compared to one featuring procedural level generation techniques.

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

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

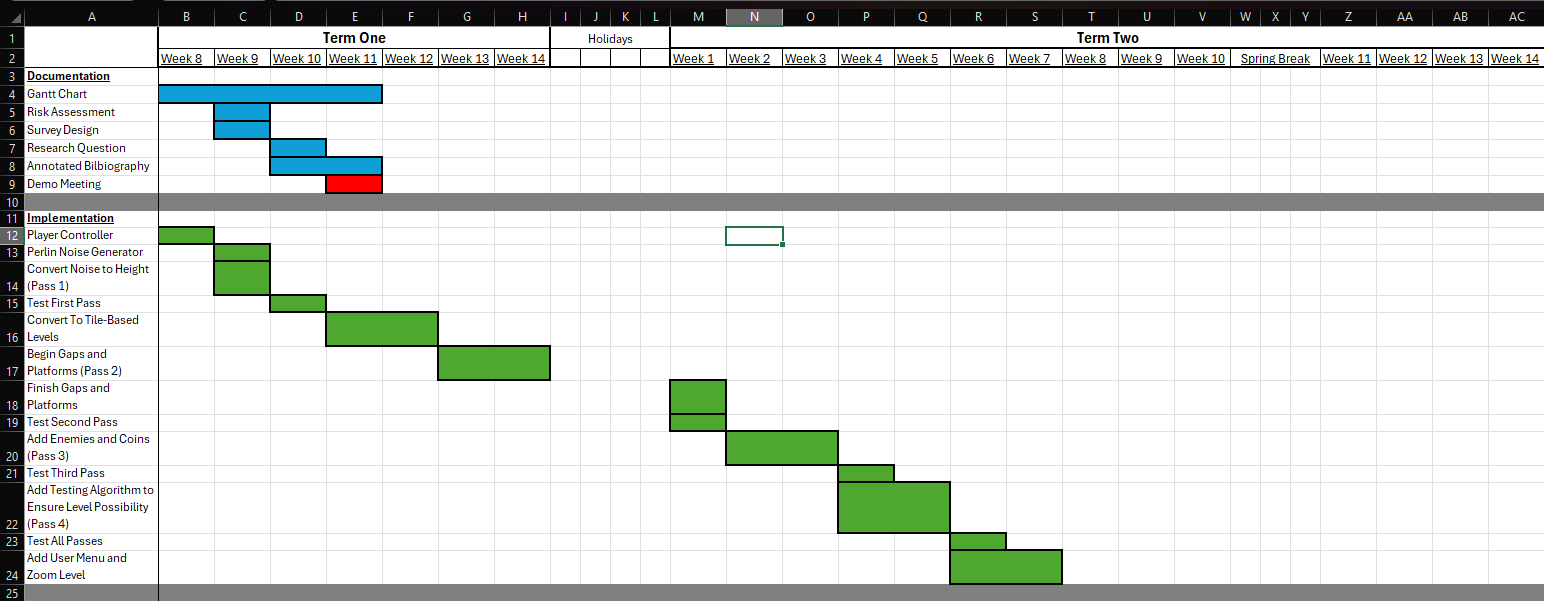
**Statement of Changes**

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

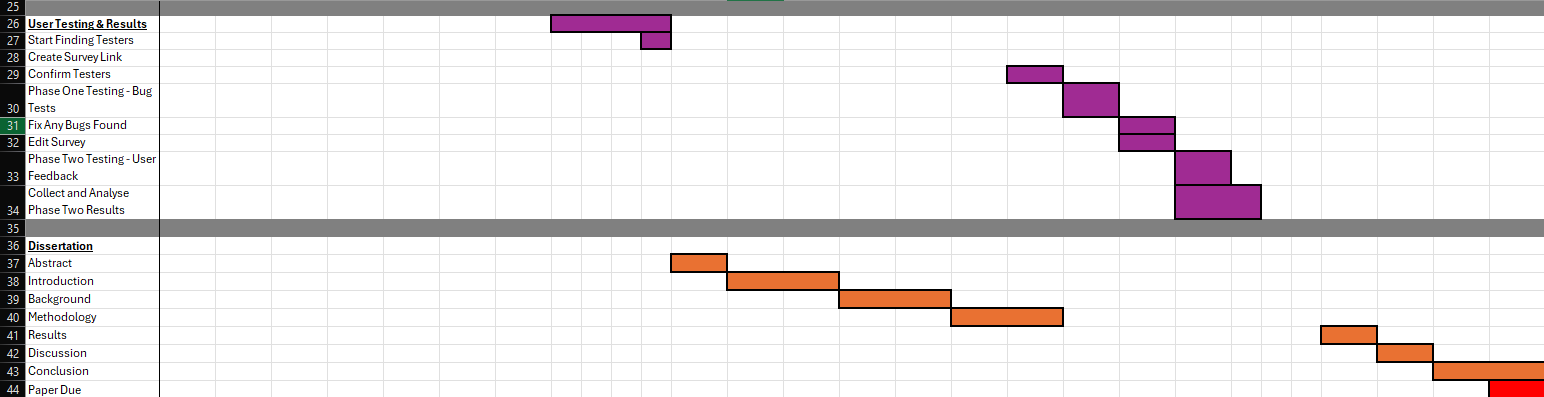
**Gantt Chart**

I have included a file for the Gantt Chart in the .zip folder.

I will be reviewing my Gantt Chart approximately every 4 weeks and making changes as needed based on the rate of development.







**Risk Analysis**

**Time and Progress**

Likelihood: High.

The biggest concern in my project is the time limit I have, and the possibility of falling behind on development due to external circumstances, such as other module assignments or personal issues. To avoid this, I will be looking at my Gantt Chart consistently to ensure I know what I need to do for the week, and then reviewing this chart once per month to see if the deadlines are still feasible.

**Scope**

Likelihood: High.

In any development project there is always a risk of going out of scope, or feature creep, resulting in trying to add in too many extra features, causing the primary focus of the project to be unfinished or not polished enough. To avoid this, I wrote out a list of vital features for my project before starting, and anything else I think to add during the project will be added to my “stretch goals” list, which I will only try to add once I am happy with the base level generator.

**Testing**

Likelihood: Moderate.

Since my testing will take the form of a user survey, it could cause an issue if I cannot get a suitable number of candidates who are willing to give me feedback. If this issue does arise, I may have to cut some time from my stretch goals to do extensive testing myself. I also plan to mitigate this risk by finding people who are willing to participate earlier on in my project, with the intention that if it does become an issue, and I need to spend more time testing by myself, I will more easily be able to plan for this by working it into my deadline schedule.

**Sickness**

Likelihood: High.

There is a chance that I may fall ill, especially over winter, however this is not too much of a concern since my work is largely done from home therefore, I should be okay to continue working as expected. However, to mitigate this risk, I will be regularly reviewing my Gantt Chart and if my sickness causes me to fall behind, I can edit my timeframe as required.

**Burnout**

Likelihood: Moderate.

Another risk which could affect my project is burnout, i.e. losing motivation to work on the project if I give myself too much to do over a short period of time. To mitigate this risk, I will try not to do much work over the Christmas holidays and spring break, as shown in my Gantt Chart. This will give me a break from the project whilst also ensuring that I keep on top of everything and still meet my estimated deadlines.

**Machine Damage**

Likelihood: Low.

Since I will be working primarily from home on my own laptop for this project, one risk I may encounter would be my laptop breaking or corrupting. I have mitigated this risk by ensuring my files are constantly backed up to a GitHub repository. This includes both documentation and implementation work, and if for any reason I do not have access to my computer for a period, I can still access Unity using the computers on campus, and any documentation I need to edit could be done at my local library’s computers.