**An investigation into the use of procedural generation to create levels for a replayable 2D platformer**

Callum Myers

School of Design and Informatics

Abertay University

DUNDEE, DD1 1HG, UK

**ABSTRACT**

**Context:** One of the biggest factors contemplated by players when buying a new game is the value for money, and this is often dictated in part by the playtime offered. This can be difficult to fulfil by developers of 2D platformers, where there is often little reason for players to play through the game more than once since it is the same linear experience each time.

**Aim:** The aim of this project is to research the benefits of using procedural generation to improve replayability in game genres where it is often lacking, then develop an algorithm to generate 2D platformer levels with these issues in mind.

**Method:** Using C# scripts within the Unity engine, an application will be created containing a simple player controller, as well as a level manager which will procedurally generate a new, unique level each time the player completes the current one. It will also contain a menu to edit parameters related to the level generation, allowing the user to see how changing these could allow for a potentially infinitely replayable game.

**Results:** Each level created by the level generation tool should feel different and unique, and the player should have a completable path to the end goal.

**Conclusion:** This project will investigate and demonstrate the use of procedural generation to improve replayability in traditionally linear games, specifically 2D platformers.

**Keywords**

Procedural Generation; Replayability; Wave Function Collapse; Markov Chains; Level Design; 2D Platformer.

**1. INTRODUCTION**

In recent years, many game studios have started to put a focus on creating a longer gameplay experience for players. This puts pressure on development teams, particularly level designers, artists, and programmers, who must create a huge array of unique levels manually which is not only time-consuming but also resource-intensive, as each level needs to feel unique and challenging to maintain player interest over extended playtimes. This can significantly increase their workload, leading to longer development cycles and higher costs for game studios.

One solution to this problem is Procedural Content Generation (PCG), which refers to "the algorithmic creation of game content with limited or indirect user input" [PCG-G book]. By using PCG, games can create a wide range of unique levels automatically, reducing the manual effort needed from designers and allowing for more content to be created in a shorter timeframe. For instance, in a study by Balim Alpay, it was found that creating a level using PCG took only 30 minutes on average, compared to 60 minutes required for a manually designed level [Human vs PG Levels, page 45], which is a good example of how the use of PCG could help reduce this workload and allow teams to turn their focus to other aspects of the game, such as gameplay mechanics or more impressive graphics.

PCG also offers a major advantage in terms of replayability. In games designed with a set number of manually created levels, the player’s experience tends to remain the same over multiple playthroughs. However, procedural generation allows for an almost endless variety of combinations. In the same study, a game with six levels designed by a human would only give the player a unique gameplay experience for the first playthrough, while the PCG version had the potential for 720 different level permutations [Human vs PG Levels, pg 41]. With this huge variety of levels available, players will be able to play through the game many times and each run will feel different to the one before.

Despite its advantages, PCG does come with challenges, particularly regarding the balance between variety and the emotion which only a talented designer could add by hand to level. To address this, designers can implement PCG in two ways: “online” or “offline.” Online PCG creates levels whilst the player is playing through the game, such as generating level segments or “chunks” just before the player encounters them [PCG Replayability, page 4]. This method works well for games where replayability is a primary goal, but it may lack the specific design elements or narrative touches the designer could add. Offline PCG, on the other hand, generates levels before the game begins, either at the start of a new playthrough or even before release [PCG Replayability, page 4]. This approach allows for more human input, as designers can handcraft certain features or add custom elements like easter eggs or specific collectible. This paper will look to use an “online” approach, since the primary focus is to demonstrate how the use of PCG levels can be beneficial in extending the potential playtime of platformers, and dynamically created levels seems like the most appropriate way to tackle this goal.

**2. BACKGROUND**

In this section you should be giving the background to your project – what is the current state of the art or understanding. What problem are you going to address and who says that it’s a problem anyway. This section should make it clear that your project is an investigation at Honours level and follows on logically from other work that’s of relevance and importance to other workers in your field.

**2.1 Subsections**

Subsection headings are also 12 point Bold, Times New Roman but only the first letter of the title is capitalized

Remember that you should be citing references within the text in these sections using the Harvard style of referencing (Harvard (2011)) since this is the accepted method within the university.

You can find detailed guidance on how to reference all styles of material within the guide available on the Library Portal or from the Library itself in a booklet form.

**2.2 Tables**

**Table 1 – Proposed schedule for delivery**

|  |  |
| --- | --- |
| **Months** | **Goals** |
| October to December | Implement a basic level generator.  Implement a player controller. |
| December and January | Add an in-game menu to allow player to adjust parameters for level generation.  Begin to write dissertation. |
| January and February | Begin user-testing and receiving feedback.  Fix any bugs found during testing. |
| February and March | Evaluate feedback to see if the aim of the project was met.  Continue work on dissertation. |
| March and April | Complete dissertation. |

**2.3 Figures**

Figures can help describe something very effectively but be careful of just using screen shots since these can run to many MB without you noticing it.

Captions for figures should also be 9 point Bold, Times New Roman but are centred below the figure. Use the full word “Figure” in the caption



**Figure 1 – A view of the entrance on level 2 of Abertay University**

Remember that you should give due credit to figures that you use by giving a reference after the caption – the above photo is one I took and so I have given myself permission to use it

The next section is the Method section and the two-column format continues.

(Approx ¾ page)

**3. METHOD**

In this section you will describe what practical work you actually intend to carry out. You should mention what choices exist and explain why you have chosen a particular method.

Of course at this point you have not actually done anything and so you are speculating to some extent but by looking at what other people have done (and referencing that work) you should be able to identify what is possible given your own limitations in terms of time and skills.

(Approx ¾ page)

**4. Summary**

You should finish your proposal with a summary of what you see your project as contributing to the subject area. Why is it worth doing and who might benefit from your results?

(Approx ¼ page)

**5. REFERENCES**

All of the references that you have cited in the text must now be listed here in alphabetical order using the Harvard Cite Them Right style of referencing. Again we remind you that guidance is available from

<https://intranet.abertay.ac.uk/library/referencing/>

which explains how to reference all the material you are likely to come across from journal papers to blogs or youtube. The reference you give must enable anyone to obtain the reference, not just people with an Abertay University IT account for example

The reference list should be 9 point Times New Roman but only left justified:

Other, A, N, date, “Title of the article”, *Name of the Journal,* Volume number, pages

See the guides for other examples of style

Note that we expect you to have no more than 8 references but you should have at least 3 good quality references meaning that they have been published in a peer-reviewed journal or other publication (a conference proceedings for example)

Finally, note that we have tried to make the two columns on this last page roughly the same length by inserting blank lines in the first column