

# The Effect of Procedural Generation on a Game's Replay Value

Author: Andrew Aston  
Supervisor: Llyr Ap Cenydd

## Introduction

When referring to a game's replay value, or **replayability**, we refer to the potential for a game to be played repetitively after the first playthrough. For many games, this can be due to hidden easter eggs, alternate endings, speed runs and achievement hunting etc. Procedural generation is a game mechanic that can fall under this list of affecting replay value within video games. This mechanic is the basis in which an algorithm can create data such as level layouts and terrain, or types of characters. These algorithms would work hand in hand with pre-made assets to generate a final product. What my project focuses on is how this aspect of game design can impact a game's replay value?

## Examples of Procedural Generation

One of the earliest and most well known examples of procedurally generated content in video games can be found with the dungeon crawler game **Rogue (1980)**. Using ASCII with a square grid structure, the game generates a dungeon level for the user to explore. Each run of the game would create the level in a unique layout. This game led to the genre known as "Roguelike", involving randomly generated dungeon crawlers.

As video games started to become more of a forefront of entertainment, more games began to use the concept of PCG. From more dungeon crawlers such as **Diablo (1997)**, to more lower-scaled random generation algorithm such as **Tetris (1984)**.

Today, games push the concept of PCG to new heights, creating whole terrain or entire worlds, such examples are most well known in games such as **Minecraft (2011)** and **No Man's Sky (2016)**.

## Tech Behind

The project has been created using the **Unity Engine**, version 2019.4.15f1, due to previous experience within other modules involving two other projects. This has allowed for the use of C# scripts in order to generate the dungeons as anticipated, alongside the spawning of varying decorations.

This has also allowed for the use of 2D sprite assets in which have been customised using the Image Manipulation Program known as **GIMP**, taking advantage of the ability to create 16x16px images.



## Evaluation and Conclusion

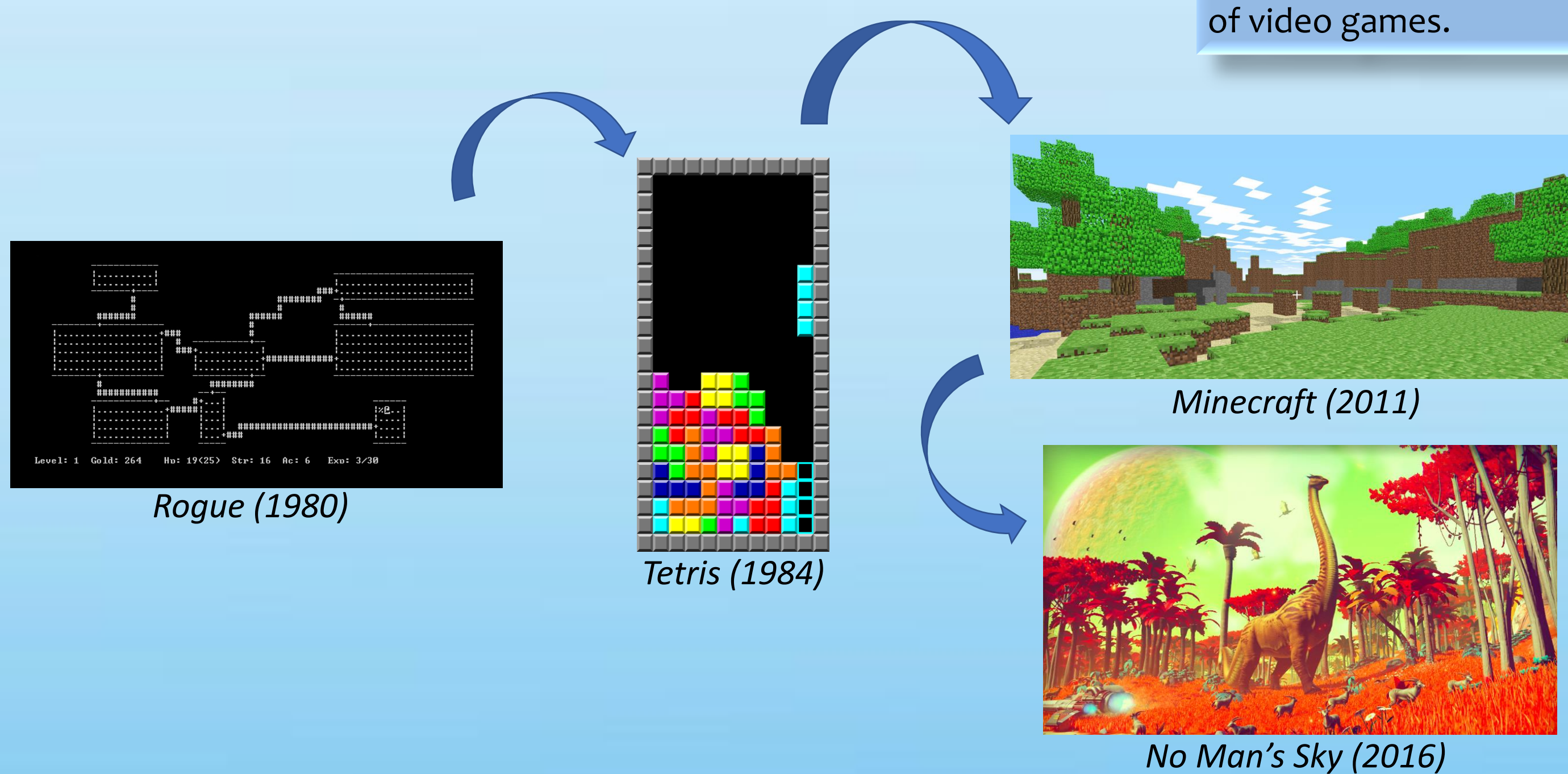
The prototype will be given to different subjects to play through multiple times. Once the players have completed their runs, they shall be given a questionnaire to fill regarding their experience. Such questions involved would include how their overall experience was, if they would play again as it were or with additional features, what they enjoyed the most, what was least enjoyed, would they have preferred if the levels were already mapped out etc.

The main focus on what is being asked is their opinion of the prototype's replay value. Using this data, we can analyse the replay value of the prototype and view the impact that PCG has as a whole on the dungeon crawler. From here, we can compare the research to other Procedurally Generated games and then apply back to the overall question of how Procedural Generation affects the replay value of video games.

## Future Work

The research on the replay impact of Procedural Generation could possibly pave the way to looking into how to improve replay value on certain games using PCG. Perhaps one could look into the limitations presented by PCG and how one could overcome them.

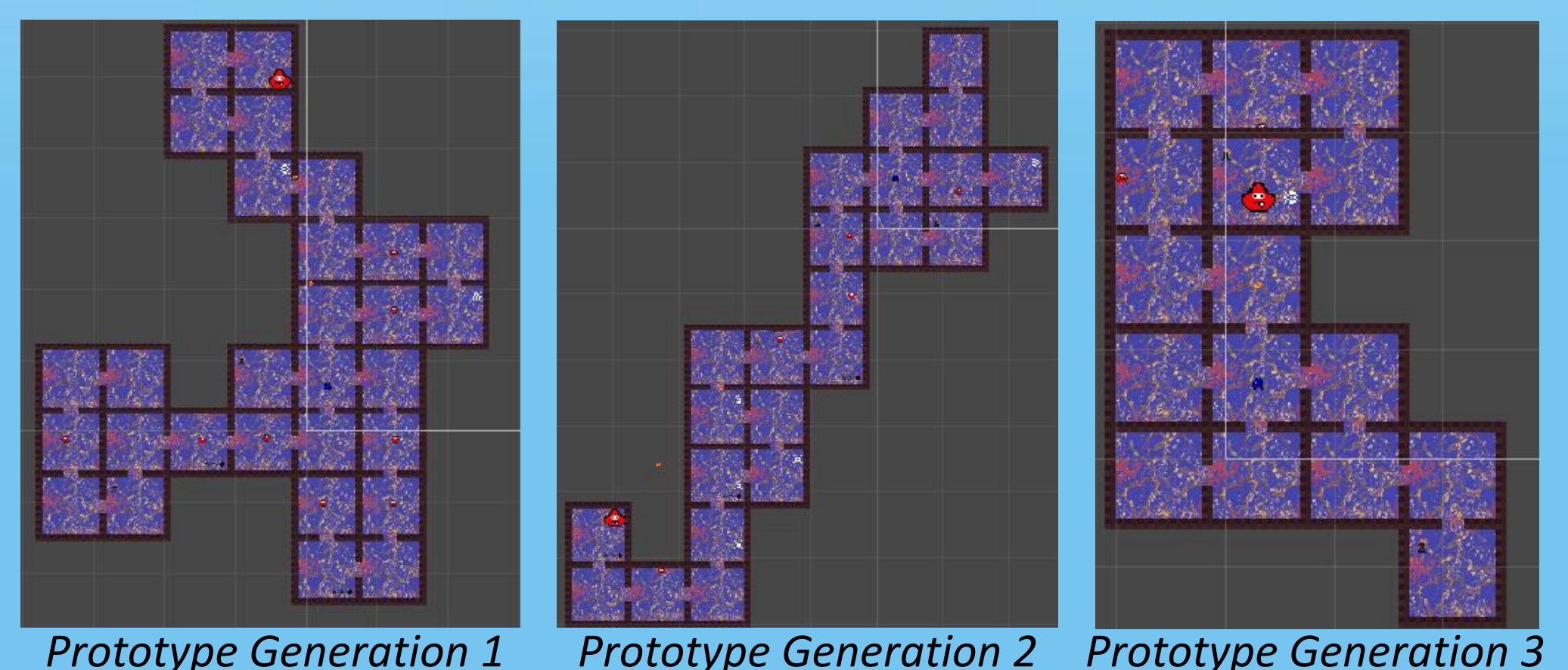
Maybe one could move from PCG and instead explore other areas that apply to a game's replay value, allowing for other developers to take note and focus on their future projects.



## Project Work

To determine what aspect of Procedural Generation actually impacts replay value, the project focused on creating a 2D dungeon crawler prototype. The main aspect of the prototype would be the generation of the dungeon being different each time, giving a new experience every time. Some iterations would be more challenging to play than others. The dungeon prototype objective is simply for the player to reach the boss room as he consistently shoots at you whilst you navigate the level; each hit sends the player to the spawn junction room.

The prototype is built off a simple algorithm tutorial series (**Blackthornprod: [tinyurl.com/4ahe48s4](https://tinyurl.com/4ahe48s4)**) that has been adapted to add extra features, with rooms containing more enemies that send the player to spawn on touch. Such rooms will also have chances of spawning random decorations from torches to spider webs, using PCG for additional atmosphere.



Each run of the prototype generates an entirely unique level. Some can be bigger than other and some can hold different decorations inside certain rooms