



Procedural Dungeon Generator

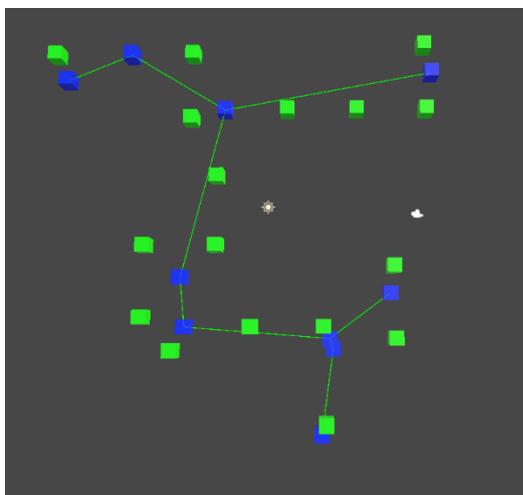
Overview

During development, we conceptualized and experimented with procedurally generated levels to assess their usefulness and viability. We found they offered several benefits, including increased replayability, enhanced player engagement, modularity, and the ability to implement a scoring system that encourages players to improve over time.

Research & Experimentation

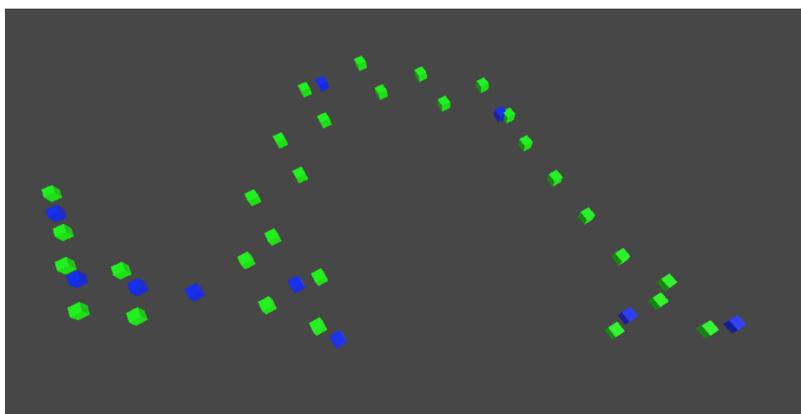
Initially, my first step was researching the different types of Procedural Generation and the accompanying methods for implementing it. Considering my aim was to create an interconnected dungeon using rooms and hallways that helped focus my search until I found two potential options: A-Star Pathfinding and a simple doorway connection system.

A* Pathfinding:

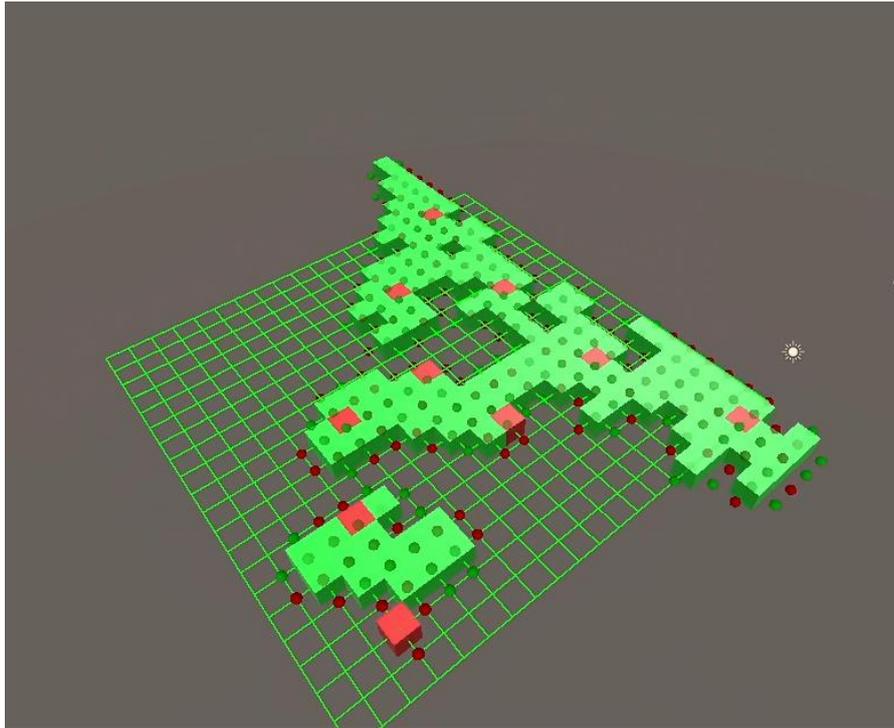


Through placing “Rooms” (Blue Squares) and attempting to connect them with “Hallways” (Green Squares), the A* Pathfinding method tries to find the shortest distance between two or more rooms and connect them up.

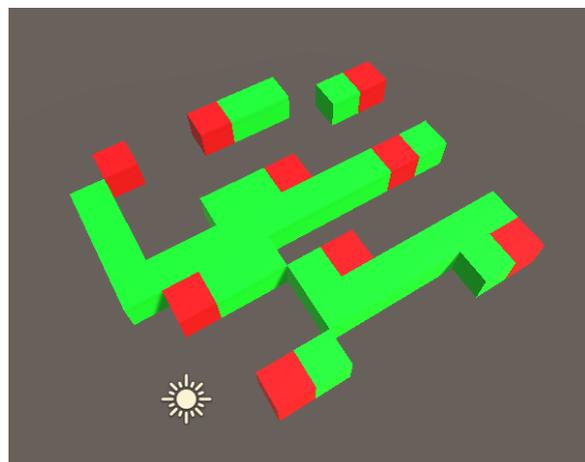
As shown, this method contained mixed results, this is largely due to the lack of a correct grid layout for the algorithm to use.



Following on from this, I researched a way to use a grid to improve upon this which presented me with a more promising result.



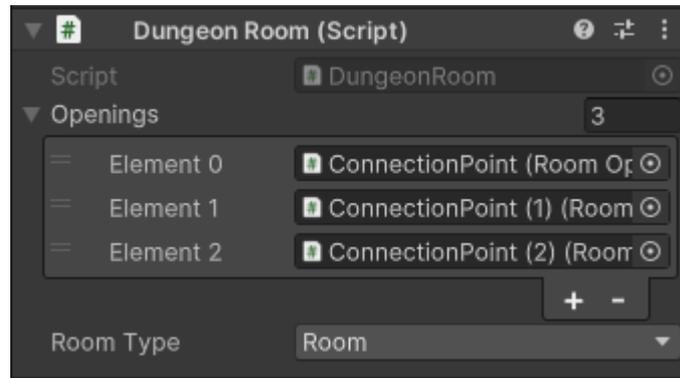
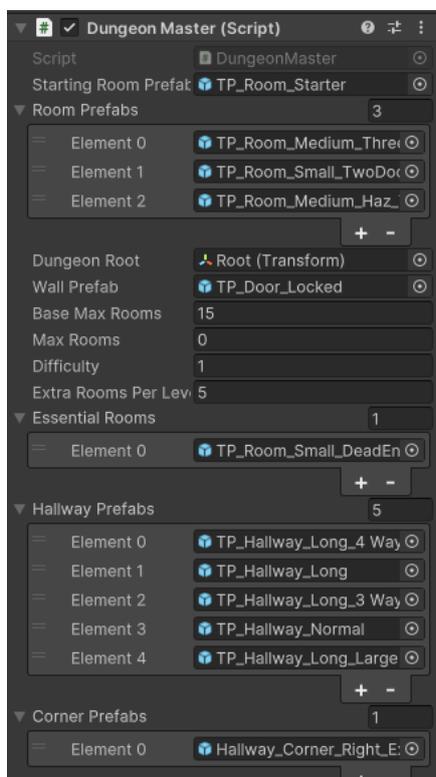
As shown in the image, the hallways are connected and some are correctly finding a way to the rooms (Red Squares) however there are still major flaws. The hallways are not correctly pathfinding and instead bunch up or even move off in random directions, alongside this not all rooms become connected and so it would cause hard locks or be way to unpredictable.



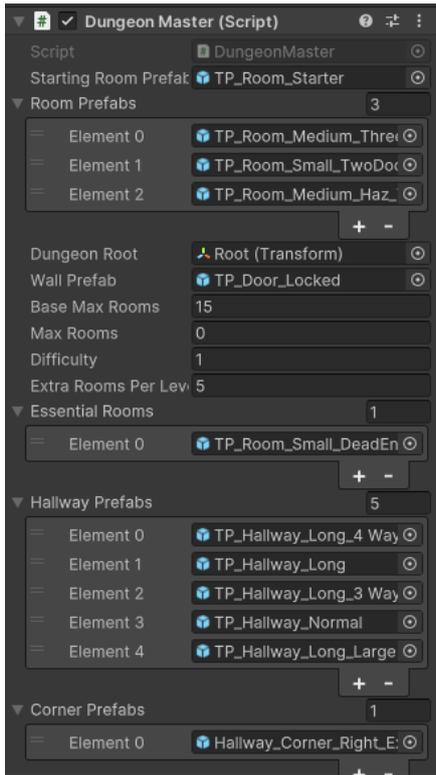
Overall, due to the complexity of this method, I determined it would not be worth pursuing this as I would not be able to make it customizable enough or to the standard for our game.

Doorway Connection Points:

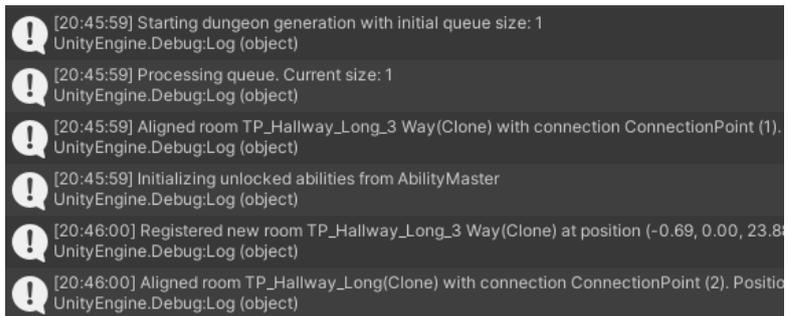
Developed by researching a method by [BrokenKnightGames](#), this new algorithm features a much less complex set of rules and steps which helped to ensure rooms would be connected, customizable and offer a both exploration and replayability experiences for the player. This system features three major components which were the Dungeon Master, Room and Connection Point.



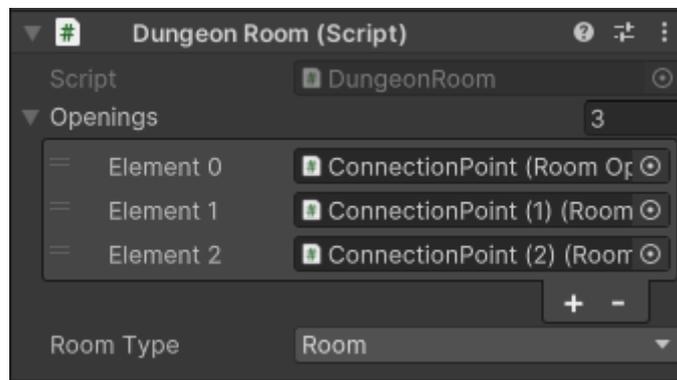
Largely, these scripts and systems were developed by trial and error with iteration being the main method of improving upon the key features.



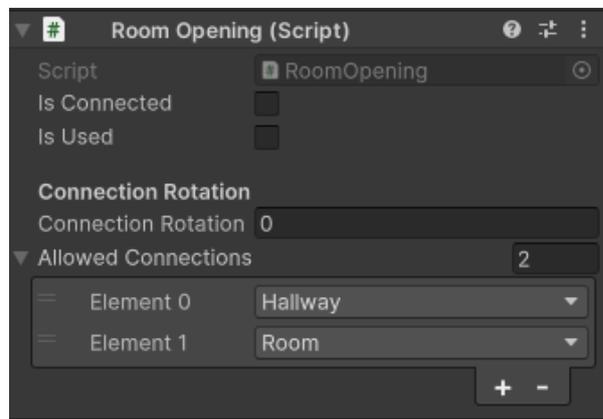
Serving as the master script, the Dungeon Master is where all the prefabs are held alongside the variables that are used to customise how big the dungeon is. This script is also responsible for starting the process of creating the dungeon and includes debugging messages to help spot any issues.



Alongside this, the Dungeon Room script was developed as a way to determine what type of room the prefab is and where the “Openings” are which will be where the connections to new rooms will be made.



As the final puzzle piece, the “Room Opening” script dictated what each “Opening” was allowed to connect to room wise. This ultimately helped make the system both procedural and customizable so we as developers maintained a level of control.



The Decision

Ultimately, it was decided that using door openings and customizable doorways was the better suited model for procedural generation for our project. This was largely due to the complexity of A* Pathfinding and also the lack of control developers had over the system. Furthermore, being able to determine room types and have a modular doorway system allowed for an easier design process with the rooms themselves which impacted both player game feel and quality.

