Evaluation

The system I intended to create during its design states ended up being difficult to implement and many issues were encountered. Although, the current version of the system has many of the customisable aspects I sought to implement at the beginning it still has its limitations. In my design plans, I planned to allow the user to create custom rooms, with varying sizes and spawn rates. This was a very hard thing to implement with Unity's Tile and Tilemap system having specific rules to follow. For example, the position of each tile was located to the bottom left of the tile resorting to many errors. With the room creation system, I ended up having the user create custom rooms within separate tile maps that the system clones and places within the grid. Although it does not cause significant performance issues, if a user were to scale up their project with 20 room prefabs or a map thousands of tiles wide, the system could suffer greatly. To fix this problem, I could have had the room prefabs copy their tiles onto one singular tilemap in a scene, however with time constraints and lack of knowledge in Unity's Tile system this was not possible during the project.

Additionally, corridor creation was a very big problem I faced with my planned algorithm not working as intended and significantly slowing the engine down. This led to lots of wasted time in attempted to fix the corridor's generation. Eventually, I rewrote the generator to use each room's midpoint in a scene and build corridors between each room's closest room. Although this method was vastly superior to my previous algorithm, improvements still could be made. Dungeon generator methods using Delaunay triangulation or binary space partitioning (BSP) would have greatly improved the system's performance and accuracy. They would have also allowed for a door generation system to be implemented which is missing from the project's current version. Due to the multiple tile maps and tile rules, a door generation system became too difficult to implement.

Overall, the project functions and meets the basic requirements I intended from the beginning. Including a playable game using the demo assets and scene to showcase what and how to use the system for. It has the player spawn in a premade starting room and navigate the dungeon to find the treasure chest. When the end room is reached, the player will be shown a UI panel where they can restart or quit the game. Although this clearly indicates the potential uses of the dungeon generation system it is still lacking. Due to a lack of knowledge in Unity's 2D system with tile maps, tiles, etc., many features that I wanted to include could not be implemented. The system also performs well enough performance-wise but could still be very much improved with different algorithms and better object handling.