

Project Proposal: 3D World Game

**Isaac Sexe,
Zach Summers**

4/18/2022

—

CMP_SC 4610

—

Professor Ye Duan

Overview

The project we choose to do was a 3D World Game called "name". This game will be rogue-like which has the player going deeper into the dungeon until defeated. We will procedurally generate a dungeon for the player to traverse. This dungeon will include enemies for the player to defeat, loot to collect, and a boss that must be located and defeated. After defeating the boss, the player will have the option to proceed to the next floor. These are the bare minimum features intended for the game and maybe expanded upon in the final version of the project.

Methods

The dungeon will be procedurally generated using an algorithm covered by vazgriz on his website [1]. Their algorithm itself was based on a different algorithm featured in the game TinyKeep, which is explained by A. Andonaac's in his blog [2]. The algorithm functions by creating different aspects of the dungeon and storing its relative information in data structures for later use. The first data structure is a 2D grid of the types of rooms and the other data structure stores information about individual rooms. This algorithm features four main processes: generating the rooms, locating the hallways, generating the hallways, and populating the rooms.

Generating Rooms

This algorithm generates rooms by selecting a random position and size with parameters given by the user. It will then create a room object in the scene and a room object in the script. The positions and size of this room are converted into (x,y) coordinates and changed to rooms in the grid.

Locating Hallways

The algorithm our's is based on uses something called Delaunay Triangulation to form a graph with the room centers as vertices. A minimal spanning tree is then calculated from this graph to form short logical paths from one vertex to another. A few of the removed edges are added back to the graph to create more loops in the graph to create more pathways for the player.

Generating Hallways

Rooms' information has been transformed into this graph a pathfinding AI creates the actual hallways for the dungeon. This is done by having the AI start from the edge of the room where an edge of the graph starts and move towards the room where that edge finishes. The heuristic function for the AI rewards the AI for going in a straight line and merging into existing hallways. The end result is a fully generated dungeon that features rooms of random sizes and semi-straight hallways connecting them.

Populating Rooms

Now that the dungeon is fully generated we can populate them to make the player feel less lonely. Firstly, we randomly pick a starting room for the player. This room should be close to the center of the dungeon position. Next, we will randomly pick a boss room. This room should be far away from the starting room and should not block off progress to another room. This means that if a room has a single entry point the room connecting shouldn't be a boss room. Then, we will select a reward room randomly. This can be any of the rooms Lastly, we randomly create some enemies. This should be done by randomly selecting a few of the remaining rooms and populating them with some sort of enemy.

References

1. <https://vazgriz.com/119/procedurally-generated-dungeons/>
2. <https://www.gamedeveloper.com/programming/procedural-dungeon-generation-algorithm>