Procedurally Generated Tower Defense 3D

Tori Broadnax[1], Jeffrey Do[1], Richard Roberts[1], and Minh Vu[1]

[1]George Mason University

December 9, 2021

The overall goal of this project was to create a 3D tower defense game that has procedurally generated levels. In this technical report, we will describe what we planned to do, how we went about it, and whether we completed it. Much of the technical report will be quoting our Project Proposal; it is all in here so there is no need to refer back to that document.

Requirements

Procedurally Generated Level

> The tower defense level will be procedurally generated. There should be at least one path from enemy spawn to the tower. The tower will be placed randomly. There should be tiles that cannot have weapons placed on them (like trees), and tiles that can. Weapons should not be able to be placed on the path.

User Interface

There should be a tower health indicator. There should be a menu for selecting weapons to build. Once selected, the player should be able to select a tile, and the weapon should build if funds are sufficient. There should be a currency indicator. There should be buttons for starting and stopping the movement of enemies (like a pause).

Weapons

There should be mutiple types of weapons that vary in damage, rate of fire, and range. Weapons will fire projectiles at enemies in their range. The player should be able to build these weapons on tiles in exchange for currency.

Enemies

Enemies will have pathfinding towards to tower. When enemies reach the tower, a corresponding tower health deduction should take place. Enemies will have varying health, speed, and currency drops. Enemies should detect collisions with projectiles and take damage from them.
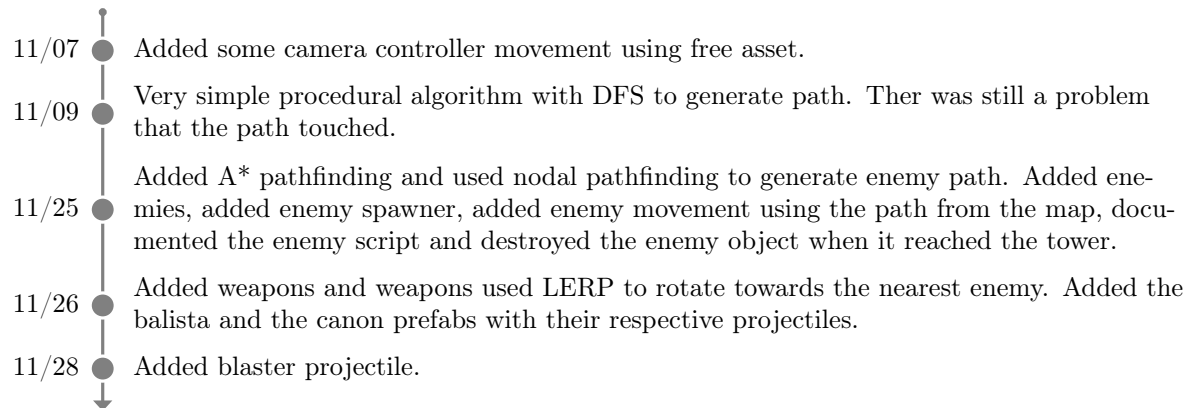
Reach Requirements

- Upgradable Weapons

- Weapon Selling

Technologies

- Unity

Timeline

11/07	Added some camera controller movement using free asset.
11/09	Very simple procedural algorithm with DFS to generate path. Ther was still a problem that the path touched.
11/25	Added A* pathfinding and used nodal pathfinding to generate enemy path. Added enemies, added enemy spawner, added enemy movement using the path from the map, documented the enemy script and destroyed the enemy object when it reached the tower.
11/26	Added weapons and weapons used LERP to rotate towards the nearest enemy. Added the balista and the canon prefabs with their respective projectiles.
11/28	Added blaster projectile.

References

- 3D Tilemap in Unity
- Brackeys Unity Tower Defense
- Manual Tilemap Unity
- Procedural Tilemap
- Kenney Tower Defense
- Unity GitHub Actions