Chirag Nanda

Design Document

Introduction

My level is called "Flypoint". The main objective is to collect all collectibles on the map without depleting your health completely. The player loses the game if they lose all their health or fall below a certain height. Upon losing, the text "Game Over" is flashed on the screen before the level resets, and the player restarts at the starting location with full health. Once all the collectibles are found, the text "Level Complete" is flashed on the screen, and the level ends.

Level Design and Mechanic

Since the level had to accommodate four different enemy classes, I decided to have one enemy per island to showcase all of them. Upon starting the level, the player first lands on a barren island with only one collectible to get room to get used to the controls. I decided to keep the flyer midair between the first island and the biggest/nearest island to keep the player engaged as they fly. The biggest island consists of two islands joined together. Since the area was large, I added a pursuer as it benefits from having large space due to its movement component. The other islands are smaller and have stationary enemies (mortar/proximity canon). Around each enemy, there are four collectibles. Additionally, there is only one health pack near each enemy to keep the game challenging. The level is also bounded using a trigger box. If the player falls below this boundary, the game is reset.

To aid the player in dodging, I decided to add a roll mechanic that the player can perform by pressing the "R" key. This mechanic makes the player move faster and reduces their footprint to dodge attacks. I implemented the roll mechanic by using a roll animation which I then used to create a montage. I then added a default component to the state machine to blend this animation. The mechanic only works if the player is not flying or falling.

Overall, my game has five enemies: one pursuer, one flyer, one mortar, and two proximity canons. Each enemy can also be killed by jumping on its head (pursuer) or center (proximity canon/mortar). The flyer cannot be destroyed as it will remain above the player or move away from the player. Also, the player takes damage and is knocked back upon colliding with the enemy (excluding the head/center). For the knockback, I use the launch character module. Lastly, I used various widgets for the UI and game over/level complete screens, including a progress bar and an internal health count to implement the health system.

Enemy Classes

1. **Pursuer:** The main objective of the pursuer is to chase the player away and return to their original patrol once the player is far enough away. I implemented this using UE4's pawn seeker and AI_MoveTo modules. The pursuer returns to its original patrol if the player is too far away or goes out of sight.
2. **Flyer:** The flyer remains above the player and shoots projectiles that lead the player's movements. I accomplished this by reading the player's direction vector and calculating an offset using the player's speed and location. I then launch the projectile at the player's approximate "next" location using this information. The flyer also uses UE4's pawn seeker. So, the player must be within the flyer's line of sight for it to attack.
3. **Mortar:** The mortar launches projectiles around itself randomly. Upon colliding with anything, these bombs explode and knock back the player by launching the player with a specific velocity. During this short period, the player cannot move
4. **Proximity cannon:** If the player moves within a specific distance from the cannon, it automatically rotates to look and shoot at the player. This feature was implemented using a trigger box. I calculated the degree of rotation based on the player's location versus the cannon's head. I decided to make another stationary enemy as there were already two mobile enemies in the game.