Name\_\_\_\_Miles Kison\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

The feature that I will incorporate into our game is enemy AI, interaction, and spawner.

When a player enters combat zones, my job is to have a spawner that finds the appropriate enemies from a pool and spawns them into the world. Once spawned, all enemies need an AI model that determines their behavior. The number of enemies and difficulty of the encounter will be resolved before the player enters, based on hard coded values.

The AI model will deal with all the decisions the enemies could make while alive. Their primary objective is to chase after the player until they reach a certain distance away, and then they will attempt to attack. The pathfinding for the chase and the distance at which they attack will be handled differently for each enemy

## Use case diagram with scenario \_\_14

### Use Case Diagrams

A notebook with writing on it

AI-generated content may be incorrect.

### Scenarios

**Name:** Spawn an Enemy

**Summary:** The enemy spawner in a zone places enemies near the player, with specific types and pathfinding capabilities. The spawn location will be at a point that is safe for the enemy to exist at until it tries to fight the player. It will also keep track of the number of enemies spawned to determine when they no longer need to be added.

**Actors:** Enemy Spawner

**Preconditions:** Player is in a combat zone that they have not entered before in the level

**Basic sequence:**

**Step 1:** The enemy spawner is called after the player enters a combat zone

**Step 2:** The number and difficulty of the enemies are decided by the enemy type

**Step 3:** Enemies are spawned in around the player at legal positions

**Step 4:** Check if the player has killed all enemies and if more need to be spawned

**Exceptions:**

**Step 1:** The player may have already gone into the zone, and the spawner would not be called

**Post conditions:** enemies are spawned and are attacking the player

**Priority:** 1\*

**ID:** ES1

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

**Name:** Set Enemy Type

**Summary:** Sets enemies’ abilities, health, and damage of difficulty and variety. Finds the player in the zone and path finds its way close enough to attack the player

**Actors:** Enemy

**Preconditions:** Spawned in my spawner

**Basic sequence:**

**Step 1:** The enemy gets a health and a damage value

**Step 2:** Enemies call for the player’s location and path find to the player

**Step 3:** Enemies check if the player is in range of an attack. If so, attack

**Exceptions:**

**Step 1:** The enemy has been spawned

**Step 2:** The player did not leave the combat zone

**Post conditions:** Enemies are chasing and attacking the player

**Priority:** 1\*

**ID:** ME1

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

### Data Flow Diagrams

A diagram of a diagram

AI-generated content may be incorrect.

### Process Descriptions

A paper with writing on it

AI-generated content may be incorrect.

## Acceptance Tests \_\_\_\_\_\_\_\_9

This feature will be tested for correct spawn locations, number of enemies, types of enemies, and pathfinding. Each will be tested in a different way and in different phases.

**Spawning Locations:**

Every combat zone will be tested 100 times to ascertain if the enemies are spawning in on legal coordinates. The test will look for whether the enemy tries to spawn on the player or another enemy, or in a spot that would trap them.

**Types of Enemies:**

The type of each enemy will be cross-checked with the difficulty, ensuring that they are appropriate to the level.

**Pathfinding:**

Every enemy needs to try to attack the player, so a simple check is needed to determine whether an enemy is successfully attacking the player or, at the very least, moving towards that goal. This check will come in the form of checking the enemy's position and if it is changing often enough to get to the player. A nonmoving enemy that is not attacking is stuck.

**Spawn Location Tests:**

|  |  |  |  |
| --- | --- | --- | --- |
| Spawn location | Player Location | Spawn Area | Result |
| 40,30,0 | 46,70,0 | 100,100,50 | correct |
| 46,70,0 | 46,70,0 | 100,100,50 | On player incorrect |
| 70,30,40 | 46,70,0 | 100,100,50 | Too high incorrect |
| 120,30,0 | 46,70,0 | 100,100,50 | Out of area incorrect |
| 20,50,0 | 46,70,0 | 100,100,50 | correct |

## Timeline \_\_\_\_\_\_\_\_\_/10

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Difficulty design | 1 | - |
| 2. Enemy type designs | 1 | 1 |
| 3. Enemy pathfinding program | 1 | 1 |
| 4. Enemy art | 1 | 2 |
| 5. Level design | 2 | 3 |
| 6. Programming | 4 | 3 |
| 7. Testing | 2 | 6 |
| 8. Installation | 2 | 4,7 |

### Pert diagram

A drawing on a piece of paper

AI-generated content may be incorrect.

### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  | 1 |  |  |  |  |  |  |  |  |
| 3 |  | 1 |  |  |  |  |  |  |  |  |
| 4 |  |  | 2 |  |  |  |  |  |  |  |
| 5 |  |  | 3 |  |  |  |  |  |  |  |
| 6 |  |  | 3 |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  | 6 |  |  |  |
| 8 |  |  |  |  |  |  |  |  | 7 |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |