Name: Savannah Johnson Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature will be level generation, which includes enemy spawning and creating interactable terrain features on the static map. Level generation is called in two major instances: the first is when gameplay is initiated (player starts the game), and when the player progresses to the next level upon finishing the current one.

When a new game is started, the map is displayed. All entities are spawned on “walkable” areas. Enemies are spawned semi-randomly throughout the map. Enemy behavior and spawn rates are affected by the current level. Cover will be spawned in fixed areas, but they can be damaged and destroyed by player and enemy attack.

When a new level begins, all instances of any enemy will be removed and replaced with a new wave, which may include a special, unique boss enemy. Any damaged or destroyed cover will return to their original undamaged state.

## Use case diagram with scenario \_\_14

### Use Case Diagrams

Diagram

Description automatically generated

### Scenarios

**Name:** Enemy Generation

**Summary:** Spawns a given number of enemies in walkable areas; type and frequency are pseudo-random and weighted by current level value.

**Actors:** Game Start/Level Progression

**Preconditions:** Level instance (not filled) and Map created

**Basic sequence:**

1. Get enemy spawn capabilities from Enemy feature
2. Determine what enemies are available at the current level
3. From available enemies, determine an amount of each type to spawn
4. Place enemy instances onto the walkable parts of the map

**Exceptions:**

1. Level unavailable or spawn unavailable: spawn no enemies
2. Spawn on unwalkable terrain: destroy enemy instance

**Post conditions:** Enemies are spawned with their set behavior and weapon

**Priority:** 2

**ID:** C01

**Name:** Interactables Generation

**Summary:** Determines where static and dynamic items that can be interacted with are placed.

**Actors:** Game Start/Level Progression

**Preconditions:** Level instance (not filled) and Map created

**Basic sequence:**

1. Remove any damaged terrain objects from the scene (cover, fences)
2. Access a storage of all static-positioned objects in the map
3. Spawn in any object that needs to be replaced
4. In walkable areas, generate items if the level allows it

**Exceptions:**

1. Terrain object is not damaged: do not respawn terrain object
2. Item spawns over terrain object: count terrain object as unwalkable space

**Post conditions:** Terrain objects are spawned and interactable, other items are spawned under certain conditions

**Priority:** 2

**ID:** C02

**Name:** Current Level

**Summary:** Keeps a record of what levels the player has completed, and which level they are currently on. Displays current level on UI

**Actors:** Game Start/Level Progression

**Preconditions:** Level instance (not filled) and Map created

**Basic sequence:**

1. Level begins at 1 from Game Start
2. On level completion (all enemies defeated), level will increase

**Exceptions:**

1. Player loses: their level resets back to 1, and a new game begins

**Post conditions:** The Level UI is updated, and new level information is sent to the generator.

**Priority:** 1

**ID:** C03

**Name:** Interactable Terrain

**Summary:** Splits all map items into either walkable or not walkable sprites

**Actors:** Game Start/Level Progression

**Preconditions:** Map, static objects generated

**Basic sequence:**

1. Map is created from sprites in static positions
2. Walkable sprites are tagged as such
3. All walkable sprites are given physics that allow a player to move over them
4. All unwalkable sprites are given physics that doesn’t allow a player to move through them

**Exceptions:**

1. Sprite is not available: display error sprite + message

**Post conditions:** The Map is given collision physics

**Priority:** 1

**ID:** C04

**Name:** Level Generation

**Summary:** All interactable entities in a level are generated and placed in the map, along with the player, the game level, and their score.

**Actors:** Game Start/Level Progression

**Preconditions:** Enemy, interactables, and terrain are generated and ready to be placed.

**Basic sequence:**

1. Clear all entities from the map
2. Level is either created or updated
3. Terrain is regenerated if it was destroyed/nonexistent
4. Enemies are generated based on level
5. Items are generated based on level

**Post conditions:** The Level now contains interactable entities and a walkable area for the player.

**Priority:** 1

**ID:** C05