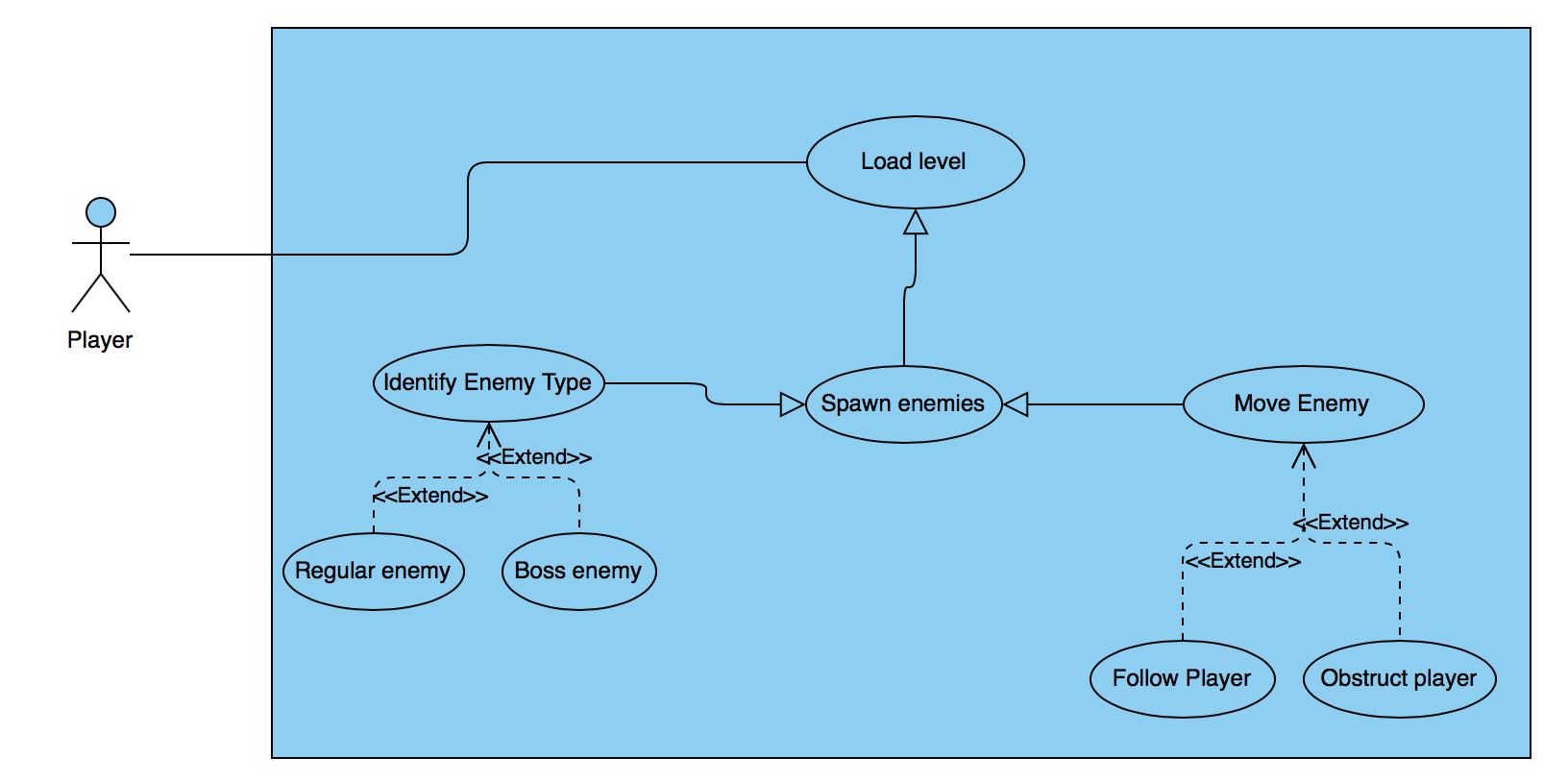
Name: Edgar Velazquez Colin Mark \_\_\_\_/50

## Brief introduction \_\_/3

I will be responsible for creating and spawning the enemies**.** These enemies will be placed in different parts of the scene which will serve as an extra obstacle for the player. Some will move faster than others. Some will have set movements on parts of the map and others will follow the player.

## Use case diagram with scenario \_\_/14



### Scenarios

**Name:** Spawn Enemy

**Summary:** Different types of enemies will be loaded onto different parts of the map.

**Actors:** Player

**Preconditions:** Player and scene must be loaded first

**Basic sequence:**

**Step 1:** Get players position and level

**Step 2:** Spawn the correct type of enemy and assign movement

**Step 3:** Update the position of enemy

**Exceptions:**

**Step 1:** Cheat codes are enabled

**Post conditions:** The correct type of enemy will spawn on the map and a movement type will be assign to it

**Priority:** 1\*

**ID:** C01

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

**Name:** Identify Enemy Type

**Summary:** There will be 2 types of enemies: Regular enemies and boss enemies

**Actors:** Player

**Preconditions:** Player and obstacles

**Basic sequence:**

**Step 1:** Determine player location and level

**Step 2:** Determine (with position and level info) if a normal enemy or a boss enemy needs to spawn

**Step 3:** Spawn correct enemy

**Exceptions:**

**Step 1:** Cheat code is enabled

**Step 2:** All enemies will be slower than normal

**Post conditions:** The correct enemies will spawn for different levels of the game

**Priority:** 3\*

**ID:** C01

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

**Name:** Move Enemy

**Summary:** There will be two types pf movements that the enemies can do

**Actors:** Player

**Preconditions:** Player and obstacles

**Basic sequence:**

**Step 1:** Determine what level the player Is on and position

**Step 2:** Spawn correct enemy and enemy type

**Step 3:** Assign enemy movement

**Exceptions:**

**Step 1:** Cheat code is enabled

**Step 2:** All enemies will be slower than normal

**Post conditions:** The enemy will be assigned a type of movement depending on the level the player is on

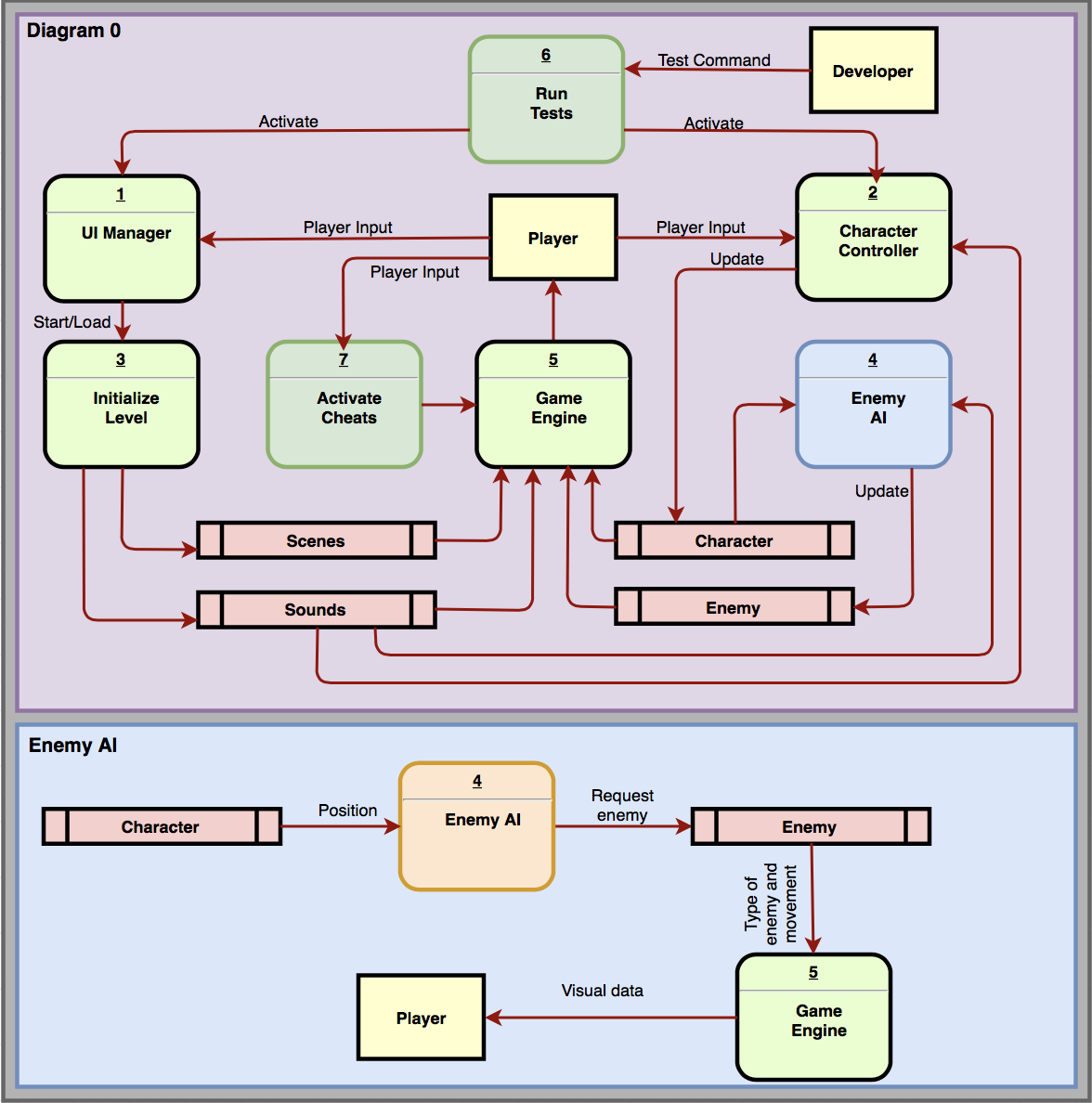
**Priority:** 2\*

**ID:** C01

\*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



### Process Descriptions

Enemy AI:

If the player is in x position and y level

Spawn boss enemy

Else

Spawn regular enemy

If cheat code is true,

Spawn regular enemies

Set speed=speed/2

**\*Notes**: Depending on the position and the level the player is, there will be statements that will determine the type of enemy and the movement is should be assigned. The only exception to the rule is when the cheat code is enabled, then the enemy will always be of type normal and the speed will be speed/2.

## Acceptance Tests \_\_\_/9

Test 1:

There will be normal enemies and boss enemies, I will check that the right enemy is loaded for the right level. I will create a test which loads a level and make a comparison to know if the right enemy was loaded onto the scene.

Test 2:

Have a character that moves through the levels and expects an enemy to collide with him on certain positions of the map where there should be enemies. Another form of testing can also be by having the player walk to certain areas of the map stop there and wait for an enemy that follows to spawn and collide with it.

The output file will have the following characteristics:

* **Number of enemies**: The number of enemies that have been originally created for that level
* **Number of collisions**: The number of collisions should equal the number of enemies in order for the test to pass
* **Special occasions:** Only when the I am testing for enemies that move, the number of collisions will be greater than the number of opponents in order for the test to pass. This is because the enemies that follow can have collisions with the player on different areas of the map.

**Example for Enemies per Level**

|  |  |  |  |
| --- | --- | --- | --- |
| Output | Number of enemies | Number of collisions | Notes |
| 3 | 3 | 3 | Test should have the number of collisions equal the number of enemies |
| 5 | 5 | 5 |  |
| 5 | 1 | 5 | This can test if the enemy follows the player that they will collide at different locations |

## Timeline \_\_/10

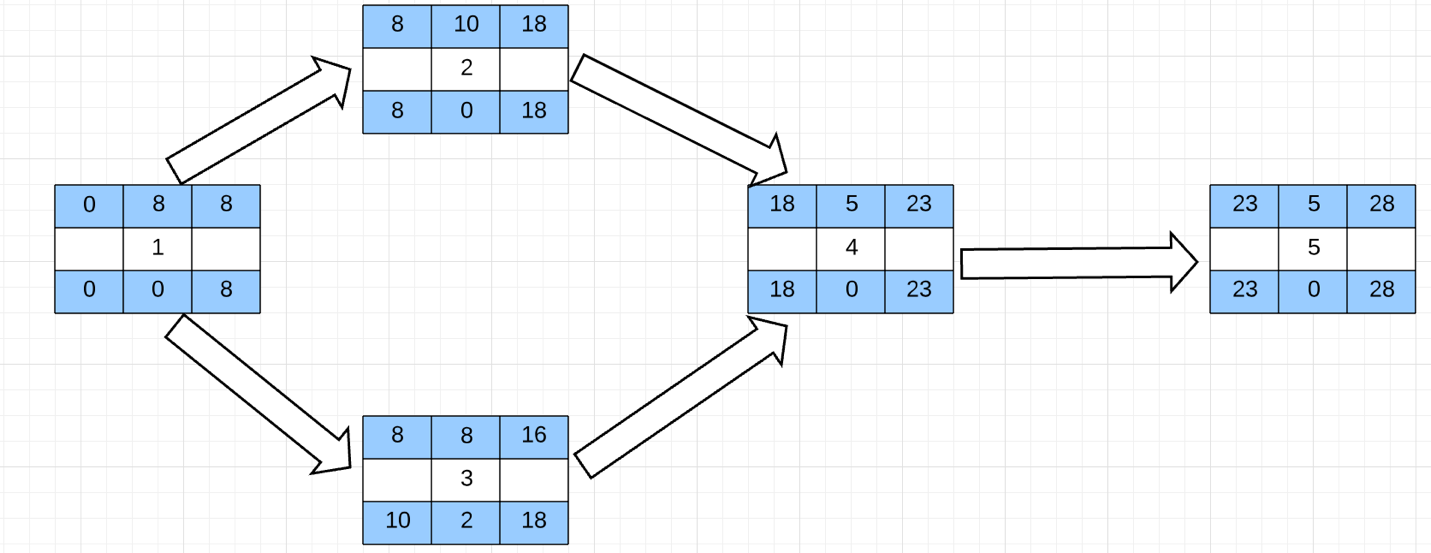
[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| **Task** | **Duration (hrs)** | **Predecessor Task(s)** |
| 1. Design enemies | 8 | - |
| 2. Design enemy movement | 10 | 1 |
| 3. Place enemies on map/program spawn locations | 8 | 1 |
| 4. Create Documentation | 5 | 3 |
| 5. Testing | 8 | 4 |

### Pert diagram



### Gantt timeline

