Name: Shreeya Pradhan Mark: \_\_\_\_/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

## Brief introduction \_\_/3

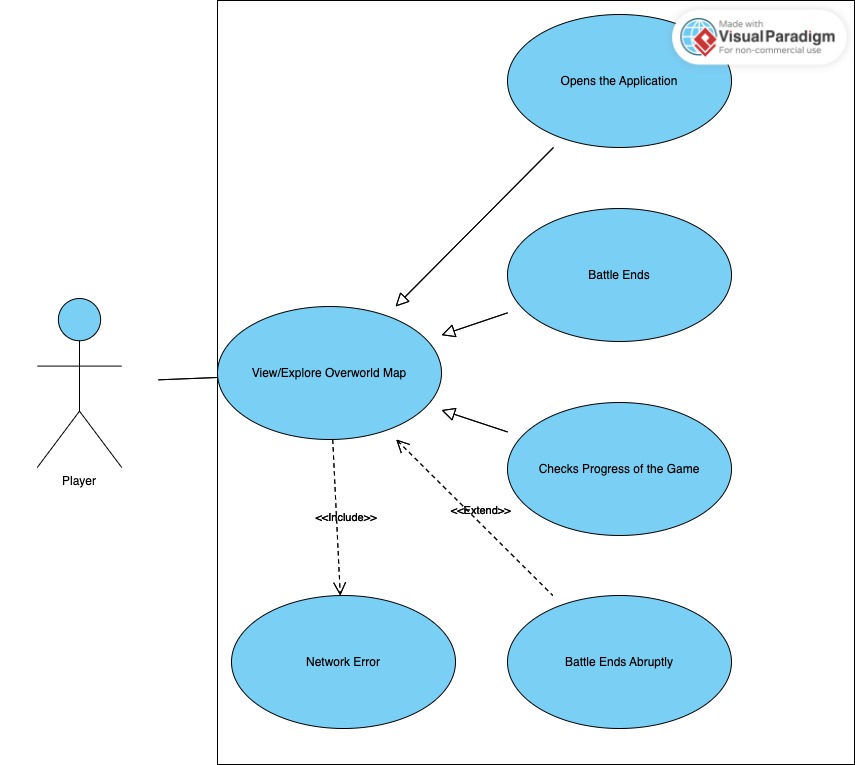
My feature for the game Corsair Clash is Overworld Map and Battle Map/s. When you first open or log in into your game, you will be able to see the world in which all battle will be taking place. As you advance more into the game, the location will be changing hence your map for different level will be different.

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

### Use Case Diagrams

### Scenarios

**Scenario 1 (1st Use Case Diagram):**

****

**Name:** View/Explore Overworld map

**Summary:** The player explores the Overworld Map.

**Actors:** Player

**Preconditions:** Player has the game installed and is logged into the game.

**Basic sequence:**

**Step 1:** Player opens the game application.

**Step 2:** Player selects “Overworld Map”.

**Step 3:** The game displays the current overworld map.

**Step 4:** Player can interact with the map to choose a location. This step is bit tricky as player are required to have certain rank to access specific map/s.

**Exceptions:**

**Step 1:** Player encounters a network issue.

* Display a message “Network Error. Please check your connection.”

**Step 2:** Player closes the game before selecting a location.

* End the use case?

**Post conditions:** Player explores and selects a location for a battle.

**Priority:** 1\*

**ID:** C01

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

**Scenario 2 (2nd Use Case Diagram):**

**Name:** View/Engage in a Battle Map

**Summary:** The player engages in the battle in a selected battle map.

**Actors:** Player, Enemy player

**Preconditions:** Player has selected the battle map to go against the enemy.

**Basic sequence:**

**Step 1:** Player selects a location with the enemy.

**Step 2:** The game loads the Battle Map for that location.

**Step 3:** Player and enemy engage in the battle.

**Step 4:** Battle ends when the player or the enemy wins.

**Exceptions:**

**Step 1:** Player wins the battle.

* Display a message “CONGRATUALTIONS!”. It will be displayed with the points earned form this battle.

**Step 2:** Player loses the battle.

* Display a message “DEFEATED! Try harder next time, LOSER.”

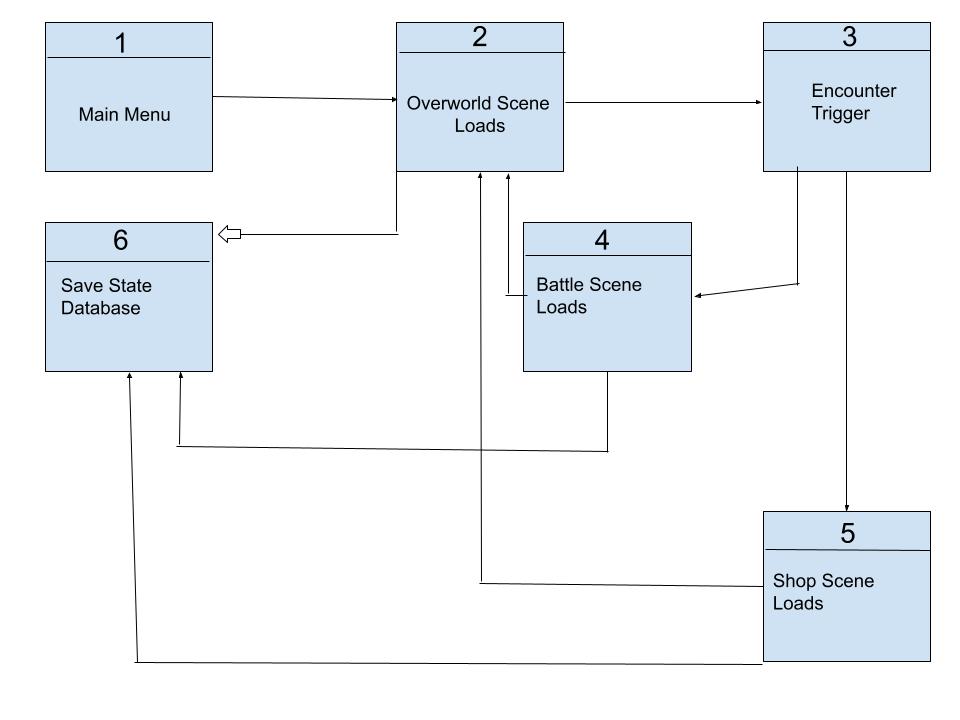
**Post conditions:** Player exits out of the map and is brought to the Overworld Map. They will see the progress points in this point.

**Priority:** 1\*

**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



In the data flow diagram below, I will be working on the overworld scene loads and battle scene loads.

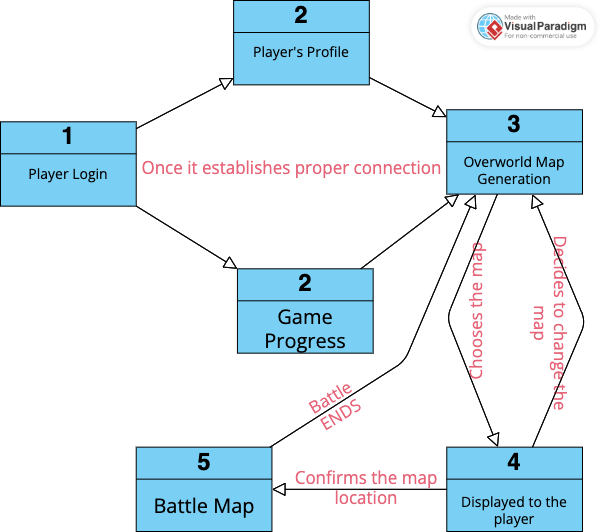
In the first diagram, I will show how the overworld map and its sub-processes will work.

In the second diagram, I will show how battle map is dependent on other features. The end case of the battle map will most likely be overworld map as well.

### Process Descriptions

1. Overworld Map:

First, it will load the player’s profile. Then, it will determine the progress in the game which will show the overworld map based on the player’s progress. It should also allow the player to interact with the map.



1. Battle Map:

Before generating the battle map, the player’s choice on the Overworld Map triggers the location of map. Then, it should load the assets and settings for the selected location to generate the battle Map. After this, the player and the enemy data will be fetched and put on the selected map. Finally, the battle BEGINS!

A diagram of a game

Description automatically generated

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

**Overworld Map**

The data based on which the overworld map will be generated is the player’s login profile and the game progress of the player. The player’s profile includes the level and progress whereas the game progress includes completed quests, victories, weekly missions and more. After these are determined, the Overall Map gets displayed to the player. We can create boundary cases by testing with various player progress levels, such as a new player (level 1), a mid-level player (level 10), and an advanced player (level 20).

Input: Player's profile, game progress

Output: Generated Overworld Map, displayed to the player

Boundary cases: Test with various player progress levels.

**Battle Map:**

Input: Location selected by the player, assets, settings, player, and opponent data

Output: Generated Battle Map, loaded for the battle

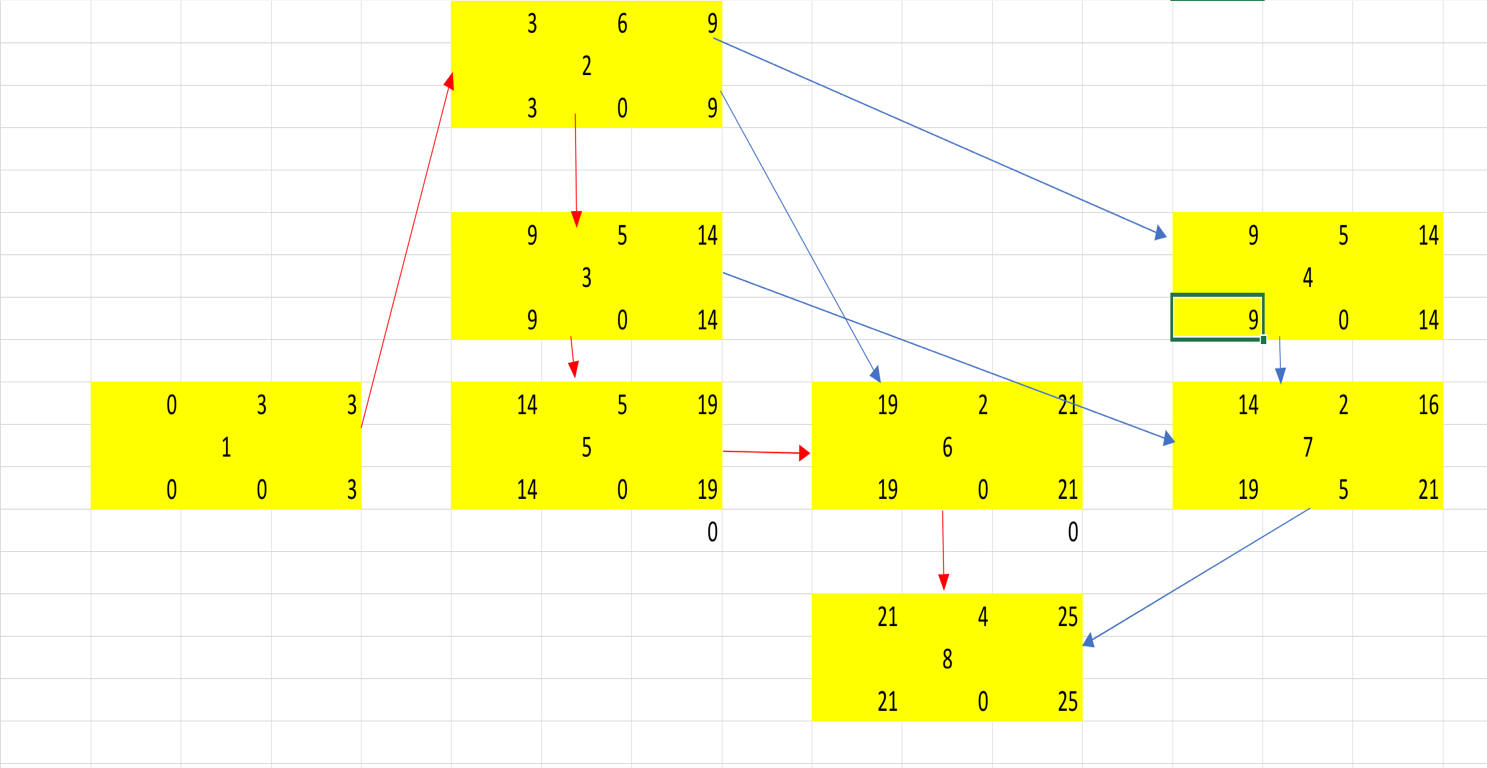
Boundary cases: Test with different locations and opponents.

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

### Work items

|  |  |  |
| --- | --- | --- |
| Tasks | Duration (Hour/s) | Predecessor Task/s |
| 1. Get Player’s Information | 3 | - |
| 1. Build the map | 6 | 1 |
| 1. Add assets and features for both | 5 | 1,2 |
| 1. Link both the maps | 5 | 2 |
| 1. Coding and Programming | 5 | 1,2,3 |
| 1. Documentation | 2 | 2,5 |
| 1. Testing | 2 | 3,4 |
| 1. Final | 4 | 6,7 |

### Pert diagram



### Gantt timeline

A graph with numbers and a line

Description automatically generated with medium confidence