50.003 Elements of Software Construction

Project Meeting 2 Report

Group 12

**Rookie Cookie**

**Prepared by:** Chok Hao Ze 1003071

Gong Chen 1002870

Lim Theck Sean 1002777

Oh Huan Xuan 1003022

Tang Mingzheng Paul 1002768

**Prepared on :** 1 March 2019

**Report Content**

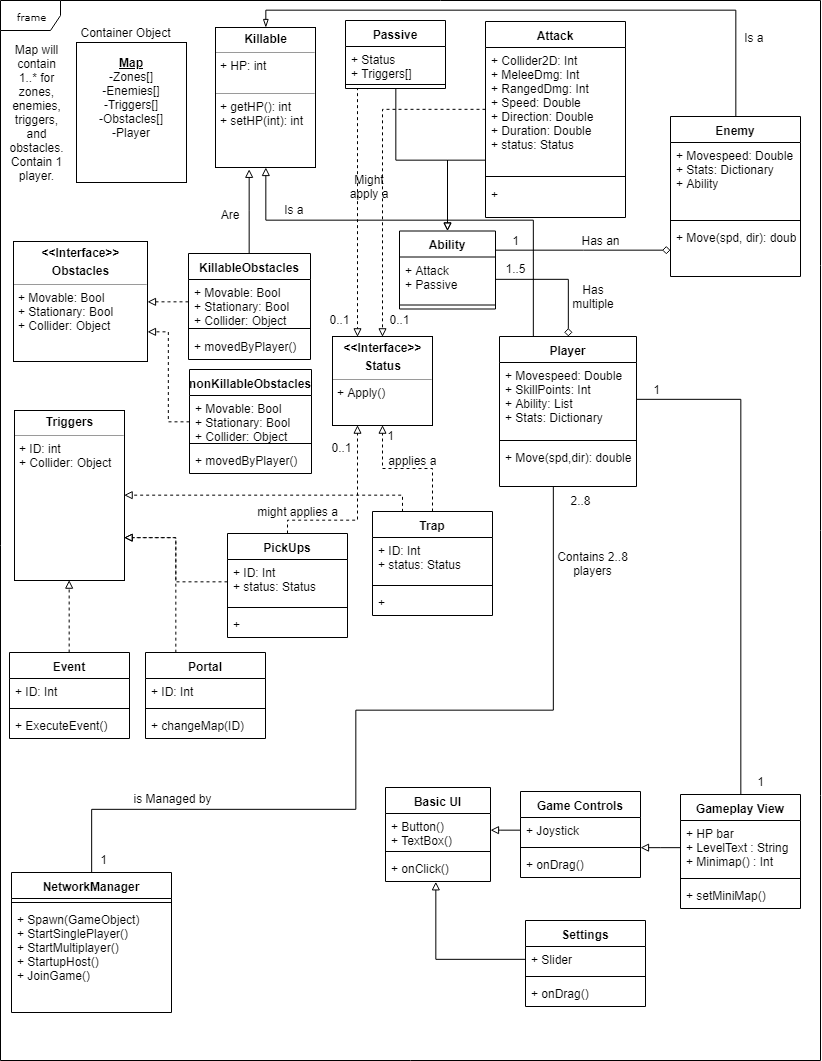
|  |  |  |
| --- | --- | --- |
| **Index** | **Description** | **Page Number** |
| 1. | Changes in Project Requirements | 3 |
| 2. | Documentation of System using UML Diagrams  Class Diagram | 4 |
| 3. | Use Case Diagram | 5 |
| 4. | Use Cases Table and Misuse Cases Table | 6 - 7 |
| 5. | Sequence Diagrams   1. Logging in and Starting the Game 2. Single Player 3. Multiplayer | 8  9  10 |
| 6. | Commit Records | 11 |
| 7. | Workload Distribution | 12 |
| 8. | Testing Plans | 13 |

**Changes in Project Requirements**

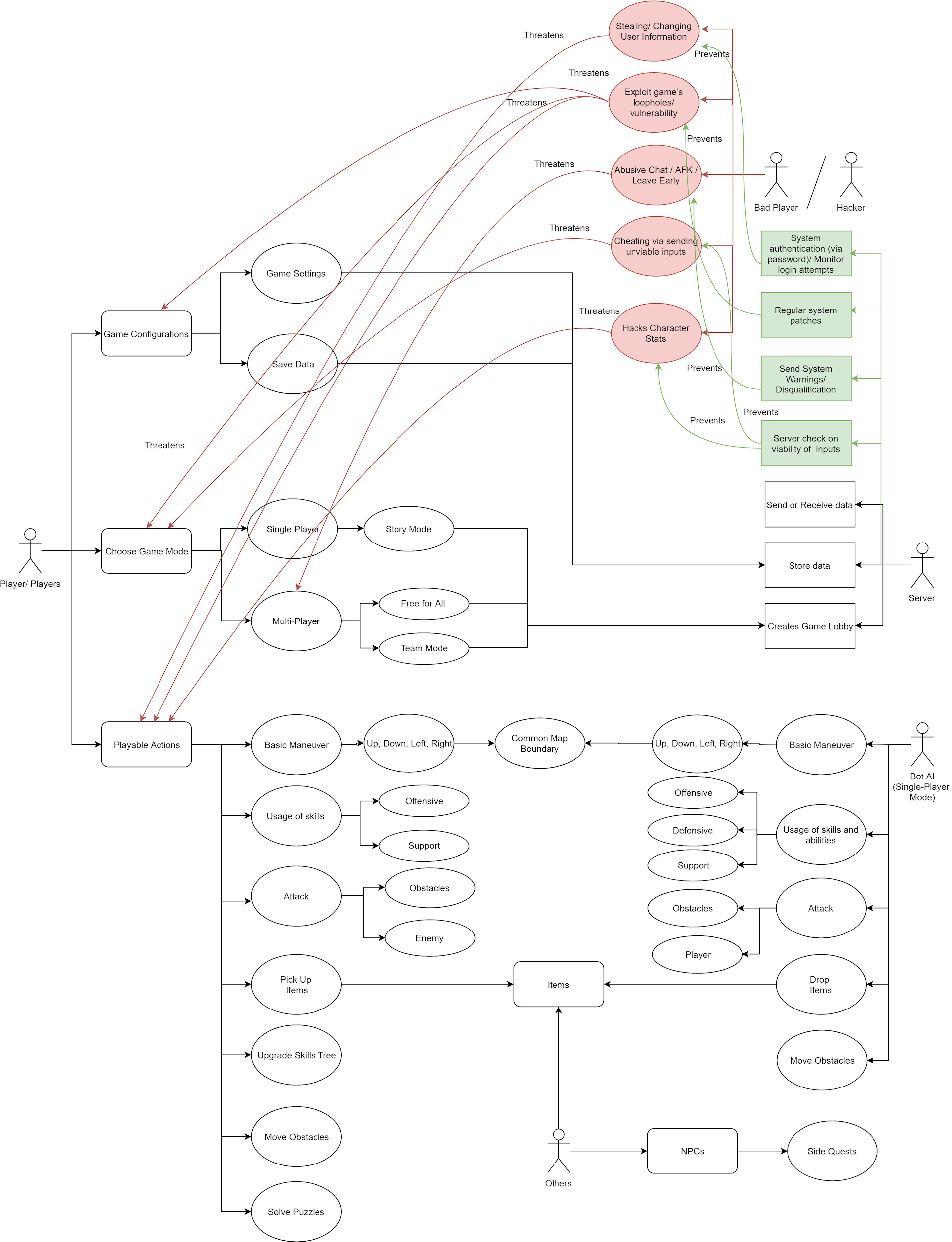
|  |  |
| --- | --- |
| **Changes Made** | **Reasons** |
| Game genre was re-scoped. New story premise is more volatile and adaptable. | 1. To appeal more to target audience. SUTD Students and Profs. 2. Easier creation of assets. |
| Game storyline was modified and shortened. | 1. A different gameplay while reusing of assets. 2. Development time constraints |
| Focus more on game mechanics and UI. | Time target audience spends on the game is limited. To leave as good of an impression as possible. |
| Not using web game platform anymore. | Skip the process of web hosting. Unity has the capability of exporting game in android /IOS format. |

**UML Diagrams**

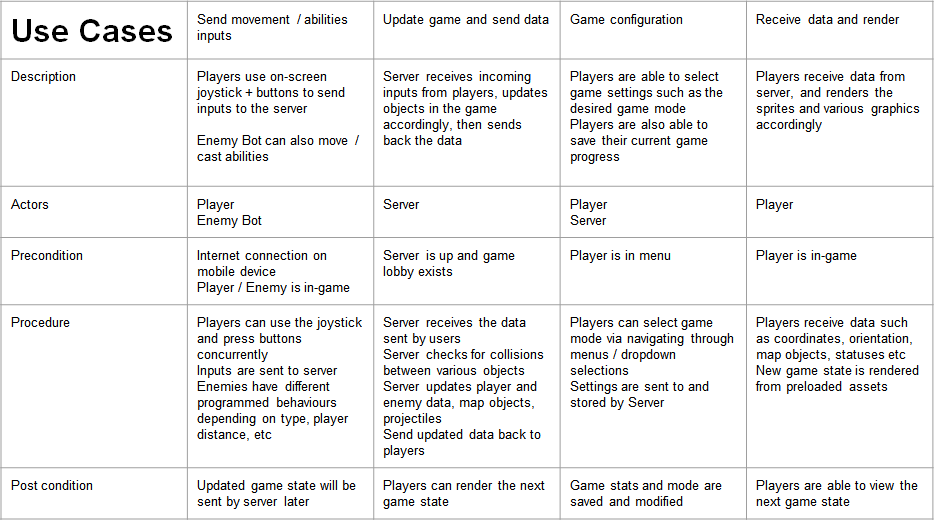
Class Diagram



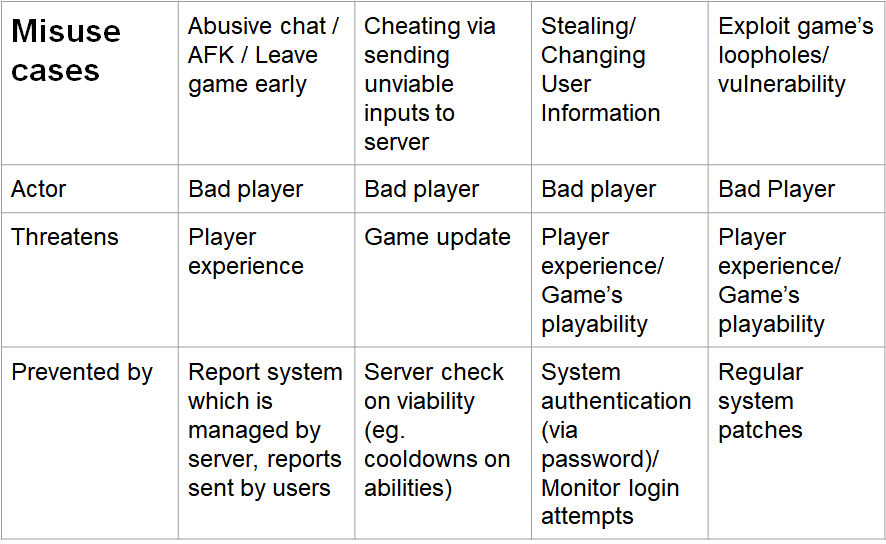
Use Case Diagram



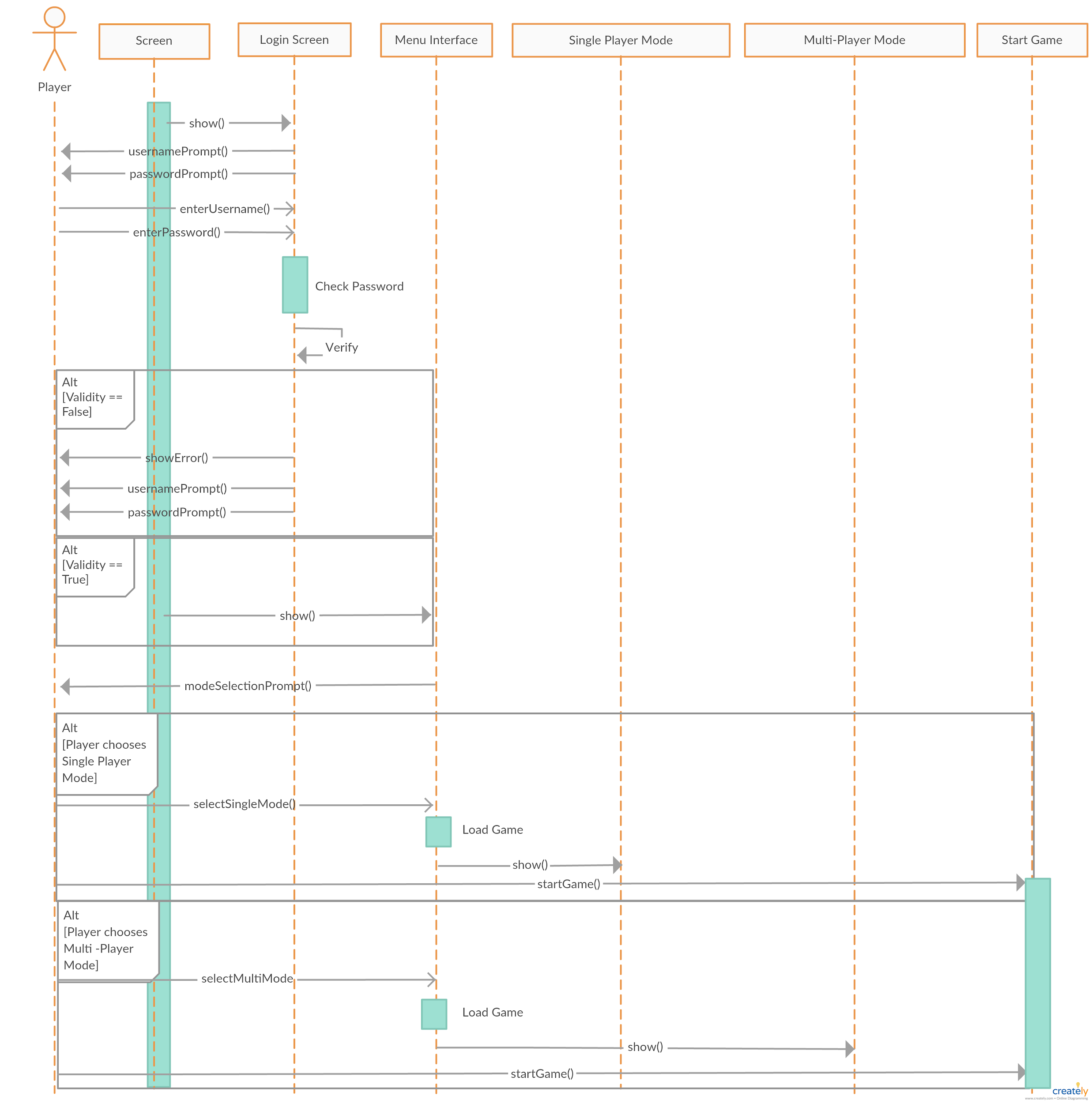
**Use Cases Table**



**Misuse Cases Table**

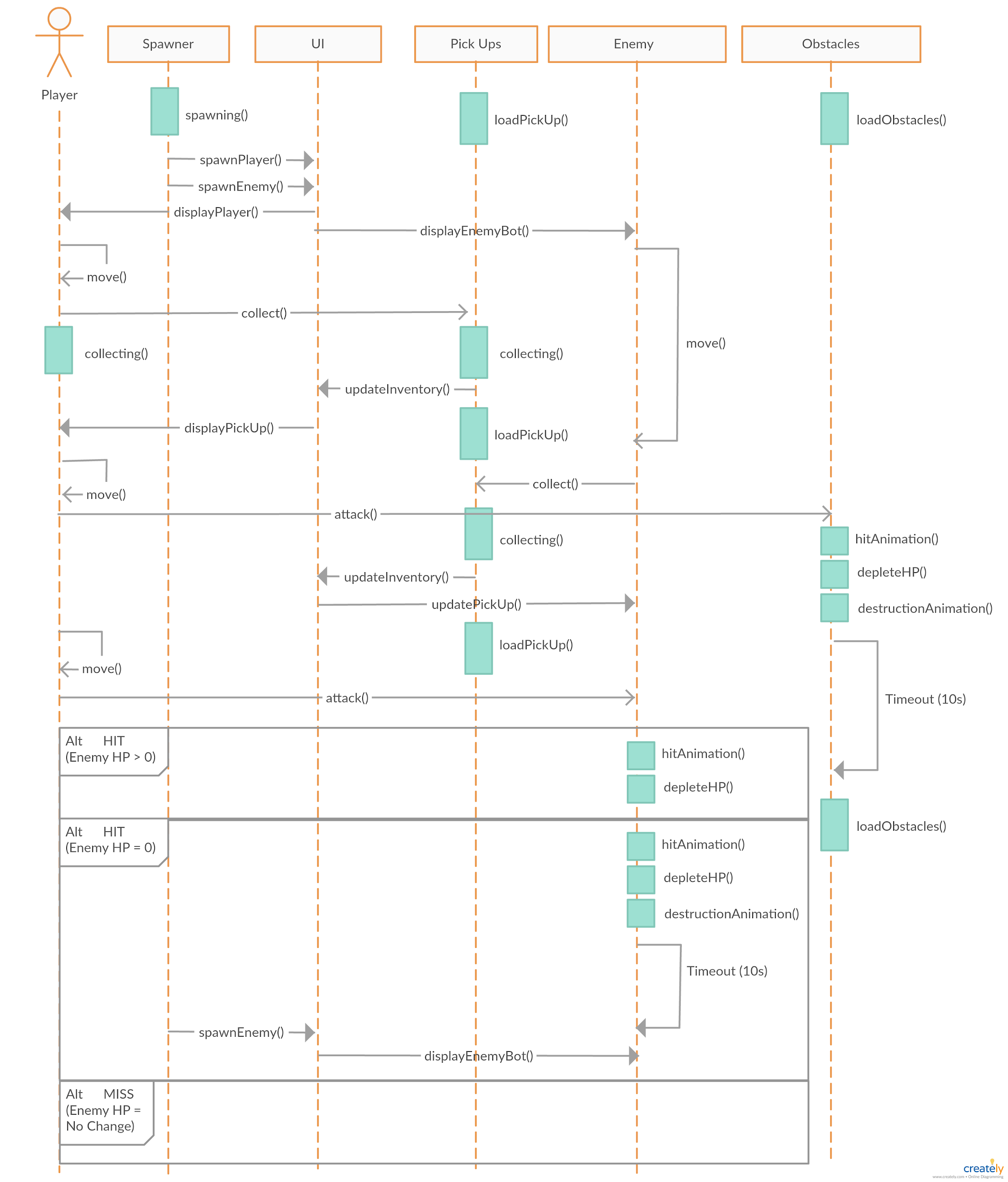


Sequence Diagram (Logging in and Starting the Game)



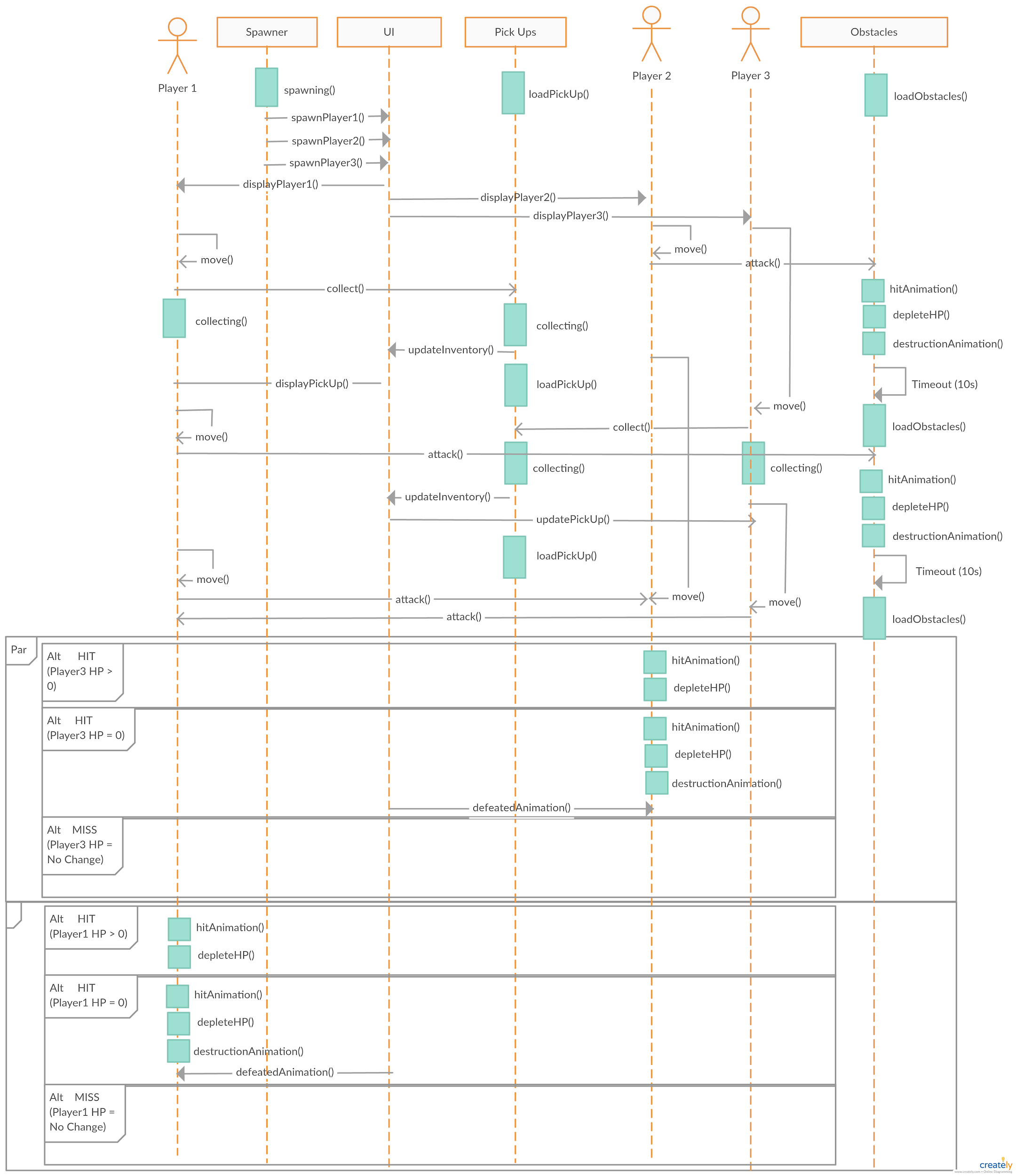
This sequence diagram describes the event flow during the process of logging into the game. The player first keys in his username and password while the system checks for its authenticity. Upon an unsuccessful log-in, the player would be prompted to re-enter his/her username and password. Otherwise, the player would be directed to the game menu page, where he/she can choose between the single player or multiplayer mode. Once the game is loaded, the desired game mode starts when the player hits the ‘Start’ button.

Sequence Diagram (Single Player)



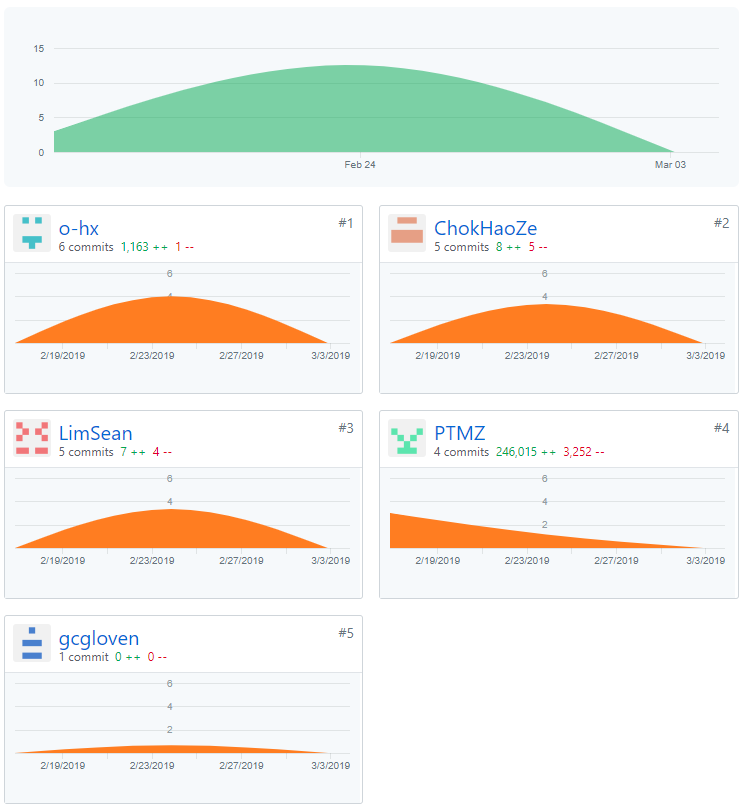
This sequence diagram describes the flow of events when the player is in the single player mode. It depicts the scenario of player 1 being spawned by our game spawner, picking up in-game pick-ups, attacking destructible obstacles before attacking the enemy bot. Once the attack was made, there could be three outcomes - the enemy got hit and its health points depleted, the enemy got hit to the extent that it got destroyed and lastly, the player missed the enemy bot completely.

Sequence Diagram (Multiplayer)



This sequence diagram describes the flow of events when the player is in the multiplayer mode. It depicts the scenario where the players engaged in different activities at the start - player 1 was collecting pick-ups and destroying obstacles, player 2 was also destroying obstacles before exploring the map while player 3 was exploring the map and collecting pick-ups. Upon the sight of one another, player 1 launched an attack on player 2 while player 3 launched an attack on player 1. All possible outcomes are documented.

**Commit Records**

****

**Workload Distribution**

**Paul** - incharge of the multiplayer system and game mechanics, and feasibility study.

**Gong Chen** and **Huan Xuan** - in charge of drawing game art. Gong chen did up the Map assets of the Anti-Virus world, while Huan Xuan did sprite drawings for our virus cookie.

**Sean** and **Hao Ze** - in charge of game design and development, developing the game storyline and logic and preparation for project meeting.

Feel free to look into our project’s github to further understand each individual member’s contribution.

https://github.com/PTMZ/ESC-Project

**Testing Plan**

**Tools:**

Unity Test Tools framework - NUnit

Google Forms - Player feedback form

**[Week 9-11]**

**Alpha Testing:**

Black-box Tests:

1. Game User Interface(Buttons, joystick, controls) Testing (Unit Testing)
2. Game Save Testing (Unit Testing)
3. Game Pickup Testing (Unit Testing)
4. Game Play Testing (Unit Testing)
   1. Move
   2. Attack
   3. Change Ability
   4. Damage
   5. HP
   6. Puzzles (Integration Testing)
5. Game Maps & Boundary Testing (Integration Testing)
6. Multiplayer Testing (System Testing)
7. Game Scenario Testing (System Testing)
8. Latency/Stress Testing (Acceptance Testing)

White-box Tests:

100% Method Coverage

100% Statement Coverage

***Note:*** As our project is a game, most of the alpha testing will be done by us the developers playing the game, and beta testing will be players, outside of the development team, playing the game. More focus on black-box testing rather than white-box testing.

**[Week 12-13]**

**Beta Testing:**

Stress Testing: Have multiplayers in a game at once

Robustness testing: Have players play the game in an unexpected way.

User Study

1. After Game Survey for impression of game at different phases.
2. Questions to get general feedback of game
3. Learning curve of game
4. User’s gameplay intuition. What user focus on in playing. E.g. intuitive response towards scenarios in the game or how to progress the game. Intuitive thoughts.