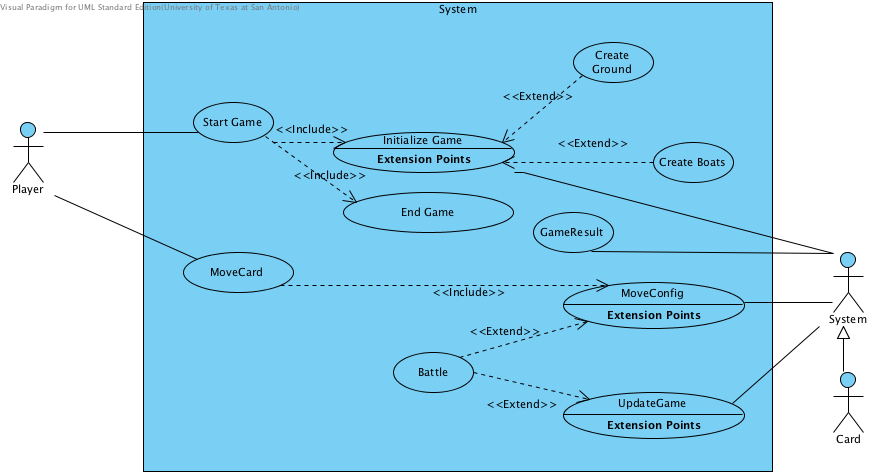
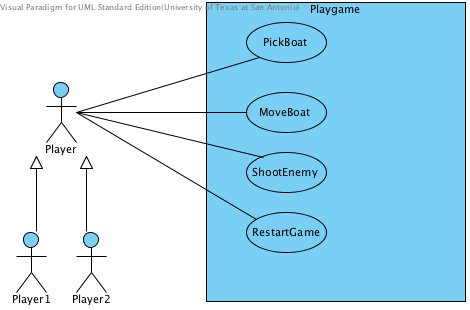
****

****

**Actors**

1. Player: Is the person who plays the game, can start the game, end the game, choose boats for the battle, control the boat and shoot other boats at the battle.
2. System: Is the system that interact and response to the player input.
3. Card: Is the marker or simply the control device that the player uses to control in the game.

**Brief Description Use Cases**

1. Use case Start Game: This starts the program.

2- Use case Initialization: This initializes the game, the computer creates the ground and creates the type of boat the Player chooses. You also initialize the specifics of the game.

3- Use case Ground: This is where the Players Boat appears.

4-Use case Boats: This is what the Player move in the game, it also has guns points to the enemy boats if the enemy is within its range.

5-Use case Game Results: This is where at the end of the game it shows the results of what occurred during the game. Such as who won and who lost.

6-Use case Move Config: When the Player moves the card it updates the result upon the Players movements.

7- Use case Battle: This is where the system will take the input and calculate who hit whom and with what force, movements based on the cards or makers.

1. **Specific requirements**
   1. **Use case scenarios**

|  |
| --- |
| **Use case:** Initialization |
| **ID: UC2** |
| **Actors:**  Player  System |
| **Preconditions:** |
| **Flow of Events:**   1. This use case begins when the Player runs the "Battle Boats" program. 2. The system begins the process of loading needed information to run the program. 3. The system loads modules needed to display the graphical user interface. 4. The system constructs all event handlers and adds them to the system. 5. The system loads all image and sound files. 6. The system displays graphical user interface. 7. The system displays startup screen and menu 8. Each player choose type of a boat   <initializeGround>  <initializeBoat(s)> |
| **Alternative Flow:**   1. The Player can leave the system at anytime once the menu has been displayed. 2. The Player can exit the game at any time. |
| **Postconditions:**   1. The application is initialized, players have been chosen, and game is ready to be played. |

|  |
| --- |
| **Use case: MoveCard** |
| **ID: UC3** |
| **Actors:**  Player |
| **Preconditions:** The program is running |
| **Flow of Events:**   1. The player choose the card. 2. The player moves the card in front of the camera. 3. The computer interacts with the movement of the card.   <initializeGround>  <initializeBoat(s)> |
| **Alternative Flow:**   1. The Player can choose different marker at anytime for different tasks. |
| **Postconditions:**   1. The computer updates the movements. |

|  |
| --- |
| **Extension use case: Ground** |
| **ID: UC4** |
| **Actors:**  Player |
| **Preconditions:**   1. The players has been chosen. |
| **Flow of Events:**  1-This use case starts after number of players has been chosen.  2-The system displays a Ground Initialization screen.  3-The system will display which player was chosen to play the Ground. |
| **Alternative Flow:**  n/a |
| **Postconditions:**   1. The Earth and its components have been initialized. |

|  |
| --- |
| **Extension use case: Boats** |
| **ID: UC4** |
| **Actors:**  Player  System |
| **Preconditions:**  Game Initialized |
| **Flow of Events:**  1-The system displays a Boat Initialization Screen.  2-The Player selects which sections of the boat. |
| **Postconditions:**   1. The Boats(s) have been initialized. |

|  |
| --- |
| **Use case: Game Results** |
| **ID: UC6** |
| **Actors:**  System |
| **Preconditions:**   1. One Player remains. |
| **Flow of Events:**  1-This use case begins when only one Player is left.  2-The system displays a screen congratulating the winner.  3-The system displays a screen with all Players scores.  4-The system displays "Game Over." |
| **Postconditions:**   1. Game Over. |

|  |
| --- |
| **Use case: MovConfig** |
| **ID: UC7** |
| **Actors:**  System |
| **Preconditions:**  The Players card is configured. |
| **Flow of Events:**  1-The use case begins when the player moves the card.   1. The system calculates the differences. |
| **Postconditions:**  1- The new position is been updated instantly. |

|  |
| --- |
| **Use case: Battle** |
| **ID: UC9** |
| **Actors:**  Player |
| **Preconditions:**   1. Game is initialized 2. Player choose a boat |
| **Flow of Events:**  1- The system implements the move requested by the Player in use case 8.   1. The system makes the determination about whether the shot was a hit or a miss using the predetermined range of the boat. 2. If the system determines the move was a hit   3.1 The system calculates the decrease in the life for the Player who has been hit.   1. The system records the information until each player has attacked. |
| **Postconditions:**  1- The life will be updated. |