Phase 1 Requirements Document: Battleship

Group Members (Salamander):
Mahim Chaudhary #301463607
Ryan Ly #301381311
Nicolas Ramirez #301379403
Sreeja Veeraghanta #301455569

Contact:
mca265@sfu.ca
rla168@sfu.ca
nra25@sfu.ca
sva36@sfu.ca

Submitted To:
CMPT 276 D100
Tao Wang
Simon Fraser University

Issue Date:
June 6, 2023

Table of Contents

Introduction

Our group is creating an online version of the board-game Battleship with many additional features, for example, action cards. The game will be implemented in Python and deployed online so players can connect to a game while also hiding the location of their fleet from their opponent.

In this version of Battleship, a player can make a choice between a two-player game or a game against the AI. The two-player game allows two human players to play against each other. One player would create the game which generates a code that the other player uses to connect. The AI will have three different difficulty levels, and each time a player beats a level they will be awarded with an achievement. The player that sinks all of the opponent's ships first will be declared the winner.

Rules

In an effort to increase the complexity of the game and make it more enjoyable for the player, the following additions to the rules have been made:

Item	Description
Win Condition:	All player's ships are destroyed
Action Cards:	To be drawn on wheel spin: - **Shields**: The user can block a missile from the enemy during the enemy's turn. - **Revelation card**: Opponent must disclose the location of one of their ships, or disclose a column or row on their board. - **Multishot**: The user can have 1-3 consecutive shots in the turn.

	- **Reinforcements**: Get a free ship. (Must be of a class of ships that has already been sunk.) - **+1 Card**: Fire & Draw again - ± 50% turn time

Table 1 - Rules

User Requirements

Story 1:
As a player, I want to be able to connect online to play against my friend quickly. I want the game to be fun, and more interesting than the original Battleship game. I want to have more strategy involved in the game. I want to be able to come back to a game and finish it later. I want to be able to play against different difficulty levels of a computer player.

Story 2:
As a player, I can create a private game room, so that my friend and I can play against each other without interruptions. Each time I create a game room, an alphanumeric code of length four is generated, and my friend can use this code to join my game. If I accidentally create too many game rooms, I need an easy way to navigate through them to delete the ones that I do not need.

Once a game is created, my friend and I are able to play it on two different computers through an online network. When playing the game, I want to be able to chat with my friend, and I want the chat box to be easy to locate and use. The game itself should be interesting with additional features like action cards which can be drawn to be used against your opponent.

Story 3:
As a user, I can choose to play the game with AI rather than another person. This computer player has three levels to play against; easy, medium or hard. No matter which computer level I win against, I want to be awarded an achievement for my victory. At the end of the game, I would like the winner's screen to show a big victory banner and a defeat banner on the loser's screen. When I go into my profile, I want to be able to see the achievements I have received. I would also like to see the number of games that I won displayed on my profile. When I play the game, there should be sound effects and animations that indicate whether I got a hit or a miss.

Story 4:
As a novice player, I can watch a tutorial and demo on how to navigate through the game and how to play. I want this tutorial to be accessible throughout the game, and not just in the lobby, in case I forget how to play. If I want to take a break from the game I can save the current game state as it is, and come back to it later. Furthermore, I want to be able to name the lobby so that if I have multiple active sessions, I know which one I need to go back to. When I load the saved game state it displays everything exactly as it was before.

System Requirements

The following is a list of functional and non-functional requirements for our system:

Functional Requirements:

Requirement ID	Requirement Description
Req. 1	System objects shall be able to load and save game state by recording the location of the ships of each player, the hand of cards they had, which player has the current turn and how much time is left from that turn.
Req. 2	System should include a Messaging service that allows sending strings from one user to the opponent.
Req. 3	System shall be capable of being uploaded to the web and played within a web browser.
Req. 4	System shall allow for two human players to play concurrently against each other.
Req. 5	System must be able to detect a hit or miss by comparing the user-entered target coordinates on the board and comparing that with the locations of the opponent ships and find if there is a match or no match.
Req. 6	If the system is able to detect hit or miss, it must also store the result of the attack (hit or miss) and indicate it on the board.
Req. 7	Where possible, the system should produce a sound and animation of the attack. If the attack hits, there should be a small explosion animation and sound. If the attack is missed, then there should be no sound.
Req. 8	The system shall prevent players from taking action when it is not their turn unless they have a “Blocker” card.
Req. 9	At the beginning of each turn, the system shall display to the user the wheel of cards and provide a button for the user to click and cause the wheel to turn. The wheel shall turn and land randomly on a card. This card will then be added to the user’s hand of cards and then the system will remove the wheel of cards and let the user continue their turn.
Req. 10	The system shall check after each turn if the user’s opponent has no ships left, thereby detecting that the win condition has been met and indicating to the winner that they have won.

Req. 11	Upon initiation of the game, the user shall be first presented with a main menu with three buttons; Join or start a new game lobby, play against the AI, or play a tutorial of the game to learn how to play.
Req. 12	The system shall be able to create at least one virtual lobby for players to join, hosted on the computer of the player who initiated the lobby. The lobby should have a unique access code that the user can share with their opponent for them to join the lobby.
Req. 13	If the user has chosen to play against the AI, the system shall present to the user 3 difficulty levels that they can choose between and will set the AI difficulty to the selection provided (clicked button).
Req. 14	The system should keep a live count of the score of each player, which is calculated as the $\frac{Number\ of\ landed\ attacks\ by\ the\ user}{Total\ Number\ of\ coordinates\ occupied\ by\ opponent\ ships.}$ and display it on the top right of the screen.
Req. 15	The system should detect if all opponent boats of a given kind have been destroyed and display an achievement award to the user, for each class of destroyed boats.
Req. 16	The system should also display background art behind the board during the course of the game.
Req. 17	Once the win condition has been met, the system shall prompt the user to replay the game or return to the main menu through two buttons shown on the screen.

Table 2 - Functional Requirements

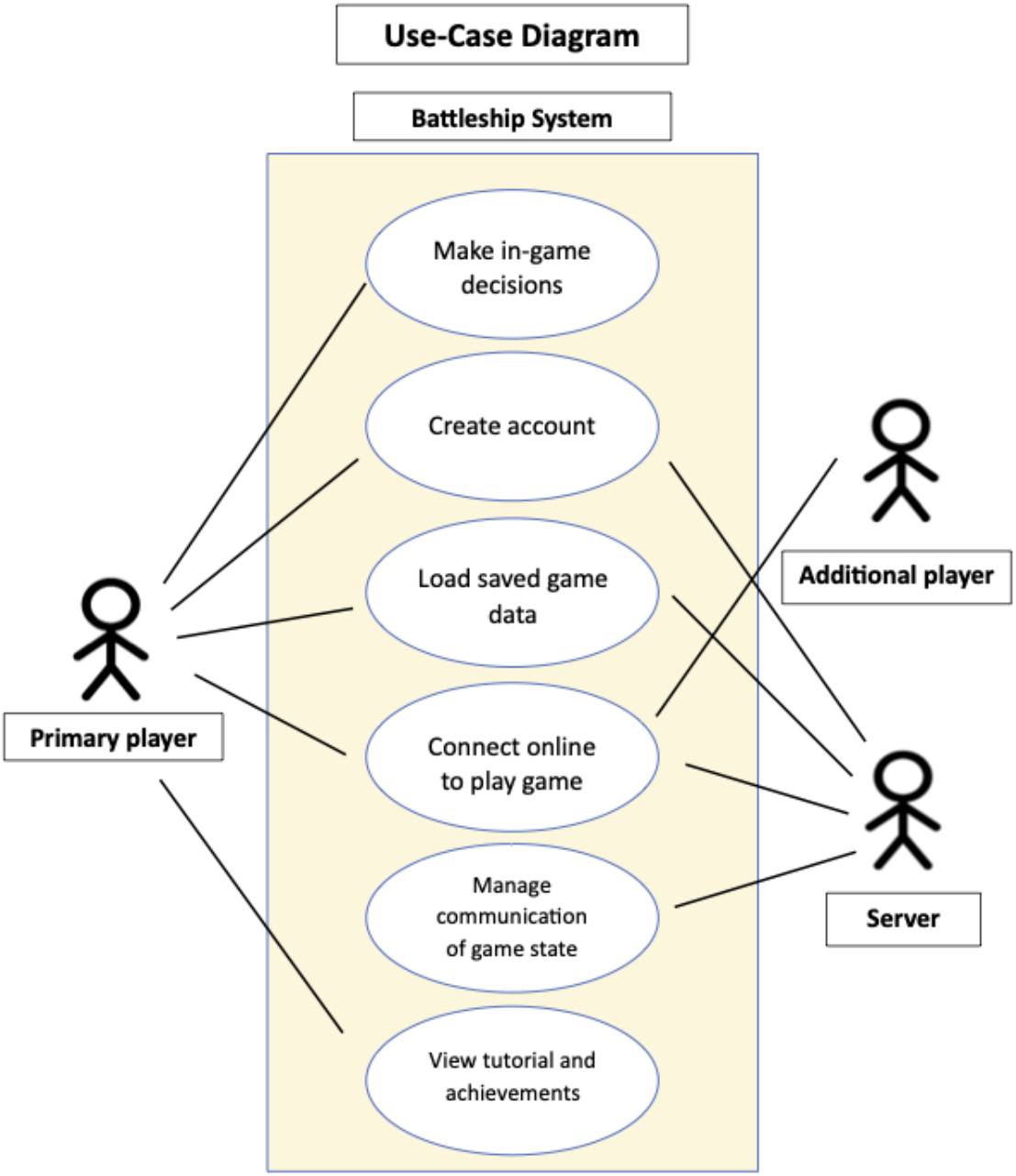
Non-functional:

Requirement ID	Requirement Description
Req. 1	The system shall be able to store user information safely by storing it locally on the computer of the user.
Req. 2	Given a latency between computers no greater than 80ms, there shall be a maximum 5-second delay between turns.
Req. 3	The state of the game shall be updated locally no longer than 50ms after input
Req. 4	The system should protect the user from unwanted users by authenticating the game lobby with a unique lobby code, and by presenting the option to accept or decline a new user from entering a lobby.
Req. 5	The system shall store user-created usernames and allow users to sign back into a game after an unexpected disconnection.
Req. 6	The user interface of the game should follow the design considerations presented in class, including color theory, gestalt principles, accessibility and customization, and present a small error message upon the detection of an error.

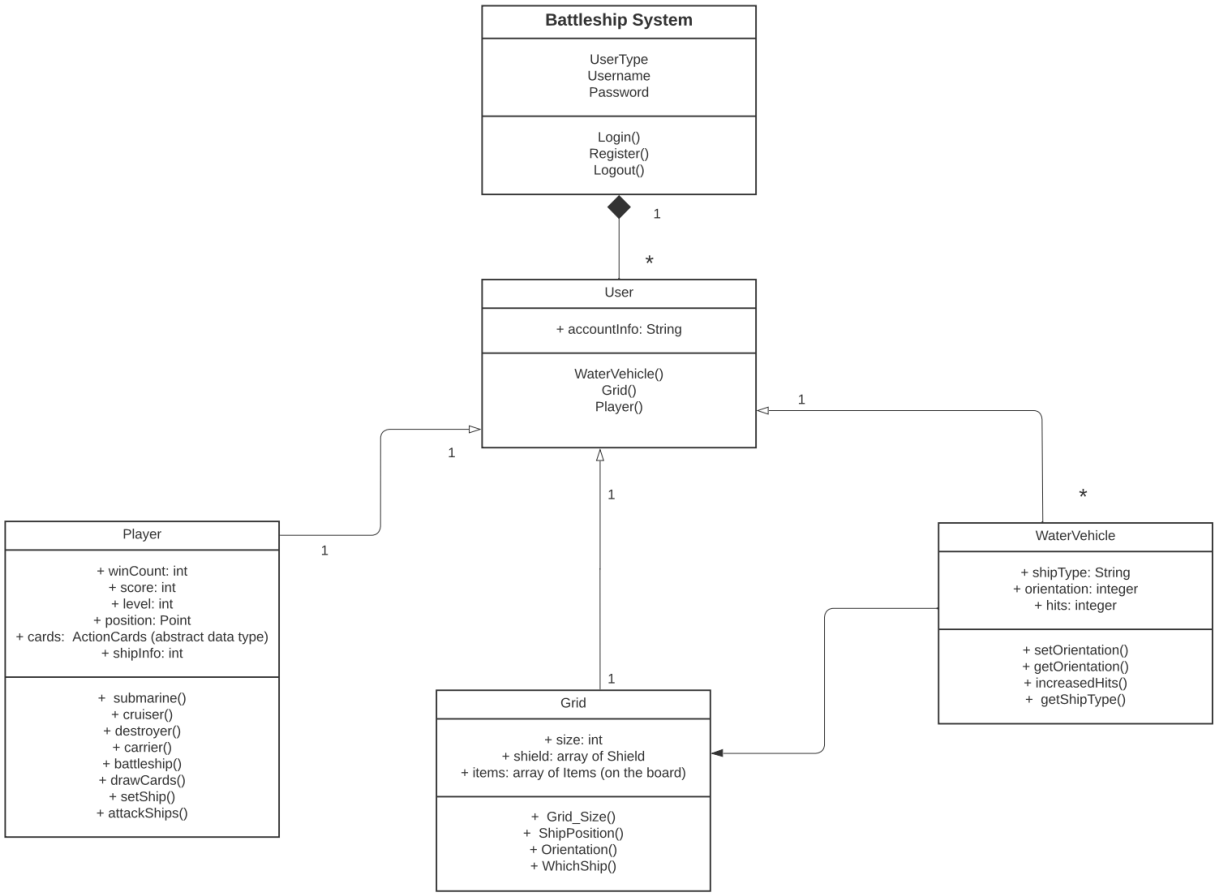
Table 3 - Non-Functional Requirements

System Architecture & Models

UML Use Case Diagram:



UML Class Diagram:



GitHub:

The repository corresponding to this project can be found here.
The professor and all Teaching assistants have been added to the repo.