Assignment 1.1: Game Proposal – Battleship

DYLAN BOYLING – ID: 041042484

Explanation

The purpose of this document is to define the elements and behavior of the GUI application to be used in the implementation of the game Battleship. This document also includes a mock-up for the game UI and it also outlines functionalities, behaviors, and use-cases.

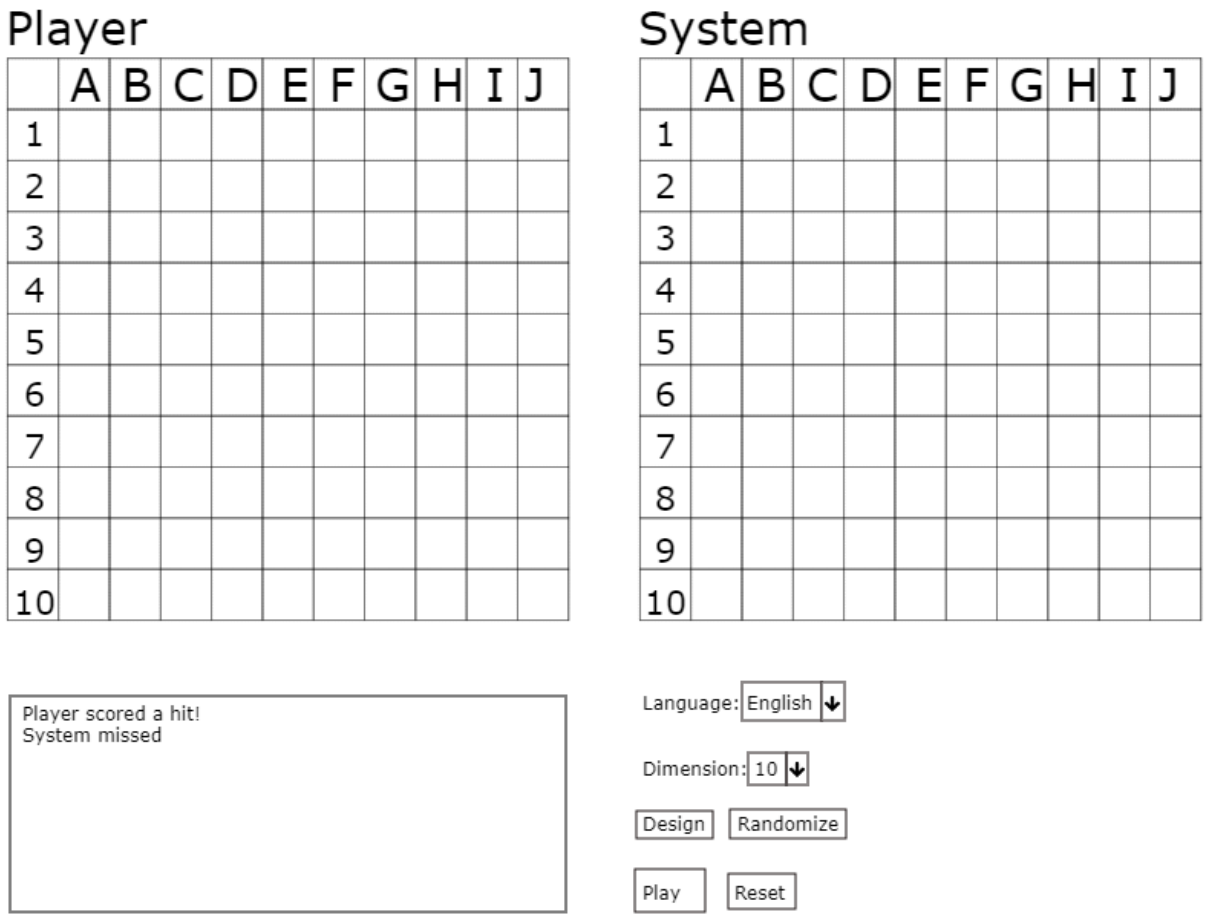

Figure 1 UI mockup for Battleship

Defining the Functionalities

Main Behavior

The functionalities of the game will be provided by Swing. Functionalities to be included will be selection boxes, buttons, a scrollable grid, time tracker, and a grid representing the gameboard which will allow highlighting of squares on the grid when the cursor is placed over them.

Functionalities and Behaviors

The game will be played by a human player who will play against the system. The player can place their ships and select locations on the grid for their guess on their turn but the system will randomize it's own

ship locations and guesses. Before placing the player's ships, they must have just started the application or clicked reset game. The precondition for starting the game is that the player must have either manually or randomly placed their ship locations onto the game board.

Once they have been placed, the player can click 'play' to start the game and then the time tracking for the current session will automatically start. When the game has started, the player can no longer modify their ship positions. Also, they can only change the dimensions of the board in the setup phase before the game has started but they can change the game's language at any time.

The 'reset' button can be clicked anytime to restart the game. Once a game has finished, the player will no longer be able to guess ship locations and the game will display a message indicating the results of the game.

Languages

English will be the first language supported for battleship as it is a mandatory requirement. The second language to be included will be French because although I unfortunately only speak English, French is the second official language in Canada and it makes sense to support it given that information.

Use-Case Diagram

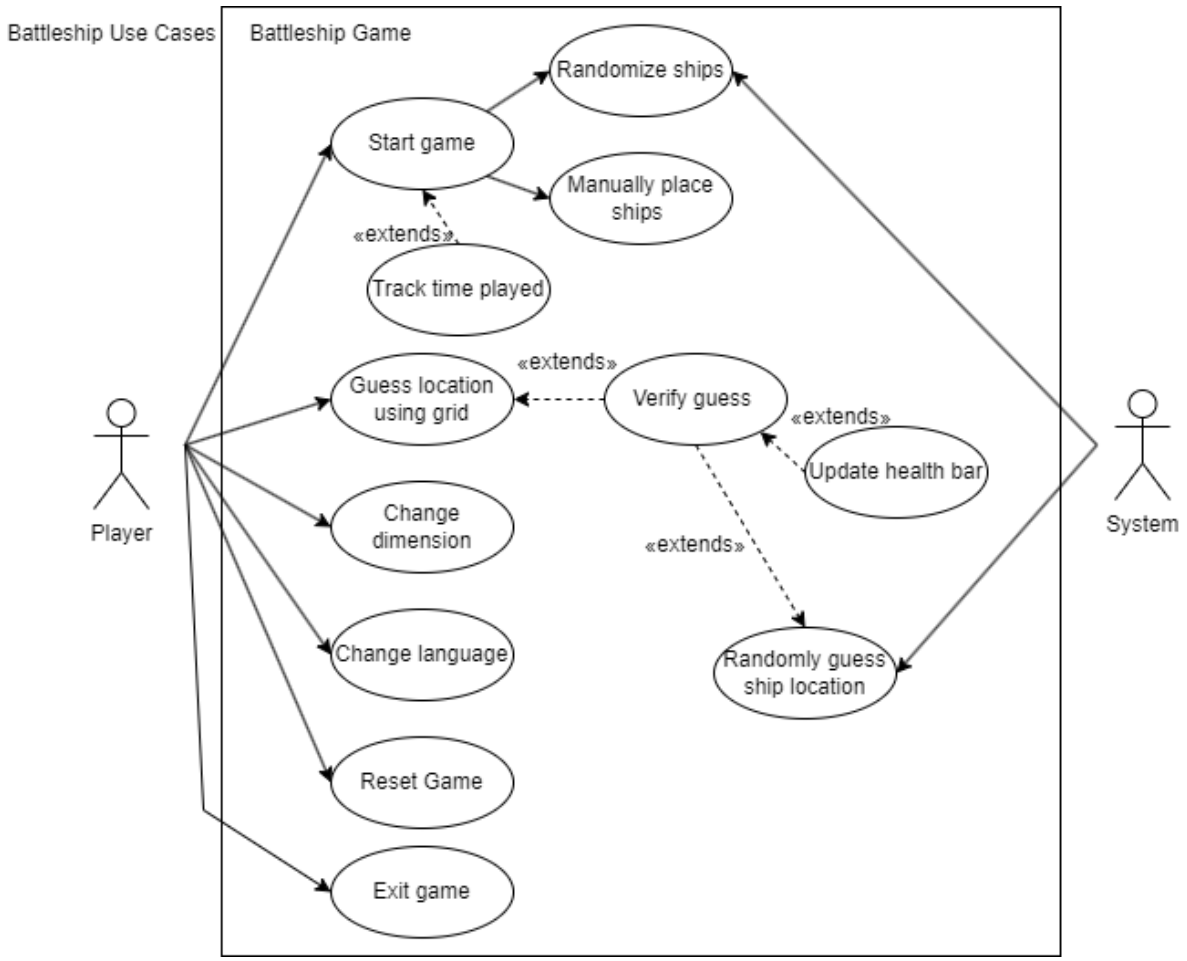

Figure 2 Use-Case diagram for Battleship

Table 1 Actors table

Actors	Explanation
Player	This actor represents the human player playing a game of battleship. They will pick their ship locations and guess the system's ship locations. They do not know the position of enemy ships.
System	This actor represents the system who will oppose the player in battleship. They will randomly guess the location of the player's ships and also randomize the player's ship locations when selected.

Table 2 Use-Case table

Use Cases	Explanation
Start game	Player begins the game of battleship after they have placed their ships. The game will automatically begin tracking the time played when the player is finished placing ships.
Randomize ships	Player can randomize the location of their ships if they want to. System has to randomize location of ships when beginning a game.
Manually place ships	Player can manually place the location of each ship onto the board.
Guess location using grid	Player guesses system ship location on grid. The player's guess will then be verified if it is a hit, or a miss and the system's health bar will be updated accordingly.
Randomly guess ship location	System will randomly guess a location on the grid where they think the player might have a ship. The system's guess will then be verified if it is a hit, or a miss and the player's health bar will be updated accordingly.
Change dimension	Player can change the dimension of the board before starting a game.
Change language	Player can change the language of the game's menus.
Reset game	Player can reset the state of the game and start a new one
Exit game	Player can exit the game's window and stop playing

Basic Cycle

When the user launches the application, they will be presented with the primary window for playing the game which contains the grid for the player and the opponent. They will click either 'Design' or 'Random' to manually place their ships on the board or let the system randomize their locations. Once they are satisfied and all their ships are placed, they can click start game which will also start tracking the time played. The player can guess where an enemy ship is, and the system will then randomly guess where a player's ship is. Upon a successful hit, the player's or system's health bar will be reduced. This process will repeat until either the player or system wins the game. At any point in the game, the player can choose to restart the game which will bring them back to the ship placement process.