Battleship Software Requirements Document

Team 42

Project Overview

Administrative

Instructor: Prof. David Johnson
GTA Supervisor: Tianxiao Zhang
Semester: Fall 2024
University: University of Kansas

Project Contributors:

- Manoj Turaga
- Connor Forristal
- Ceres Botkin
- Henry Marshall
- Elizabeth Chanel

For further information about this project's development, please contact the contributors listed above.

Game Rules

Project 1 is an implementation of the classic two-player game "Battleship" where players are tasked to destroy all enemy ships before their own ships are destroyed. The game shall take place on a 10 x 10 board with columns labeled with the letters A - J and the columns labeled 1 - 10.

Starting a Game

The application shall greet the user with the ability to select whether they will be hosting or joining a game. Upon the selection, the user shall be prompted to determine the number of ships placed on the board. The prompts shall remain until the user is able to make a valid selection.

Upon completion of "configuration" prompts, the user shall be greeted with a page that will allow them to place their ships on the board. The minimum number of ships that the user can place is 1 and the maximum number of ships that can be placed is 5. The user shall be prompted to to determine the starting point of a ship and what the orientation of the ship is. The ship's orientation shall be restricted to be the upwards direction, or in the rightwards directions.

Invalid ship placements shall be the following:

- Ships coordinates that exceed the bounds of the grid

- Ship coordinates that conflict with the placement of another ship.

If the user is the host user, the game shall wait until a client connects to the host. If a client attempts to join without a corresponding host game, the client shall immediately error out.

Gameplay

Upon the start of gameplay, if the user is hosting the game, then the user shall have the ability to to make the first move. Otherwise, the user shall wait for their opportunity to make a move until it is their turn.

The game shall continue in this fashion until all the ships that are placed on the board are destroyed for either opponent

Win Conditions

The winner status shall be granted to the player who is able to sink the opposing player's ships before one's own ships are sunk. Upon completion of the game, the game shall inform the player of their win status

Moves

When a player turn is initiated, the user shall have the ability to fire an attack on their opponent's board. The user shall specify both the column and the row of the attack.

The application shall force the user to select only valid coordinates. Valid coordinates shall be defined as follows:

- The coordinate is within the bounds of the board.
- The coordinate is targeting a location that has not been attacked before.

If the previous conditions fail, the user shall be forced to re-enter the coordinates until the attack can be successfully triggered.

Status

Whenever a player makes a move, the game shall provide the status of the move to the user. The following status shall be triggered:

- If the user attacks a coordinate that does not have a ship, the game shall inform both players that the attack at the specified coordinate missed.
- If the user attacks a coordinate that does have a ship, the game shall

inform both players that the attack at the specified coordinate hit a ship.

    - If the coordinate that was hit is the last available coordinate for a particular ship, both players shall be informed of the ship that was destroyed.

- If the user attempts to attack a coordinate that has already been hit, only the user shall be informed of an invalid attack

    - The opposing player shall wait for their turn until the current player is able to make an attack. The game shall inform the player that they are waiting for their opponent to make a move

Interface Requirements

The codebase shall implement a “plug and play” system where developers can create new components that should be easily integrated into the codebase. The codebase shall employ a Model - View - Presenter architecture. For any new additions, the developer shall ensure that the additions satisfy the following requirements

Model Interface

Any models shall provide the following functions:

- The model shall provide a means to get the state of a coordinate for a given player’s board.
- The model shall provide a means to update the state of a coordinate for a given player’s board.
- The model shall provide a means to determine if an action taken on a player’s board for a particular coordinate is valid.
- The model shall provide a means to determine if there are any ships still alive for a given player’s board.
- The model shall provide a means to determine if a particular ship is still alive in a given player’s board
- The model shall provide a means to get the internal representation of a given players board

View Interface

Any models shall provide the following functions:

- The view shall provide a means to draw the start page for the application.
- The view shall provide a means to prompt the initialization of the board.
- The view shall provide a means to allow the user to input the attack.
- The view shall provide a means to display a “wait for opponent” page.
- The view shall provide a means to draw a “game over” page.
- The view shall provide a means to draw a grid on the page.

Presenter Interface

Any presenters shall provide the following functions:

- The presenter shall provide a function to trigger events

Interaction Interface

Any interaction communications shall provide the following functions

- The interaction interface shall provide means to open a connection to another player.
- The interaction interface shall provide a means to close a connection.
- The interaction interface shall provide a means to send a message.
- The interaction interface shall provide a means to wait for a message from the opponent.

Model Requirements

Game Model

Data Requirements

The Game Model satisfies the following data requirements:

- The model shall store a board for the user, as well as their opponent

Behavior Requirements

The Game Model satisfies the following behavioral requirements:

Get Coordinate

- The model shall return the information from the board of the specified player.
- The model shall return the ID of the ship, and the state of the cell, at the indicated coordinate

Update Coordinate

- The model shall update the information in the board of the specified player.

- The model shall update the ID of the ship, and the state of the cell, at the indicated coordinate.

Is Valid Coordinate

- The model shall determine if a particular coordinate is valid for the specified player

- If the user is placing ships, a coordinate is valid if the coordinate is within the range of the board and no other ships have been placed at the given coordinate

- If the user is making an attack, a coordinate is valid if the coordinate has not already been attacked.

Are All Ships Alive

- The model shall if there are any ships alive for the specified player

- If there is a cell that has a ship and has not been attacked, the model shall indicate that at least one ship is alive.

- If the model cannot find any ships alive, the model shall indicate that there are no ships alive for the given player.

Is Ship Alive

- The model shall determine if a particular ship is alive for a given player

- If a particular ship still has a coordinate that is alive, the model shall indicate that the entire ship is still alive.

- If a particular ship has no coordinates that are alive, the model shall indicate that the entire ship is destroyed.

Get Visual Board

- The model shall return a visual representation of a given player's board.

- The model shall return an N x N grid with the following values at a given coordinate:

    - If a coordinate has been attacked and the attacked hit, the cell shall have a -1.

    - If a coordinate has been attacked and the attacked missed, the

cell shall have a -2.

- If the coordinate does not have any ships in it, the cell shall have a 0.
- If the coordinate does have a ship on it, the cell shall have the numeric id of the ship.

View Requirements

Data Requirements

The Game View will not store any data.

Behavior Requirements

The view implements the view interface by using the command line interface as the medium of visual representation.

Presenter Requirements

Game Presenter

Data Requirements

The Game Presenter satisfies the following data requirements:

- The presenter shall store an instance of the view.

Behavior Requirements

The Game Presenter satisfies the following behavioral requirements:

Trigger Event

- The presenter shall provide a means to trigger events.
- For the following events, the presenter shall trigger specific view related functionality to allow the fetching and publishing of data
    - Initialization
    - Placing Ships

- Making Attack
- Waiting for opponent
- Game End

Interaction Requirements

Data Requirements

The Game Interaction satisfies the following data requirements:

- If hosting, the interaction shall store the port and socket of the host. If joining, the interaction shall store the address of the host and the socket of the client.

Behavior Requirements

The client implements the interface specified by using TCP over a network connection.