Joseph DiPalma, Annan Miao, and Ben Xu

May 2, 2017

Professor Brian King

CSCI 205 Final Project

User Manual

Introduction:

Our project is a GUI based version of the board game Battleship. Our game allows 2 people to play together on a LAN. The main benefit is that it allows people to play Battleship without owning a physical version of the game. This is useful for people who enjoy this game but may not own it or are unable to find other people to play against. The overall goal of this project is to create an enjoyable game for people with a target audience of children.

Background:

We considered many different options for this project before deciding on the Battleship game. We always knew we wanted to do a game, but we also realized that we had a limited amount of time. This meant that the project had to be difficult but not too complex or else we would be unable to finish. We eventually decided upon Battleship while researching different types of board games. Battleship is easy to play but still was complex enough to need a month of work to complete. Additionally we had to consider that our team consisted of 3 people so we would each have to take on more work. So, we choose Battleship since it is a fun game and was also a feasible project for 3 people to create in a month.

Problem Domain:

The game of Battleship presented us with multiple problems to deal with. We needed to make an intuitive GUI representation of the game so that users could easily navigate the interface. This was critical because terrible GUI design would cause players

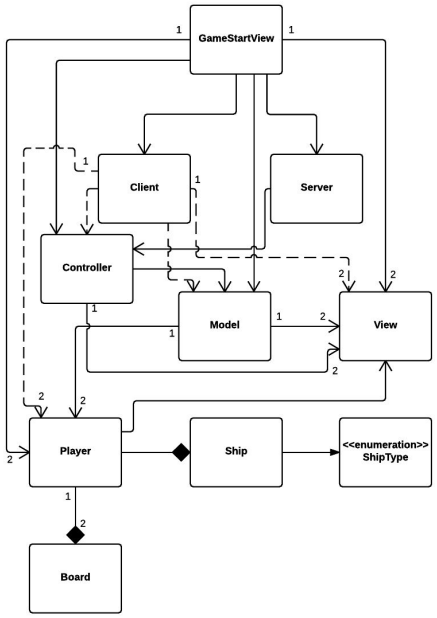
to stop playing the game. Additionally, we needed to establish network connections between 2 computers in order to implement the multiplayer functionality. In summary, we had to deal with the following problems:

1. How can we properly establish a network connection between 2 computers?
2. How can we create a GUI that is intuitive for the user but also has a lot of functionality?
3. How can we ensure smooth gameplay for both players?

Implementation:

We decided to break up the entire game into objects that are present in the physical game Battleship. Some of these objects are: Board, Player, and Ship. This made our design phase much easier and also allowed us to create good object-oriented code. Since we created the objects in the physical version of Battleship, we only had to deal with interactions in the same way as the board game. This made the game much easier to implement during the implementation phase of the project.

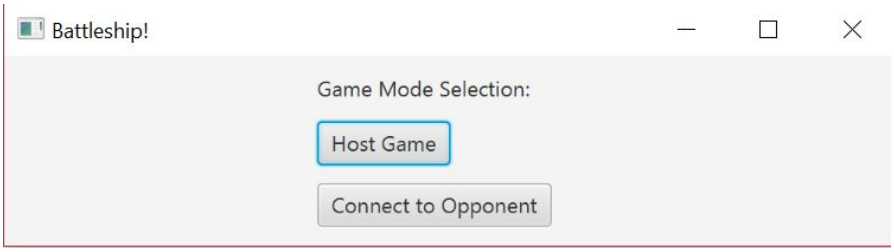
Basic UML Class Diagram:



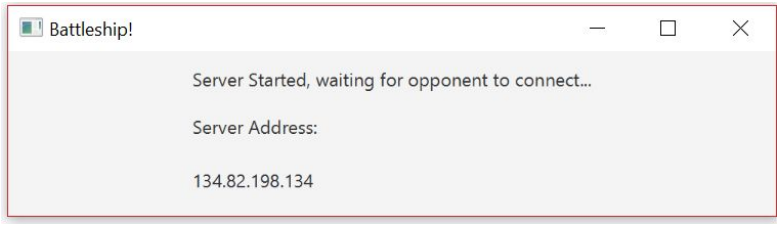
Each player has a board to show the ships as well as an array that stores the ships and their locations. Each ship is classified by an enumerated type representing the type of the ship. The interactions between these objects is encapsulated in the GUI which contains the Model, View, and Controller classes.

Instructions:

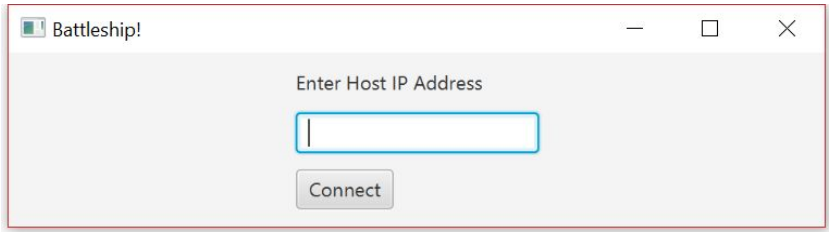
Let p_1 and p_2 be players. Each player can choose to host the game. Suppose p_1 is the host. To run the game follow the 6 stage process below:



1. p_1 needs to run the application "csci205finalproject.jar" and select the "Host Game" button. Then they need to tell p_2 the IP address of their computer that is shown in the window.



2. p_2 needs to run the application "csci205finalproject.jar" and select the "Connect To Opponent" button. Then p_2 needs to enter the IP address provided by p_1 and click the "Connect" button.



3. Once the connection is established between p_1 and p_2, they both need to place their ships on their boards.

4. Once each player is done selecting their ship locations, click the "Confirm Ship Selection" button.
5. After both p_1 and p_2 have clicked the "Confirm Ship Selection" button, the game starts. The host will always go first. So, p_1 would go first in this case.
6. The game ends when one of the players has no ships remaining.
7. If the players want to play again then they have to repeat this process.

Below is a picture of the GUI for our project: