CS 342 – Software Design   Project 4   Fall 2017

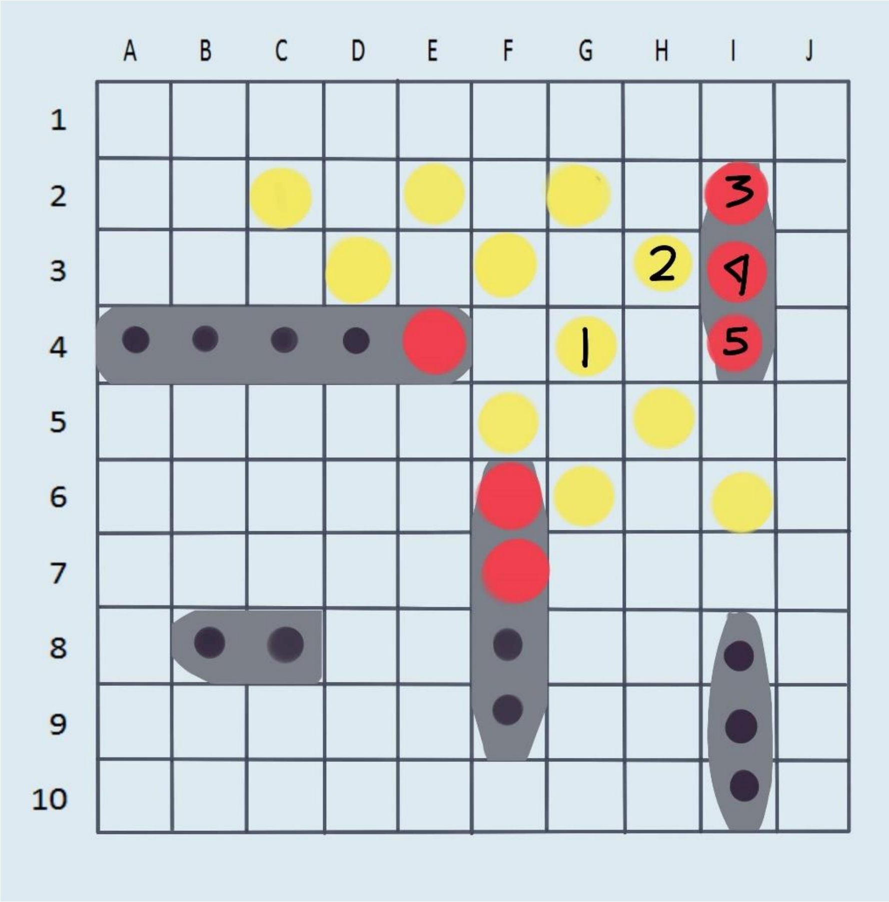
Networked Battleship

Due: November 16, 2017 at 11:59 pm

Group Selection Deadline: Monday, 11/6/17 at 11:59pm

For this project, all CS undergrad students must work in teams of 2 or 3 students. All members of the each team must be currently taking CS 342. Students will be allowed to select team members themselves. Students in the class who are not CS undergrad students may work on their own or in a group of 2 or 3 students. Groups must be selected and reported in Piazza by Monday, 11/6/17 before 11:59pm. The reporting in Piazza must be done as a REPLY to the instructor's original post for Program 4 Group Selection. Students may not work with any of the members of their group for Project 3 (i.e. group membership MUST change).

For this project, you are to use the Java Swing Libraries to create a socket based two player version of the game Battleship. If you are unfamiliar with the game, see the Wikipedia page.



The game is played on four square grids, two for each player. The grids are to be 10x10 square and the individual squares in the grid are identified by letter and number. On one grid the player arranges ships and records the shots by the opponent. On the other grid the player records their own shots.

Before play begins, each player arranges a number of ships secretly on the grid for that player. Each ship occupies a number of consecutive squares on the grid, arranged either horizontally or vertically. The number of squares for each ship is determined by the type of the ship. The ships cannot overlap (i.e., at most one ship can occupy any given square in the grid) nor can a ship go outside of the grid. The types and numbers of ships allowed are the same for each player. Note, the design of this part of the project is completely up to the programmer.

We will use the following ships:

Type of Ship	Size
Aircraft Carrier	5
Battleship	4
Destroyer	3
Submarine	3
Patrol Boat	2

After the ships have been positioned, the game proceeds in a series of rounds. In each round, each player has a turn. During a turn, the player selects a target square in the opponents' grid which is to be shot at. If a ship occupies the squares, then it takes a hit. When all of the squares of a ship have been hit, the ship is sunk. If at the end of a round all of one player's ships have been sunk, the game ends and the other player wins.

The following images can be used for this program. You can create your own images if you would like. I am not sure how image batt101.gif is supposed to be used. Nor do I know why you need multiple "midship" images for the unhit ships. These images were originally acquired from http://javascript.internet.com/games/battleship.html, but this page no longer exists.

Water Images

batt100.gif

batt101.gif

batt102.gif

batt103.gif

Hit Ships

batt201.gif

batt202.gif

batt203.gif

batt204.gif

batt205.gif

batt206.gif

Unhit Ships

batt1.gif

batt2.gif

batt3.gif

batt4.gif

batt5.gif

batt6.gif

batt7.gif

batt8.gif

batt9.gif

batt10.gif

Note that the Hit Ships Images (batt201.gif – batt206.gif) are displayed in the grid showing the player's boats when a boat is hit. The Water Image, batt103.gif is displayed in the grid showing the opponents area when a boat is hit.

Your program is to first allow the user to establish a connection with another person. One person is be the "server" while the other person is to be the "client". **The same program must be used for the server and the client.** So your program will have to switch from “server” mode to “client” mode while the connection is being established. Once a connection is established, the players are allowed to place their ships on a grid. Once the ships have been placed, the guessing of locations begins. When a game is over, ask the players if they want to play another game. If they do, allow the users to place their ships on the grid. Otherwise, break the connection between the client and server.

The program us to have menu bars that will allow the user to:

- exit the program,
- view an about box,
- view a help box (while only one is required, it is suggested that you have two - one for help on the connection side of the program and one for the battleship game play/rules),
- establish the connection between the two players (this may require multiple menu items), and
- a simple statistics box.

The about box must clearly list the names of NetId’s of all members of your team. For the statistics box, the exact statistics are left up to you but they could include the percent of hits to misses, total number of shots needed to finish the game, etc.

A “status bar” is to be used to inform the user of the various game “states”, such as:

- when to connect to server (or to become the server)
- when the server is waiting for the client to connect
- when to place ships on the player’s grid
- when it is his/her turn to fire at the opponent
- when a hit or miss has occurred
- when the game is over

Your program must be written in good programming style. This includes (but is not limited to):

- meaningful identifier names,
- use of multiple source code files
- a file header at the beginning of each source code file ,
- a method header at the beginning of each method,
- proper use of blank lines and indentation to aide in the reading of your code,
- explanatory "value-added" in-line comments, etc.

Submission of the Project

You are to submit your program via GitHub following the directions provided by the TA. Thus you are required to make sure your program is accessible via the GitHub interface in Eclipse.