CS 340 - Project 5

Battleship

Due: May 2, 2013 at 11:59 pm

For this project, you are to use the Qt Libraries to create a socket based two player version of the game Battleship

You may work in a group of two students for either this project. If you are working with a partner, only one of you needs to submit the project code, but both people must be clearly identified in the README file, header comments of the files and in the About Box for the project.

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 announces 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 muliple "midship" images for the unhit ships. The images are also in a zip file at battleship.zip. These images were originally acquired from http://javascript.internet.com/games/battleship.html

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

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. 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 "side"),
- establish the connection between the two players (this may require multiple menu items), and
- a simple 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.

This program will also require a 1-2 page write up of the data structures used in the program and the logical division of your program into multiple source code files (i.e. which routines are where). Remember that this write-up is to be written in ASCII format and is to be electronically turned in with your program. The name of this file should be "readme.txt".

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

- meaningful identifier names,
- a file header at the beginning of each source code file,
- a function header at the beginning of each function or 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 this project using the Assignment Link for Project 5 in the Blackboard site for the course. If you are working with a partner, only one of you needs to submit the project code, but both people must be clearly identified in the README file, header comments of the files and in the About Box for the project. Be sure to submit all source code, header files, makefile as well as your program description. Failure to turnin all required pieces will result in a lower grade for the assignment.

For the Qt Files, it is probably easiest to create a .zip file with all of the needed files and then submit the .zip file.