CS162 - Assignment 1: Battleship (Part 1)

This assignment is intended to give you practice implementing a Java class.
This assignment is worth 100 points. The due date is 12 August, 11:59 pm.

Assignment Overview

In this assignment, you will be implementing a single class which represents a board in the Battleship game (if you don't know what the official game of Battleship is, read about it on the Wiki page here).

This class is intended to encapsulate the logical idea of a Battleship board and it stores information about the locations of the battleships, along with other behaviors (ie. public functions) that will be discussed in detail below. This BattleshipBoard class is only a part of the Battleship game. You will **not** be implementing the entire game; instead, you are implementing a single class. In a future assignment, you will be building the other parts of the Battleship game, such as the graphical user interface.

We will provide a stub of the BattleshipBoard class. Your job is to finish implementing the class by filling in the contents of the public functions in the stub. You may add further member variables and methods, but **do not change the definitions of the public functions in the BattleshipBoard.java file**. Remember to follow Java coding conventions.

You only need to fill in the functions in the BattleshipBoard class. In future assignments, you will be building the rest of the Battleship game.

Specifically, you will complete the following functions:

1. **public BattleshipBoard(int numRows, int numCols)**
   This is the constructor for the BattleshipBoard class. It requires you to specify the number of rows and number of columns on the board.
2. **public int getNumRows()**
   This function returns the number of rows on the board.
3. **public int getNumCols()**
   This function returns the number of columns on the board.
4. **public void placeShip(int startCol, int startRow, int endCol, int endRow) throws Exception**
   This function places the battleship starting at position (startCol,startRow) and ending at position (endCol,endRow) inclusive. For example, if you place a ship at

(startCol,startRow) = (2,3) to (endCol, endRow) = (4,3), the ship will be at (2,3), (3,3) and (4,3). You should throw an exception if:
   - Any of the coordinates are out of bounds. A coordinate is in bounds if 0 <= col < (number of columns - 1) and 0 <= row < (number of rows - 1).
   - startCol > endCol or startRow > endRow
   - The ship is placed diagonally. You are only allowed to place battleships vertically or horizontally.
   - The ship cannot be placed successfully because it intersects a board cell that is part of a battleship that has already been placed.
5. **public boolean fireShot(int col, int row) throws Exception**
   This function fires a shot at a Battleship located at position (col,row). It returns true if you hit a battleship and false otherwise. The function throws an Exception if the coordinates are out of bounds. You are allowed to fire more than once at the same spot. For each time you fire at the same spot, your code should return the appropriate value for that spot (If there is no battleship at that spot, you should return false each time. If there is a battleship at that spot, return true each time).

Please read the comments above each public function and make sure your implementation satisfies the requirements in the comments.

If you would like to test that your code works, you should write a main function and write code to instantiate a BattleshipBoard and execute the functions you implemented. We do not require you to write a main function and we will not grade the main function (the main function is for your own testing purposes).

Bonus

For up to 10 bonus marks, add the following:

1. **public int getNumBattleshipsLeft()**
   This function keeps track of the number of battleships that are still standing.
2. **public boolean isGameOver()**
   This function returns true if the game is over, and false otherwise
3. You will need to update the fireShot() function so that it decrements the number of battleships once a battleship has been sunk.

There is a fair bit of bookkeeping you will need to do to get the bonus to work.

Files you will need (available on blackboard until the website gets put up)

- BattleshipBoard.java

What to Turn In

Please hand in the .java files in your assignment. You probably will only have one file -- the BattleshipBoard.java file. Please turn them in via Blackboard.

Grading Scheme

We will grade you on the correctness of the public functions in the BattleshipBoard.java class.