ITEC 120: Principles of Computer Science I

Homework 8 – Battleship simulator

>>>>>>>> Due Date: Friday, March 23rd at 10:00 PM via Desire2Learn <<<<<<<

In doing this homework, remember to abide by the RU Honor Code.

Problem 1

30 points

Board Games Inc. has hired you to create a simple command line battleship simulator so people can experience how old computers games were played before modern operating systems were created. Your boss wants an interactive command line simulator that allows users to create a two dimensional grid of characters of varying size. The grid starts with every character being W which represents water. Users can then choose to be red or blue. Users then place ships on the two dimensional grid either horizontally or vertically. Ships always cover two grid units (and represented with R or B for red or blue), and cannot be placed next to each other. I.E. water or the array boundary always surrounds a ship. Once ships have been placed, players can shoot the ships on the board. Shooting involves targeting one spot in the two dimensional grid. If that place is a ship, then it is changed to represent it being shot (F for red ships, H for blue ships). If both elements in the grid representing a ship are shot, it is sunk and one point is given to the person who fired. The goal of the game is to sink the most ships.

Input

Your program will be given a series of commands to manipulate the grid representing the playing field. The commands are as follows:

Command	Description
create	Creates a two dimensional grid of characters of the requested size.
shoot	Shoots a particular target on the grid and potentially sinking a ship.
print	Prints out the grid showing ships and water.
hship	Place a ship horizontally on the screen
vship	Place a ship vertically on the screen.
side	Change whether red or blue ships are to be placed.
score	Print out each side's score.
help	Prints out all of the commands that the program can accept.
quit	Stops the simulator.

Computation

All of your commands should ensure the array representing the spreadsheet exists if applicable, and ensure the values being accessed/set are within the range of the created array.

For the create command, your program will create a two dimensional array of characters of a size specified by the user. Its values are initialized to W. The created array will then be used with the other commands the simulator accepts.

For the shoot command, your program will ask the user for the position in the array to shoot, then will update a hit on a ship, or will report that a ship is sunk.

For the print command, your program will print out the contents of the spreadsheet in a grid like format.

For the shipv and shiph commands, your program will place a red or blue ship represented by the characters R or B at a specific location in the 2D grid.

For the score command, your program will print out the number of ship sunk by each side.

For the side command, prompt the user for side 1 (red) or side 2 (blue).

For the quit command, your program will stop.

For the help command, your program will print out the commands that the simulator will accept.

Output

Make sure your program outputs the proper prompts for the user so they know what information is being requested. Also, make sure the output of the calculations is easily identifiable.

Sample input:

```
create
10
10
vship
1
1
hship
1
3
side
2
vship
8
1
hship
8
3
print
```

```
shoot
1
3
shoot
8
1
shoot
1
4
shoot
9
1
print
score
```

Output from previous commands <omits input prompts>:

```
Blue sunk a ship
Red sunk a ship
W W W W W W W W W W
W R W F F W W W W W
W R W W W W W W W W
W W W W W W W W W W
W W W W W W W W W W
W W W W W W W W W W
W W W W W W W W W W
W W W W W W W W W W
W H W B B W W W W W
W H W W W W W W W W
Red has sunk 1 ship(s).
Blue has sunk 1 ship(s).
```

Testing note: your program must allow for a file to be redirected as input to the program (java programName < fileName.txt). If it does not, then you will receive a 0 on this program. As long as you use one scanner and pass it around as a parameter, this will not be a problem.

Constraints:

You must use at least two functions in your solution. You must also submit a test case that is similar but not identical to the previous usage scenario. By doing so, you demonstrate that you have tested your program beyond the basics.

No fields, instance, or class variables may be used in this program. If you don't know what they are, don't worry about it. The purpose for this rule is to make sure you must pass information through the program with parameters.

The reference solution for this project is 153 lines of code without comments. Feel free to use less or more code in your program.

Submission requirements:

You must submit the .java file containing your program to Desire 2 Learn under the Homework #8 assignment. If your submitted file does not compile, it will receive a 0. You can demo the homework the next school day after it is due, or it will be graded automatically. If you choose not to demo your homework you cannot contest the grade you receive.

Grading Rubric

12 Points – Correctly detects when ships are sunk.
8 Points – Places ships properly on the 2D grid.
5 Points - Is the program commented using inline and javadoc comments?
3 Points – Was a suitable test case submitted with the program?
2 Points - Does it use 2D arrays properly (are they initialized properly)?