CS 120: Introduction to Computer Programming II

Battleship Short (Design)

due at 5pm, **Tue** 30 Nov 2021

1 Overview

This assignment requires you to work on the design for the Battleship program. Your TA will give you feedback on your design - letting you know if they see any noteworthy issues that you need to address. For instance, you may not have carefully thought through the algorithm for a given operation, or you may not have detailed how the classes will interact.

You will not be graded on whether your design is "right" or "wrong" - it's a preliminary design, after all. Instead, you will be graded on your **engagement and effort** - so don't slack off!

1.1 Not Binding

Although you are required to think carefully during the Design phase, please note that your Design is **not binding.** Feel free to change it when you implement your code. In fact, I think it's **likely** that you will find some issues that you need to change!

1.2 Document Format

You should turn in a PDF file. However, we don't have any particular requirements about how you format the text inside it. Some students may prefer to write paragraphs; some students may use bulleted lists. Some students may find it useful to include diagrams; you have freedom. Just make it **clear** and make sure that you cover all of the issues listed below.

2 Key Elements to Discuss

The rubric is broken into four categories. Make a point of carefully addressing each one:

2.1 Category 1: Class Design

This project requires that you build at at least two classes: `Board` and `Ship` (you are allowed to create more if you wish). You should list out the methods of each one; while you don't have to mention all of the helper functions you might use, list any major ones that you can forsee.

In addition to methods, do your classes have any public data fields? (You will, of course, have private data fields - but we'll discuss that a little later.)

Finally, make a point of discussing how the two classes interact. **This is one of the mst challenging parts of this project, and so it's worth careful attention.** What will your Board need to know about the Ships? And what will the Ships need to know about the board? Make a point of listing any **methods or data fields** that the two classes use when they are interacting with each other.

2.2 Category 2: Data Design

Your classes will have to store data about the world. How will your Board class be implemented? How can it tell the difference between spaces that contain ships, and spaces that do not? How can it tell which ones have been shot at, and which ones have not?

Likewise, how will your Ship store its internal state? Handling straight ships (like the standard ones used in the real game) are fairly easy to implement, but this program doesn't assume that all ships are straight. How would you store the shape of a complex ship, like thes shown in the Battleship Code spec?

Take a moment, in this section, to re-consider your design from Category 1 above: does your code have enough information to handle a situation when a ship gets hit - will your Board have enough information to update the correct Ship object?

2.3 Category 3: Printing the Board

How will you print out the board? Will you have all the information you need in the Board class, or will it need to also get some information from the Ship objects?

If the board is larger than 10 spaces, meaning that you must support two-digit numbers, how would you modify your print logic to make it work?

2.4 Category 4: `rotate()`

How will you implement the rotation of a ship? A good trick is to look at the pictures in the Battleship Code spec, and compare the `rot=0` case with the `rot=1` case. When you move in the +x direction in `rot=0`, how must you move in the `rot=1` case? How about when you move in the +y direction?

Also, can you come up with a simple strategy for handling larger rotations? Is there a way to calculate the new locations, without having to implement three different types of rotation?

Finally, think carefully about how you implement the "shape" array. It's quite possible that there might be multiple Ship objects, which use the same shape array. Yet if we only rotate one of those ships, we don't want to modify all of the ships. What does this tell us about the shape array, when you rotate?

3 What You Must Turn In

You must turn in a PDF to GradeScope. Other file formats will not be accepted.