

## Battleship Design

### Category 1:

Class **Board** will have the following methods:

A method for printing the board

A method to receive ship coordinates and add it to the board

A method to check if the place shot at has already been shot before

A method to make sure that the user is shooting at the correct and available coordinates

Class **Ship** will have the following methods:

A method to create a ship

A method to set a coordinate for a ship

A method to rotate a ship by 90 degrees clockwise however many times possible

A method to keep track of the state of the ship – if it is damaged or not, and if it has sunk or not

To be honest, for now, I do not have any idea what kind of public and private fields I am going to use.

I think the way both classes will interact is as follows:

Board will receive a ship's shape and coordinates and it will print it on the board on the given coordinates. The user makes some interactions with the board (shooting) and if they can hit it or make it sink, the board will report this information to the class Ship and it will make changes to it accordingly. The interaction will probably include the methods: receiving the ship coordinates, sending the ship state from Board to Ship, keeping track of the ship state.

### Category 2:

My Board class will have a list in which each element is another list with 10 elements in it, and each element will be either a simple dot (.) or a letter expressing the ship. If there is a dot, it means the place is empty. If it has a letter, it has a ship, and once the user interaction starts, there may be an asterisk (\*) to indicate the damaged part of the ship, a cross (x) to show that the ship has sunk, or a lowercase o to indicate that the user missed the target.

My Ship class will hold string variables for each ship and the length of the ship will determine the length the variable itself. The ship will be displayed the same as its string name and any change in the ship will be reflected in its string variable name. The variable will be a reference to a list which holds the coordinates of each letter of its string name in a double tuple form (x, y).

### **Category 3:**

I will print my board using the 2D list that I will create in my Board class – I have already mentioned above what kind of 2D list it is and its content. I think, from time-to-time, Board and Ship classes will have to communicate with each other in order to submit the updates on the states of the ship on the board. Board class will mostly have all the information needed to print the board. The only time it needs to communicate with Ship class will be when delivering status update on the states of the ship. In the project overview video, Russ did not mention anything about the size of the board being larger than 10x10, but even if it is, I do not think it will be much of a problem. I guess it will have to do with changing the size of my 2D list.

### **Category 4:**

From what I understood, rotate() turns the ship by 90 degrees clockwise any time it is run. Basically, what happens is that the head of the ship does not ever change its coordinate, but its joint parts change coordinates according to how many times rotate() is run. The method rotate() functions as follows: it identifies in what position it is and where the head of the ship is, from there it makes 90-degree clockwise turn and depending on its position (horizontal or vertical), it changes either its x or y coordinates. I think it is possible to come up with a formula to rotate the ships and change their coordinates easily. We just need to first identify its position and the head. Though the shape array may contain same length ships with same position alignments, they will have different coordinates, so I think maybe the elements in the shape array will be dictionaries, the keys of which will be ship names and the values being the coordinates.