

Single Player Battleship Game
David Russell
5/8/20

Description: This program runs a single player version of the classic tabletop game battleship. The player faces off against the computer to see which one can sink the other's 5 ships first. The player's ship coordinates and orientation are first read in from an external csv file and then the computer's ship positions are randomly generated. The player and computer take turns guessing a coordinate which may contain a ship.

Input Requirements:

Player's ships: .csv file. Five rows with 3 fields in each row. The first field is the name of the ship (data type: string) (valid range: Carrier, Battleship, Cruiser, Submarine, or Destroyer), the second is its starting coordinate (data type: string) (valid range: A1 - J10), and the third is its orientation (char)(valid range: V or H).

Player attack guess: Keyboard input. First input (data type: char) (valid range: A - J). Second input (data type: int) (valid range: 1 - 10).

Player quit: Keyboard input. (data type char) (valid range: Q) This input allows the user to quit the game at any time.

Output Requirements:

Boards to console. The player's ship board and attack board will be output to the console each turn. They will be output as a 10 x 10 grid with letters A - J on the x-axis along the top and 1 - 10 on the y-axis on the left. Each successful "hit" from a torpedo will be marked as an O in a box and an unsuccessful "hit" will be marked as an X. On the ship board, each of the player's ships will be shown on the board with a corresponding letter in each space it occupies (Carrier: C, Battleship: B, Cruiser: R, Submarine: S, Destroyer: D).

Prompt for user Input: Each turn a prompt for the player to input a guess will be output to the console. This will be done in the format "First coordinate:" (data type char) (valid range: A - J) and "Second coordinate:" (data type int) (valid range: 1 - 10).

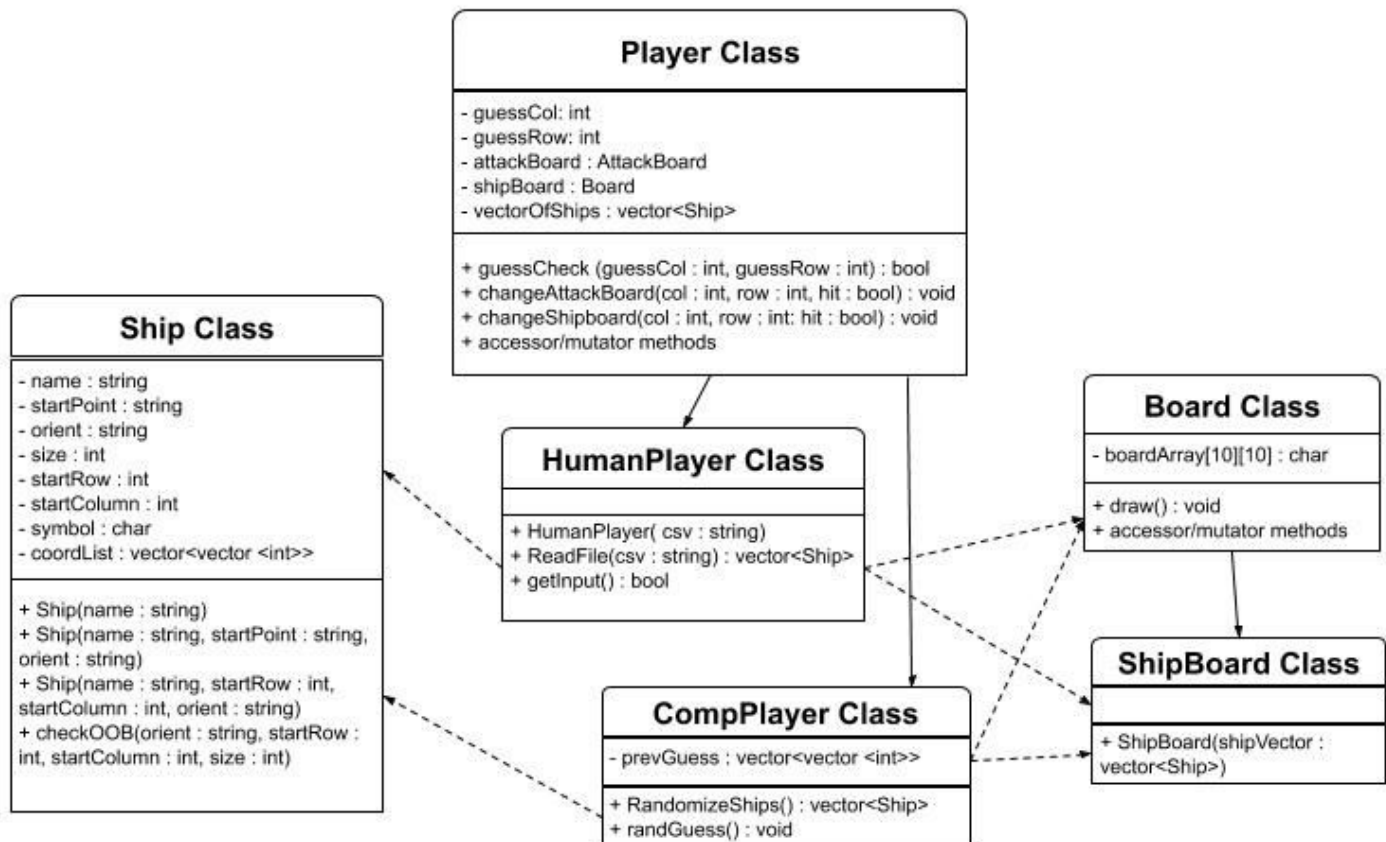
Computer guess: The computer's guess will be output to the console in the format of "A5", with the first coordinate being a (data type char) (valid range: A - J) and the second being a (data type int) (valid range: 1 - 10).

Problem Solution Discussion:

The key algorithms for this program are within the ReadFile() function and the RandomizeShips() function. The ReadFile() function will evaluate if the input file matches the correct format. After reading in and parsing the information on each ship, this function calls the ship constructor and stores the ship object in a vector which it returns. If it does not match the correct number of rows, columns, and ship names then it will throw an exception. The RandomizeShips() function contains a random number generator and an algorithm which iterates through each ship's coordinate points to look for overlapping points. If there are no overlapping points, and the ship's starting point, size and orientation do not cause it to go off the

board then a ship object is instantiated and stored in a vector. This function is called from within the CompPlayer constructor.

Classes, Inheritance:



Overall Software Architecture:

First, the player's ship information will be read from the .csv into the program using the `ReadFile()` function. If the file does not match the correct format then the program will throw an exception and ask the user to correct the file. Once a valid file is read into the program, Ship class objects will be constructed from the data. At this point, if any of the ships have incorrect placement (out of bounds or overlapping another ship) the program will throw an exception asking the user to correct the file. Once the 5 Ship class objects are created, a Board class object will be created from an array of the Ship objects as the player's ship board. A player attack board will also be created.

The program will now use the default Ship constructor to randomly create 5 ship objects for the computer using the `Ship()` default constructor. These will be used to instantiate a Board class object. An attack board will then be instantiated for the computer.

Once the 4 game boards are created, the game loop will begin. The game loop will continue until either all 5 of one player's ships are sunk or the player chooses to quit. It will

begin by asking the player to input a coordinate to torpedo, one coordinate at a time. If it is a hit, an O will appear on the attack board, a miss will appear as an X. Then, the computer's coordinate guess is printed to the screen. A hit will be recorded on the player's ship board with an O, a miss with an X.

Current Status

The program runs successfully. The program was developed and tested on Visual Studio version 16.2.3. It was compiled, run, and tested on gcc version 6.3.1 on csegrid.cudenver.pvt.