***Battleship Game – Assignment 1***

This program simulates a battleship game in 3 different search strategies. I have implemented 8 classes and one interface in order to create this program.

**Games Class:** This class contains the methods that simulate a battleship game. The class takes a text file and set the points (x, y) on the board. It sets a total of 3 game rounds where in each round, a new search strategy runs.

**Design Decision:**

I decided to create each game at this class where the board is set, the file that contains the shots is read and the main function executes all 3 games.

**Battleship Search Class:** This class contains the methods that stores the results and print the game scores: the total number of shots, the coordinates where the ships are located on the board and the type of strategy.

**Design Decision:**

I decided to set 3 various search strategies:

1. Diagonally Search Strategy – search for ships diagonally on the board with the coordinates 0, 0.
2. Horizontal Sweep Strategy – search for ships horizontally on the board with the coordinates 0, 0.
3. Random Search Strategy – search for ships randomly on the board.

After running all 3 strategies, I have noticed that the Random Strategy is found to be the most inefficient as it can take a long time to perform a simulation. The main reason is that it may hit the same point more than once.

The rest of the classes are standard and implement methods that the interfaces provide. I have handled expected errors in case of an overlap or reading the output file.

Note about Diagonally Search Strategy: due to grid points that might be shot diagonally, it might be a possibility that the ships locations will not be set at the initial locations on the board. Another note about Random Strategy: due to the fact that the same point might be shot more than once, the simulation might take more time. Therefore in the games class, I have set the total rounds to 2 where the running time will be shorter.