**Battleship Assignment Project Manual**

**What is this about:**

Battleship is a turn-based two-player game.

**Project Requirement:**

* Visual Studio 2017 and above.
* .Net Framework 4.7.2

**Steps to run the solution:**

* Clean and build the solution. If you face any build error, then restore NuGet packages installed in the project. Make sure your system is connected to the Internet.
* After successful build, you can run the project with F5 or by clicking Start button from the visual studio control panel.
* To play the game follow the instruction displayed in the console.

**Description:**

1. Battleship Assignment has 3 layers.
   1. BattleShipAssignment is UI layer.
   2. BattleShip.BLLis business layer for handling all business logic
   3. BattleShipTestis test layer i.e. handing unit test cases for the project.
2. Program.csis the starting class of the project. Project has 2 input options
   1. Read input from file

**(**Input will be read from Input.txt present in Bin folder and output will be written to Output.txt present in bin folder of BattleShipAssignment)

* 1. Read input from user

**Class and Interfaces in Project:**

**BattleShipAssignment – UI Layer of the project**

Interpreter: Responsible for Input output operations. (I/O Operations)

1. Interpreter responsible for parsing the inputs and setting up the next input to be read from user.
2. It also checks when to exit or continue based on user input.
3. Interpreter also contains GamePlayCompleted event handler. Responsible for generating output.

FileUtil: Concrete class of IFileOperations Interface. It is Singleton class responsible for handling file operations

1. It has 2 methods ReadFromFile and WriteToFile, responsible for reading and writing to file respectively.

GamePlay: Concrete class of IGamePlay Interface. Responsible for executing the game.

1. It as one method called Start, that will be called only after all the input is validated. And it is called from interpreter.

ExitCommand: Concrete class of ICommand Interface. And is responsible for exiting from game.

ContinueCommand: Concrete class of ICommand Interface. And is responsible to continually play game.

**BattleShip.BLL**

InputValidator: Concrete class of IInputValidator Interface. And is handling validation of Input.

BattleshipBoardService: Concrete class of IBattleshipBoardService Interface.

1. Used to setup BattleshipBoard, which includes generating players and setting current input for game.

GamePlayService: Concrete class of IGamePlayService Interface. Handling battleship game core logic.

1. Responsible for finding winner of the game.
2. It also fills battleship board with the battle ships.
3. It also Generate GamePlay Board.

**BattleShip.Test**

It’s Unit test project of **BattleShipAssignment** game.

Its 2 Test classes

1. BattleShipInputTest: For testing battleship game input
2. BattleShipTest: For testing battleship complete game flow.