



**Advanced Programming Practices (SOEN 6441)**  
**Summer 2019**

**Project - Build1**  
**Battleship Game**  
**Coding Standards**

**Submitted By: Team1**

Aruni Patel - 40077021

Dhvani Agola - 40081593

Sanjana Udar - 40094029

Sahil Savaliya - 40080380

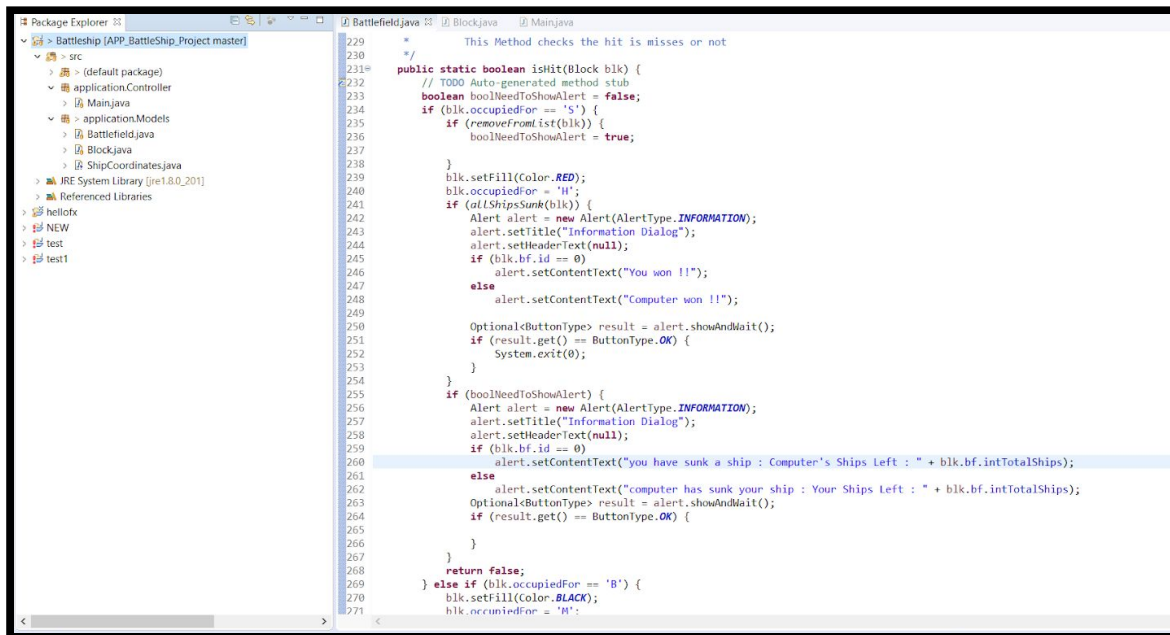
Sahana Shankar - 40092026

**Submitted To:**

Nagi Basha

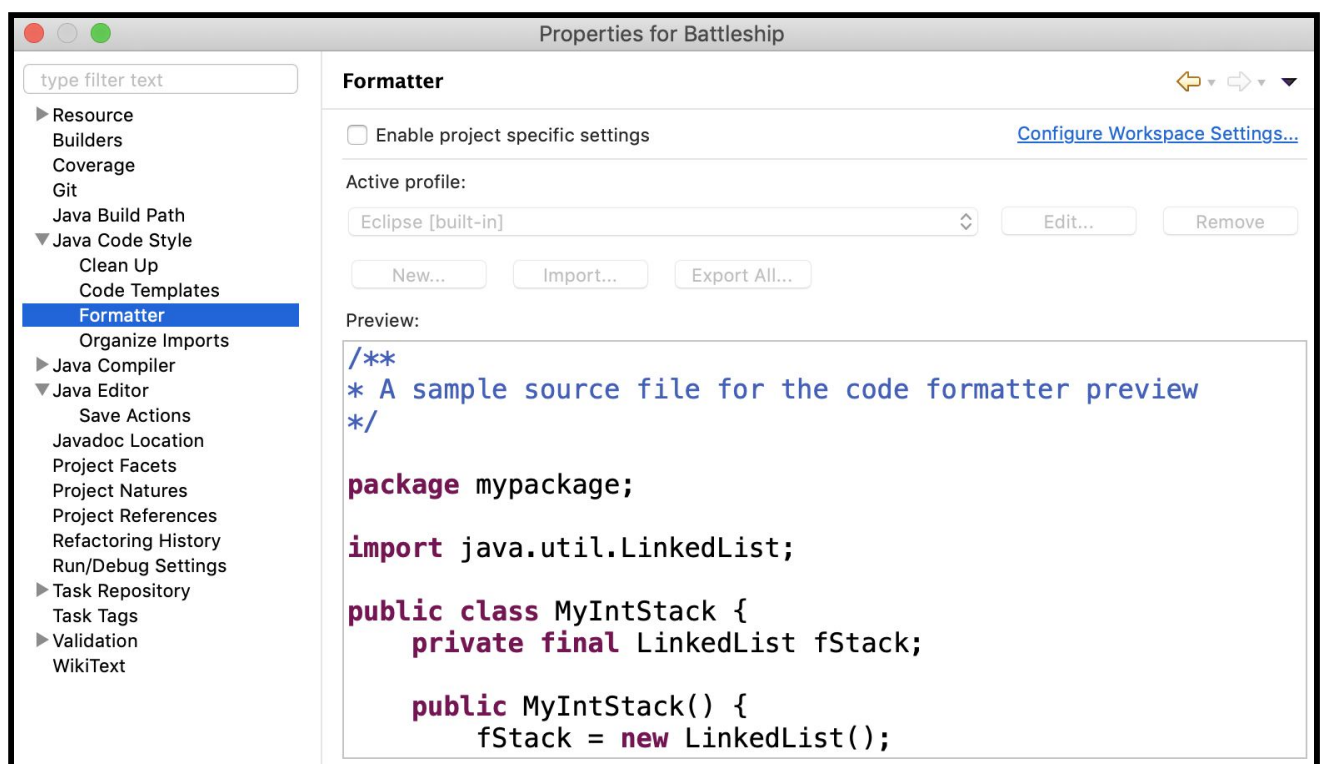
## • File Structure

- The packages 'Controller' and 'Models' are included inside the 'application' package.
- The 'Controller' package has controller file 'Main.java'.
- The 'Models' package has two models files, 'Battlefield.java' and 'Block.java'.



## • Code Style :

- Open curly braces written in the same line of the respective code block to reduce the number of lines of code.
- Close curly braces written in the line to overcome the issue of readability.
- Indentation : After every open curly brace, the new code block will start by left one tab space.



## ● Naming Conventions :

- Package Name : All package names starts with uppercase letter and the first letter of each word is in capital letter
- Class name : All class names starts with uppercase letter and first letter each word is in capital letter.
- Variable and parameter name : All variable names are in camelcase and starts with its datatype.
- Method name : All method names are in camelcase.

```
34     public static List<List<Battlefield.Block>> lstAllShips_Player = new ArrayList<>();
35     public static List<List<Battlefield.Block>> lstAllShips_Computer = new ArrayList<>();
36     private boolean boolIsGameStart = false;
37     private Battlefield bfComputer;
38     private Battlefield bfPlayer;
```

## ● Comments :

- To fulfill the purpose of readability and creating a javadoc, we have put detailed comments over each and every class, constructor and methods.
- Written comments to each class, constructor and methods to describe which functionality(ies) is included in the class, constructor and methods.
- These comments includes
  - Brief description of the class, constructor and method.
  - The list of parameters and also brief description of those parameters.
  - What the method is returning.
  - Author(s) of the class.

```
56         blkTemp.setOnMouseClicked(handler);
57         hboxlor_Y.getChildren().add(blkTemp);
58     }
59
60     vboxVert_X.getChildren().add(hboxlor_Y);
61 }
62
63     getChildren().add(vboxVert_X);
64 }
65
66 /**
67  *
68  * @param intShipLength Length of a ship which going to be placed
69  * @param boolIsHorizontal Checking the ship is horizontal or vertical which is
70  *   going to be placed
71  * @param x x-coordinate where ship is going to be placed
72  * @param y y-coordinate where ship is going to be placed
73  * @param boolIsPlayer Checking is the player is a computer or a human
74  * @return Returns true if ship placed successfully else returns false.
75  */
76 public boolean startPlaceShip(int intShipLength, boolean boolIsHorizontal, int x, int y, boolean boolIsPlayer) {
77     List<Block> lstTemp = new ArrayList<>();
78
79     System.out.println("line37:" + this.id);
80
81     if (isValidToPlaceShip(boolIsHorizontal, x, y, intShipLength)) {
```

- **References :**

- <https://www.geeksforgeeks.org/java-naming-conventions/>
- <https://github.com/AlmasB/Battleship/tree/master/src/com/almasb/battleship>
- <https://battleship-game.org/en/>
- <https://stackoverflow.com/questions/10872444/mvc-pattern-in-javafx-with-scene-builder>
- <https://www.thesprucecrafts.com/the-basic-rules-of-battleship-411069>