UNITAR GRADUATE SCHOOL

Course: ITWM5113 Software Design and Development

Course Instructor: Dr. Simon Lau Boung Yew

Assignment Submission (Individual)

Assignment Title: Assignment 1 – Battle Ships

mc220316759@stundent.unitar.my

SOFTWARE DESIGN AND DEVELOPMENT (ITWM5113)

ASSIGNMENT 1 TITLE
MODULE 1 PROJECT: BATTLESHIPS

MOHAMMAD HAZAL BIN HANIFAH
(STUDENT ID: MC220316759)

DR. SIMON LAU BOUNG YEW

mc220316759@stundent.unitar.my

Table of Contents

mc220316759@stundent.unitar.my

OBJECTIVE

The purpose of this Report assignment is to demonstrate the JAVA coding or script based on instruction given by Lecturer based on Open Learning module (Assignment 1).

INTRODUCTION

This project will help you get more familiar with arrays. You will be recreating the game of battleships. A player will place 5 of their ships on a 10 by 10 grid. The computer player will deploy five ships on the same grid.

Once the game starts the player and computer take turns, trying to sink each other's ships by guessing the coordinates to "attack". The game ends when either the player or computer has no ships left.

ENVIRONMENT

Before the start of this programming, relevant Java util package is imported using “import java.util.*). It contains collection framework and classes related to scanner, random, arrays etc which are utilized in this program.
Then public class called BattleShipsGame is declared. In here, global variables (i.e. NumUserShips and NumComputerShips) are initialized and it will be used in various part of the methods in this program.

Assignment Steps	Methods	Purpose
Step 1: Create ocean map	makeEmptyOcean(ocean) printOcean(ocean)	Print welcome messages and build initial 2D array
Step 2: Deploy user’s ships	placeUserShips(ocean)	To deploy 5 user ships
Step 3: Deploy computer’s ships	placeComputerShips(ocean)	To deploy 5 computer ships
Step 4: Battle	printOcean(ocean) startBattle(input, ocean)	Print the OceanMap with user location ‘@’ only and start the battle
Step 5: Game Over	GameOver() Input.close()	Print the results of the user’s game with computer and close the scanner

ASSIGNMENT INSTRUCTION -

The high level instruction for Battleship JAVA coding as below:-

1. Step 1 – Create the Ocean Map.
2. Step 2 – Deploy Player's Ships.
3. Step 3 – Deploy Computer's Ships.
4. Step 4 – Battle.
5. Computer's Turn.
6. Step 5 – Game Over

CODING PROGRAMMED

```
package battleship;


import java.util.Random;

import java.util.Scanner;


public class assignment1 {



	public static void main(String[] args) {

		// TODO Auto-generated method stub

		System.out.println("***Welcome to Battle Ships game");
```

```
                System.out.println("\nRight now, the sea is empty.\n");
//              Step 1 - Create the ocean map
                String[][] coordinates = setCoordinates();
                printCoordinates(coordinates);

//              Step 2 - Deploy player's ships
                Scanner input = new Scanner(System.in);
                int x, y, count = 0;

                while (count < 5) {
                        System.out.print("\nEnter X coordinate for your "+(count + 1)+". ship: ");
                        x = input.nextInt();
                        System.out.print("\nEnter Y coordinate for your "+(count + 1)+". ship: ");
                        y = input.nextInt();

//                      y - row
//                      x - column

                        if (x > 9 || x < 0 || y > 9 || y < 0) {
                                System.out.println("You can't place ships outside the 10 by 10 grid");
                        } else if (coordinates[y+1][x+2].equals("1")) {
                                System.out.println("You can not place two or more ships on the same
location");
                        } else {
                                coordinates[y+1][x+2] = "1";
                                count++;
                        }
                }

                printCoordinates(coordinates);

                count = 0;
```

```
                Random rand = new Random();

//              Step 3 - Deploy computer's ships
                while (count < 5) {
                        x = rand.ints(0,9).findFirst().getAsInt();
                        y = rand.ints(0,9).findFirst().getAsInt();

//                      y - row
//                      x - column

                        if (!coordinates[y+1][x+2].equals("1") && !coordinates[y+1][x+2].equals("2"))
{
                                System.out.print("\n"+(count + 1)+". ship DEPLOYED");
                                coordinates[y+1][x+2] = "2";
                                count++;
                        }
                }

                System.out.print("\n\n");
                printCoordinates(coordinates);

//              Step 4 - Battle
                int computerShips = 5;
                int userShips = 5;

                boolean pass = false;

                while (computerShips > 0 && userShips > 0) {
                        System.out.println("YOUR TURN");
                        System.out.print("Enter X coordinate: ");
                        x = input.nextInt();
                        System.out.print("Enter Y coordinate: ");
```

```
                    y = input.nextInt();

                    if (coordinates[y+1][x+2].equals("1")) {
                System.out.println("Oh no, you sunk your own ship :(");
                coordinates[y+1][x+2] = coordinates[y+1][x+2].replace("1", "x");
                userShips--;
            } else if (coordinates[y+1][x+2].equals("2")) {
                System.out.println("Boom! You sunk the ship!");
                coordinates[y+1][x+2] = coordinates[y+1][x+2].replace("2", "!");
                computerShips--;
            } else {
                System.out.println("Sorry, you missed");
                coordinates[y+1][x+2] = "-";
            }

//                  ////////////////////////////////////////////////////////////////////////////

                    System.out.println("COMPUTER'S TURN");
                    while(!pass) {
                        x = rand.ints(0,9).findFirst().getAsInt();
                        y = rand.ints(0,9).findFirst().getAsInt();

                        if (!coordinates[y+1][x+2].equals("#")) {
                            pass = true;
                        }

                    }

                    pass = false;

                    if (coordinates[y+1][x+2].equals("1")) {
```

```
                System.out.println("The Computer sunk one of your ships!");
                coordinates[y+1][x+2] = coordinates[y+1][x+2].replace("1", "x");
                userShips--;
        } else if (coordinates[y+1][x+2].equals("2")) {
                System.out.println("The Computer sunk one of its own ships");
                coordinates[y+1][x+2] = coordinates[y+1][x+2].replace("2", "!");
                computerShips--;
        } else {
                System.out.println("Computer missed");
                coordinates[y+1][x+2] = "#";
        }


                        System.out.println();
                        printCoordinates(coordinates);
                        System.out.println("\n\nYour ships: "+userShips+" | Computer ships:"+computerShips);
                        System.out.println("-------------------------------------------------------\n\n");
                }


//              Step 5 - Game Over
                if(computerShips > 0) {
                        System.out.println("Your ships: "+userShips+" | Computer ships:"+computerShips);
                        System.out.println("Computer wins the battle :)");
                } else if (userShips > 0){
                        System.out.println("Your ships: "+userShips+" | Computer ships:"+computerShips);
                        System.out.println("Hooray! You win the battle :)");
                } else if (userShips == computerShips) {
                        System.out.println("Your ships: "+userShips+" | Computer ships:"+computerShips);
                        System.out.println("No winner");
                }
```

```
}

public static String[][] setCoordinates() {
        int rows = 12;
        int columns = 14;
        String[][] coordinates = new String[rows][columns];

        for(int i = 0; i < rows; i++){
    for(int j = 0; j < columns; j++){
        if(i == 0 || rows-i == 1) {
                if(j == 0 || j == 1 || j == (columns - 1) || j == (columns - 2)) {
                        coordinates[i][j] = " ";
                } else if (j != 0 && j != 1 && j != (columns - 1) && j != (columns - 2)) {
                        int coordinate_x = j - 2;
                        coordinates[i][j] = Integer.toString(coordinate_x);
                }
        }

        else if(i > 0 && i < rows - 1) {
                if (j == 0 || j > (columns - 2)) {
                        int coordinate_y = i - 1;
                        coordinates[i][j] = Integer.toString(coordinate_y);
                } else if (j == 1 || j == (columns - 2)) {
                        coordinates[i][j] = "|";
                } else if (j != 1 && j < (columns - 2)) {
                        coordinates[i][j] = " ";
                }
        }
    }
  }
```

```
                return coordinates;
        }

        public static void printCoordinates(String[][] coordinates) {
                int rows = coordinates.length;
           int columns = coordinates[0].length;

                for(int i = 0; i < rows; i++){
              for(int j = 0; j < columns; j++){

                if (coordinates[i][j].equals("1") && i > 0 && i < rows - 1 && j > 1 && j < columns - 1) {

                        System.out.print("@");
                } else if (coordinates[i][j].equals("2") && i > 0 && i < rows - 1 && j > 1 && j < columns
- 1) {
                        System.out.print("x");
                }
                else if (coordinates[i][j].equals("#")) {
                        System.out.print(" ");
                } else {
                        System.out.print(coordinates[i][j]);
                }
              }

              System.out.println();
                }

        }

}
```

Game Over

mc220316759@stundent.unitar.my