

Assignment Report

Course Members:	JC2002
Student Name:	Hong, Cao
UoA Student id:	50079760
Major:	Computing Science
Assignment Github Adress:	github.com/1zero224/Assigment_Java2022.11

1 Task completion status

Task	
Task 1 : Create Battle Ship	Completed
Task 2 : Implement the functionality to display the game grid	Completed
Task 3 : Create a mechanism for playing game in rounds and handling the attacks	Completed
Task 4 : Complete your code to implement remaining game controls	Completed
Task 5 : Report and Code Comments	Completed

Test results obtained using maven

```
[INFO] Results:
[INFO]
[INFO] Tests run: 11, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.386 s
[INFO] Finished at: 2022-11-07T22:56:29+08:00
[INFO] -----
```

2 Task completion method

2.1 Task 1

1.1)

Create a direction array, and use the random function to randomly pick a number in 0 and 1 as the index of the array to complete the random decision of the ship's direction.

1.2)

The gettr methods all return the desired variables directly, while the setter methods inherit and assign the input values to the desired variables.

1.3)

The diff array is defined to add or subdivide the abscissa or ordinate of the position of the ship according to the different direction to access the three coordinates of the ship. At the same time, whether the current coordinate is the ship has received the attack.

2.2 Task 2

2.1)

After checking the validity of the input, a new binary array variable is generated according to the three input variables to realize the generation of the grid. And through the loop to complete the initialization of the grid.

2.2)

You loop through the input and assign ships to an array of your new AbstractBattleShip class.

2.3)

The coordinates of the center position among the three positions of a ship are randomly generated according to the direction of the ship. Ships are subsequently placed in the grid according to the center coordinates in and ship directions.

2.4)

PlayerGameGrid's printGrid() method prints the player's grid directly through the loop. OpponentGameGrid's printGrid() method prints the opponent's grid through the loop. OpponentGameGrid's printGrid() method prints the elements that are not "*", while those that are are ".".

2.3 Task 3

3.1)

The gamecontrol interface is implemented according to the type of the method and the type of the passed parameters, and the three passed parameters are used to generate the grid and ship of the player and opponent, respectively. The getter method is implemented by returning the generated PlayerGameGrid class and OpponentGameGrid class variables.

3.2)

Pass input to the function and use the contains() method to check for exit in the string. If it does, print the message and end the program.

3.3)

This process is divided into two steps. We first iterate over all the ships and count how many ships have been destroyed, and then use the result to determine whether all the ships of one side have been destroyed. To check if either side won.

3.4)

First, we check that the input is valid. The input is then transformed into coordinates. The attack method implements the inspection of the attack, the output of the attack result and the update of the grid by traversal and checkAttack function. The robotStrategy method implements a random attack on the player's ship through the Attack method and random. When the attack is complete, we check to see if either side won the victory by using the checkVictory method.

2.4 Task 4

4.1)

Check that args is valid. We then convert args to three variables and run the game via game.

4.2)

The number of rounds depends on the number of grid elements, and the program can be terminated by exitGame or game victory.

4.3)

The first termination condition uses the exitGame method

The second termination condition uses the checkVictory method

4.4)

Take a line as input, and keep putting input into playRound runs as long as the input is valid.

4.5)

Using if, throws an exception if the input is not exit or formatted coordinates.

3 How do you run the code

Make sure you are in the assignment directory

You can compile exec with **mvn :java -D"exec.mainClass"="abdn.scnu.cs.RunGame" -D"exec.args"="arg1 arg2 arg3"** Run the program in a command-line window (arg1 is the height of the grid, arg2 is the width of the grid, arg3 is the number of ships)

When running the game, you can

- 1) Enter the coordinates you want to attack in the form x,y
- 2) Type exit to quit the game

You can also choose to run the program tests via the **mvn clean test** command