Report

task1 status: completed

1.1-1.2

In this task I created the class " BattleShip "which is the subclass of the abstract class "AbstractBattleShip", this class has many variables like**"hits，name, shipCoordinate, shiporientation"** and so on. So I also provide access to the class variables by using method "**getName** (), **getHits**(), **getShipOrientation**(), **sethits**(), **getShipCoordinates**(), and **setShipCoordinates**()."

```
public BattleShip(String name) {
        //to set ship Orientation randomly
        String[] index = {"vertical", "horizontal"};
        Random r = new Random();
        this.name = name;
        this.shipOrientation = index[r.nextInt(2)];
}
```

In order to set the ship Orientation randomly, I used the **Random** class and **nextInt** () method to generate random number as index to choose "vertical" and "horizontal" in a String list.

1.3 CheckAttack()

```
public boolean checkAttack(int row, int column) {
        //to check whether the ship get hit
        if (row==this.shipCoordinates[0][0]&&column==this.shipCoordinates[0][1]){
            //if so hits +1, when the ship's hits over 3 the ships was destroyed
            this.hits=this.hits+1;
            if (this.hits > 3) {
                return false;}
            else {
                return true;
            }
        }
        else {return false;}
```

In this method, I try to match the **row** and **column** parameters to shipCoordinates respectively to check whether the ship get hits.

task2 status: completed

2.1 InitializeGrid()

```
public void initializeGrid () {
        //create the game grid by deflating "."
        for (int i = 0; i < gameGrid.length; i++) {
            for (int j = 0; j < gameGrid[i].length; j++) {
                gameGrid[i][j] = ".";
            }}}
```

To initialize grid I use two **for loop** and insert the "." in gameGrid which is an 2 dimension arraylist

2.2 generateShips()

```
 public void generateShips ( int numberOfShips){
        //set the space of ships array
        ships = new BattleShip[numberOfShips];
        //instantiate the ship and store it in ships array
        for (int i = 0; i < numberOfShips; i++) {
            BattleShip ship = new BattleShip("Ship " + (i + 1));
            this.ships[i] = ship;
        }
    }
```

In this small section, I just instantiate BattleShip objects which amount is required by numberOfShips parameter and store them in ships Arraylist

2.3 placeShip()

I think this may be not the optimal code. I create the center coordinate of the ship initially, then considering shipOrientation , I can get the other two coordinate of the ship. After that, I update the shipCoordinates and update gameGrid too.

2.4 Player&Opponent printGrid()

Opponent printGrid()

```
public void printGrid() {
        System.out.print("Opponent's Grid");
        for (int i = 0; i < gameGrid.length; i++) {
            String subArray[] = gameGrid[i];
            System.out.println();
            for (int j = 0; j < subArray.length; j++) {
                //to hide the ship location
                if (subArray[j]=="*"){
                    System.out.print(".");
                    continue;
                }
                System.out.print(subArray[j]);}}}
```

printGrid() method both used two **for loop** structure.

as to Opponent printGrid() to hide the ship location , I replace"*" to "."

Player printGrid() is familiar except the hide part.

task3 status: completed

3.1 Construct Game.java

```
public Game(int row, int column, int NumberOfShip){
        //Construct the basic variable
        this.row = row;
        this.column = column;
        this.NumberOfShip = NumberOfShip;
```

take 3 parameters: row, column, NumberOfShip

3.2 exitGame()

```
@Override
    public void exitGame(String input) {
        if(Objects.equals(input, "exit")){
            System.out.println("Exiting game_thank you for playing");
            System.exit(0);
        }}
```

if program get the input is "exit", It would use method **System.exit(0);** to end the program

3.3 checkVictory()

In this task, I just need to check all the ship in ships arraylist in both side respectively, whether all of them are destroy. And return corresponding message.

3.4 playround()

When it comes to playround. Firstly, I used the split() and Integer. parseInt() method to get the useful Input message and turn it from String type to Int type. Then, I check the whether the ships get hit. if so, turn update the gamegird of oppoent to "X" and hits plus 1, if not update the gamegrid of opponent to "%"

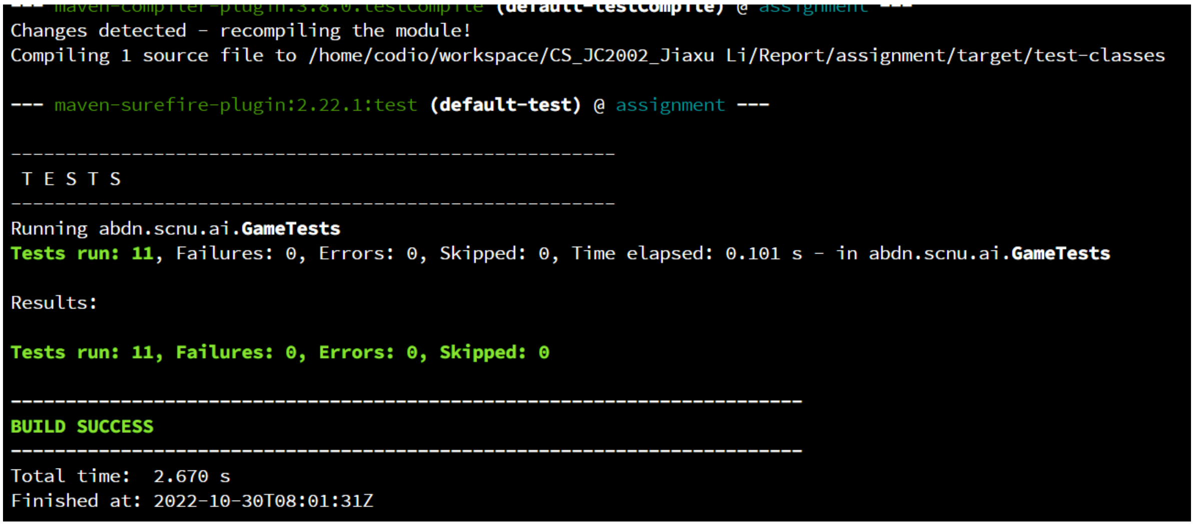
Last but not least, I also create a playround() for the robot as opponent, to attack the player's ship, which will be seen at the code.

task4 status: completed

4.5 RunGame.java try exception

Catch an exception thrown with an unexpected input (i.e, if users enters anything else apart

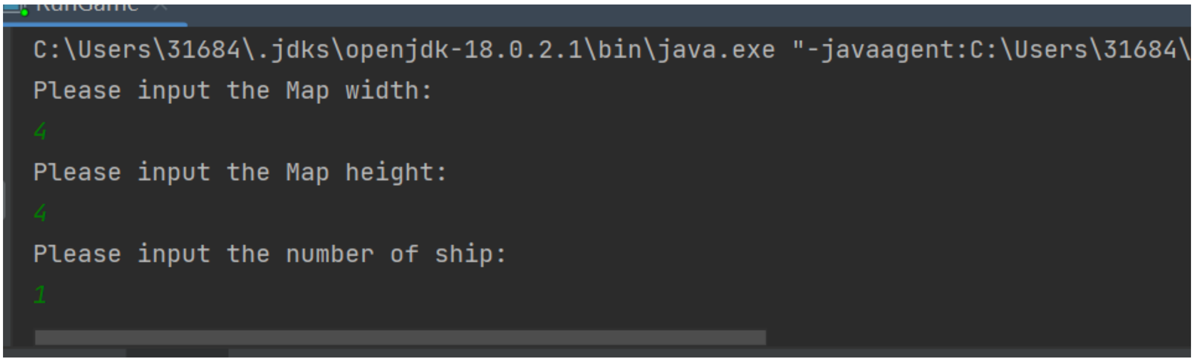
from “exit” or “row,column” coordinates. When the error occurs, print “Incorrect input”.



```
--- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ assignment ---
Changes detected - recompiling the module!
Compiling 1 source file to /home/codio/workspace/CS_JC2002_Jiaxu Li/Report/assignment/target/test-classes

--- maven-surefire-plugin:2.22.1:test (default-test) @ assignment ---

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running abdn.scnu.ai.GameTests
Tests run: 11, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.101 s - in abdn.scnu.ai.GameTests

Results:

Tests run: 11, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  2.670 s
Finished at: 2022-10-30T08:01:31Z
```

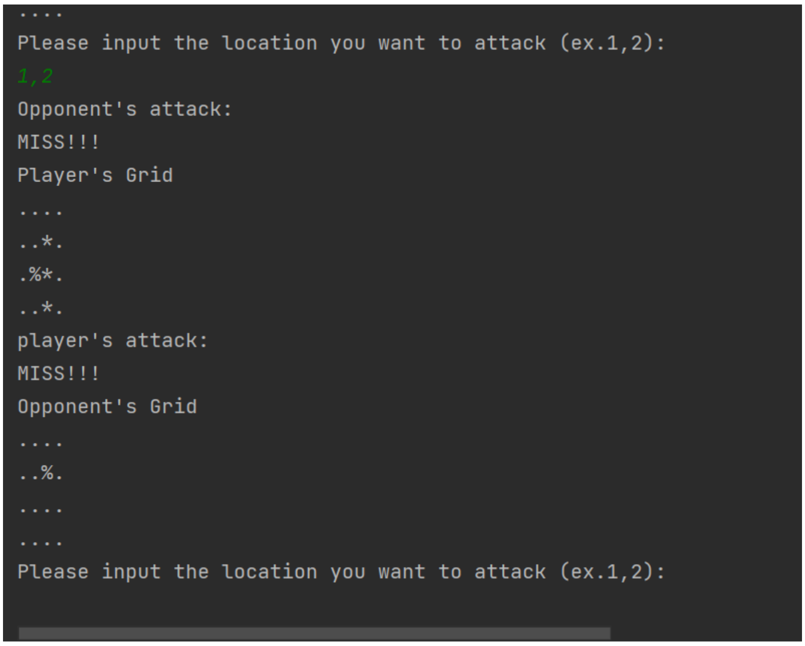
How to run the game

After starting the code. you are required to set the row, column and number of ship for game. (press "enter" after insert the number)



```
C:\Users\31684\.jdks\openjdk-18.0.2.1\bin\java.exe "-javaagent:C:\Users\31684\
Please input the Map width:
4
Please input the Map height:
4
Please input the number of ship:
1
```

Then, the program will initialize the player's grid and opponent's grid automatically. Then you can insert the coordinate you want to attack in grid. (use "," to apart row and column) for instance input 1,2 mean is want to attack row 2 and column 3.)

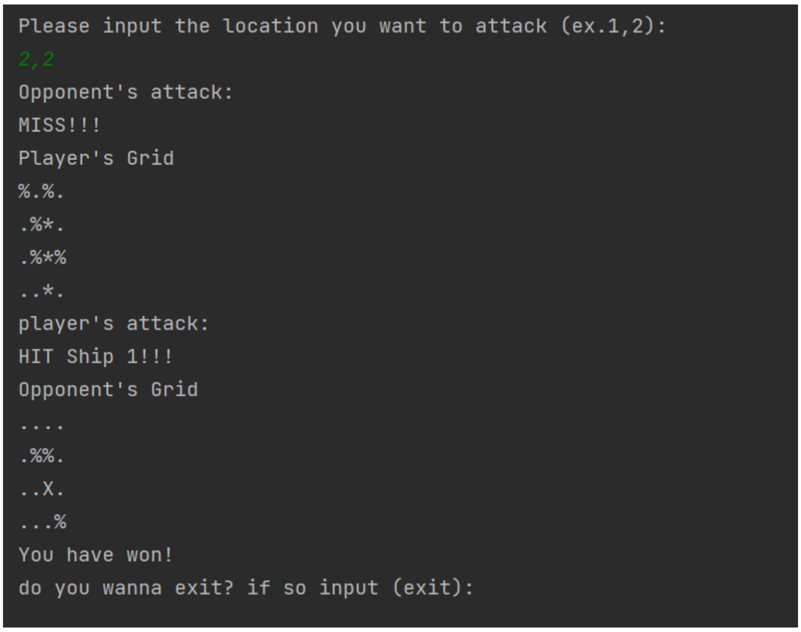


```
....
Please input the location you want to attack (ex.1,2):
1,2
Opponent's attack:
MISS!!!
Player's Grid
....
..*.
.%*.
..*.
player's attack:
MISS!!!
Opponent's Grid
....
..%.
....
....
Please input the location you want to attack (ex.1,2):
```

(use "," to apart row and column) for instance input 1,2 mean is want to attack row 2 and column 3.)

As you can see, row 2 column 3 was attacked.(unfortunately, enemy battleship is not here) and you can see the enemy didn't find your ship too.

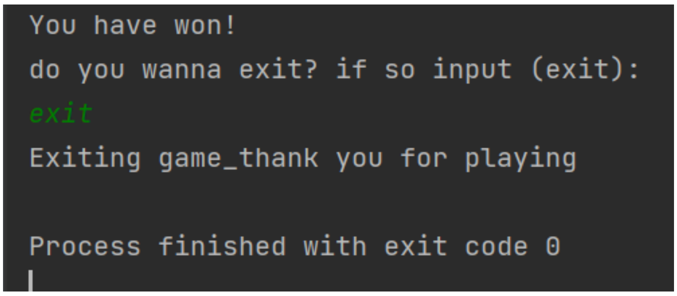
The program will keep asking you to insert attack coordinate until the game finish(Either player’s or opponent’s ships are destroyed.)



```
Please input the location you want to attack (ex.1,2):
2,2
Opponent's attack:
MISS!!!
Player's Grid
%.%.
.%*.
.%*%
..*.
player's attack:
HIT Ship 1!!!
Opponent's Grid
....
.%%.
..X.
...%
You have won!
do you wanna exit? if so input (exit):
```

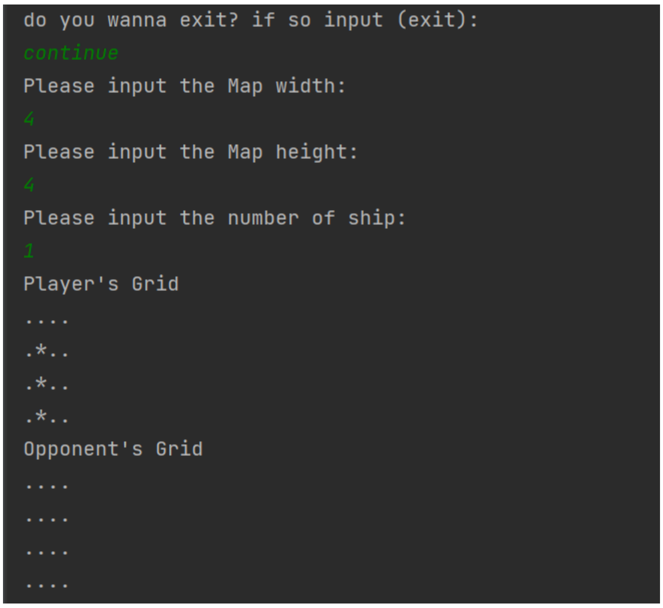
For instance, here I won the game, and the program asked me whether I want to exist.

If I enter "exit", the program will end.



```
You have won!
do you wanna exit? if so input (exit):
exit
Exiting game_thank you for playing

Process finished with exit code 0
```

If not it will restart the game.



```
do you wanna exit? if so input (exit):
continue
Please input the Map width:
4
Please input the Map height:
4
Please input the number of ship:
1
Player's Grid
....
.*..
.*..
.*..
Opponent's Grid
....
....
....
....
```