

Psuedocode

Class TestAviJoshua

main();

Game = newMinesweepergame(scanner);

Game.play();

Class MinesweeperAviJoshua

MinesweeperAviJoshua(scanner);

play();

While game not over()

PrintBoard();

RevealCell(userInput);

printboard();

getUsermove();

RevealCell(input);

GameOver();

Class CellAviJoshua

//cell representation things

ERRORS:

Error in Scanner, not taking proper values throwing a InputMismatchException

```
Run Arguments
int row = scanner.nextInt();
int col = scanner.nextInt();
System.out.println("Flag cell? (y/n): ");
String flagInput = scanner.next();

if (flagInput.equalsIgnoreCase("y")) {
    System.out.println("Enter row and column to flag: ");
    int flagRow = scanner.nextInt();
    int flagCol = scanner.nextInt();
}

Compile Messages | JGRASP Messages | Run IO | Interactions

Test
Clear
Help
Enter row and column (e.g., 1 2):
----JGRASP: process ended by user.

----JGRASP exec: java TestAviJoshua
>>> Enter board width: 2
>>> Enter board height: 2
>>> Enter number of mines: 1
>>>
>>> Enter row and column (e.g., 1 2): 1.1
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:943)
    at java.base/java.util.Scanner.next(Scanner.java:1598)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2263)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2217)
    at MinesweeperAviJoshua.play(MinesweeperAviJoshua.java:149)
    at TestAviJoshua.main(TestAviJoshua.java:20)

----JGRASP wedge2: exit code for process is 1.
----JGRASP: operation complete.
```

Incorrect Logic in CheckWinner(), always returning as I win

```
Run Arguments
import java.util.Scanner;

public class TestAviJoshua {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter board width: ");
        int width = scanner.nextInt();
        System.out.println("Enter board height: ");
        int height = scanner.nextInt();
        System.out.println("Enter number of mines: ");
        int numMines = scanner.nextInt();
        MinesweeperAviJoshua game = new MinesweeperAviJoshua(width, height, numMines);
        game.play();
        scanner.close();
    }
}

Compile Messages | JGRASP Messages | Run IO | Interactions

Test
Clear
Help
----JGRASP exec: java TestAviJoshua
>>> Enter board width: 2
>>> Enter board height: 2
>>> Enter number of mines: 1
>>>
>>> Enter row and column (e.g., 0 1): 0 0
>>> Flag cell? (y/n): n
>>> You win!

----JGRASP: operation complete.
```

Calling printBoard incorrectly, causes board to be printed twice and incorrectly displayed

```

// Run Arguments
}
}

// find all (row, col);
}

if (gameBoard[row][col].isFlagged()) {
    System.out.println("This cell is flagged, please select a different cell.");
    continue;
}

if (row < 0 || row >= boardWidth || col < 0 || col >= boardHeight) {
    System.out.println("Invalid cell, please try again.");
    continue;
}

if (gameBoard[row][col].hasMine()) {
    gameBoard[row][col].setRevealed(true);
    printBoard(true);
    System.out.println("Game over! You stepped on a mine.");
    gameOver = true;
} else {
    gameBoard[row][col].setRevealed(true);
    printBoard(false);
}

if (checkWinner()) {
    System.out.println("You win!");
    gameOver = true;
}
}

// CellAviJoshua.java // MinesweeperTestJoshua.java // TestWinJoshua.java
//
// Compile Messages // JGRASP Messages // Run IO // Interactions
//
// End
// Clear
// Help
//
//
// jGRASP exec: java TestAviJoshua
//> Enter board width: 2
//> Enter board height: 2
//> Enter number of mines: 1
// #
// #
//> Enter row and column (e.g., 0 1): 0 0
//> Flag cell? (y/n): n
// 1 #
// #
// #
// #
//> Enter row and column (e.g., 0 1): [

```

Incorrect Printing when gameOver;

```
        }
    }
}

private void flagCell(CellAviJoshi[][] game, int row, int column) {
    if (!game[row][column].isRevealed()) {
        game[row][column].toggleFlag();
        //printBoard(game, false); // Print the updated board after flagging
    } else {
        System.out.println("Cannot flag a revealed cell. Choose a different action.");
    }
}

}
```

Incorrect Cell Assignment; everything put as 0.

```
        secondsPassed++;
    }
    }, 1000, 1000);
}

private String formatTime(int seconds) {
    int minutes = seconds / 60;
    int remainingSeconds = seconds % 60;
    return String.format("%02d:%02d", minutes, remainingSeconds);
}

private void revealCell(CellAviJosh[][] game, int row, int column) {
    if (game[row][column].hasMine()) {
        gameOver = true;
        printBoard(game, true);
        System.out.println("Game over! You stepped on a mine.");
    }
}
```

CellAviJosh.java MinesweeperAviJosh.java TestAviJosh.java

Compile Messages jGRASP Messages Run I/O Interactions

End
Clear
Help

```
#####
#####
#####
#####
#####
#####
Do you want to flag a cell? (Yes/No): no
Enter row and column to reveal (e.g., 1 2): 1 1
0 0 0 0 0 0 0 0
0 _ 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
You win!
Your total steps: 1
jGRASP: operation complete.
jGRASP exec: java TestYourname
#####
```

TEST RUNS:

```
Do you want to reveal (R) or flag (F) the cell? r
1# #####
#####
#####
#####
#####
#####
12###21111
-1#211-
-111-
-
Enter row and column (e.g., 1 2): 4 3
Do you want to reveal (R) or flag (F) the cell? r
1# #####
#####
#####
#####
#####
#####
12###21111
-1#211-
-111-
-
Enter row and column (e.g., 1 2): 0 8
Do you want to reveal (R) or flag (F) the cell? r
1# #####1#
#####
#####
#####
#####
#####
12###21111
-1#211-
-111-
-
Enter row and column (e.g., 1 2): 0 9
Do you want to reveal (R) or flag (F) the cell? r
1# #####11
#####
#####
#####
#####
#####
12###21111
-1#211-
-111-
-
Enter row and column (e.g., 1 2):
```

```

Messages | jGRASP Messages | Run I/O | Interactions
--
-- 1 1 3 # # 1 -
-- 1 # # # 2 1 -
-- 1 1 2 1 1 -
-- 1 1 2 1 1 -
-- 1 # # # 2 1 -
-- 1 1 2 # 2 1 -
-- 2 # # 2 2 2
-- 1 # # # # #
>> Enter row and column (e.g., 1 2): 5 4
>> Do you want to reveal (R) or flag (F) the cell? r
Cell already revealed or flagged, choose a different action.
-- 1 # # # 1 -
-- 1 1 3 # # 1 -
-- 1 # # # 2 1 -
-- 1 1 2 1 1 -
-- 1 1 2 1 1 -
-- 1 # # # 2 1 -
-- 1 1 2 # 2 1 -
-- 2 # # 2 2 2
-- 1 # # # # #
>> Enter row and column (e.g., 1 2): 4 5
>> Do you want to reveal (R) or flag (F) the cell? r
Cell already revealed or flagged, choose a different action.
-- 1 # # # 1 -
-- 1 1 3 # # 1 -
-- 1 # # # 2 1 -
-- 1 1 2 1 1 -
-- 1 1 2 1 1 -
-- 1 # # # 2 1 -
-- 1 1 2 # 2 1 -
-- 2 # # 2 2 2
-- 1 # # # # #
>> Enter row and column (e.g., 1 2): 9 9
>> Do you want to reveal (R) or flag (F) the cell? r
-- 1 * 2 1 -
-- 1 1 3 3 * 1 -
-- 1 * 2 * 2 1 -
-- 1 1 2 1 1 -
-- 1 1 2 1 1 -
-- 1 * 2 * 2 1 -
-- 1 1 2 # 2 1 -
-- 2 3 * 2 2 2
-- 1 * 2 2 * *
Game over! You stepped on a mine.
Your total steps: 1
Time taken: 04:40

```

[illegible]

