

## Software Tools 4

### Assignment 2 Part 2

Krunal Rank  
U18C0081

Minesweeper: Write a program that takes 3 arguments from user M, N, and p and produces an M-by-N boolean array where each entry is occupied with probability p. In the minesweeper game, occupied cells represent bombs and empty cells represent safe cells. Print out the array using an asterisk for bombs and a period for safe cells. Then, replace each safe square with the number of neighboring bombs (above, below, left, right, or diagonal) and print out the solution.

Try to write your code so that you have as few special cases as possible to deal with, by using an (M+2)-by-(N+2) boolean array.

#### Answer:

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Random;
import java.util.Scanner;

public class Minesweeper {
    static class Game {
        private int[][] arr, mask;
        private String[][] board;
        private int m, n;
        private double p;
        private int score;
        private boolean gameOver, gameWon;

        Game(int m, int n, double p) {
            this.m = m;
            this.n = n;
            this.p = p;
            this.score = 0;
            this.gameOver = false;
            this.gameWon = false;
            arr = new int[this.m][this.n];
            mask = new int[this.m][this.n];
            board = new String[this.m][this.n];
            for (int i = 0; i < this.m; i++) {
                for (int j = 0; j < this.n; j++) {
```

```

        double r = Math.random();
        arr[i][j] = (r > p) ? 0 : 1;
        board[i][j] = "□";
        mask[i][j] = 0;
    }
}

public void resetBoard() {
    this.score = 0;
    this.gameWon = false;
    this.gameOver = false;
    arr = new int[this.m][this.n];
    mask = new int[this.m][this.n];
    board = new String[this.m][this.n];
    for (int i = 0; i < this.m; i++) {
        for (int j = 0; j < this.n; j++) {
            arr[i][j] = (Math.random() > p) ? 0 : 1;
            board[i][j] = "□";
            mask[i][j] = 0;
        }
    }
}

public void playMove(int i, int j) throws Exception {
    if (this.gameOver == true) {
        throw new Exception("Invalid Move Error: Game Already Over!");
    }
    if (!(i >= 0 && i < this.m && j >= 0 && j < this.n)) {
        throw new Exception("Invalid Move Error: Please enter valid Move!");
    }
    if (this.mask[i][j] == 1) {
        throw new Exception("Invalid Move Error: Cell already unlocked!");
    }
    if (this.arr[i][j] == 1) {
        this.gameOver = true;
        this.revealBoard();
        return;
    }

    Queue<ArrayList<Integer>> q = new LinkedList<>();

    ArrayList<Integer> start = new ArrayList<Integer>();
    start.add(i);

```

```

start.add(j);
q.add(start);
while (q.size() != 0) {
    ArrayList<Integer> node = q.peek();
    q.remove();
    int count = 0;
    int x = node.get(0);
    int y = node.get(1);
    if (this.mask[x][y] == 1)
        continue;
    this.mask[x][y] = 1;
    for (int a = -1; a <= 1; a++) {
        for (int b = -1; b <= 1; b++) {
            if (a == 0 && b == 0)
                continue;
            if (x + a >= 0 && x + a < this.m && y + b >= 0 && y + b <
this.n) {

                if (this.arr[x + a][y + b] == 1)
                    count++;
                else {
                    if(Math.abs(a)+Math.abs(b)>1) continue;
                    ArrayList<Integer> t = new ArrayList<Integer>();
                    t.add(x + a);
                    t.add(y + b);
                    q.add(t);
                }
            }
        }
    }
    this.board[x][y] = Integer.toString(count);
    this.score = this.score + count;

}

int pass = 1;
for (int a = 0; a < this.m; a++) {
    for (int b = 0; b < this.n; b++) {
        if (this.arr[a][b] == 0 && this.mask[a][b] != 1) {
            pass = 0;
        }
    }
}
if (pass == 1) {
    this.gameWon = true;
}

```

```

        this.gameOver = true;
        this.revealBoard();
    }

    return;
}

private void revealBoard() {
    for (int i = 0; i < this.m; i++) {
        for (int j = 0; j < this.n; j++) {
            if (this.arr[i][j] == 1 && this.gameWon == false) {
                this.board[i][j] = "💣";
            } else if (this.arr[i][j] == 1 && this.gameWon == true) {
                this.board[i][j] = "💣";
            }
        }
    }
}

public void printBoard() {
    System.out.println("Board Status:");
    for (int i = 0; i < this.m; i++) {
        for (int j = 0; j < this.n; j++) {
            System.out.print(this.board[i][j]);
        }
        System.out.println("");
    }
    System.out.println("Score: " + this.score);
}

public boolean getGameOver() {
    return this.gameOver;
}

public boolean getGameWon() {
    return this.gameWon;
}

public int getScore() {
    return this.score;
}

```

```

}

public static void main(String[] arr) {
    try {
        if (arr.length != 3) {
            System.out.println("Minesweeper Help");
            System.out.println("Three Arguments Required:");
            System.out.println("M - No. of rows (Integer)");
            System.out.println("N - No. of columns (Integer)");
            System.out.println("P - Probability of Mines (Double)");
            System.out.println("Example: >java ./Minesweeper.java <M> <N> <P>");
            throw new Exception("Insufficient Arguments error!");
        }

        int m = Integer.parseInt(arr[0]);
        int n = Integer.parseInt(arr[1]);
        double p = Double.parseDouble(arr[2]);
        if (p <= 0 && p >= 1)
            throw new Exception("Invalid Argument Error! Probability should be
between 0 and 1 exclusive!");
        Scanner sc = new Scanner(System.in);

        Minesweeper.Game newGame = new Minesweeper.Game(m, n, p);
        while (!newGame.getGameOver()) {
            try {
                newGame.printBoard();
                System.out.print(">Your Move: ");
                String[] res = sc.nextLine().split(" ");
                if (res.length != 2)
                    throw new Exception(
                        "Invalid Input Error: Please enter Input of the form
<X> <Y> (X between 0 and M; Y between 0 and N) !");
                int i = Integer.parseInt(res[0]);
                int j = Integer.parseInt(res[1]);
                newGame.playMove(i, j);
            } catch (Exception e) {
                System.out.println(e.getMessage());
            }
        }
        newGame.printBoard();
        if (newGame.getGameWon() == true) {
            System.out.println("Game Over! You Won!");
        } else {
            System.out.println("Gamve Over! You Lost!");
        }
    }
}

```

```

    }

    sc.close();

} catch (Exception e) {
    System.out.println("Program exited due to unforeseen error!");
}

}
}

```

```

krhero@hellblazer: /mnt/0FB812900FB81290/BTech/Assignments/3rd_Year/ST4/Assignment2$ java ./Minesweeper.java 10 10 0.4
Board Status:
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
□□□□□□□□
Score: 0

```

```

>Your Move: 2 2
Board Status:
□22□□□□□
□34□□□□□
□□4□□□□□
□34□□□□□
12□□□□□□
023□□□□□
13□□□□□□
1□□□□□□□
13□□□□□□
013□□□□□□
Score: 43
>Your Move: 5 5
Board Status:
□22□□□□□
□34□□□□□
□□4□□□□□
□34□□□□□
12□□□□□□
023□□4□□□
13□643234□
1□□□2102□
13□□□3111
013□□□100
Score: 81

```

```
>Your Move: 6 6
Invalid Move Error: Cell already unlocked!
Board Status:
02200000
03400000
00400000
03400000
12000000
023004000
1306432340
100021020
130003111
013000100
Score: 81
>Your Move: 7 0
Invalid Move Error: Cell already unlocked!
Board Status:
02200000
03400000
00400000
03400000
12000000
023004000
1306432340
100021020
130003111
013000100
Score: 81
```

```
>Your Move: 0 7
Board Status:
0220322222
0340600200
004006560
034000000
120000000
023004000
1306432340
100021020
130003111
013000100
Score: 119
>Your Move: 6 6
Invalid Move Error: Cell already unlocked!
Board Status:
0220322222
0340600200
004006560
034000000
120000000
023004000
1306432340
100021020
130003111
013000100
Score: 119
>Your Move: 7 7
```

>Your Move: 4 9

Board Status:

□22□322222

□34□6□□2□□

□□4□□656□

□34□□□□□□

12□□□□□□4

023□□4□□3

13□643234□

1□□□2102□

13□□□3111

013□□□100

Score: 126

>Your Move: 9 4

Board Status:

☀22☀322222

☀34☀6☀☀2☀☀

□☀4☀☀☀656☀

☀34☀☀☀☀☀☀☀☀

12☀□☀☀☀□☀☀4

023☀☀☀4☀☀☀3

13☀643234☀

1☀☀☀☀☀2102☀

13☀☀☀□☀3111

013☀☀☀☀☀100

Score: 126

Game Over! You Lost!