Isaac Ray Shoebottom
CS 1083
Assignment 8
3429069

Source Code:

Painter.java:

```java
import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;


/**

 * Painter object that paints the room and reads the input file

 * @author Isaac Shoebottom (3429069)

 */


public class Painter {

    Scanner sc;

    int height;

    int width;

    char[][] room;


    /**

     * Constructor for painter

     * @param fileIn The file containing the information to construct the room

     * @throws FileNotFoundException When file does not exist

     */

    Painter(File fileIn) throws FileNotFoundException {

        sc = new Scanner(fileIn);

        height = Integer.parseInt(sc.nextLine());

        width = Integer.parseInt(sc.nextLine());

        room = new char[height][width];


        for(int i = 0; i < height; i++) {

            String currentLine = sc.nextLine();

            for(int j = 0; j < width; j++) {

                room[i][j] = currentLine.charAt(j);

            }
        }
    }

    /**
     * Determines the colors in the room and calls to paint the room
     */
    public void paint() {
        char[] colors = new char[]{'R', 'B', 'G'};
        int indexLine = 0;
        int indexCell = 0;

        for(char colorName: colors) {
            for(char[] line: room) {
                for(char cell: line) {
                    if (cell == colorName) {
                        paint(indexLine, indexCell, colorName);
                    }
                    indexCell++;
                }
                indexLine++;
                indexCell = 0;
            }
            indexLine = 0;
        }
    }

    /**
     * The method that paints the room
     * @param row The row in which to be painted
     * @param col The column in which to be painted
     * @param color The color to be painted
     */
    private void paint(int row, int col, char color) {
            room[row][col] = color;
```

```
            if (row == 0) {
                if (col == 0) {
                    //case for when the row and column is zero. (first position)
                    if (room[row + 1][col] == '0') //down
                        paint(row + 1, col, color);
                    if (room[row][col + 1] == '0') //right
                        paint(row, col + 1, color);
                }
                else if (col == width - 1) {
                    //case for when the row is zero and the column is max (top right)
                    if (room[row + 1][col] == '0') //down
                        paint(row + 1, col, color);
                    if (room[row][col - 1] == '0') //left
                        paint(row, col - 1, color);
                }
                //case for when the row being operated on is zero
                else {
                    if (room[row + 1][col] == '0') //down
                        paint(row + 1, col, color);
                    if (room[row][col - 1] == '0') //left
                        paint(row , col - 1, color);
                    if (room[row][col + 1] == '0') //right
                        paint(row, col + 1, color);
                }
            } else if (row == height - 1) {
                if (col == width - 1) {
                    //case for when the row and the column are at the max value (last
position)
                    if (room[row - 1][col] == '0') //up
                        paint(row + 1, col, color);
                    if (room[row][col - 1] == '0') //left
                        paint(row - 1, col - 1, color);
                }
                else if (col == 0) {
```

```
                //case for when the row is max and the column is zero (bottom left)
                if (room[row - 1][col] == '0') //up
                    paint(row + 1, col, color);
                if (room[row][col + 1] == '0') //right
                    paint(row, col + 1, color);
            }
            else {
                //case for when the row is at its maximum value
                if (room[row - 1][col] == '0') //up
                    paint(row - 1, col, color);
                if (room[row][col + 1] == '0') //right
                    paint(row, col + 1, color);
                if (room[row][col - 1] == '0') //left
                    paint(row, col - 1, color);
            }
        } else if (col == 0) {
            //case for when just the column is zero
            if (room[row - 1][col] == '0') //up
                paint(row + 1, col, color);
            if (room[row + 1][col] == '0') //down
                paint(row + 1, col, color);
            if (room[row][col + 1] == '0') //right
                paint(row, col + 1, color);
        } else if (col == width - 1) {
            //case for when the column is at its last position
            if (room[row - 1][col] == '0') //up
                paint(row - 1, col, color);
            if (room[row + 1][col] == '0') //down
                paint(row + 1, col, color);
            if (room[row][col - 1] == '0') //left
                paint(row, col - 1, color);
        } else {
            //case for any middle position on board
            if (room[row - 1][col] == '0') //up
```

```java
                    paint(row - 1, col, color);
                if (room[row + 1][col] == '0') //down
                    paint(row + 1, col, color);
                if (room[row][col - 1] == '0') //left
                    paint(row, col - 1, color);
                if (room[row][col + 1] == '0') //right
                    paint(row, col + 1, color);
            }
    }

    /**
     * toString method for Painter object. Prints the room as was in the text file
     * @return String containing the room
     */
    public String toString() {
        int indexCell = 0;
        StringBuilder returnString = new StringBuilder();
        for (char[] line : room) {
            for (char cell : line) {
                indexCell++;
                if (indexCell == width) {
                    returnString.append(cell).append("\n");
                }
                else {
                    returnString.append(cell);
                }
            }
            indexCell = 0;
        }
        return returnString.toString();
    }
}
```

PaintDriver.java:

```java
import java.io.File;
import java.io.FileNotFoundException;

/**
 * Driver for the painter method
 * @author Isaac Shoebottom (3429069)
 */

public class PaintDriver {
    public static void main(String[] args) {
        File input = null;
        if (args.length > 0) {
            input = new File(args[0]);
        }
        else {
            System.out.print("File does not exist");
            System.exit(1);
        }
        Painter painter = null;
        try {
            painter = new Painter(input);
        } catch (FileNotFoundException exception) {
            System.out.print("File does not exist");
            System.exit(1);
        }

        System.out.println("Before Paint:");
        System.out.print(painter.toString());
        painter.paint();
        System.out.println("\nAfter Paint:");
        System.out.print(painter.toString());
    }
}
```

Test 1:

Input 1:

```
8
7
1000000
11111G0
00B0101
1111111
101R101
1010101
0001111
0B010G0
```

Output 1:

```
Before Paint:
1000000
11111G0
00B0101
1111111
101R101
1010101
0001111
0B010G0

After Paint:
1GGGGGG
11111GG
BBBB1G1
1111111
1B1R101
1B1R101
BBB1111
BBB1GGG
```

Test 2:

Input 2:

```
4
6
100000
11111G
00B010
111111
```

Output 2:

```
Before Paint:
100000
11111G
00B010
111111

After Paint:
1GGGGG
11111G
BBBB1G
111111
```

Test 3:

Input 3:

```
2

28

10B011100B01011G1R1010001111

11111G0111111110101010B010G0
```

Output 3:

```
Before Paint:

10B011100B01011G1R1010001111

11111G0111111110101010B010G0


After Paint:

1BBB111BBBB1011G1R101BBB1111

11111GG11111111G1R101BBB1GGG
```