

IMPLEMENT THE PAINTFILL
FEATURE FROM MS PAINT

IMPLEMENT THE PAINTFILL FEATURE FROM MS PAINT

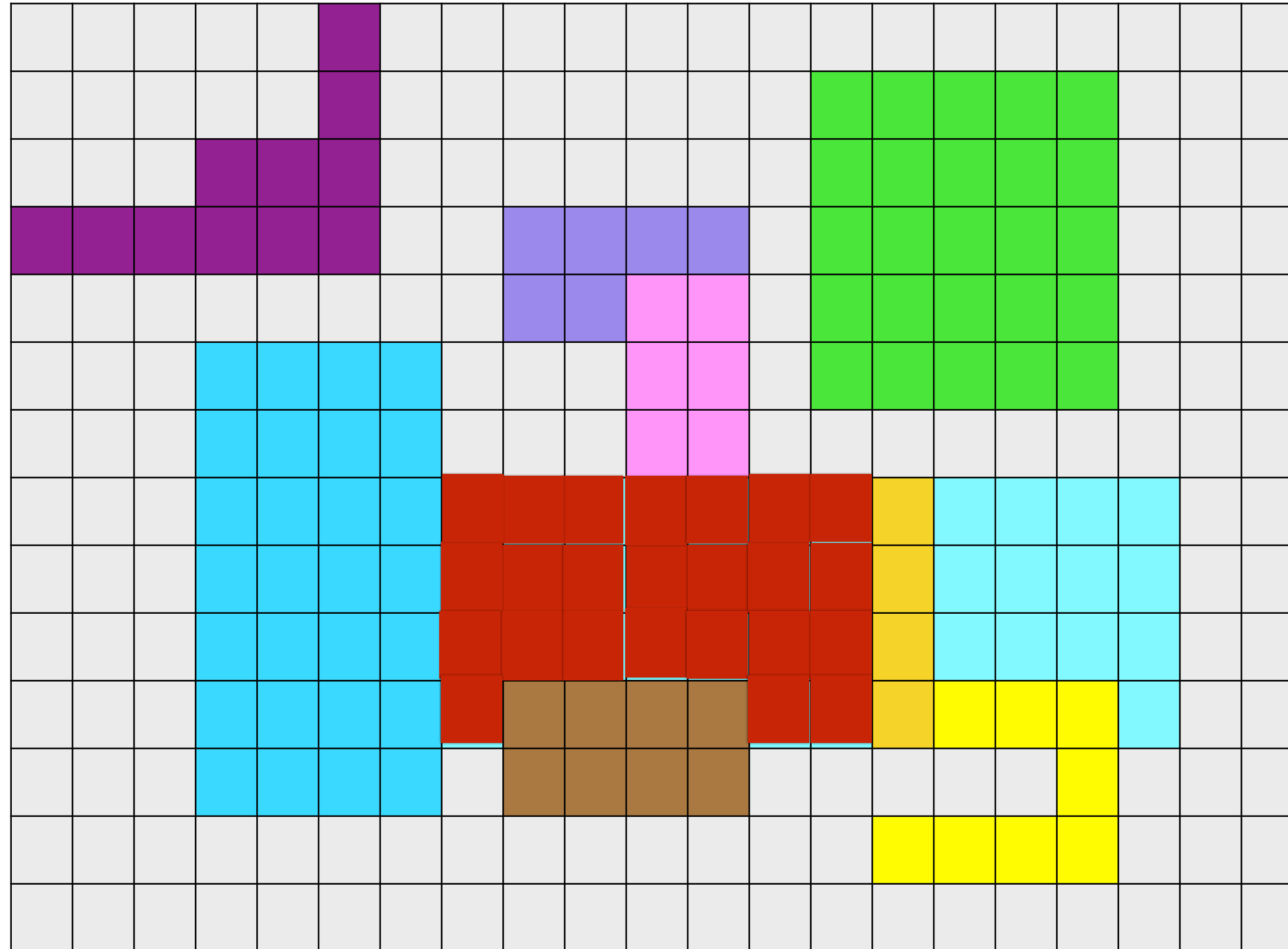
THE PAINT STARTS FROM ONE
PIXEL WHERE YOU CLICK AND
MOVES OUTWARDS

ALL ADJOINING PIXELS WITH THE
SAME ORIGINAL COLOR GET THE
COLOR FILL WITH THE NEW COLOR

THE COLOR FILL MOVES
OUTWARDS TILL IT REACHES A
BOUNDARY WHICH HAS A
DIFFERENT COLOR

PAINTFILL

SAY YOU WANT TO
COLOR ONE PORTION
OF THIS TABLE **RED**



START WITH ONE PIXEL
WHICH HAS SOME ORIGINAL
COLOR

MOVE OUTWARDS, IF THE
NEIGHBORING PIXELS HAVE
THE SAME ORIGINAL COLOR,
COLOR THEM AS WELL

REPEAT TILL THE BORDERS
ARE REACHED

WHAT IS THE BASE CASE?

1. THE CURRENT PIXEL DOES NOT HAVE THE SAME ORIGINAL COLOR, SO REPRESENTS A BORDER
2. THE CURRENT PIXEL IS BEYOND THE SCREEN BOUNDARIES

WHAT IS THE RECURSIVE CASE?

MOVE OUTWARD FROM THE START
PIXEL COLORING INDIVIDUAL PIXELS

A SINGLE PIXEL

THIS REPRESENTS A PIXEL, THE ONLY THING WE CARE ABOUT IS THE COLOR, SO KEEP THE CLASS SIMPLE

```
public static class Pixel {  
    private String color;  
  
    public Pixel(String color) {  
        this.color = color;  
    }  
  
    public String getColor() {  
        return color;  
    }  
  
    public void setColor(String color) {  
        this.color = color;  
    }  
}
```

HELPER METHODS TO GET AND SET THE COLOR

PAINTFILL

THE DISPLAY SCREEN WHICH HOLDS THE PIXELS, A SIMPLE MATRIX REPRESENTS IT

```
public static void paintFill(Pixel[][] displayScreen, int x, int y,
                             String originalColor, String newColor) {
    if (x < 0 || y < 0 || x >= displayScreen[0].length || y >= displayScreen.length) {
        return;
    }
    if (displayScreen[y][x].getColor() != originalColor) {
        return;
    }
    if (displayScreen[y][x].getColor() == originalColor) {
        displayScreen[y][x].setColor(newColor);

        // Pixel on the left
        paintFill(displayScreen, x - 1, y, originalColor, newColor);
        // Pixel on top
        paintFill(displayScreen, x, y - 1, originalColor, newColor);
        // Pixel on the right
        paintFill(displayScreen, x + 1, y, originalColor, newColor);
        // Pixel on the bottom
        paintFill(displayScreen, x, y + 1, originalColor, newColor);
    }
}
```

THE COORDINATES OF THE FIRST PIXEL TO BE COLORED, THE ORIGINAL COLOR AND THE NEW COLOR

DON'T GO BEYOND THE SCREEN

MOVE OUTWARDS COLORING NEIGHBORING CELLS

UPDATE THE COLOR OF THE CURRENT CELL

IF THE CURRENT CELL IS OF A DIFFERENT COLOR THEN WE'VE REACHED A COLOR BOUNDARY

NOTE THAT A PIXEL IN THE
SCREEN ARRAY IS ACCESSED [Y]
[X], SINCE Y IS THE DISTANCE
FROM THE X-AXIS IT IS ACTUALLY
THE ROW IN THE ARRAY

WE HAVE TO VISIT EVERY CELL
TO COLOR IT, THE COMPLEXITY
IS $O(N)$, WHERE N IS THE
NUMBER OF CELLS