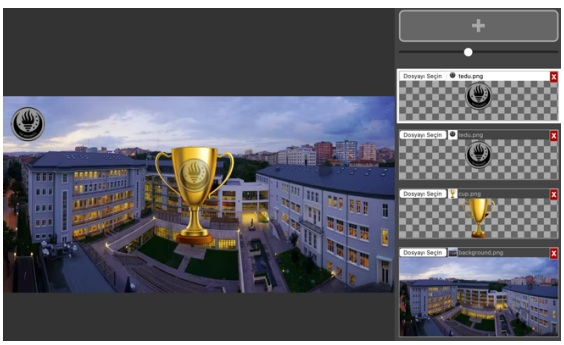
Neslihan Pelin Metin
71047171244

Part 1: Images

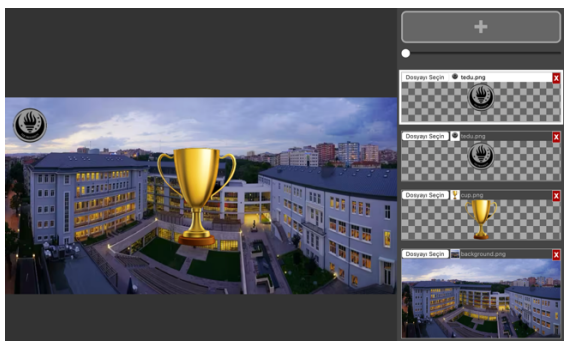
Image 1 cup.png and background.png

Image 2 cup.png, tedu.png and background image

To be able to change the transparency, we can change the alpha value between 0 and 1 by moving the point located on the strip which is under the plus sign. When the alpha is equal to 0, the image is fully transparent and when it is 1, the png is opaque.



(alpha < 0.5)



(alpha = 0)

Neslihan Pelin Metin
71047171244

Image 3 cup.png, tedu.png and background image with changing transparency

Image 4 cup.png, tedu.png and background.png and any additional.png

Part 2: Code Explanation

```
function composite(BackGround, ForeGround, ForeGroundOpacity, ForeGroundPosition) {
    var bgData = BackGround.data;
    var fgData = ForeGround.data;
    var width = BackGround.width;
    var height = BackGround.height;

    var x = ForeGroundPosition.x;
    var y = ForeGroundPosition.y;

    var startX = Math.max(x,0);
    var endX = Math.min(x + ForeGround.width, width);
    var startY = y;
    var endY = y + ForeGround.height;

    for (var i = startX; i < endX; i++) {
        for (var j = startY; j < endY; j++) {

            var bgIndex = (j * width + i) * 4;
            var fgIndex = ((j - y) * ForeGround.width + (i - x)) * 4;
            var alpha = ForeGroundOpacity * fgData[fgIndex + 3] / 255;

            for (var channel = 0; channel < 3; channel++) {
                bgData[bgIndex + channel] = bgData[bgIndex + channel] * (1 - alpha) + fgData[fgIndex + channel]
* alpha;
            }

            bgData[bgIndex + 3] = 255; }}}
```

Neslihan Pelin Metin
71047171244

In the function, I have separated the code into 5 parts, and I will explain them one by one.

1) Position and Range Calculation

Firstly, I am storing the x and y coordinate information of the foreground position as variables x and y. After that, I am calculating the starting and ending coordinates for the compositing operation. For the starting point of x value, I had to use min function from Math library so that the foreground image does not exceed the boundaries of the background image and does not lead to any unintended behavior. Similarly, this applies for the ending position too. So, I used max function for the ending position. When I remove the specified functions, the output result is shown in the below image (Image 5).

Image 5 When the specified functions are not used

However, I did not use these functions for the coordination of y because I did not encounter this problem on the y-axis.

2) Pixel Iteration

I have iterated over the pixels within the specified range of the foreground image to apply alpha blending technique. The first for loop iterates over the x axis while the second loop iterates over the y axis.

3) Pixel Index & Alpha Value Calculation

I determined the position in the image arrays where the pixel values for the background and foreground images can be found. I am calculating the linear index corresponding to the current pixel's position. Then, I multiplied the index by 4 to correctly access the RGBA data for the current pixel.

The variable alpha is calculated next, and it will be used in alpha compositing. It considers both the specified the opacity of the foreground image and the alpha channel of the current pixel in the image.

4) Color Channel Blending

This is the part where the alpha compositing is done. The for loop iterates over the 4 channels, RGBA. It uses the formula of alpha blending which is shown in the image below.

$$c = \propto c_f + (1 - \propto) c_b$$

5) Alpha Channel Update

In the last part, I am setting the alpha channel of the background pixel back to 255, indicating that it's fully opaque and that the alpha compositing operation is complete for that pixel.