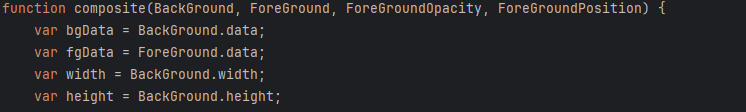
**CMPE 360 HOMEWORK1**

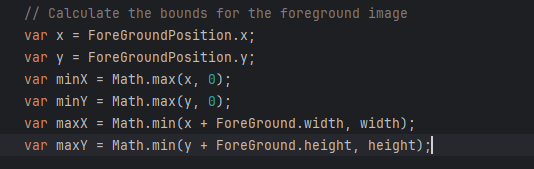
**Introduction**

This report offers a thorough description of the composite JavaScript function's operation. The composite function uses alpha (transparent) blending to combine background and foreground images. Four input parameters are required by the function: BackGround (the background image), ForeGround (the foreground image), ForeGroundOpacity (the foreground's opacity), and ForeGroundPosition (the foreground's location). Let's examine this function's operational logic.

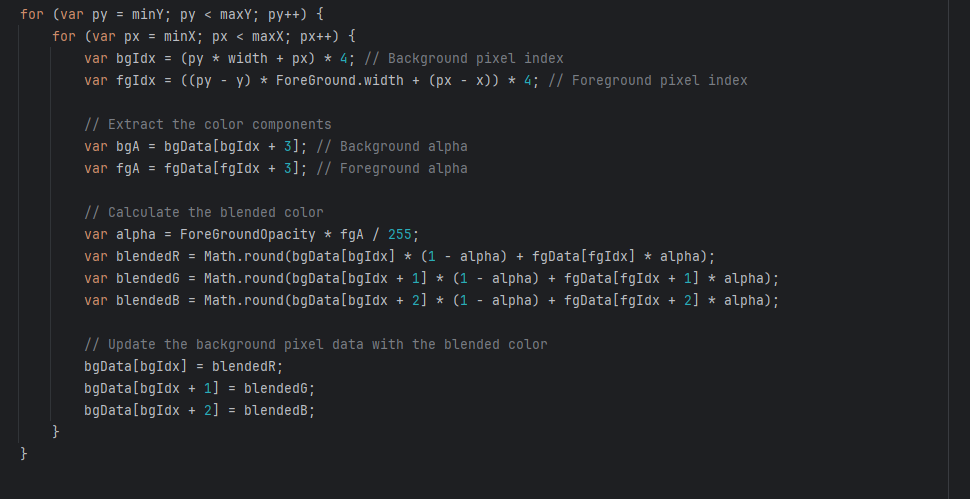
**Functions**



To access the pixel information of the BackGround and ForeGround images, the function initializes the arrays bgData and fgData.The background image's width and height are stored in the width and height variables.



The foreground's position in relation to the backdrop is determined by the coordinates x and y.The foreground zones that will be blended are determined by the minX, minY, maxX, and maxY variables. By doing this, it is made sure that the foreground does not go over the limits of the background. Each pixel is processed through two nested loops (py and px) to blend the foreground and background together.



To access the pixel data of the background and foreground, bgIdx and fgIdx are computed.The foreground's (fgA) and background's (bgA) alpha values are then determined. The pixel's transparency is determined by alpha.Alpha mixing is taken into consideration as the color components of the pixels (bgR, bgG, bgB, fgR, fgG, fgB) are mixed.The background's pixel data is lastly updated with the blended color values.

**Conclusion**

In this manner, the composite function utilizes alpha blending between the background and foreground photos to produce the desired outcome. The resultant image is created by suitably merging transparent foreground pixels (i.e., pixels with a non-black background) with the background. An essential idea in image processing is implemented in this piece of code.