In WebGL, the fragment shader is actually the main part that determines the color of the shape. The fragmentShaderSource in the Redbox.js file is where the actual fragment shader is stored. So gl\_FragColor = vec4(1.0, 0.0, 0.0, 1.0); We can understand that the code is actually where the color comes from. This vec4 function shows color ratios in RGBA format. Of the 4 digits in this vec4, the first digit indicates red, the second digit indicates green, the third digit indicates blue and the last digit indicates opacity. Therefore, the change we make in the values in this section will change the color in the box. The blue color is represented in RGBA format (0.0, 0.0, 1.0, 1.0), so our code is gl\_FragColor = vec4(0.0, 0.0, 1.0, 1.0); By doing this, we actually instruct the fragment shader in WebGL to display the blue color.

A screenshot of a computer program

Description automatically generatedA computer screen shot of a message

Description automatically generated

After we save redbox.js, when Assignment2.html calls the fragmentshadersource, the box will now be rendered blue instead of red. Thus, we will reach our goal.

A red square with white border

Description automatically generatedA blue rectangle with white background

Description automatically generated