CMPE 360 HOMEWORK6

Part 1: Running Local Web Server

To execute and observe the WebGL OpenGL Primitives, a local web server was utilized. The server setup enabled the visualization of graphics in a controlled environment.

Image of the Running Local Web Server

Local Web Server

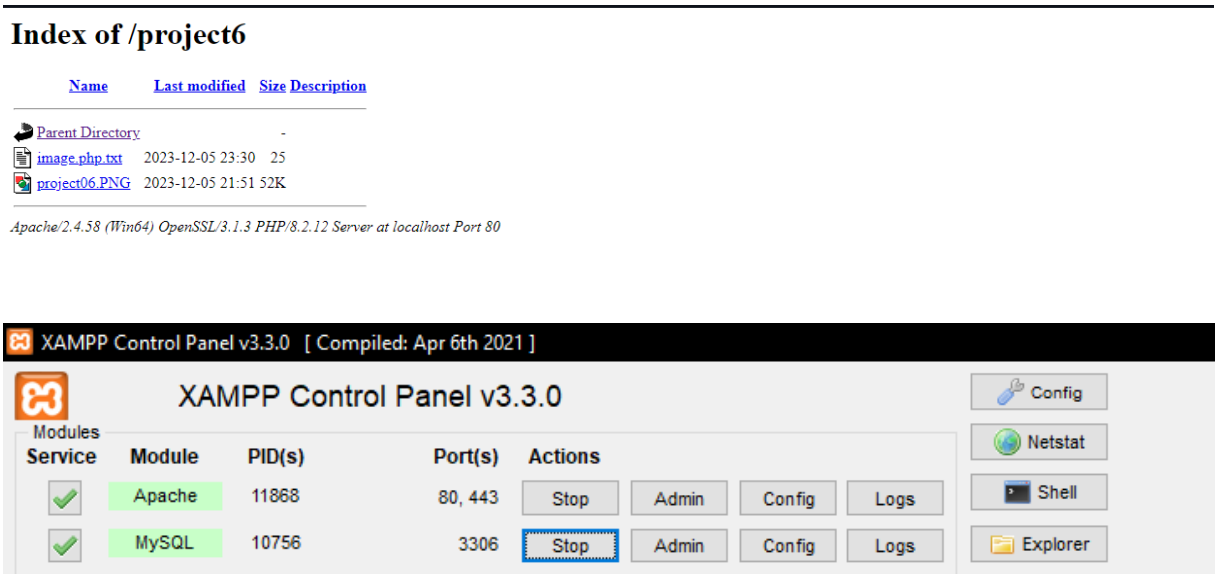

Figure 1 *Local Web Server Image*

Part 2

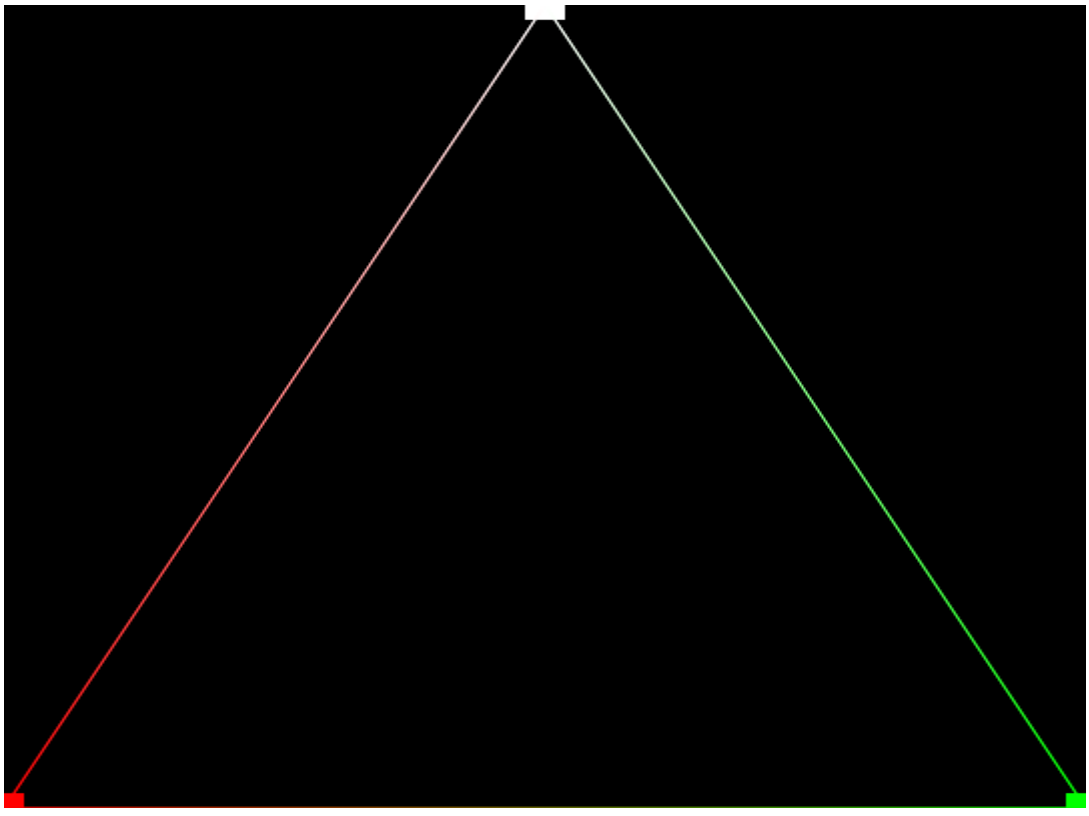

Figure 2 Image

Part 2: WebGL OpenGL Primitives Implementation

HTML Code (WebGlTemplate.html)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>OpenGL Primitives</title>
    <style>
        body { margin: 0; }
        canvas { display: block; }
    </style>
</head>
<body>
<canvas id="myCanvas" width="540" height="540"></canvas>
<script src="Lab1Demo.js"></script>
</body>
</html>
```

JavaScript Code (Lab1Demo.js)

```javascript
function initBuffers(gl) : {color: any, position: any}  {
    const positionBuffer : WebGLBuffer | AudioBuffer  = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);

    const positions : number[]  = [
        0.0,  1.0,
        -1.0, -1.0,
        1.0, -1.0,
    ];

    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(positions), gl.STATIC_DRAW);

    const colorBuffer : WebGLBuffer | AudioBuffer  = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, colorBuffer);

    const colors : number[]  = [
        1.0, 1.0, 1.0, 1.0,  // white
        1.0, 0.0, 0.0, 1.0,  // red
        0.0, 1.0, 0.0, 1.0,  // green
    ];

    gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(colors), gl.STATIC_DRAW);

    return {
        position: positionBuffer,
        color: colorBuffer,
    };
}
```

Explanation of Image and Process

The implemented WebGL OpenGL Primitives satisfy the homework requirements. The image represents a canvas with points and a loop of lines drawn on it. The process can be broken down as follows:

Canvas Initialization: The canvas is set up with dimensions 540x540 pixels.

WebGL Context: WebGL context is initialized, and potential errors are handled.

Shader Programs: Vertex and fragment shader programs are defined to handle vertex positions, colors, and point sizes.

Buffers: WebGL buffers for positions and colors are created and filled with data.

Drawing Primitives: Points and a loop of lines are drawn on the canvas, demonstrating the use of various OpenGL primitives.

Additional Files

Lab1Demo.js: The complete JavaScript file containing the WebGL implementation.

WebGlTemplate.html The complete HTML file containing the JavaScript and WebGL implementation.

Conclusion

The WebGL OpenGL Primitives have been successfully implemented, meeting the requirements specified in the homework. The provided code and image demonstrate an understanding of WebGL concepts, including shader programs, buffers, and drawing primitives. The included files ensure that the code can run seamlessly on other computers.