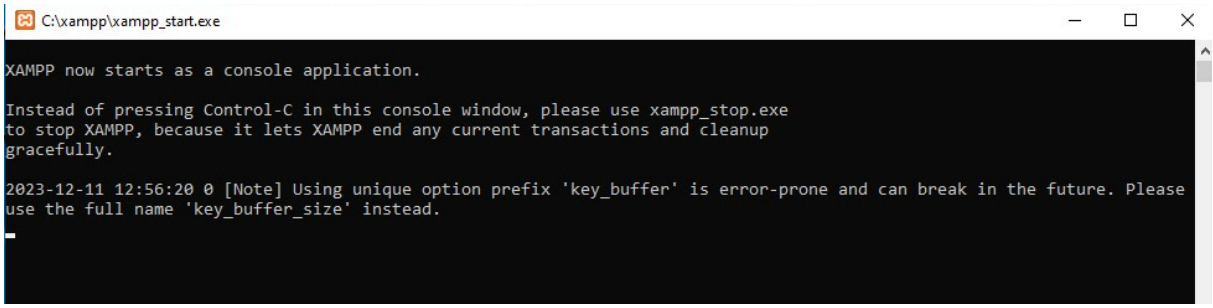
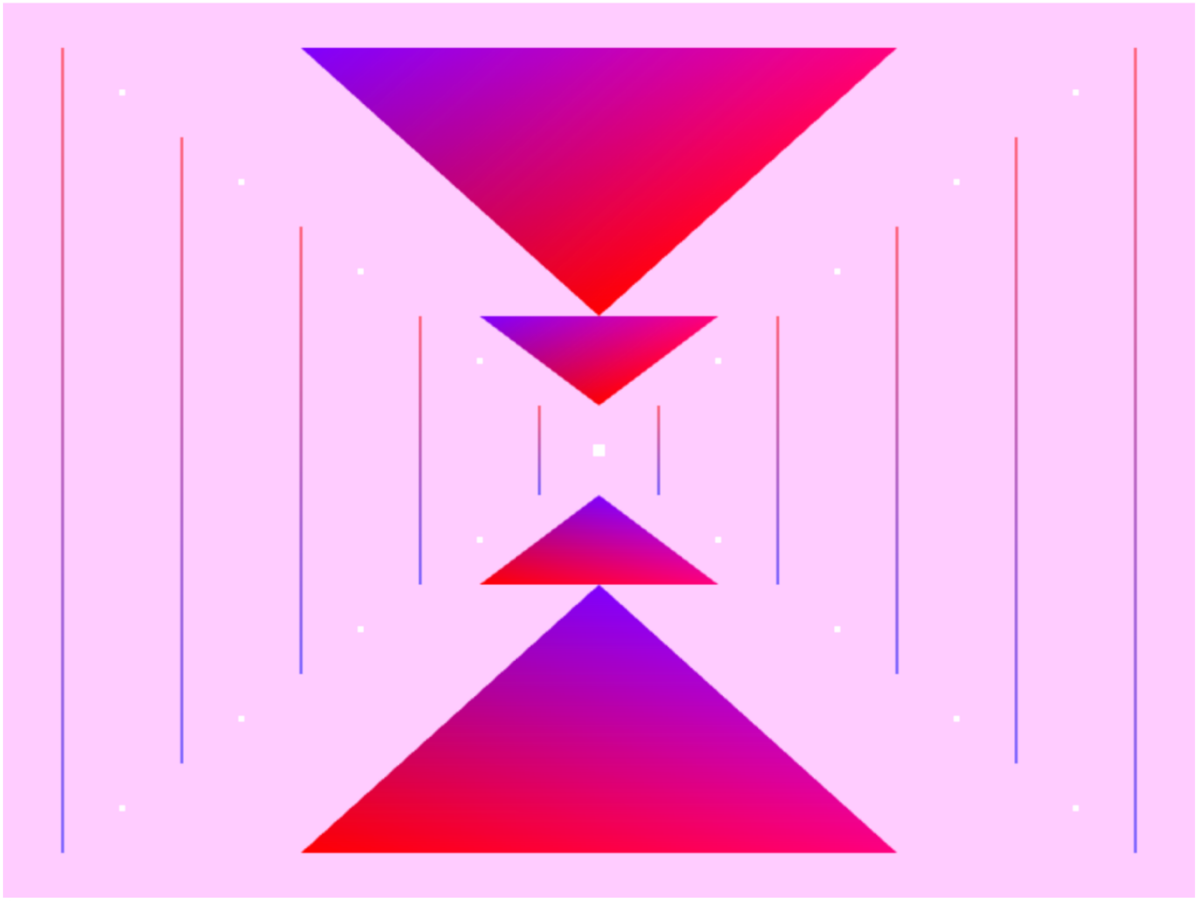
Neslihan Pelin Metin
71047171244

Part 1



Part 2



by Neslihan Pelin Metin

This project aims to demonstrate fundamental WebGL concepts by rendering basic shapes on a canvas using JavaScript. It asks for us to use points, lines, and triangles with different sizes and colors in our drawings. I think that my drawing meets the requirements in that sense.

The HTML file contains a canvas element for rendering WebGL graphics and references an external JavaScript file (project6.js) responsible for handling WebGL operations.

Neslihan Pelin Metin
71047171244

```
<!DOCTYPE html>
<html>
<head>
    <title>WebGL – Cmpe 360 Project 6</title>
</head>
<body onload="InitDemo();">
    <canvas id="game-surface" width="800" height="600">
        Your browser does not support HTML5
    </canvas>
    <br />
    <i>by Neslihan Pelin Metin</i>
    <script src="project6.js"></script>
</body>
</html>
```

The JavaScript code starts with initializing and executing the WebGL rendering. The vertex shader computes vertex positions and colors, while the fragment shader determines how pixel colors are received. After that, the **InitDemo** function respectively initializes the WebGL context, compiles the shaders, defines the vertex data, and organizes the rendering. There are three drawing operations for different shapes.

- **Points**: Draws various points of different sizes using **gl.POINTS**.
- **Lines**: Renders lines using **gl.LINES**.
- **Triangles**: Draws triangles using **gl.TRIANGLES**.

```
// Vertex shader code defining vertex positions and colors
var vertexShaderText = /* ... */

// Fragment shader code defining how fragments receive color
var fragmentShaderText = /* ... */

var InitDemo = function () {
    // ... WebGL context initialization ...

    // Compile shaders, create shader program, and link it
    // ... Shader compilation, linking, and validation ...

    // Define vertices for different shapes with colors
    var vertices = /* ... Definition of vertices for points, lines, and triangles
... */

    // Create and bind buffer for the vertices data
    // ... Buffer creation and binding ...

    // Specify attribute locations for position and color
    // ... Attribute location definitions ...

    // Configure interpretation of attribute data
    // ... Attribute data interpretation ...

    // Enable vertex attribute arrays
    // ... Enabling attribute arrays ...

    // Set point size using a uniform
    // ... Setting point size using a uniform variable ...

    // Perform rendering of points, lines, and triangles
    // ... Rendering operations ...
};
```