

Atividade 1: Introdução aos Sistemas Gráficos com WebGL

Discente: Nicolas Expedito Lana Mendes

Matrícula: 22.1.4028

Docente: Rafael Alves Bonfim de Queiroz

Universidade Federal de Ouro Preto

Departamento de Computação

BCC327 - Computação Gráfica

Fevereiro de 2025

1 Introdução

Este relatório descreve a implementação de uma aplicação gráfica utilizando a API WebGL, conforme solicitado na Atividade 1 da disciplina BCC327 - Computação Gráfica. O objetivo foi criar um programa que renderiza um quadrado azul centralizado sobre um fundo preto, explorando os conceitos de shaders, buffers e posicionamento em WebGL. A escolha do WebGL foi motivada por sua integração com navegadores web, oferecendo uma plataforma prática para experimentação gráfica.

2 API Utilizada

A API empregada foi o **WebGL**, baseada em OpenGL ES, projetada para renderização gráfica 2D e 3D em navegadores. Ela utiliza shaders programáveis para processar vértices e fragmentos, permitindo controle preciso sobre a renderização.

3 Trechos de Código

A seguir, são apresentados trechos relevantes do código implementado:

3.1 Definição dos Shaders

Os shaders foram definidos no código HTML:

```
const vertexShaderSource = `
    attribute vec2 a_position;
    void main() {
        gl_Position = vec4(a_position, 0.0, 1.0);
    }
`;
```

```
const fragmentShaderSource = `
    precision mediump float;
    void main() {
        gl_FragColor = vec4(0.0, 0.0, 1.0, 1.0); // Cor azul
    }
`;
```

3.2 Configuração do Canvas e Vértices

O canvas ocupa a tela inteira, e os vértices definem um quadrado central:

```
canvas.width = window.innerWidth;
canvas.height = window.innerHeight;
gl.viewport(0, 0, canvas.width, canvas.height);

const vertices = new Float32Array([
    -0.5, 0.5, // Canto superior esquerdo
    -0.5, -0.5, // Canto inferior esquerdo
    0.5, 0.5, // Canto superior direito
    0.5, -0.5, // Canto inferior direito
    -0.5, -0.5, // Canto inferior esquerdo
    0.5, 0.5, // Canto superior direito
]);
gl.drawArrays(gl.TRIANGLES, 0, 6);
```

4 Instruções de Execução

Para executar a aplicação:

1. Salve o código em um arquivo `.html` (ex.: `quadrado-azul.html`).
2. Abra o arquivo em um navegador web moderno (como Chrome ou Firefox) com suporte a WebGL.
3. O resultado será um quadrado azul centralizado sobre um fundo preto.

5 Funcionalidade

A aplicação inicializa um contexto WebGL, configura o canvas para ocupar a tela inteira e renderiza um quadrado azul no centro, formado por dois triângulos. O vertex shader posiciona os vértices em coordenadas normalizadas (-0.5 a 0.5), enquanto o fragment shader aplica a cor azul. O fundo preto é garantido pelo `clearColor`, e o uso de buffers otimiza a renderização.

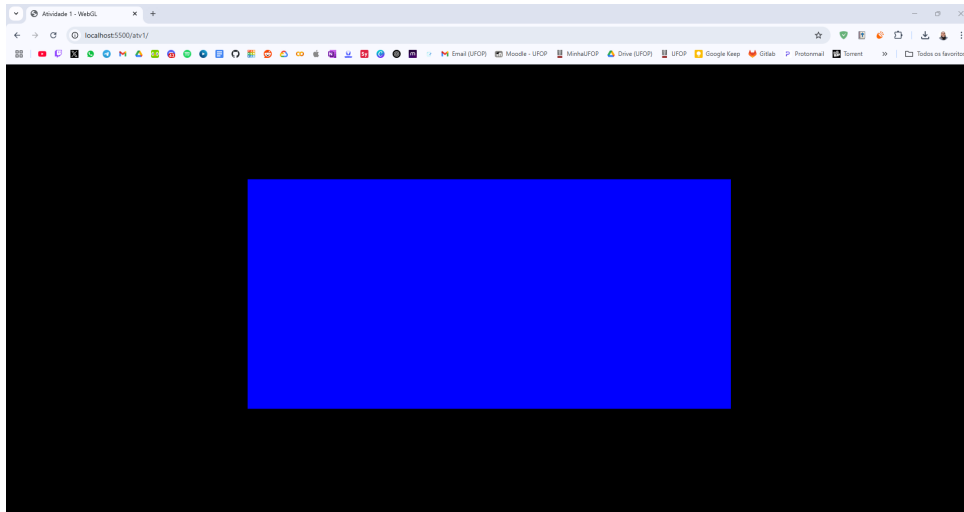


Figure 1: Resultado da aplicação: quadrado azul centralizado com fundo preto.

6 Conclusão

A atividade proporcionou uma introdução prática ao WebGL, destacando a configuração de shaders, o uso de buffers e o posicionamento de elementos gráficos. O resultado foi uma aplicação simples que exibe um quadrado azul centralizado sobre um fundo preto, ilustrando os fundamentos da computação gráfica em um ambiente web. Este exercício foi valioso para entender como manipular cores e posições em WebGL.