

Aluna: Rebecca Nery de Lima Baltoré
Ciência da Computação - 5º período - 2024.2 - Noite

Objetivos

A atividade desenvolvida tem por objetivo utilizar o framework fornecido pelo professor para aplicar o conceito do Algoritmo de Bresenham para o desenho de linhas. O programa tem por objetivos: exibir um ponto, receber dois pontos e traçar uma linha entre eles e desenhar um triângulo dados 3 pontos.

Estratégias

A principal estrutura de dados utilizada foi o array, utilizado para localizar o pixel de referência e aplicar o valor da componente RGB desejada; ou seja, semelhante ao color buffer.

As funções utilizadas foram PutPixel, que recebe as coordenadas e exibe o pixel na cor vermelha; DrawLine, que recebe as coordenadas de dois pontos e, através do Algoritmo de Bresenham, desenha uma linha entre eles; DrawTriangle, que recebe as coordenadas de 3 pontos e, dois a dois, utilizando a DrawLine, desenha linhas entre eles, formando um triângulo.

Resultados

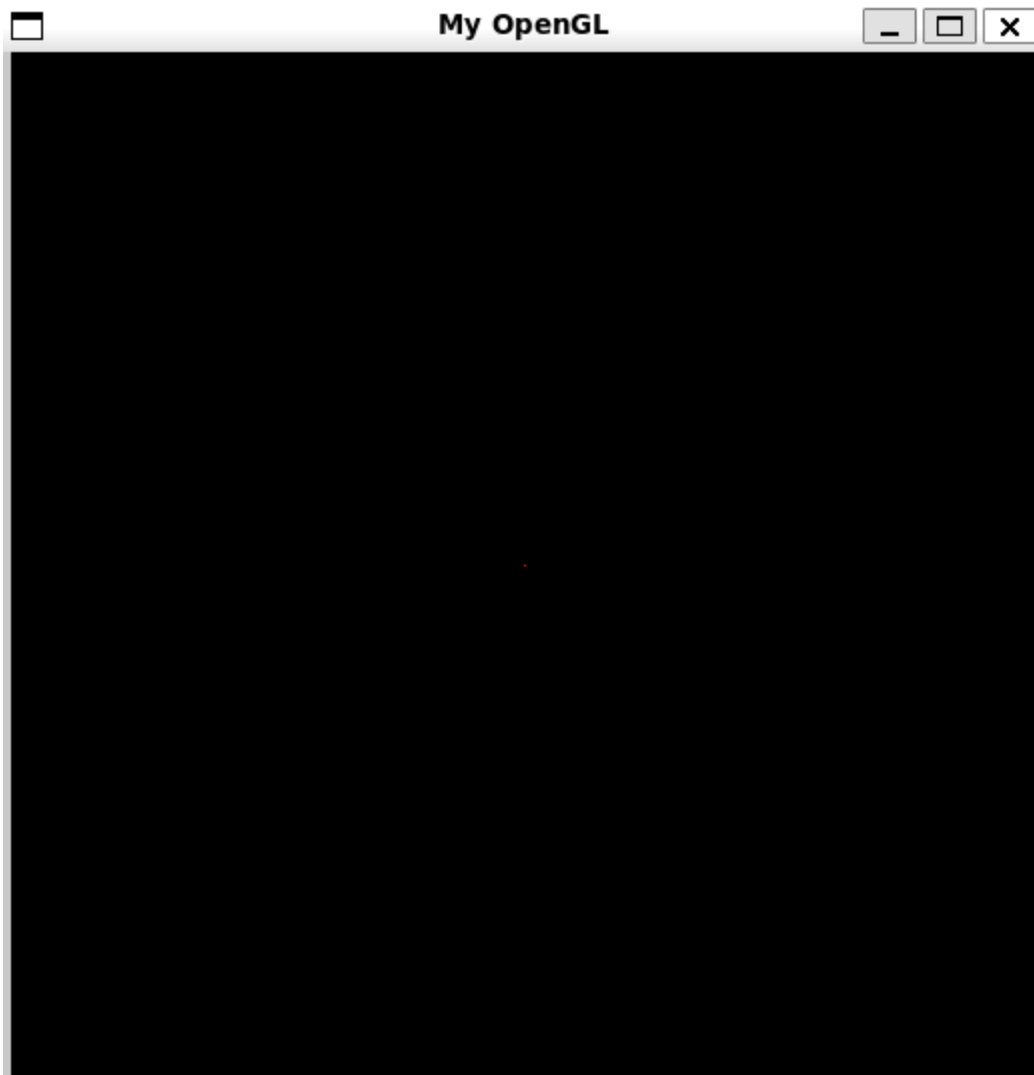
A maior dificuldade foi a instalação do wsl (Ubuntu), juntamente a outras sub-instalações necessárias, que tomaram a maior parte do tempo. Após isso, foi necessário aprender alguns comandos para utilizar no terminal, o que também trouxe alguns empecilhos. Feita toda essa parte, a adaptação do código fornecido pelo professor foi simples.

Possível melhoria: aprimorar o recebimento das variáveis, permitindo que o usuário digite os valores sem, necessariamente, precisar acessar a função diretamente.

Função: PutPixel(int x, int y)

```
void PutPixel(int x, int y) {  
    // Escreve um pixel vermelho na posicao (x, y) da tela:  
    FBptr[4*x + 4*512*y + 0] = 255; // componente R  
    FBptr[4*x + 4*512*y + 1] = 0;    // componente G  
    FBptr[4*x + 4*512*y + 2] = 0;    // componente B  
    FBptr[4*x + 4*512*y + 3] = 255; // componente A  
}
```

Resultado: pequeno ponto vermelho desenhado nas coordenadas fornecidas (x=256, y=256), exatamente no centro da tela (512x512).



Função: DrawLine(int x1, int y1, int x2, int y2)

```
void DrawLine(int x1, int y1, int x2, int y2) {
    int dx = abs(x2 - x1); // Diferença absoluta em x
    int dy = abs(y2 - y1); // Diferença absoluta em y

    int sx = (x1 < x2) ? 1 : -1; // Direção de incremento para x
    int sy = (y1 < y2) ? 1 : -1; // Direção de incremento para y

    int erro = dx - dy; // Valor do erro inicial

    while (true) {
        // Desenha o pixel atual (cor: vermelho, RGBA = 255, 0, 0, 255)
        PutPixel(x1, y1);

        // Se alcançou o ponto final, interrompe o loop
        if (x1 == x2 && y1 == y2) {
            break;
        }

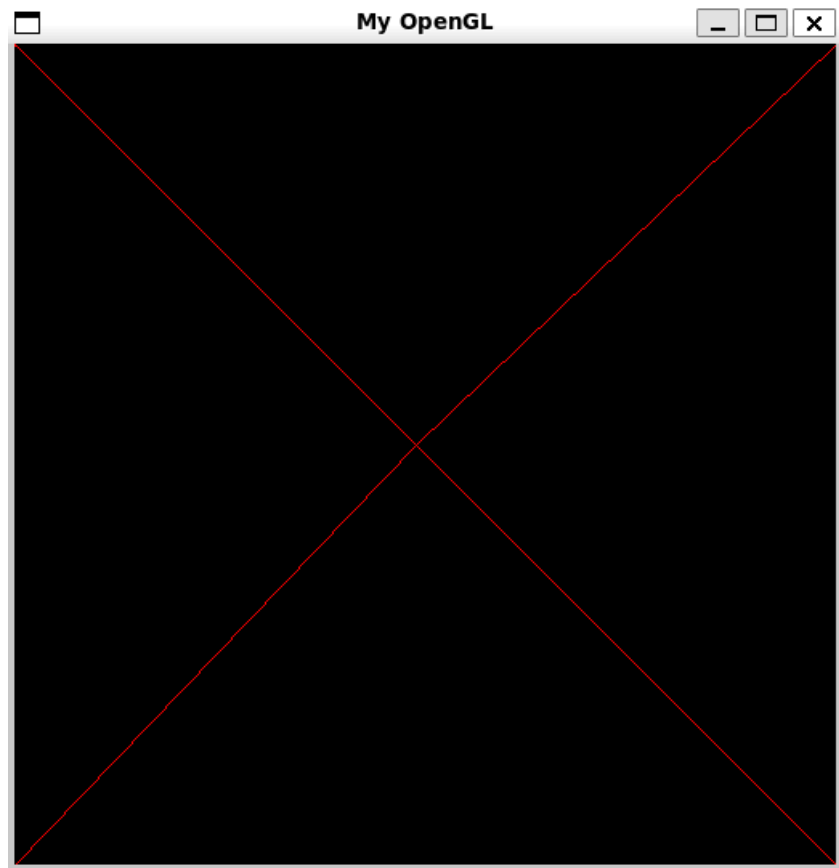
        int e2 = 2 * erro;

        // Ajusta o erro e as coordenadas x
        if (e2 > -dy) {
            erro -= dy;
            x1 += sx;
        }

        // Ajusta o erro e as coordenadas y
        if (e2 < dx) {
            erro += dx;
            y1 += sy;
        }
    }
}
```

Resultado: 4 linhas desenhadas, dadas as seguintes coordenadas:

```
DrawLine(250, 250, 512, 0);
DrawLine(250, 250, 0, 512);
DrawLine(250, 250, 512, 512);
DrawLine(250, 250, 0, 0);
```



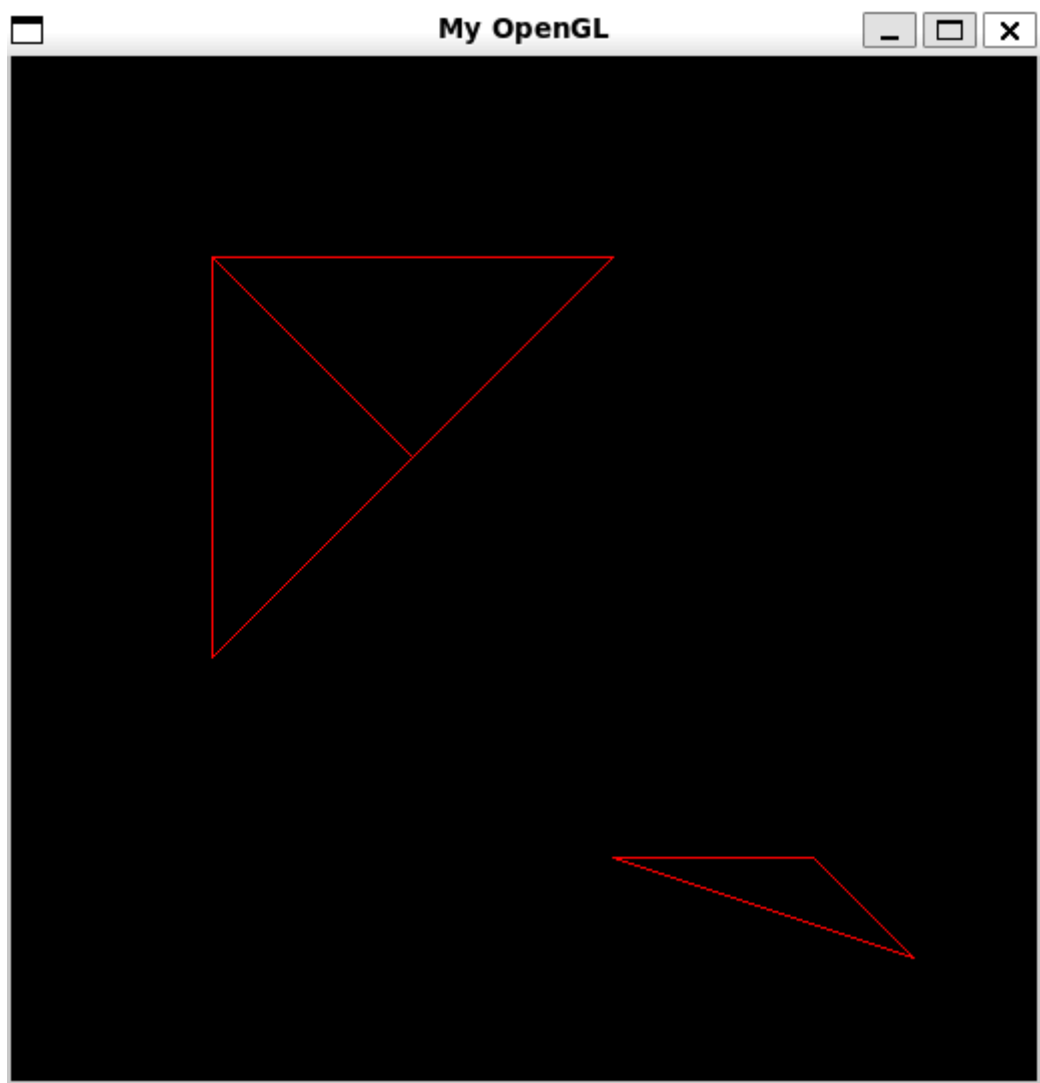
Função: DrawTriangle(int x1, int y1, int x2, int y2, int x3, int y3)

```
void DrawTriangle(int x1, int y1, int x2, int y2, int x3, int y3) {  
    DrawLine(x1, y1, x2, y2);  
    DrawLine(x2, y2, x3, y3);  
    DrawLine(x3, y3, x1, y1);  
}
```

→ Fazendo uso da função DrawLine

Resultado: aplicando as seguintes coordenadas:

```
DrawTriangle(100, 100, 200, 200, 300, 100);  
DrawTriangle(100, 100, 200, 200, 100, 300);  
DrawTriangle(400, 400, 450, 450, 300, 400);
```



Bibliografia

RODRIGO GOMES. *Como instalar o WSL2 no Windows 10/11 - Linux e Windows Lado a Lado para Iniciantes*. 2024. Vídeo. Disponível em: <https://www.bing.com/videos/riverview/relatedvideo?q=como+configurar+e+instalar+o+wsl&mid=84BFC2F0E6B38B76D2CF84BFC2F0E6B38B76D2CF&FORM=VIRE>. Acesso em: [17 set. 2024].