PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE RORAIMA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

DCC703 - COMPUTAÇÃO GRÁFICA

RELATÓRIO DO PROJETO DE RECORTE

ALUNOS:
Marcos Vinícius Tenacol Coêlho - 2021000759

Março de 2025
Boa Vista/Roraima

Resumo

Este relatório descreve a implementação de um programa em Python utilizando o algoritmo de recorte de polígonos de Sutherland-Hodgman, o programa visa demonstrar a aplicação do algoritmo de Sutherland-Hodgman para recortar polígonos conforme uma janela de recorte definida pelo usuário.

Estrutura do Código

dentro: Verificar se um ponto está dentro da janela de recorte;

intersecao: Calcular a interseção entre um segmento e uma borda da janela de recorte;
recortar: Inicia o recorte, passa pela lista de polígonos checando o caso em que ele se encaixa.

Como funciona o algoritmo de Sutherland-Hodgman ?

O algoritmo funciona cortando o polígono iterativamente contra cada uma das bordas da janela de recorte (esquerda, direita, topo e fundo). Para cada borda:

- Identifica se um vértice está dentro ou fora da região.
- Calcula os pontos de interseção quando uma aresta atravessa a borda.
- Mantém apenas os vértices dentro da região de recorte.

Implementação e testes

```
def dentro(p, borda, janela):
    x, y = p
    x_min, y_min, x_max, y_max = janela
    return (x >= x_min if borda == "esquerda" else
            x <= x_max if borda == "direita" else
            y >= y_min if borda == "topo" else
            y <= y_max)
def intersecao(p1, p2, borda, janela):
    x1, y1 = p1
    x2, y2 = p2
    x_min, y_min, x_max, y_max = janela

    if x1 == x2:
        if borda == "topo":
            return (x1, y_min)
        elif borda == "fundo":
            return (x1, y_max)
    if y1 == y2:
        if borda == "esquerda":
            return (x_min, y1)
        elif borda == "direita":
            return (x_max, y1)

    m = (y2 - y1) / (x2 - x1)
    b = y1 - m * x1

    if borda == "esquerda":
        return (x_min, m * x_min + b)
    elif borda == "direita":
        return (x_max, m * x_max + b)
    elif borda == "topo":
        return ((y_min - b) / m, y_min)
    elif borda == "fundo":
        return ((y_max - b) / m, y_max)

    return None

def recortar(poligono, janela):
    for borda in ["esquerda", "direita", "topo", "fundo"]:
        novo_poligono = []
        s = poligono[-1]
```

```
        for p in poligono:
            s_dentro = dentro(s, borda, janela)
            p_dentro = dentro(p, borda, janela)

            if s_dentro and p_dentro:
                novo_poligono.append(p)
            elif s_dentro and not p_dentro:
                i = intersecao(s, p, borda, janela)
                if i:
                    novo_poligono.append(i)
            elif not s_dentro and p_dentro:
                i = intersecao(s, p, borda, janela)
                if i:
                    novo_poligono.append(i)
                novo_poligono.append(p)

            s = p

        poligono = novo_poligono

    return poligono
```

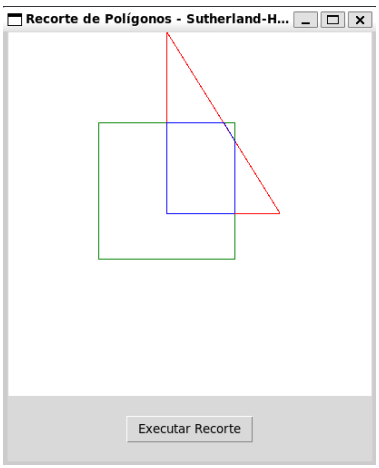
Essa função verifica se os pontos estão dentro ou fora da janela de corte. Caso estejam fora, ela determina em qual setor se encontram e, se necessário, realiza o cálculo da interseção, que pode ser vertical, horizontal ou diagonal.

A partir daqui, serão apresentados os testes com as respectivas coordenadas, onde:

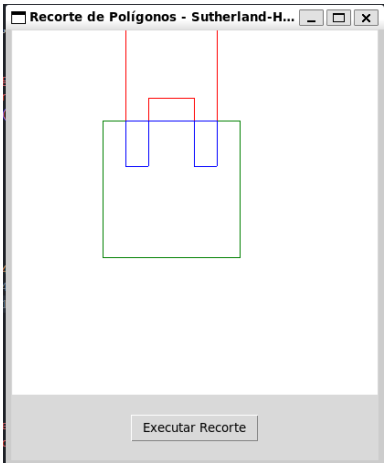
- A área verde representa a janela de corte,
- A vermelha corresponde à figura completa,
- A azul indica a parte recortada.

Sendo as coordenadas:

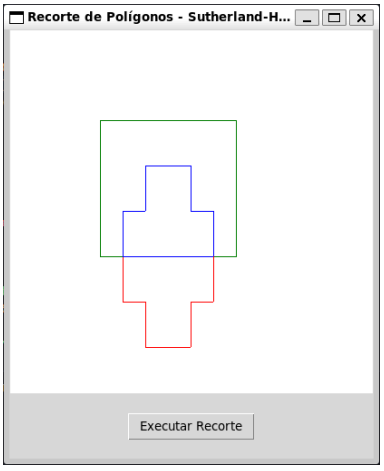
- (3.5,0),(3.5,4),(6,4):



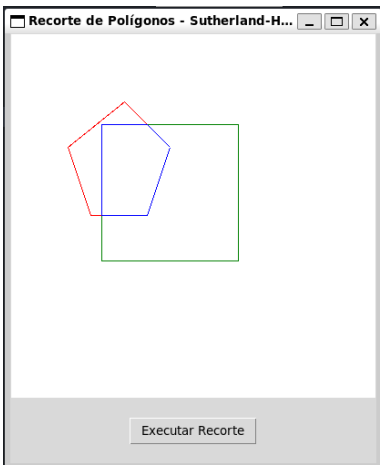
- (2.5, 0), (4.5, 0), (4.5, 3), (4,3), (4, 1.5), (3, 1.5), (3, 3), (2.5, 3):



- (3,3),(4,3), (4,4),(4.5,4),(4.5,6),(4,6),(4,7),(3,7),(3,6),(2.5,6),(2.5,4),(3,4):



- (2.5,1.5),(3.5,2.5),(3,4),(1.75,4),(1.25,2.5):


Recorte de Polígonos - Sutherland-H...
Executar Recorte