
Instituto Federal do Norte de Minas Gerais

IFNMG – Campus Montes Claros

Bacharelado em Ciência da Computação

Disciplina de Computação Gráfica

Manual de utilização do 3º Trabalho Prático

Discente João Gabriel Ferreira

Sumário

1	Alteracoes para a etapa 3	1
1.1	Rotação	1
1.2	Clipping	1
1.2.1	Algoritmos de Clipping	2
2	Programa	3
2.1	Visão Geral	3
2.2	Linguagem e Bibliotecas Utilizadas	3
2.3	Arquivos	3
2.4	Utilização	3
2.4.1	Operações	4

Capítulo 1

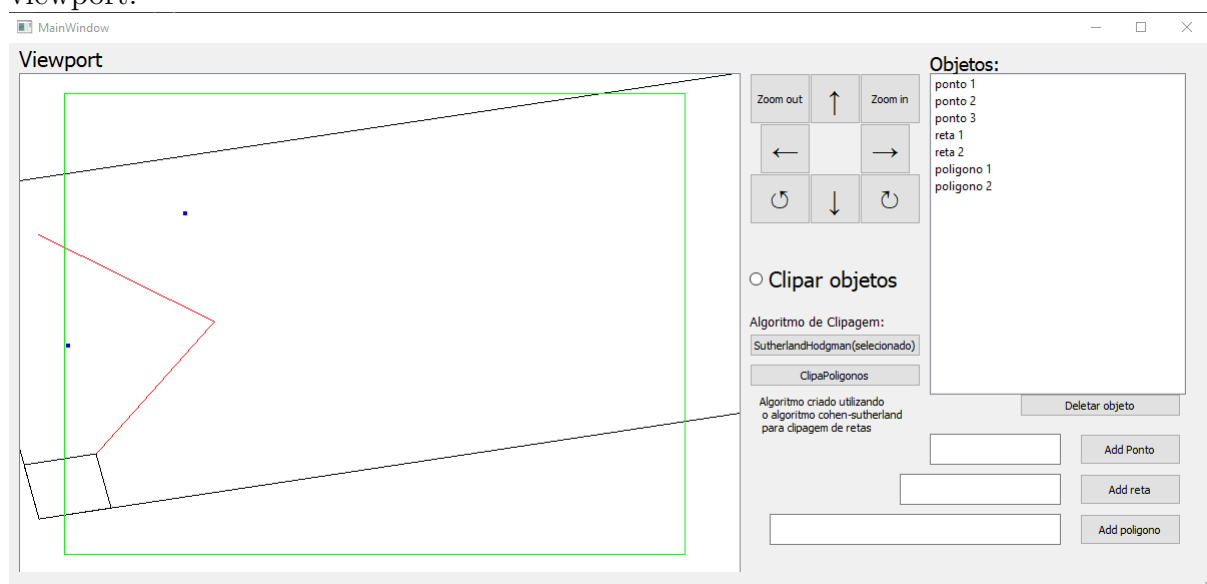
Alteracoes para a etapa 3

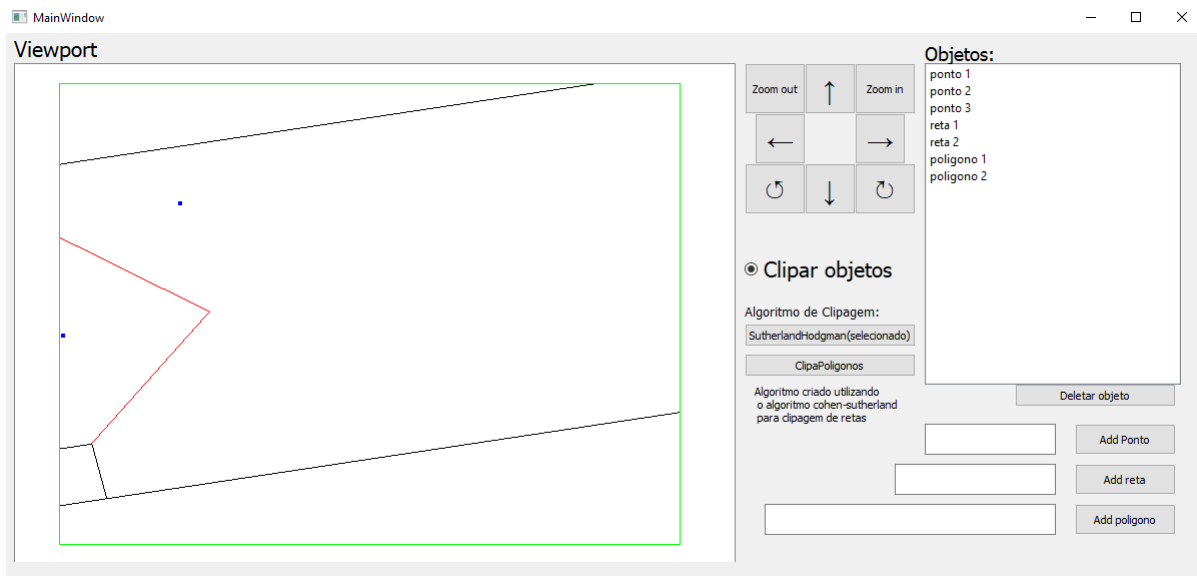
1.1 Rotação

Para a etapa 3, primeiro foram adicionadas as operações de rotação da window. Foi adicionado uma função para multiplicação de matrizes para que as transformações possam ser feitas.

1.2 Clipping

A principal adição foi a possibilidade de realizar o clipping dos objetos na cena. Por padrão, o clipping é desativado. Para ativa-lo, basta marcar a opção "Clipar Objetos". A partir disso, todos os objetos em cena serão clipados, assim como todos os objetos que forem sendo adicionados. Para desativar o clipping, basta desmarcar a opção. A borda em verde representa a viewport.





Além disso, é possível escolher qual algoritmo para clipar polígonos vai ser usado.

1.2.1 Algoritmos de Clipping

Foram implementados os seguintes algoritmos: Cohen-Sutherland, para clipagem de retas, Sutherland-Hodgman para clipagem de polígonos e o algoritmo ClipaPoligonos, desenvolvido utilizando o algoritmo de cohen-sutherland como subrotina.

Além disso, alguns algoritmos auxiliares foram implementados: Um algoritmo para ordenar os pontos em sentido horario e um algoritmo para encontrar o ponto central de um polígono.

Capítulo 2

Programa

2.1 Visão Geral

Este documento serve como guia para utilização do programa desenvolvido como trabalho prático 3 para a disciplina de Computação Gráfica, ministrada pelo Prof. Dr. Wagner Ferreira de Barros. Neste capítulo são incluídos explicações sobre a forma como o programa foi feito, os arquivos que acompanham o programa principal, e a forma correta de execução do programa.

2.2 Linguagem e Bibliotecas Utilizadas

Este trabalho prático foi desenvolvido utilizando a linguagem Python, utilizando a biblioteca PyQt5, e com a ferramenta QTDesigner.

2.3 Arquivos

A pasta enviada contém cinco arquivos, o programa principal, com nome de "tp3cg", um arquivo auxiliar com nome de "Classes", onde foram implementadas as classes para pontos, retas e polígonos, um arquivo auxiliar com nome de "Clipping", onde foram implementados todos os algoritmos referentes ao clipping dos objetos, o arquivo XML de exemplo disponibilizado pelo professor, e este relatório no formato PDF.

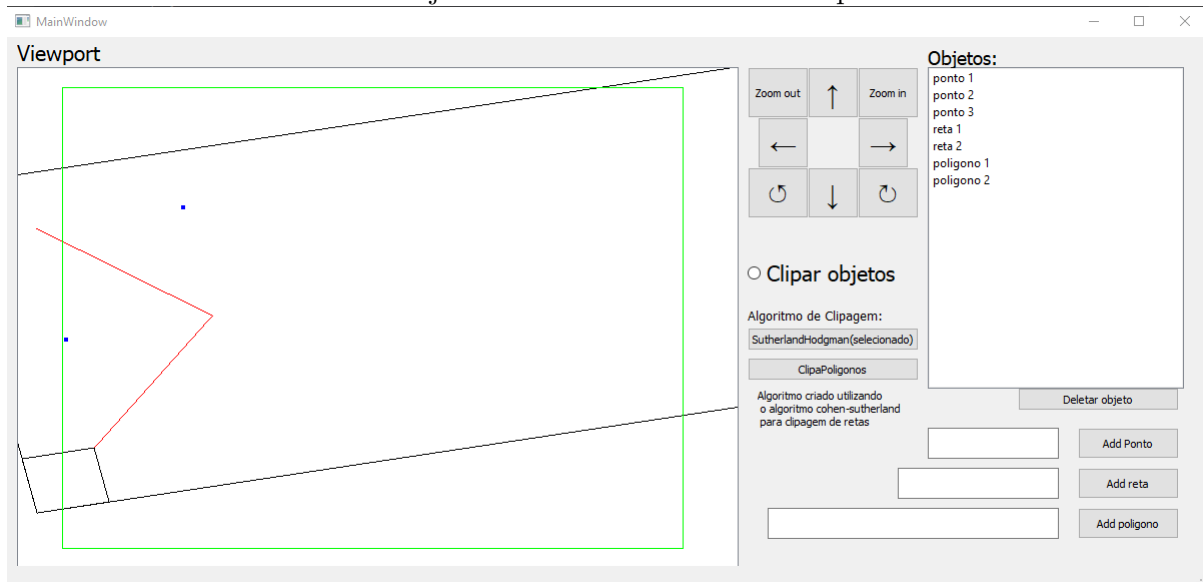
2.4 Utilização

Tendo garantido que a biblioteca PyQt5 está instalada, basta executar o arquivo principal com nome de "tp3cg.py". Além disso, é necessário que o arquivo "tp3cg.py" e os arquivos "Classes.py" e

"Clipping.py" estejam na mesma pasta, pois as definições das classes dos pontos, retas e polígonos e algoritmos de clipping se encontram neles. Há duas formas de rodar o programa: passando ou não um arquivo xml com dados sobre a window, viewport e possíveis pontos, retas e polígonos como argumento. Para executar o programa principal pelo terminal, passando um arquivo xml como argumento, usamos o comando:

```
python tp3cg.py "entrada.xml"
```

Nesse caso, os valores da window e viewport usados serão os definidos no arquivo xml. Além disso, todos os objetos definidos no arquivo xml serão mostrados na viewport, já com suas coordenadas transformadas da window para a viewport. Além disso, todos os objetos serão listados na lista de objetos. Além disso, esse arquivo de entrada precisa conter obrigatoriamente os valores mínimos e máximos para a window e a viewport. O arquivo pode ou não conter dados sobre os objetos. A borda em verde representa os limites da viewport.



Para executar o programa principal sem um arquivo xml, basta usar o comando:

```
python tp3cg.py
```

Nesse caso, os valores da window e viewport serão setados para os valores padrão: (0,0) até (10,10) para a window, e uma janela de visualização de 640 x 480 pixels para a viewport.

2.4.1 Operações

Adicionando objetos

É possível adicionar pontos, retas e polígonos manualmente pela interface do programa. As coordenadas passadas devem ser em relação a window. Ou seja, se os valores mínimos e máximos

para a window forem (0,0) e (10,10), as coordenadas devem estar incluídas nesse limite para aparecerem dentro da viewport. Caso contrário iram aparecer fora. Assim que forem incluídos, as coordenadas dos objetos serão passadas para as coordenadas da viewport e serão corretamente exibidos na viewport.

Adicionando Pontos

Para os pontos, o formato para entrar com um ponto é o seguinte: (x , y). ou seja, abre parenteses, valor do eixo x, vírgula, valor do eixo y, e fecha parenteses. É esperado somente um ponto no input, mais de um ponto acarretará numa falha que encerrará o programa.

O espaçamento não importa. Ou seja, para o input dos pontos, são exemplos de entradas válidas :

(5 , 7)

(5,7)

(5 , 7)

Adicionando Retas

Para o input das retas, é esperado dois pontos. Mais de dois pontos acarretará numa falha que encerrará o programa. O espaçamento entre os dois pontos não importa. Não é esperado nenhum caractere entre os pontos. São exemplos de entradas válidas:

(3,7)(10,7)

(3,7) (10,7)

(3 , 7)(10,7)

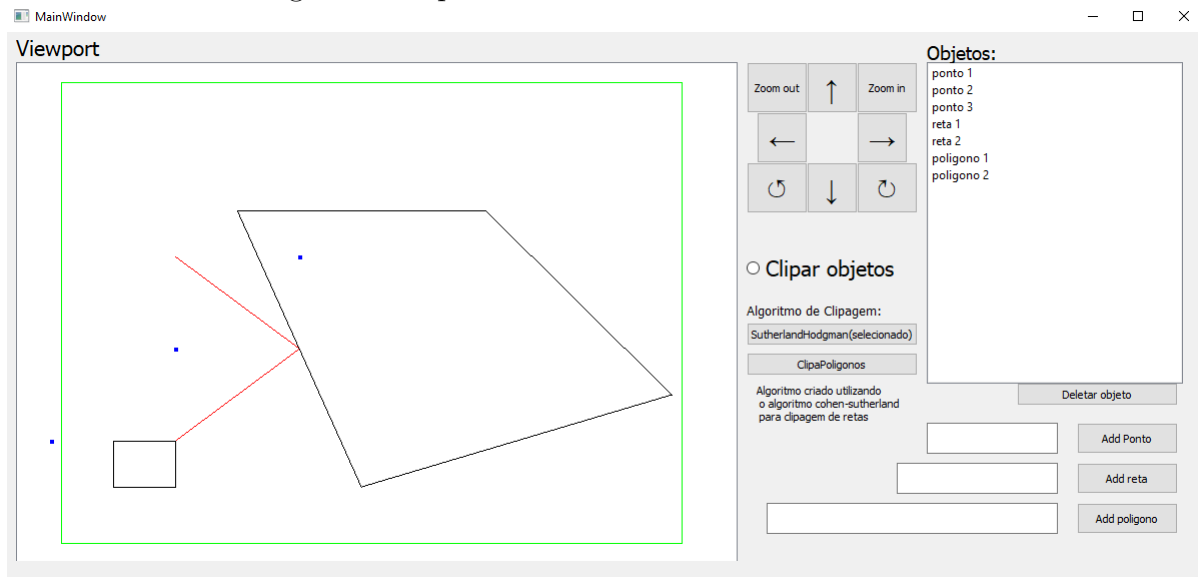
Adicionando Polígonos

Para o input dos polígonos, são esperados pelo menos dois pontos. O espaçamento entre quaisquer dois pontos não importa. Não é esperado nenhum caractere entre os pontos. São exemplo de entradas válidas:

(5,1) (10, 3)(7,7) (3,7)

(5, 1) (10,3) (7 , 7) (3, 7)

Tela após inserir o polígono definido pelos pontos (5,1)(10,3)(7,7)(3,7), com as coordenadas mínimas e máximas para a window sendo (0,0) e (10,10), e com a viewport com visualização de 640x480 com margem de 10 pixels :



Assim que um objeto for inserido, ele aparecerá na lista de objetos.

Deletando objetos

Para deletar um objeto, basta selecionar um deles na lista de objetos e clicar no botão Deletar objetos. O objeto será apagado da viewport.

Aumentar ou diminuir escala

Para aumentar ou diminuir a escala da window, basta usar os botões + e - da interface. Cada ponto da window aumentará ou diminuirá por um fator de 0.1 cada vez que os botões forem clicados.

Transladar viewport

Para transladar a window basta usar os botões com as setas. Cada ponto será transladado por um fator de 0.1 numa determinada direção.