

Documentação do Código: Computação Gráfica - Trabalho Prático 1

1. Introdução

Este documento descreve a implementação de um aplicativo de desenho básico utilizando a biblioteca [Tkinter](#) em [Python](#). O aplicativo permite ao usuário desenhar formas geométricas simples, como linhas, círculos e polígonos, além de aplicar transformações geométricas e recortes a estas formas.

2. Requisitos

Para executar este programa, é necessário ter *Python* instalado na máquina, além das bibliotecas *Tkinter*. Não são necessárias instalações adicionais de bibliotecas externas, uma vez que o *Tkinter* é incluído por padrão nas instalações do *Python*.

3. Estrutura do Programa

O programa é estruturado em torno de uma classe principal chamada *TP_01_CG*, que inicializa a interface gráfica do usuário e define uma série de métodos para desenhar formas geométricas e aplicar transformações. A interface gráfica é criada usando a biblioteca *Tkinter*, e os eventos do mouse são usados para capturar entradas do usuário e clicar na tela e nos botões.

4. Detalhes de Implementação

Cada funcionalidade do programa é implementada como um método dentro da classe *TP_01_CG*. Métodos como *draw_line*, *draw_circle* e *draw_polygon* são usados para desenhar formas

geométricas, enquanto métodos como *translate*, *rotate* e *scale* aplicam transformações geométricas. Há também métodos específicos para implementar algoritmos de recorte que são: *clipping_c* e *clipping_l* para retas, e *clipping_s* para polígonos onde seus algoritmos respectivamente são Cohen-Sutherland, Liang Barsky e Sutherland-Hodgman. Quando foi feito algum procedimento de mudança como transformação e recorte os objetos transladados ou recortados vão aparecer em vermelho e no caso da janela do recote vai aparecer pontilhada em azul para melhor visualização. Em questão da visualização, temos o eixo x e o eixo y. Há *feedbacks* para o usuário quando tem algo de errado, ou que algo ocorreu com sucesso.

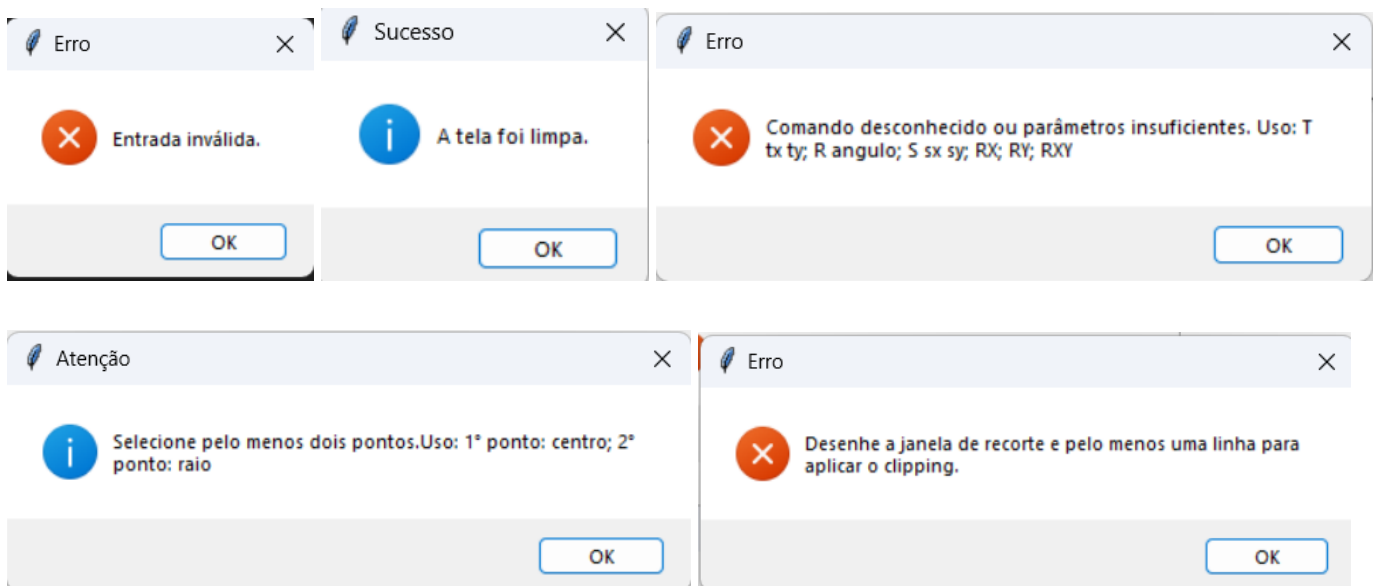


Figura 1

5. Guia de Uso

Para utilizar o programa, execute o script *Python*. A interface gráfica exibirá uma janela onde os pontos podem ser desenhados clicando no botão direito do mouse. Transformações podem ser

aplicadas através da inserção de comandos na caixa de texto disponível na interface, como mostra na figura 2. Já outros comandos, os quais são: desenho, recorte e limpar tela cada um tem seu botão especificado. O botão desenhar reta é o único que vai abrir uma aba menor, como na Figura 3, que pede para escolher um dos procedimentos de desenho de reta.

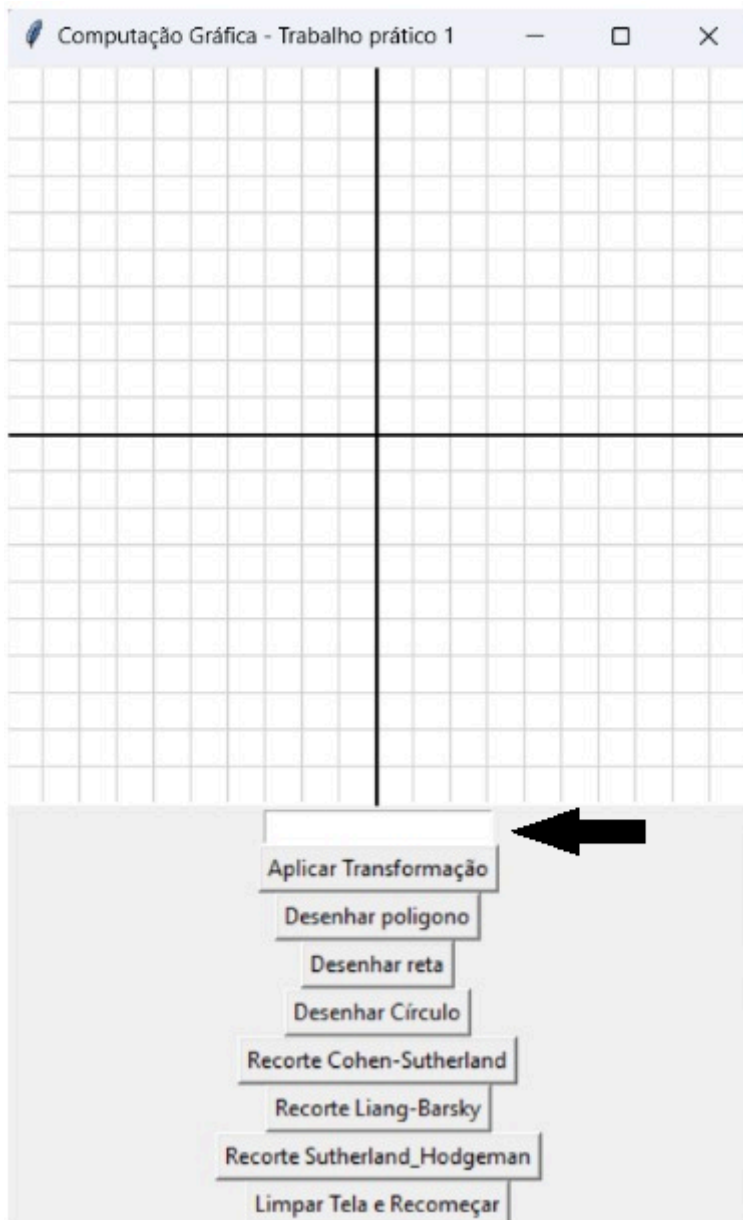


Figura 2



Figura 3

6. Exemplos de Uso

Exemplo 1: Para desenhar uma linha, clique para desenhar dois pontos na tela e selecione o botão desenhar reta e depois o método escolhido.

Exemplo 2: Para desenhar um polígono, clique para desenhar 3 ou mais pontos na tela e selecione o botão desenhar polígono.

Exemplo 2: Para desenhar uma circunferência, clique primeiro onde você quer que seja o centro e depois selecione o botão desenhar círculo.

Exemplo 3: Para aplicar uma translação, insira o comando T seguido pelos valores de deslocamento dx e dy na caixa de texto e pressione Enter.

Exemplo 4: Para aplicar uma rotação, insira o comando R seguido pelo valor do ângulo na caixa de texto e pressione Enter.

Exemplo 5: Para aplicar uma escala, insira o comando S seguido pelos valores sx e sy na caixa de texto e pressione Enter.

Exemplo 6: Para aplicar uma reflexão, insira o comando RX para refletir no eixo x, RY para refletir no eixo y e RXY para refletir nos dois, na caixa de texto e pressione Enter.

Exemplo 7: Para aplicar um recorte de reta, desenhe a quantidade de retas que desejar, como explicado no exemplo 1, depois clique na tela nos locais que serão os 4 pontos que formaram a

janela. Depois clique no algoritmo de retas desejado, Cohen-Sutherland ou Liang Barsky.

Exemplo 8: Para aplicar um recorte de polígonos, desenhe-o, depois clique na tela nos locais que serão os 4 pontos que formaram a janela. Depois clique no algoritmo de polígonos Sutherland-Hodgman.

7. Conclusão e Trabalhos Futuros

Este programa fornece uma base para exploração de conceitos de computação gráfica em um ambiente interativo, com objetivo de reforçar os conceitos aprendidos em aula. Trabalhos futuros podem incluir a implementação de outros algoritmos, como o de preenchimento. Para entender melhor o código e ver os exemplos ao vivo tem o vídeo do youtube explicando isso, segue link.

8. Créditos e Agradecimentos:

Agradeço a professora Rosilane Ribeiro da Motta, por explicar todos os algoritmos implementados e oferecido o pseudocódigo em seus materiais.