Documentação do Projeto: Paint PRO - Computação Gráfica

1. Visão Geral

O Paint PRO é uma aplicação de desenho vetorial desenvolvida em Python com a biblioteca Pygame. O projeto foi criado como parte da disciplina de Computação Gráfica, implementando conceitos fundamentais como rasterização, transformações geométricas 2D e algoritmos de recorte.

A interface do programa é dividida em um painel de ferramentas à esquerda e uma área de desenho (canvas) à direita, permitindo uma interação intuitiva e focada no uso do mouse, minimizando a necessidade do teclado.

2. Estrutura do Código

O código está estruturado de forma modular dentro de um único arquivo (paint\_pro.py), contendo classes principais que organizam a lógica do programa.

Classes Principais

PaintCG: A classe central que gerencia todo o programa. É responsável por:

Inicializar a janela e o Pygame.

Controlar o loop principal de eventos.

Desenhar a interface do usuário (UI) e o canvas.

Gerenciar o estado da aplicação (modo de desenho, cor selecionada, etc.).

Processar todas as interações do usuário (mouse e teclado).

Shape: Representa um objeto geométrico no canvas. Cada forma (linha, círculo, polígono) é uma instância desta classe.

Atributos: type, points (coordenadas dos vértices), color, thickness e selected.

ColorWheel: Cria e gerencia a roda de cores interativa no painel de ferramentas, permitindo a seleção de cores de forma visual.

Enums (Enumerações)

Para tornar o código mais legível e evitar o uso de "números mágicos", foram utilizadas enumerações para definir estados e opções:

DrawMode: Define as ferramentas de desenho disponíveis (SELECT, LINE, CIRCLE, CUT, CROP, etc.).

TransformMode: Define os tipos de transformações geométricas (TRANSLATE, ROTATE, SCALE, etc.).

LineAlgorithm: Permite a seleção entre os algoritmos de rasterização de linha (BRESENHAM, DDA).

3. Funcionalidades Implementadas

Ferramentas de Desenho

O usuário pode selecionar diversas ferramentas para criar formas vetoriais:

Point: Desenha um ponto.

Line: Desenha uma linha reta a partir de dois cliques.

Circle: Desenha um círculo a partir do centro e do raio.

Polygon: Cria um polígono com múltiplos vértices (clique esquerdo para adicionar, clique direito para finalizar).

Freehand: Desenha uma linha suave seguindo o movimento do mouse.

Sistema de Seleção e Transformação

Seleção (SELECT): Permite selecionar uma ou mais formas. As formas selecionadas exibem marcadores em seus vértices. A seleção pode ser feita por clique individual ou arrastando um retângulo.

Transformações: As seguintes transformações podem ser aplicadas às formas selecionadas:

Translação: Mover objetos arrastando-os com o mouse.

Rotação: Girar objetos em torno de seu centroide, com o ângulo definido no painel de controle.

Escala: Aumentar ou diminuir o tamanho dos objetos.

Reflexão: Espelhar objetos nos eixos X, Y ou ambos.

Ferramentas de Corte e Recorte

Cut: Permite ao usuário desenhar um retângulo. As partes das formas que estiverem dentro deste retângulo são removidas, dividindo as formas originais.

Crop: O inverso do Cut. O usuário desenha um retângulo e tudo que estiver fora dele é apagado do canvas.

Navegação no Canvas

Zoom: O zoom pode ser aplicado usando a roda do mouse sobre o canvas ou através dos botões no painel. O zoom é centralizado na posição do cursor.

Pan: O usuário pode navegar pelo canvas arrastando-o com o botão direito do mouse.

Interface do Usuário (UI)

Painel de Ferramentas: Contém todas as opções de ferramentas, cores, espessura e transformações.

Barra de Rolagem: O painel possui uma barra de rolagem que é ativada caso o conteúdo não caiba na altura da janela.

Controles Interativos: Campos de texto e botões permitem a entrada precisa de valores para espessura e rotação.

4. Algoritmos de Computação Gráfica

O projeto implementa diversos algoritmos clássicos exigidos na especificação do trabalho:

Rasterização

A rasterização é o processo de converter uma forma vetorial em pixels na tela.

Linhas:

DDA (Analisador Diferencial Digital): Utiliza aritmética de ponto flutuante.

Bresenham: Uma otimização que utiliza apenas aritmética de inteiros, sendo muito mais rápido.

Círculos:

Bresenham para Círculos: Versão adaptada do algoritmo para desenhar circunferências de forma eficiente.

O usuário pode alternar entre DDA e Bresenham no painel para ver que o resultado visual é idêntico, mas a base computacional é diferente.

Recorte (Clipping)

O recorte é usado pelas ferramentas CUT e CROP.

Liang-Barsky: Este é o algoritmo paramétrico utilizado para calcular com eficiência a interseção entre uma linha e um retângulo. Ele determina quais partes de uma linha estão dentro ou fora da área de corte, sendo a base para as duas ferramentas.