

# Grupo de Pesquisa em Computação Gráfica, Processamento de Imagens e Entretenimento Digital

Seja bem vindo...



## Unidade 3

**Conceitos básicos de 2D:** programação orientada a eventos (mouse), algoritmos de seleção e Bounding Box, modelagem e aplicação de texturas.

**Objetivo:** demonstrar conhecimento no desenvolvimento de aplicativos interativos com o usuário. Interpretar, especificar e desenvolver aplicativos utilizando conceitos básicos 2D.

**Atenção:** os arquivos gerados nas respostas devem ser “zipados”. O material do trabalho deve ser “postado” no AVA3 (NÃO usar links externos). Até as 24:00 do dia em que foi pedido para ser entregue (ver calendário). As cópias receberão nota zero; o professor pode a qualquer momento verificar o trabalho desenvolvido.

**Lembre:** TODAS questões deste trabalho devem ser desenvolvidas em UM único projeto (executarem juntas).

### O que deve ser feito:

O cenário consiste em implementar um Editor Vetorial 2D para “adicionar” e “manipular” polígonos abertos/fechados e convexos/côncavos utilizando somente o teclado e mouse.

### Conhecimento necessário:

- estruturas simples dinâmicas para armazenar os dados gráficos;
- algoritmo de seleção: selecionar polígonos côncavos (não entrelaçados).
- transformações geométricas 2D: movimentar, girar e redimensionar objetos;
- conceitos básicos de grafo de cena.

**Descrição:** o cenário consiste em implementar um Editor Vetorial 2D para “adicionar” e “manipular” polígonos abertos/fechados e convexos/côncavos utilizando somente o teclado e mouse utilizando a biblioteca OpenGL. A aplicação deve permitir “adicionar” interativamente

(“clitando” na tela) polígonos sem limite do número de vértices, e sem limite de polígonos. Além disso, cada polígono pode ter sua própria cor e deve ser possível: apagar/mover vértices e apagar/mover polígonos. Estas funções (apagar/mover) devem se interativas (o usuário deve clicar no polígono ou vértice).

**Observação:** as funções do OpenGL “glTranslate”, “glRotate” e “glScale” não deve ser utilizada para fazer as Transformações Geométricas, e sim deve ser usado a classe “Transform” associada ao objeto gráfico. Assim o método de desenho do objeto gráfico utiliza “Transform” junto com as funções do OpenGL “glPushMatrix”, “glMultMatrixd” e “glPopMatrix”. A justificativa, é que desta forma isolasse o render do OpenGL facilitando, se for o caso, mudar para outra forma de renderizar (por exemplo, DirectX). Para evitar problemas com diferenças de valores entre coordenadas de tela e do espaço gráfico não altere os valores usados no Ortho do exemplo fornecido. Observem que após n a matriz de transformação do referido objeto não é possível s usando os seus vértices originais. Tem de pegar o valor das converter para o espaço gráfico (do Ortho, se for o caso) e co gráfico transformado usando as matrizes de transformações o objeto desejado.

### Questões Práticas:

**Lembre** Todas as questões deste trabalho devem implementar

\_\_\_\_\_ Especificação e documentação

- 01) especificar usando Diagrama de Classes;
- 02) documentar o código (JavaDoc ou DoxyGen).

\_\_\_\_\_ Estrutura de dados dinâmica

- 03) inserir e remover polígonos;
- 04) selecionar vértice usando o mouse para pode mover ou rei caso do mover vértice o valores da coordenada é alterada, e n transformação).

\_\_\_\_\_ Visualização

- 05) exibir o “rasto” ao desenhar os segmentos do polígono.

\_\_\_\_\_ Interação com o usuário (usando teclado e mouse)

- 06) desenhar polígonos (aberto ou fechado) “clitando” na tela;
- 07) trocar as cores dos polígonos (pode ser associado a uma tecla);
- 08) seleção de polígonos côncavos usando detecção hierárquica filtrando primeiro pela BBox e, se for interno a BBox então testar usando Scan Line (exibir a BBox do polígono selecionado).

\_\_\_\_\_ Transformações Geométricas (usar matriz de transformação, não altera os valores dos vértices dos polígonos)

**Material de apoio**

- 09) translação: mover o polígono selecionado;
- 10) escala: redimensionar o polígono selecionado em relação ao centro da sua BBox;
- 11) rotação: girar o polígono selecionado em relação ao centro da sua BBox.

\_\_\_\_\_ Grafo de cena (usar matriz de transformação global para acumular transformações de acordo com o grafo de cena)

12) permitir adicionar polígonos “filhos” num polígono selecionado utilizando a estrutura do grafo de cena;

13) considerar transformação global ao transformar (translação/escala/rotação) um polígono “pai”.

Boa prática!

---

