

UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ – UNIOESTE
CAMPUS UNIVERSITÁRIO DE CASCAVEL
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Disciplina: Computação Gráfica.
Prof^o: Adair Santa Catarina.

TRABALHO 1 DE COMPUTAÇÃO GRÁFICA – TURMA 2024

Para a construção de cenas sintéticas utilizando computadores são necessários diversos algoritmos como a conversão para um sistema de coordenadas de câmera, projeção, mapeamento, recorte etc. Na etapa final deste processo são necessários algoritmos que sejam capazes de desenhar e colorizar regiões na tela do computador. O algoritmo para desenhar um polígono e preenchê-lo, em cor sólida ou não, é conhecido como **fillpoly** e é executado no universo 2D usando coordenadas no SRT.

O algoritmo **fillpoly** identifica quais *scanlines* devem ser processadas, registrando todas as interseções entre elas e as arestas que compõem a fronteira do polígono.

Para colorizar um polígono em cor sólida, a entrada de dados do algoritmo consiste nas coordenadas dos vértices do polígono (SRT) e na cor (RGB) destes vértices. Se todos os vértices têm a mesma cor, então o polígono é desenhado com cor única, mas se possuírem cores distintas a cor dos pixels internos do polígono deve ser calculada.

Como a região interna de um polígono possui muitos pixels faz-se necessário um algoritmo de rasterização preciso e rápido, o que nos leva a explorar algoritmos que trabalhem com aritmética simples. Em outras palavras, algoritmos que usem de aritmética incremental, onde um novo valor é calculado a partir de um valor anterior adicionado de uma parcela fixa.

Neste trabalho o que se pede é a implementação de um software que seja capaz de desenhar e pintar a região interna de polígonos (algoritmo **fillpoly**). Os requisitos são:

- 1) O algoritmo a ser implementado deve usar aritmética incremental (requisito obrigatório), conforme orientações apresentadas em sala de aula;
- 2) Pode ser desenhando qualquer tipo de polígono (convexo, côncavo ou com autointerseção);
- 3) O usuário pode desenhar um ou mais polígonos de qualquer formato na tela do computador, usando para isso cliques do mouse;
- 4) Cada polígono pode ser selecionado individualmente pelo usuário, usando para isso cliques do mouse sobre a interface onde estão desenhados;
- 5) O usuário pode selecionar um polígono e trocar a cor de preenchimento (requisito obrigatório);
- 6) Definida a geometria do polígono e sua cor de preenchimento deve-se pintar apenas os pixels que pertençam à região interna do polígono (requisito obrigatório);
- 7) Deve-se estar disponível ao usuário a opção de pintar ou não pintar as arestas dos polígonos. Neste caso a cor das arestas será sempre fixa (em amarelo);
- 8) Polígonos selecionados podem ser excluídos da lista de polígonos manipuladas pelo software.

Data de entrega: 23/10/2024

A entrega deverá ser realizada através do e-mail prof.stacatarina@gmail.com. Agrupem todos os arquivos do código fonte em um único arquivo .zip e façam o envio. Não incluam no arquivo .zip arquivos binários (executáveis) pois o gmail recusará o recebimento destas mensagens. **Fiquem atentos ao e-mail resposta confirmando a entrega, pois se não o receberem é porque seu trabalho não chegou na caixa postal de destino.**

Incluam na mensagem de envio seu nome completo e as instruções para conseguir executar sua aplicação (Sistema operacional, compilador e versão, bibliotecas necessárias e como instalá-las, etc.). Caso seja por demais complexa a configuração de ambiente criem um repositório no GitHub, Drive ou outro serviço online e enviem o link onde seja possível baixar tudo que é necessário para sua aplicação funcione. Isto não exime a necessidade de enviar o arquivo .zip com o código fonte, apenas simplifica o processo de preparação do ambiente para execução de sua aplicação.

Critérios de Correção (Qualidade do software – 50%):

- 1) Não cumpriu os requisitos obrigatórios (1, 5 e 6). Nota = 0 (zero pontos);
- 2) Software é executado sem erros e atende perfeitamente os 5 requisitos não-obrigatórios. Nota = 100;
- 3) Software é executado sem erros e atende 4 dos requisitos não-obrigatórios. Nota = 96;
- 4) Software é executado sem erros e atende 3 dos requisitos não-obrigatórios. Nota = 92;
- 5) Software é executado sem erros e atende 2 dos requisitos não-obrigatórios. Nota = 88;
- 6) Software é executado sem erros e atende 1 dos requisitos não-obrigatórios. Nota = 84;
- 7) Software é executado sem erros e atende apenas os requisitos obrigatórios. Nota = 80.

Critérios de Correção (Arguição – 50%):

- 1) Agendar horário para apresentação do trabalho com o professor (será disponibilizado um documento compartilhado para agendamento das apresentações que deverão acontecer em Outubro/2024);
- 2) Apresentação individual e oral – trazer a aplicação funcionando em computador próprio ou instalar a aplicação em um computador de laboratório do curso (responsabilidade do apresentador). Caso o computador ou a aplicação desenvolvida não esteja funcionando no início da apresentação agendada a nota da arguição será igual a 0 (zero);
- 3) Responder corretamente e com a devida profundidade as questões realizadas pelo professor;
- 4) A leitura de comentários do código fonte, como resposta às questões realizadas pelo professor será considerada insuficiente, o que levará a nota 0 (zero) na arguição.
- 5) Será inserido, pelo professor, um erro simples no código-fonte da aplicação e que deverá ser corrigido pelo apresentador em tempo máximo de 5 minutos. A falha em localizar e corrigir o erro inserido levará a penalização de 50

pontos na nota da arguição.

Este trabalho deverá ser desenvolvido **individualmente**. Quaisquer dúvidas que persistirem poderão e deverão ser esclarecidas através de consultas ao professor da disciplina.