

CIV 2802 – SISTEMAS GRÁFICOS PARA ENGENHARIA – 2010.1

Trab4: Trabalho de Produção de Desenhos Tridimensionais

Visualização de primitivas gráficas tridimensionais

Objetivo

O objetivo deste trabalho é:

- (a) Implementar rotinas que desenham em uma janela (*canvas*) modelos tridimensionais de primitivas gráficas;
- (b) Implementar rotinas de transformações geométricas para projeção de modelos tridimensionais em um *canvas* bidimensional.
- (c) Implementar uma classe para manipular primitivas do tipo polígono quadrilateral, incluindo as funções que desenham uma versão tridimensional desse tipo de primitiva (prisma cuja base é a primitiva bidimensional).

Este trabalho é uma continuação do trabalho anterior, isto é, a estrutura de dados implementada no terceiro trabalho é utilizada neste trabalho.

Especificação

Assim como no trabalho anterior, é fornecido um programa incompleto e o objetivo do trabalho é implementar as funções que complementam o programa. O entendimento do programa fornecido faz parte deste trabalho.

A estrutura de dados utilizada para armazenar em memória um modelo de primitivas gráficas é descrita na especificação do terceiro trabalho. A solução do terceiro trabalho deve ser copiada para este trabalho. Dessa forma, já estão implementadas funções para abrir e ler um arquivo contendo um modelo e para criar a estrutura de dados. A única novidade com respeito à estrutura de dados das primitivas, é que neste trabalho a “altura” de uma primitiva selecionada pode ser manipulada por um potenciômetro (*valuator*). Dessa forma, a primitiva ganha um aspecto tridimensional, consistindo de um prisma tendo como base a primitiva bidimensional e a altura especificada.

O aluno terá implementar funções que, através de chamadas a funções do sistema gráfico OpenGL, desenham primitivas quadrilaterais tridimensionais (primitivas com “alturas”). Para fazer estas complementações é preciso que se entenda a parte do programa que desenha primitivas circulares tridimensionais (primitivas com “alturas”) em coordenadas reais (no sistema de coordenadas de modelagem) utilizando o sistema gráfico OpenGL.

O trabalho também consiste na implementação de toda a seqüência de transformações geométricas para fazer a projeção de um modelo tridimensional em um *canvas* bidimensional. Para tanto, o programa cria um diálogo secundário com um *canvas* para desenhar de forma bidimensional as projeções das primitivas tridimensionais. É fornecido um módulo que implementa o algoritmo de *z-buffer* para remoção de superfícies escondidas. O desenho bidimensional das primitivas projetadas e tratadas pelo algoritmo de *z-buffer* é feito utilizando o sistema gráfico CD (Canvas Draw).

O arquivo *trab4.led* especifica a configuração da interface gráfica na linguagem LED. Este arquivo fornecido cria um diálogo de interface onde já são especificados uma área de desenho (*canvas*), uma área de mensagens para o usuário, um *valuator* para modificar a altura de uma primitiva selecionada e a associação de teclas do teclado a funções do programa.

Abaixo estão relacionadas as teclas usadas e as correspondentes funções acionadas:

- “H” (*Help*): Informações sobre as ações do programa.
- “Ctrl+Q” (*Quit*): Saida do programa.
- “Ctrl+I” (*Info*): Informações sobre o trabalho.
- “Ctrl+O” (*Open*): Carrega um novo modelo de arquivo.
- “Ctrl+S” (*Snap Shot*): Desenha o modelo em um *canvas* à parte usando o CD.
- “R” (*Reset View*): Reinicializa a visualização.
- “F” (*Fit*): Ajusta o modelo no *canvas*.
- “P” (*Perspective*): Seleciona projeção em perspectiva.
- “O” (*Orthographic*): Seleciona projeção ortográfica.
- “X” (*Proj. X*): Projeção ortográfica no plano $x = 0$.
- “Y” (*Proj. Y*): Projeção ortográfica no plano $y = 0$.
- “Z” (*Proj. Z*): Projeção ortográfica no plano $z = 0$.
- “Ctrl+M” (*Manip.*): Mouse controla manipulação (*screen-based*) do modelo.
- “Ctrl+N” (*Navig.*): Mouse controla navegação (*walk-through*) no modelo.
- “Ctrl+Z” (*Zoom*): Mouse controla zoom do modelo em uma área especificada.

Fornecido e Pedido

Os seguintes arquivos são fornecidos:

- *trab4.led*: Arquivo em linguagem LED que especifica a configuração da interface gráfica.
- *drv.cpp*: *Driver* que faz a interface do programa para o sistema de interface IUP e que dirige o fluxo de controle do programa.
- *prj.cpp*: Arquivo onde estão localizadas as funções de projeto do programa, isto é, as funções que distribuem as tarefas para os outros módulos do programa. Essas funções são definidas como membros globais (*static*) da classe *Prj*. O texto de uma variável *char** de informação, definida neste arquivo, deve ser alterado de forma a indicar o autor do trabalho.
- *prj.h*: Arquivo com a definição da classe *Prj*.
- *prm.cpp*: Arquivo onde se encontra a definição da estrutura de dados e onde estão localizadas as funções que manipulam a estrutura de dados de primitivas gráficas do programa. Essas funções estão organizadas na classe *Prm*. Parte das funções a serem complementadas se encontram neste arquivo (veja comentários `/** COMPLETE HERE: XX */`).
- *prm.h*: Arquivo com a definição da classe *Prm*.
- *circ.cpp*: Arquivo onde estão localizadas as funções que tratam de primitivas circulares. Essas funções estão organizadas na classe *Circ*, que é derivada da classe *Prm*.
- *circ.h*: Arquivo com a definição da classe *Circ*.
- *quad.cpp*: **Arquivo que deve ser criado para implementar a classe *Quad* que trata de primitivas quadrilaterais. Esta classe deve ser derivada da classe *Prm*.**
- *quad.h*: **Arquivo que deve ser criado com a definição da classe *Quad*.**
- *prmio.cpp*: Arquivo com as funções para ler e escrever modelos em arquivo. Essas funções estão organizadas na classe *PrimIO*.
- *prmio.h*: Arquivo com a definição da classe *PrimIO*.
- *dsp.cpp*: Arquivo onde estão localizadas as funções para desenhar o modelo nos dois *canvases*: o que desenha utilizando o OpenGL e o que desenha primitivas projetadas utilizando o CD. Essas funções estão organizadas na classe *Dsp*. Este arquivo deve ser complementado como parte deste trabalho (veja comentários `/** COMPLETE HERE: XX */`).
- *dsp.h*: Arquivo com a definição da classe *Dsp*.

- *t3d.cpp*: Arquivo onde estão localizadas as funções que implementam as transformações geométricas tridimensionais do projeto. Essas funções estão organizadas na classe *T3d*. Este arquivo deve ser complementado como parte deste trabalho (veja comentários **/**/ COMPLETE HERE: XX */**).
- *t3d.h*: Arquivo com a definição da classe *T3d*.
- *zbf.cpp*: Módulo que implementa o desenho bidimensional de primitivas projetadas utilizado o algoritmo *z-buffer* para remoção de superfícies escondidas. Esse módulo está organizado na classe *Zbf*. Este arquivo deve ser complementado como parte deste trabalho (veja comentário **/**/ COMPLETE HERE: XX */**).
- *zbf.h*: Arquivo com a definição da classe *Zbf*.
- Bibliotecas de funções do sistema de interface IUP e do sistema gráfico CD, e os correspondentes arquivos de definições e protótipos.
- Bibliotecas do pacote de manipulação VGL que controla a manipulação da visualização via *mouse*, com os correspondentes arquivos de definições e protótipos.
- Arquivos para compilação e criação do executável (Visual C versão 9).
- Arquivo com um exemplo de primitivas que compõem um quadro de Mondrian modificado.

Pede-se um executável do programa final e os arquivos que foram modificados para implementar o trabalho. Pede-se também, em papel, somente as linhas que foram adicionadas nos arquivos. Estas linhas devem ser indicadas da seguinte forma:

COMPLETE HERE: XX
LINHAS ADICIONADAS