

Computação Gráfica (CG)

Prática – Transformações Geométricas em 2D

O OpenGL consiste em um conjunto de rotinas gráficas gratuitas que serão utilizadas na disciplina. Utilizaremos como base o livro abaixo para nossa prática em OpenGL, para programação em C/C++:

M. COHEN, I. H. MANSSOUR, **OpenGL - Uma Abordagem Prática e Objetiva**, Novatec, 2006.

A. Instalação – OPENGL:

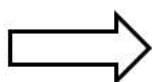
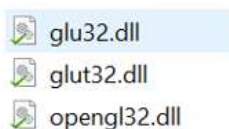
1. Baixar o arquivo com bibliotecas e exemplos sobre OpenGL da Novatec Editora, ou do Moodle da UTFPR:

<https://s3.novatec.com.br/downloads/exemplos-8575220845.zip>

<https://www.moodle.td.utfpr.edu.br/course/view.php?id=919>

2. Instalar bibliotecas necessárias para a execução:

- a) Windows: Copiar manualmente os arquivos DLL para a pasta System32



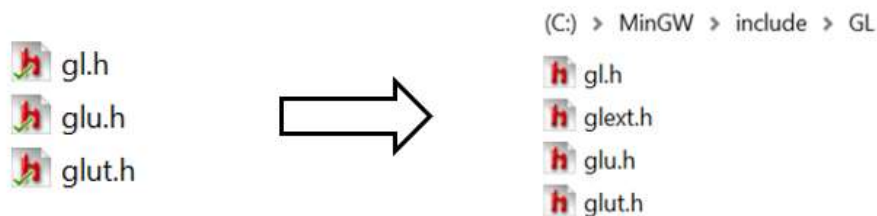
(C:) > Windows > System32

- b) Linux: Devem vir instaladas as bibliotecas por padrão em /usr/lib: libGL.so, libGLU.so e libglut.so

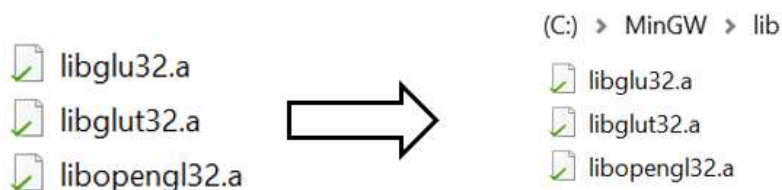
3. Instalar bibliotecas necessárias para compilação:

a) Windows (ex: Codeblocks):

→ Copiar os arquivos .h para o diretório **include/gl** :



→ Copiar os arquivos .h para o diretório **lib**:



b) Linux:

Baixar e instalar os arquivos .h conforme cada distribuição Linux. Exemplo: download Freeglut

→ Verificar se arquivos `gl.h` e `glu.h` estão disponíveis em `/usr/include/GL` ou em diretório equivalente.

→ Verificar se arquivos normalmente instalados por padrão estão disponíveis em `/usr/lib`: `libGL.so`, `libGLU.so` e `libglut.so`

4. Configurar a suíte de desenvolvimento a ser usada e criar um novo Projeto:

→ Codeblocks

→ Visual C/C++

→ Dev-C++

→ Visual Studio .Net

→ Linux...

Os passos irão variar conforme cada ambiente de desenvolvimento e não serão abordados aqui. De qualquer forma, pode-se procurar na internet, ou pedir ajuda!

B. Desenvolver um programa que utilize rotinas OpenGL e que atenda aos seguintes requisitos:

R1) Permitir ao usuário escolher no mínimo as seguintes figuras geométricas planas regulares a serem desenhadas: Triângulo, Quadrado, Hexágono.

R2) Permitir ao usuário digitar as coordenadas dos vértices de cada figura. Neste caso, os polígonos a serem desenhados poderão ser irregulares.

R3) Permitir ao usuário digitar o ponto central e o tamanho da aresta (borda) de cada polígono. Neste caso os polígonos desenhados serão regulares.

R4) Permitir ao usuário utilizar cliques de mouse para determinar os vértices e/ou o centro de cada figura, no caso de polígonos irregulares.

R5) Desenhar a figura na tela conforme parâmetros de entrada fornecidos pelo usuário (requisitos R1 a R4).

R6) O usuário poderá escolher qualquer uma das seguintes TGs a serem realizadas sobre uma figura já desenhada:

- Translação, Escala, Rotação, Reflexão, Cisalhamento

R7) Conforme a TG escolhida, o programa deverá solicitar ao usuário os parâmetros correspondentes e necessários para realizar a transformação. O usuário deverá poder escolher os parâmetros livremente por digitação (proibido fixar os parâmetros no programa). As seguintes funcionalidades devem ser implementadas em cada caso de transformação. seguintes casos específicos:

a) Translação: Deve permitir a translação positiva ou negativa em qualquer direção a escolha do usuário (eixo x, o eixo y).

b) Escala: Permitir aumentar e permitir reduzir o objeto.

c) Reflexão: Permitir fazer a reflexão em relação ao eixo x e em relação ao eixo y.

d) Rotação: O usuário deverá poder escolher o ângulo de rotação, que deve ser indicado em graus, podendo ser positivo ou negativo. A figura deve ser rotacionada na mesma posição em que se encontra (ver requisito R8 para mais detalhes).

R8) Somente a transformação solicitada pelo usuário deve ser aplicada a cada vez e mostrada ao usuário (o programa pode fazer mais transformações em segundo plano, mas para o usuário somente uma irá aparecer).

No caso específico da rotação, a figura deve ser rotacionada sobre o mesmo "local" onde se encontra. Ao aplicar a rotação, sugere-se utilizar a seguinte técnica:

Passo 1: Transladar o objeto para a origem do sistema de coordenadas.

Passo 2: Rotacionar o objeto.

Passo 3: Transladar o objeto de volta para o seu local original.

R9) Após confirmação do usuário o programa aplica a TG escolhida conforme os parâmetros informados (R5, R6 e R7), calcula e mostra o resultado na tela.

R10) Deverá ser desenhado um sistema de referência na tela de visualização, com escala indicativa das distâncias, com as seguintes características:

a) Centralizar o ponto central de coordenadas no centro da tela de visualização.

b) Deve mostrar um eixo de coordenadas x (linha contínua)

c) Deve mostrar um eixo de coordenadas y (linha contínua)

d) Dividir os eixos x e y em 10 segmentos positivos e 10 segmentos negativos, e criar pequenas marcações (um pequeno tracinho) para sinalizar cada divisão.

R11) O software deverá permitir ao usuário alternar entre tela de desenho e de entrada de parâmetros livremente. Deve permitir apagar os polígonos desenhados e redesenhar outros polígonos ou usar outras opções (aplicar mais de uma transformação por exemplo) sem sair do programa.

R12) As entradas de coordenadas devem ser validadas conforme segue:

- As coordenadas devem estar dentro dos limites do sistema de coordenadas do universo (da tela);
- Não permitir a entrada de pontos repetidos (iguais);
- Permitir a entrada de pontos alinhados;
- O desenho de polígonos convexos ou côncavos é permitido para figuras irregulares.