OpenGL

Professor Cláudio Márcio N. A. Pereira

cmnap@ien.gov.br

O que é OpenGL

OpenGL (do inglês, "**Open Graphics Library**") é uma interface de software para dispositivos (hardware) gráficos.

É uma biblioteca gráfica independente de hardware, contendo mais de uma centena de funções e procedimentos para especificação de objetos e operações necessários no desenvolvimento de aplicações gráficas 3D interativas.

Bibliotecas e Headers do OpenGL

Bibliotecas:

OpenGL32.dll- biblioteca principal

Glu.dll e **Glu**.lib - biblioteca de utilidades. Possui alguns objetos pré-definidos.

Glut.dll e **Glut**.lib - funcionalidades para sistemas de janelas

Arquivos de inclusão (headers):

Gl.h ; **Glu**.h ; **Glut**.h - possuem definições para as bibliotecas

OBS:

As DLL's devem ser colocadas no Windows\System

As LIB's são incluídas no projeto

Os headers são incluídos nos programas fonte.

A biblioteca GLUT

Por ser independente de hardware, a biblioteca básica não inclui funções para criação de janelas, pois isto depende da plataforma utilizada. Entretanto, para tornar a programação mais simples, de forma que não seja necessário utilizar as APIs específicas do sistema (para criação de janelas), existe (como um anexo do OpenGL) a biblioteca GLUT (OpenGL Utility Toolkit), que auxilia na criação de janelas básicas. Além desta facilidade, a GLUT possui algumas formas 3D pré-definidas.

Instalando a biblioteca GLUT no Dev C++

As bibliotecas básicas do OpenGL já vem incluídas no Dev C++, entretanto a Glut não vem.
Para instalar a Glut, siga o seguinte procedimento:

- Descompacte glut-devc.zip
- Copie glut.h para á pasta C:\Dev-Cpp\Include\GL
- Copie glut32.def e libglut.a para C:\Dev-Cpp\lib
- Copie glut32.dll para C:\WINDOWS\system32 ou para a pasta onde serão gerados os executáveis
- No projeto (console), vá em:
  [Project]-[Project Options]-[Parameters]
  e na caixa Linker digite: ***-lopengl32 -lglut32 -lglu32***

Criando Janelas com a GLUT

```
void main (int argc, char** argv) {
   glutInit(&argc, argv);  // Inicializa glut
   glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); // Display
   glutInitWindowSize(400,400); // Tamanho da janela
   glutInitWindowPosition(50,50); // Posição da janela
   glutCreateWindow("Hello"); // Cria a janela
   glutDisplayFunc(Display); // Executa função Display
   glutMainLoop(); // Entra em loop
}
```

A função de Exibição

```
void Display () {
    glClearColor(1, 1, 0, 1); // Escolhe cor de fundo
    glClear(GL_COLOR_BUFFER_BIT); // Limpa janela
    // Exibe algo ...
    //  .
    glFlush(); // Força a atualização da janela
}
```

```
void main (int argc, char** argv) {
    ...
    glutDisplayFunc(Display); // Executa função Display
    ...
}
```

Controlando Teclado

```
void Keyboard (unsigned char key, int x, int y) {

    if (key == 27)
    exit(0);  // Termina programa
}
```

```
void main (int argc, char** argv) {
    ...
    glutKeyboardFunc(Keyboard); // Controla teclado
    ...
}
```

Controlando Mouse

```
void Mouse (int button, int state, int x, int y) {

   switch (button) {
      case GLUT_LEFT_BUTTON:
         if (state == GLUT_DOWN) {
            cout << endl << "Mouse Left Down" << endl;
         }
         break;
   }
}
```

```
void main (int argc, char** argv) {
   ...
   glutMouseFunc(Mouse); // Controla mouse
   ...
}
```

Linhas Ligadas

```
// Cria polígono ligando linhas

glBegin (GL_LINE_LOOP);
   glVertex2f (1.25, 1.25);
   glVertex2f (1.75, 1.25);
   glVertex2f (1.75, 1.75);
   glVertex2f (1.25, 1.75);
   glVertex2f (1.25, 1.25);
glEnd();
```

Quadrilátero

```
// Cria quadrilátero

glBegin (GL_QUADS);
   glVertex2f (2.25, 0.25);
   glVertex2f (2.75, 0.25);
   glVertex2f (2.75, 0.75);
   glVertex2f (2.25, 0.5);
glEnd();
```

Triângulo

```
// Cria triangulo

glBegin (GL_TRIANGLES);
   glVertex2f (1.5, 2.0);
   glVertex2f (2.5, 2.0);
   glVertex2f (2.0, 2.8);
 glEnd();
```