



Aula Teórico-Prática nº 1

OpenGL e GLUT



Sumário

- Bibliotecas
- Programação Orientada ao Evento
- Programação com GLUT
- Esqueleto Base
- Primitivas Geométricas do GLUT
- Enunciado
- Crash Course em VS



Bibliotecas

- OpenGL (Open Graphics Library)
 - Funcionalidade Gráfica
- GLU (GL Utilities)
 - Funcionalidade Extra, ao nível gráfico
- GLUT (GL Utility Toolkit)
 - Construção de aplicações independentes do sistema de janelas (Win, XWin)
- GLUI (GLUT-Based User Interface)
 - Biblioteca para construção de interfaces gráficas



Programação Orientada ao Evento

- Define-se uma acção para cada evento relevante para a aplicação
- Exemplos de eventos:
 - Tecla premida
 - Rato movido
 - Janela redimensionada
 - Rato entrou na janela



Programação Orientada ao Evento

- A aplicação é controlada pelo sistema gestor de janelas.

É necessário **definir**
um conjunto de *funções para processar eventos*
e **registar** essas funções no sistema.



Programação com GLUT

```
#include <GL/glut.h>
```

```
int main(int argc, char **argv) {
```

```
    // inicializar o GLUT e a janela
```

```
    // registo de funções para processar eventos (callbacks)
```

```
    // entrar no ciclo de processamento do GLUT
```

```
    return 1;
```

```
}
```



Programação com GLUT

```
int main(int argc, char **argv) {  
  
    // processo de inicialização e definição da janela  
  
    //Registo de callbacks  
  
    // entrada no ciclo de processamento do GLUT  
    glutMainLoop( );  
  
    return 1;  
}
```



GLUT - Inicialização

```
glutInit (&argc, argv);
```

- Função a invocar para inicializar o GLUT.
- Os argumentos a enviar são os mesmos que são recebidos na função *main*.



GLUT - Inicialização

`glutInitDisplayMode (...);`

- Define um conjunto de características do output da janela e do processamento gráfico (mais sobre isto nas aulas teóricas)
- Por enquanto, considerar o parâmetro da função como sendo:

`GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA`



GLUT - Inicialização

```
glutInitWindowPosition(100,100);
```

- Posição do canto superior esquerdo em coordenadas do ecrã.
- Estes valores são um mero pedido, podendo este ser ignorado pelo sistema gestor de janelas.



GLUT - Inicialização

```
glutInitWindowSize(800, 800);
```

- Dimensão da área cliente da janela (largura, altura) em pixels.
- Tal como na função anterior, as dimensões fornecidas são um mero pedido.



GLUT - Inicialização

```
glutCreateWindow("CG@DI");
```

- Função que cria a janela.
- Note-se que a janela só será visível quando for invocado o método:

```
glutMainLoop( );
```



Programação com GLUT

```
#include <GL/glut.h>
```

```
int main(int argc, char **argv) {
```

```
    // inicializar o GLUT e a janela
```

```
    // registo de funções para processar eventos (callbacks)
```

```
    // entrar no ciclo de processamento do GLUT
```

```
    return 1;
```

```
}
```



Registo de Callbacks

`glutDisplayFunc (nome_função);`

- A função utilizada deve ser a responsável por desenhar o conteúdo da janela.
- É obrigatório o registo desta função.
- Assinatura da função registada:

`void nome_função(void);`



Registo de Callbacks

`glutReshapeFunc (nome_função) ;`

- Registrar a função que será responsável por manter as proporções e dimensões do conteúdo da janela em caso de redimensionamento.
- A função registada também será invocada inicialmente, quando se cria a janela.
- Assinatura da função registada:

```
void nome_função(int largura, int altura);
```



Registo de Callbacks

`glutIdleFunc (nome_função) ;`

- A função registada aqui será invocada quando o sistema não estiver a processar outros eventos.
- Pode-se utilizar para realizar animações. No entanto deve-se minimizar a carga desta função de modo a permitir a interactividade.
- Assinatura da função registada:
 - `void nome_função(void) ;`



Programação com GLUT

```
#include <GL/glut.h>
```

```
int main(int argc, char **argv) {
```

```
    // inicializar o GLUT e a janela
```

```
    // registo de funções para processar eventos (callbacks)
```

```
    // entrar no ciclo de processamento do GLUT
```

```
    return 1;
```

```
}
```



Ciclo de Processamento do GLUT

`glutMainLoop();`

- Ao invocar esta função estamos a entrar no ciclo de processamento do GLUT.
- A partir daqui o controle da aplicação é da responsabilidade do GLUT.
- O GLUT invocará as funções registadas quando for necessário ou quando estiver à espera de eventos (`glutIdleFunc`).



Esqueleto Base

- Main

```
int main(int argc, char **argv) {  
  
    // pôr inicialização aqui  
  
    // pôr registro de funções aqui  
  
    // alguns settings para OpenGL  
    glEnable(GL_DEPTH_TEST);  
    glEnable(GL_CULL_FACE);  
    glClearColor(0.0f, 0.0f, 0.0f, 0.0f);  
  
    // entrar no ciclo do GLUT aqui  
    glutMainLoop();  
    return 1;  
}
```



Esqueleto Base

- Reshape Func

```
void changeSize(int w, int h) {  
  
    // Prevent a divide by zero, when window is too short  
    // (you can't make a window with zero width).  
    if(h == 0)  
        h = 1;  
  
    // compute window's aspect ratio  
    float ratio = w * 1.0f / h;  
  
    // Set the projection matrix as current  
    glMatrixMode(GL_PROJECTION);  
    // Load the identity matrix  
    glLoadIdentity();  
  
    // Set the viewport to be the entire window  
    glViewport(0, 0, w, h);  
  
    // Set the perspective  
    gluPerspective(45.0f, ratio, 1.0f, 1000.0f);  
  
    // return to the model view matrix mode  
    glMatrixMode(GL_MODELVIEW);  
}
```



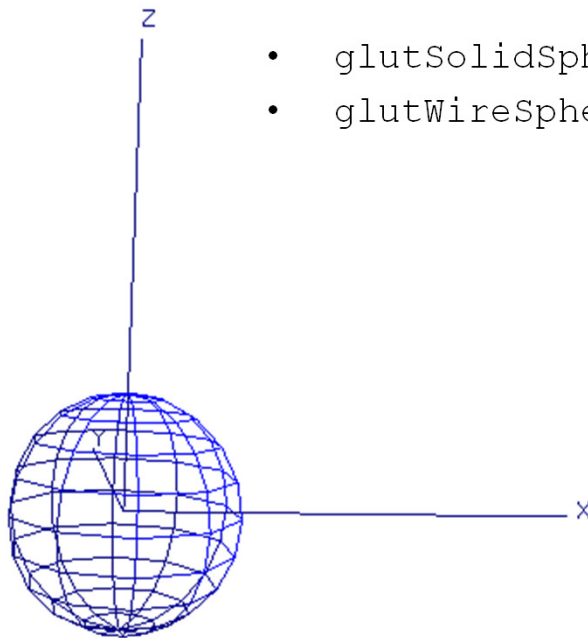
Esqueleto Base

- Display e Idle Func

```
void renderScene(void) {  
  
    // clear buffers  
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);  
  
    // set camera  
    glLoadIdentity();  
    gluLookAt(0.0f, 0.0f, 5.0f,  
              0.0f, 0.0f, -1.0f,  
              0.0f, 1.0f, 0.0f);  
  
    // pôr instruções de desenho aqui  
  
    // End of frame  
    glutSwapBuffers();  
}
```



GLUT - Primitivas Geométricas

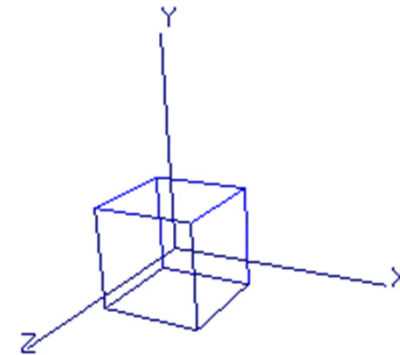


- `glutSolidSphere(float raio, int fatias, int camadas);`
- `glutWireSphere (float raio, int fatias, int camadas);`



GLUT - Primitivas Geométricas

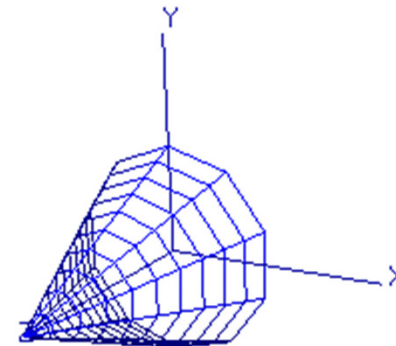
- `glutSolidCube(float dimensão);`
- `glutWireCube (float dimensão);`





GLUT - Primitivas Geométricas

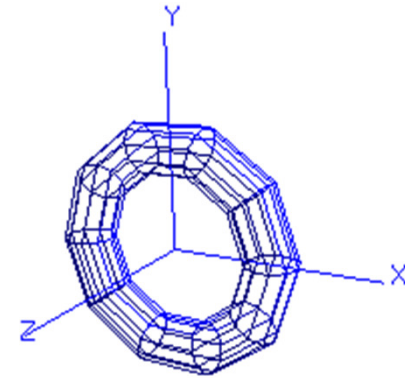
- `glutSolidCone(float raioBase, float altura, int fatias, int camadas);`
- `glutWireCone (float raioBase, float altura, int fatias, int camadas);`





GLUT - Primitivas Geométricas

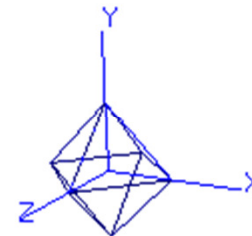
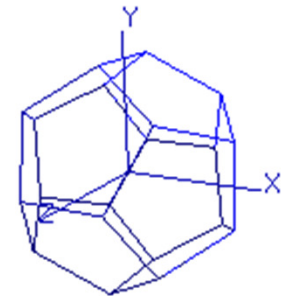
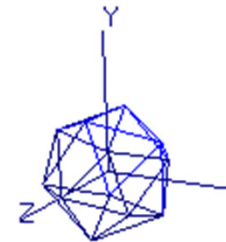
- `glutSolidTorus(float raioInterior, float raioExterior, int numLados, int aneis);`
- `glutWireTorus(float raioInterior, float raioExterior, int numLados, int aneis);`





GLUT - Primitivas Geométricas

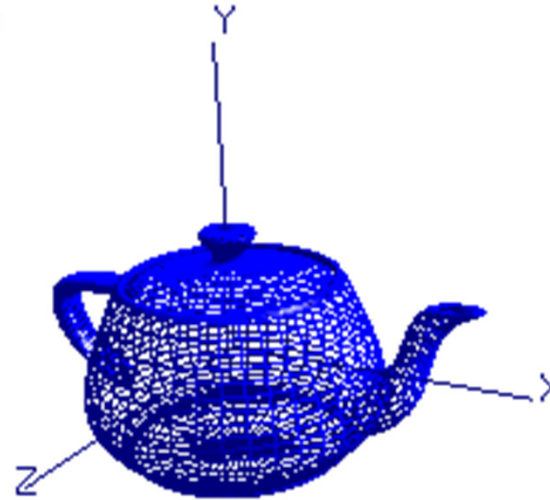
- `glutSolidIcosahedron(void);` (20 lados)
- `glutWireIcosahedron(void);`
- `glutSolidDodecahedron(void);` (12 lados)
- `glutWireDodecahedron(void);`
- `glutSolidOctahedron(void);` (8 lados)
- `glutWireOctahedron(void);`
- `glutSolidTetrahedron(void);` (6 lados)
- `glutWireTetrahedron(void);`





GLUT - Primitivas Geométricas

- `glutSolidTeapot(float dimensão);`
- `glutWireTeapot(float dimensão);`





Exercício prático

- Completar o esqueleto fornecido para criar uma aplicação em OpenGL + GLUT.
- Esta aplicação deve desenhar um bule de chá em modo *wireframe* com linhas brancas sobre um fundo preto.
- Alterar de forma contínua a dimensão do bule de chá (utilizando por exemplo a função seno), criando desta forma uma animação simples.
- Experimentar com as outras primitivas do GLUT.



Crash Course em VS

- File -> New -> Project
- Project Type: Win32
- Templates: Win32 Console Application
- Name: nome da solução (por omissão) e do projecto
- **OK**
- **Next**
- Precompiled header (unchecked)
- Empty project (checked)
- **Finish**
- Botão direito em "Source Files" na janela da esquerda
- Add->Existing Item (deve-se colocar previamente o cpp na pasta do projecto)



Crash Course em VS

- **GLUT** (Ficheiros: glut.h, glut32.lib, glut32.dll)

Sugestão para uma casa arrumada:

Criar uma pasta para pôr todos os ficheiros das bibliotecas (ex: c:\toolkits)

Criar três pastas dentro desta (includes, libs e dlls)

Criar uma pasta (c:\toolkits\includes\GL), pôr dentro desta o ficheiro glut.h

Colocar o glut32.lib na pasta libs, e o glut32.dll na pasta dlls

Adicionar o caminho para as libs e includes no VS

(VS2010: Project->Properties->Configuration Properties->VC++ Directories)

(VS2005-8: Tools->Options->Projects->VC dirs)

Adicionar à variável *Path* (environment variables) o caminho para as dlls (c:\toolkits\dlls)

(Control Panel -> System -> Advanced System Settings -> Environment Variables)