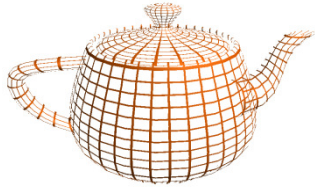




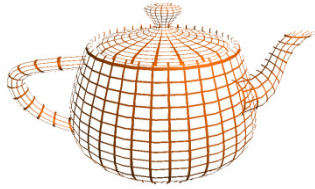
Aula Teórico-Prática nº 1

OpenGL e GLUT



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



Bibliotecas

- GLUT

<http://www.xmission.com/~nate/glut.html>

Ficheiros: glut.h, glut32.lib, glut32.dll

glut.h: colocar na directoria dos includes em *GL/*

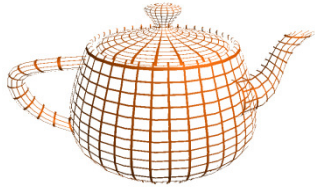
glut32.lib: colocar na directoria das libs

glut32.dll: system (win98) ou system32 (Win NT ou XP)

- GLUI

<http://www.cs.unc.edu/~rademach/glui/>

Ficheiros: glui32.lib (é necessário compilar)



Headers necessários

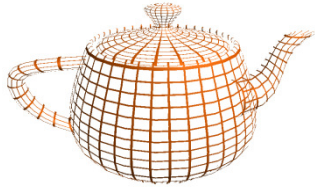
- Com GLUT

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

- Sem GLUT

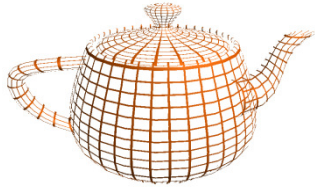
```
#include <GL/glu.h>
```

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



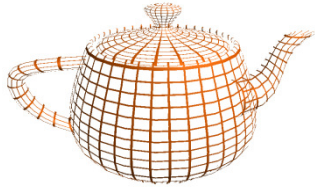
Programação Orientada ao Evento

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



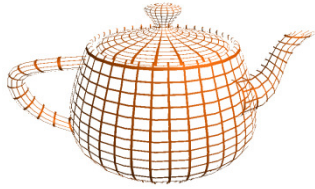
Programação Orientada ao Evento

- A aplicação é controlada pelo sistema gestor de janelas.
- É necessário definir um conjunto de funções para processar os eventos e registar as funções no sistema.



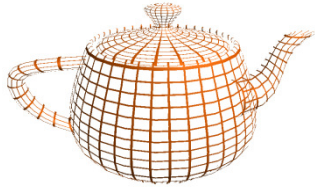
Programação em GLUT

```
void 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  
}
```



Programação em GLUT

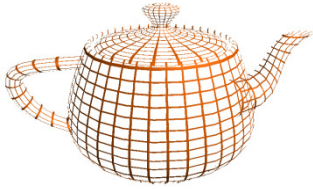
```
void main(int argc, char **argv) {  
    // processo de inicialização e definição da janela  
  
    //Registo de callbacks  
  
    // entrada no ciclo de processamento do GLUT  
    glutMainLoop();  
}
```

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*.



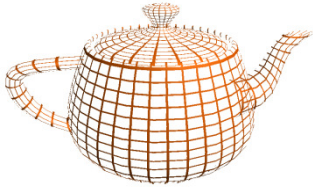
GLUT - Inicialização

- **`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`

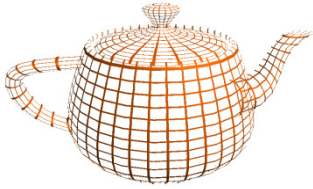


GLUT - Inicialização

- `glutInitWindowPosition(100,100);`

Posição do canto superior esquerdo em coordenadas do ecrã.

Estes valores são um mero pedido, podendo ser ignorado pelo sistema gestor de janelas.

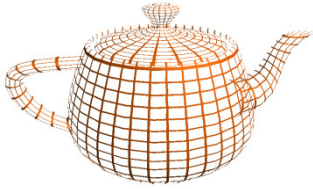


GLUT - Inicialização

- **`glutInitWindowSize(320,320);`**

Dimensão da janela (largura,altura) em pixels.

Tal como na função anterior, as dimensões fornecidas são um mero pedido.

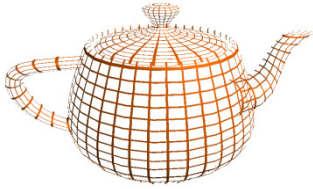


GLUT - Inicialização

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

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

```
glutMainLoop();
```



Registo de Callbacks

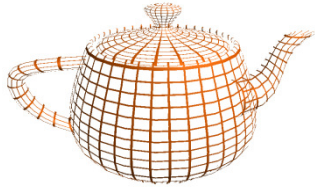
- `glutDisplayFunc(nome_função);`

A função utilizada deve ser a responsável por desenhar o conteúdo da janela.

É obrigatório o registo desta função.

A função cujo nome é fornecido tem a seguinte assinatura:

```
void nome_função(void);
```



Registo de Callbacks

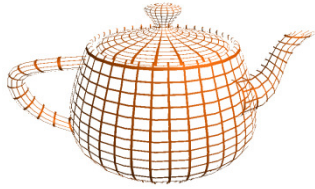
- **glutReshapeFunc(nome_função);**

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

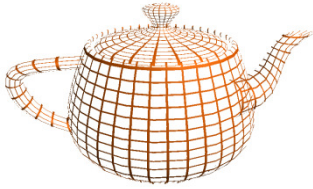
- **glutIdleFunc(nome_função);**

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);
```

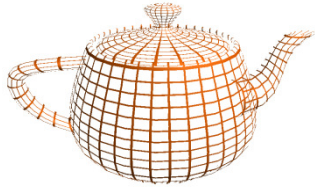
Ciclo de Processamento do GLUT

- `glutMainLoop();`

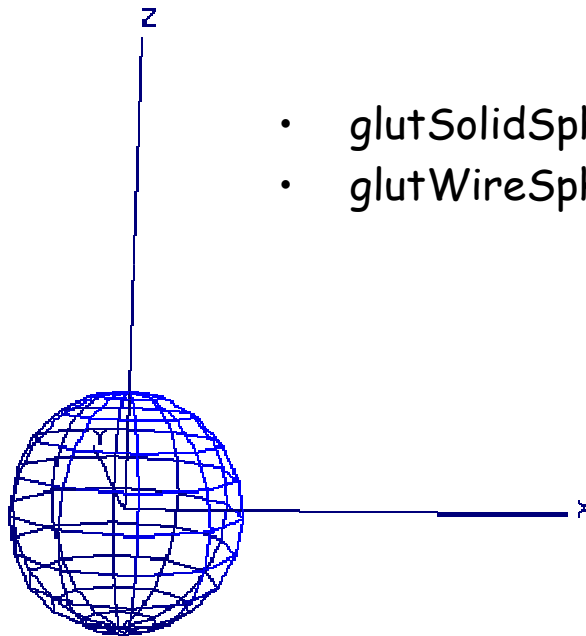
Ao invocar esta função estamos a entrar no ciclo de processamento do GLUT.

A partir daqui o controle da aplicação é da responsabilidade do GLUT.

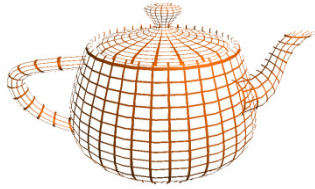
O GLUT invocará as funções registadas quando for necessário ou quando estiver à espera de eventos (`glutIdleFunc`).



GLUT - Primitivas Geométricas

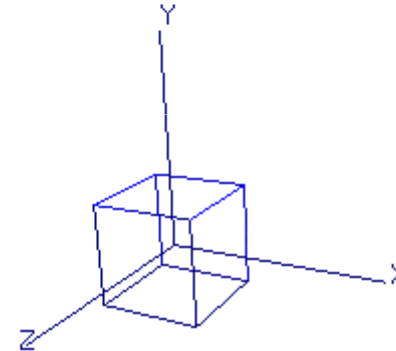


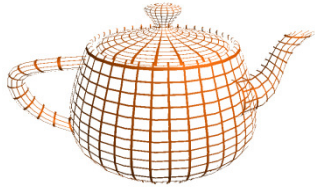
- `glutSolidSphere(float raio,int fatias,int camadas);`
- `glutWireSphere(float raio,int fatias,int camadas);`



GLUT - Primitivas Geométricas

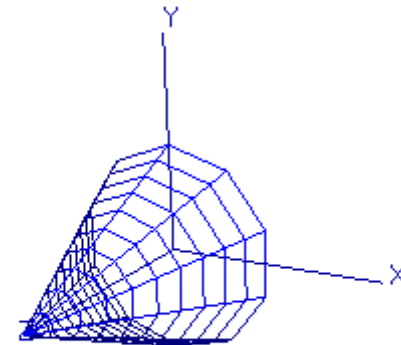
- `glutSolidCube(float dimensão);`
- `glutWireCube(float dimensão);`

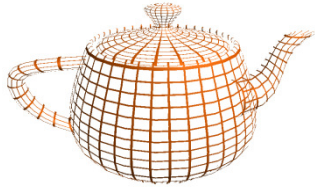




GLUT - Primitivas Geométricas

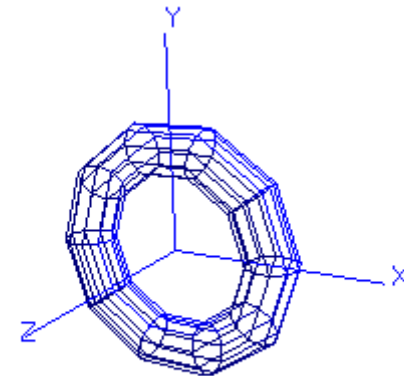
- `glutSolidCone(float raioBase, float altura, int fatias,int camadas);`
- `glutWireCone(float raioBase, float altura, int fatias,int camadas);`

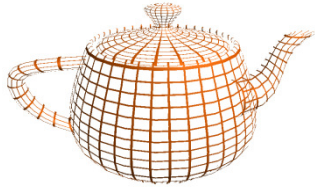




GLUT - Primitivas Geométricas

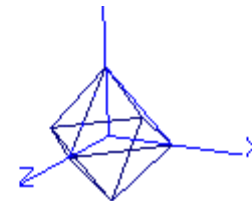
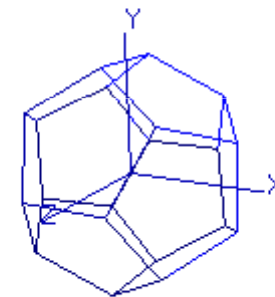
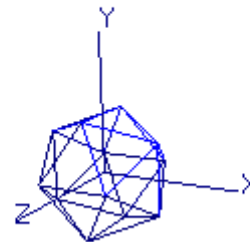
- `glutSolidTorus(float raioInterior, float raioExterior, int numLados, int aneis);`
- `glutWireTorus(float raioInterior, float raioExterior, int numLados, int aneis);`

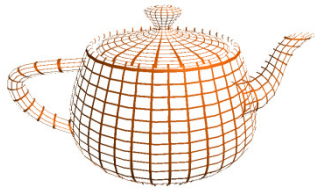




GLUT - Primitivas Geométricas

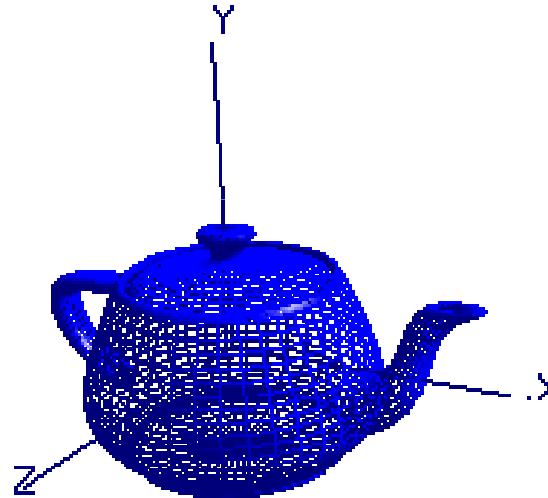
- `glutSolidIcosahedron(void)`; (20 lados)
- `glutWireIcosahedron(void)`;
- `glutSolidDodecahedron(void)`; (12 lados)
- `glutWireDodecahedron(void)`;
- `glutSolidOctahedron(void)`; (8 lados)
- `glutWireOctahedron(void)`;
- `glutSolidTetrahedron(void)`; (6 lados)
- `glutWireTetrahedron(void)`;

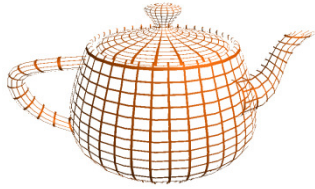




GLUT - Primitivas Geométricas

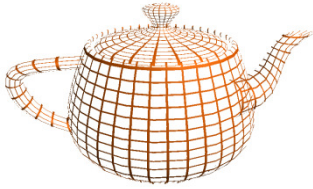
- `glutSolidTeapot(float dimensão);`
- `glutWireTeapot(float dimensão);`





Exercício prático

- Completar o esqueleto fornecido para criar uma aplicação em OpenGL + GLUT.
- Esta aplicação desenhará um pote 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 básica.
- Experimentar com as outras primitivas.



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 por questões de organização)