

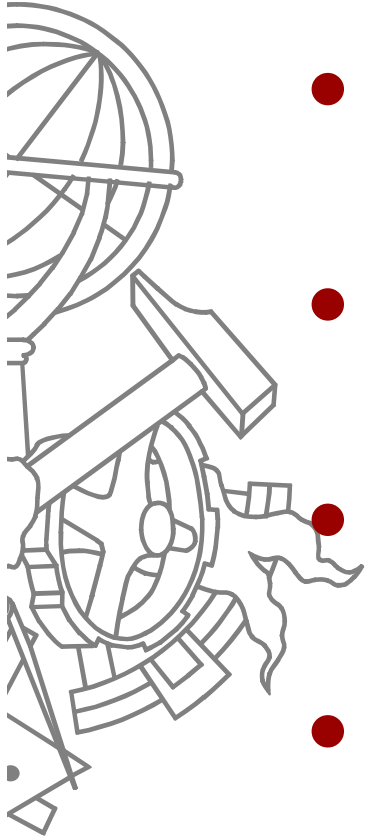
CG com OpenGL

Paulo Gandra de Sousa

psousa@dei.isep.ipp.pt

Sistemas Gráficos e Interactivos
Instituto Superior de Engenharia do Porto

Conteúdo



- O que é?
 - Exemplos
- API do OpenGL
 - demos
- Desenho, iluminação, projecções, transformações, ...
- Exemplos de aplicações



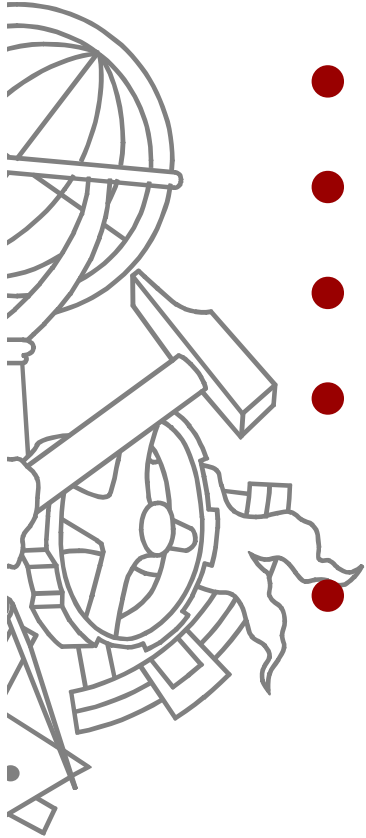
Introdução ao OpenGL

Aula 1

Sistemas Gráficos e Interactivos
Instituto Superior de Engenharia do Porto

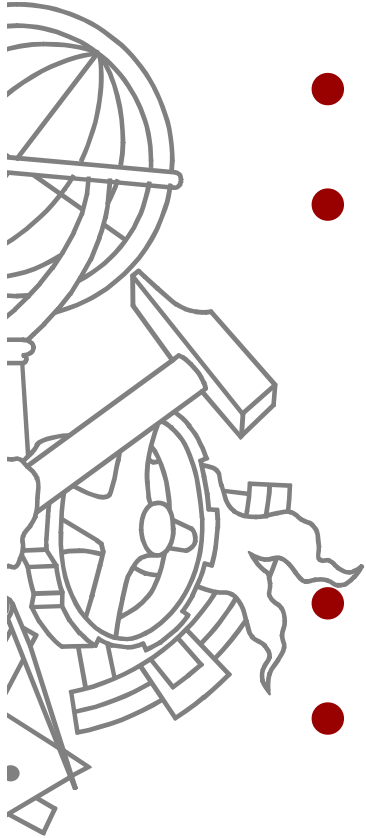
Paulo Gandra de Sousa
psousa@dei.isep.ipp.pt

Conteúdo



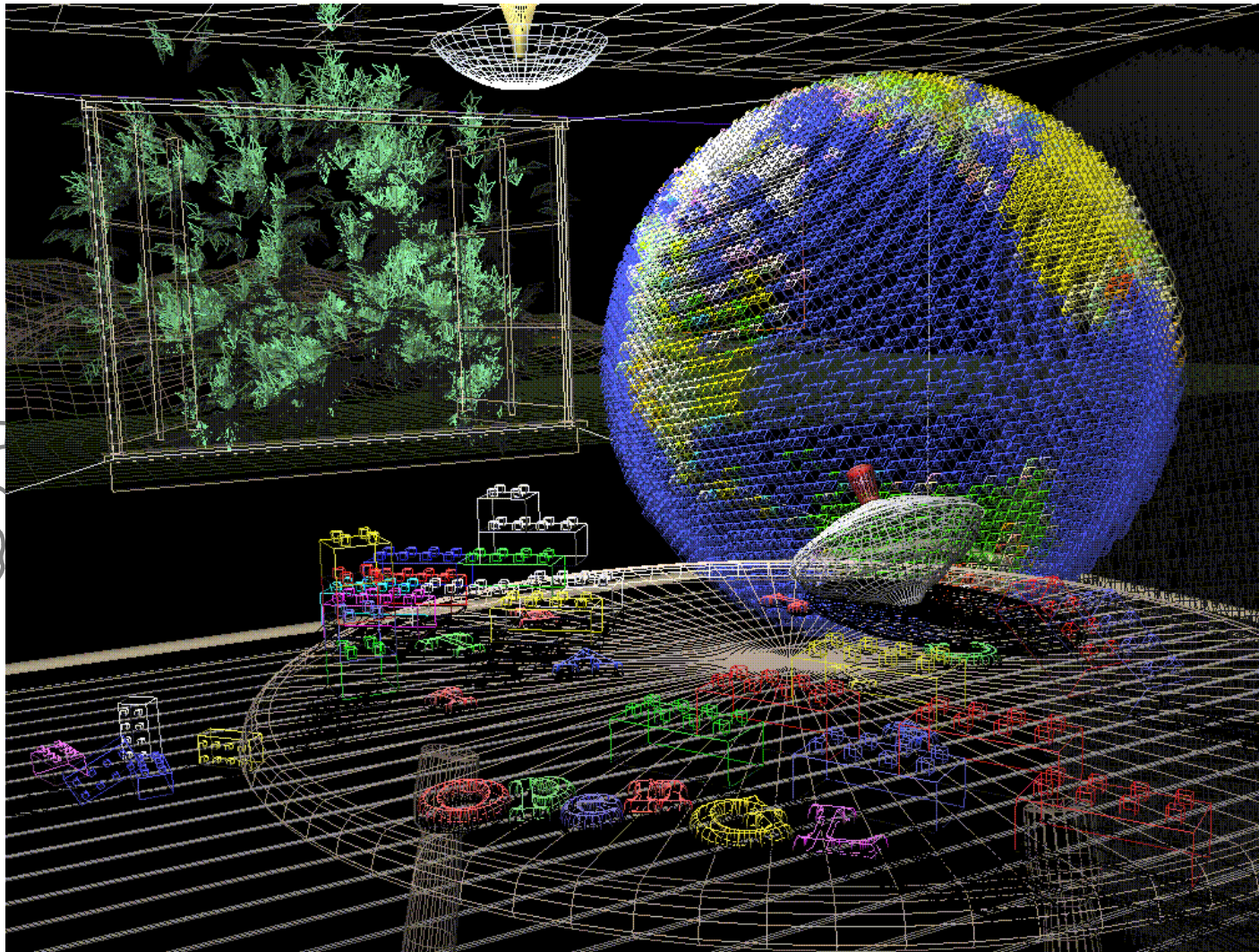
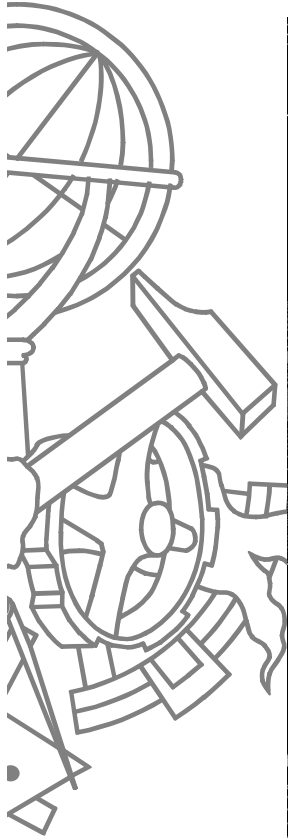
- Introdução
- Tipos de dados
- Nomenclatura de funções
- Esqueleto de programa
 - Actividades básicas
- Esqueleto de programa usando GLUT
 - Inicialização
 - *Callback* de desenho
- Exemplos de aplicações

OpenGL

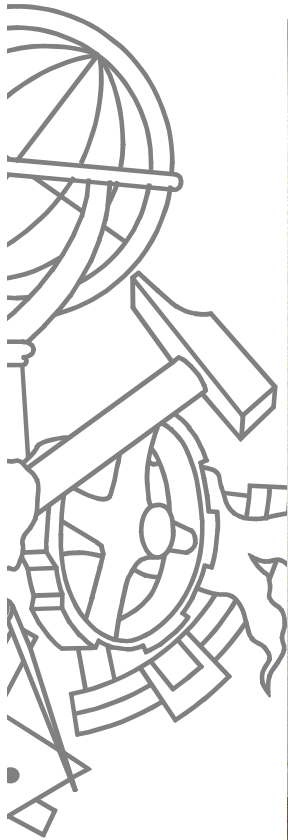


- **Open Graphics Library**
- API genérica de gráficos 2D/3D
 - Apenas contém primitivas simples
 - Modelos complexos construídos com base nestas primitivas
- Independente do sistema operativo
- Independente do sistema gestor de janelas

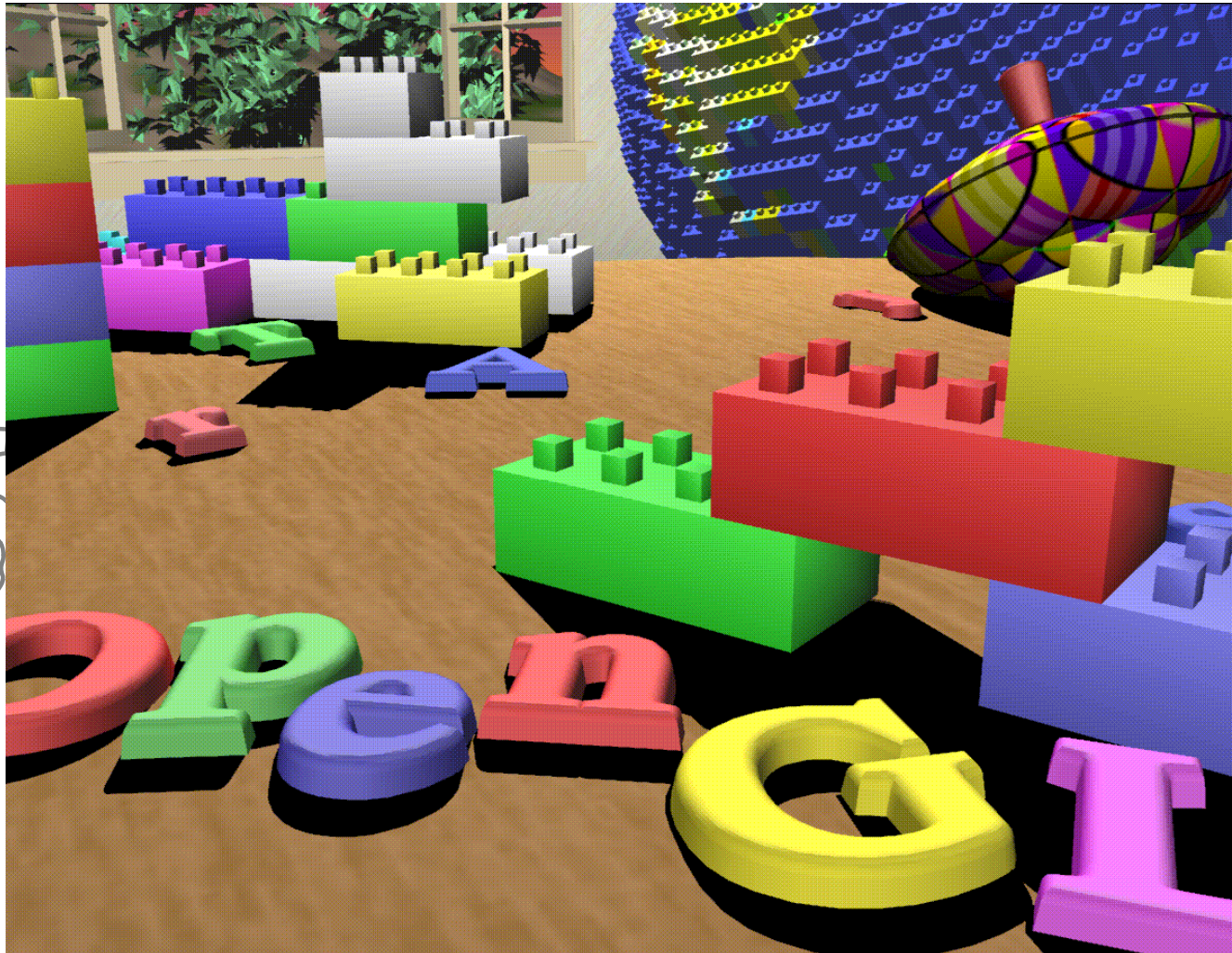
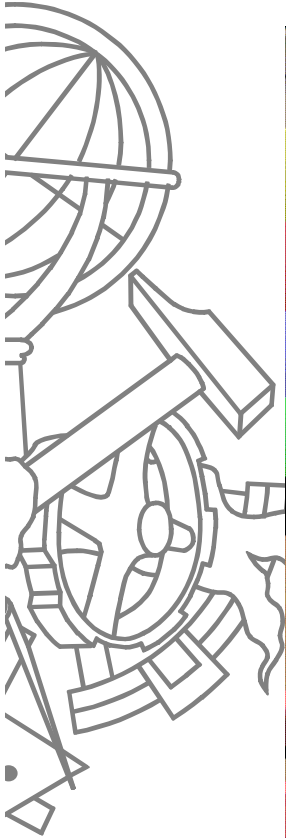
Uma cena: wireframe



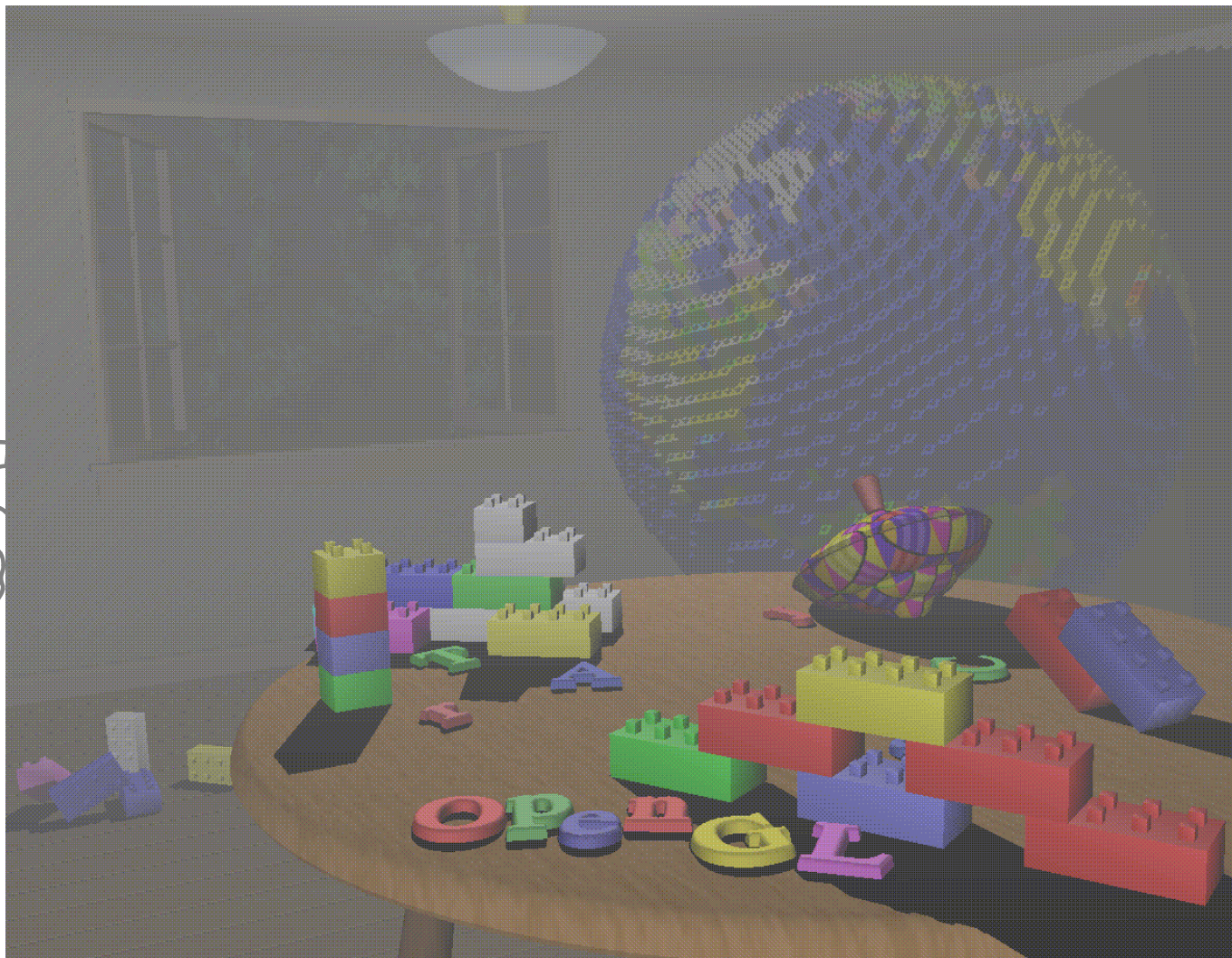
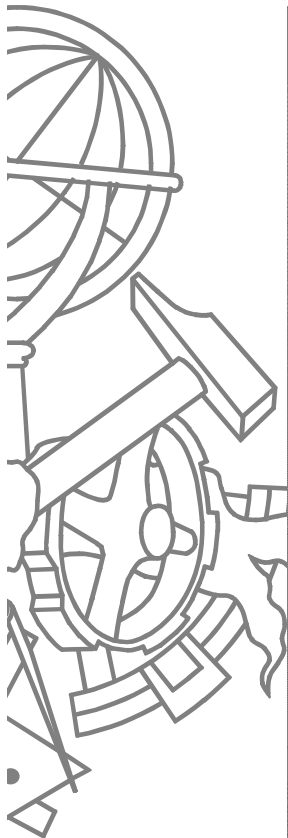
Sombras e iluminação



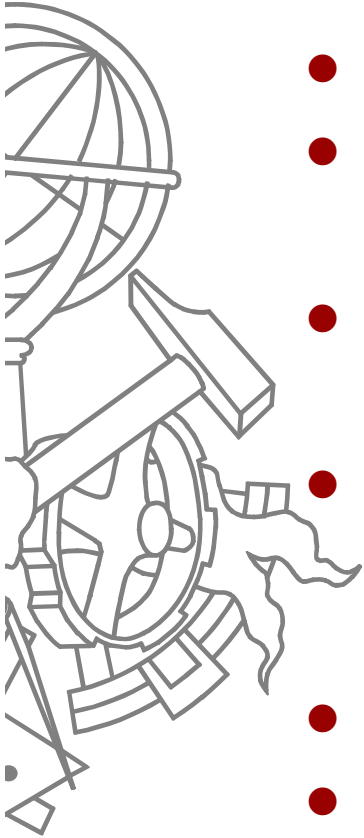
Pormenor



Efeitos ambiente

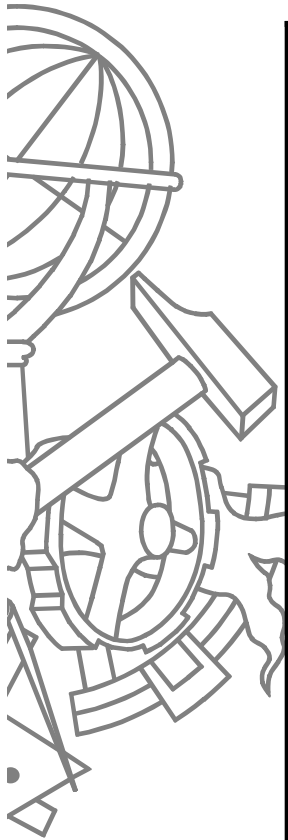


“Bits and bytes”



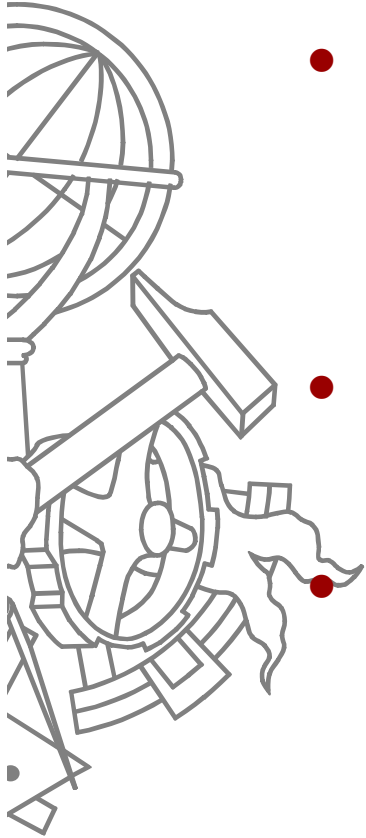
- OpenGL
- GLU – OpenGL Utility library
- GLUT – OpenGL Utility Toolkit
- GLX / WGL – OpenGL for X-Windows/OpenGL for Windows
- Java bindings
- GLAx – OpenGL ActiveX
- CsGL – OpenGL for C#

Tipos de dados



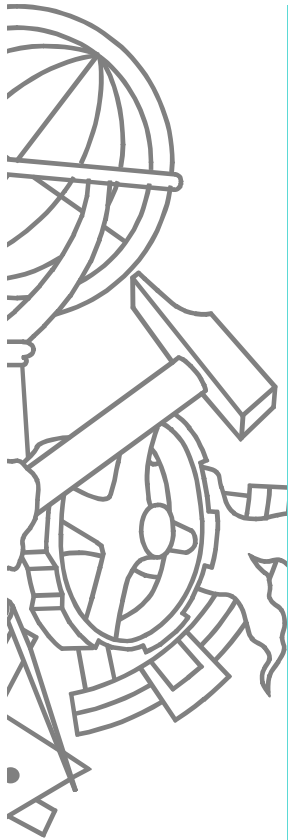
Suffix	Data Type	Typical Corresponding C-Language Type	OpenGL Type Definition
b	8-bit integer	signed char	GLbyte
s	16-bit integer	short	GLshort
i	32-bit integer	int or long	GLint, GLsizei
f	32-bit floating-point	float	GLfloat, GLclampf
d	64-bit floating-point	double	GLdouble, GLclampd
ub	8-bit unsigned integer	unsigned char	GLubyte, GLboolean
us	16-bit unsigned integer	unsigned short	GLushort
ui	32-bit unsigned integer	unsigned int or unsigned long	GLuint, GLenum, GLbitfield

Nomenclatura de funções



- Prefixos no nome das funções
 - gl
 - glu
 - glut
- { gl | glu | glut } FUNC [{ 1 | 2 | 3 | 4 } TYPE [v]]
 - TYPE := b | d | f | i | s | ub | ui | us
- Exemplos:
 - **glColor3b**, **glColor3f**, **glColor4d**, **glColor4dv**
 - **glLoadMatrixd**, **glLoadMatrixf**
 - **gluLookAt**
 - **glutSwapBuffers**

Esqueleto de programa



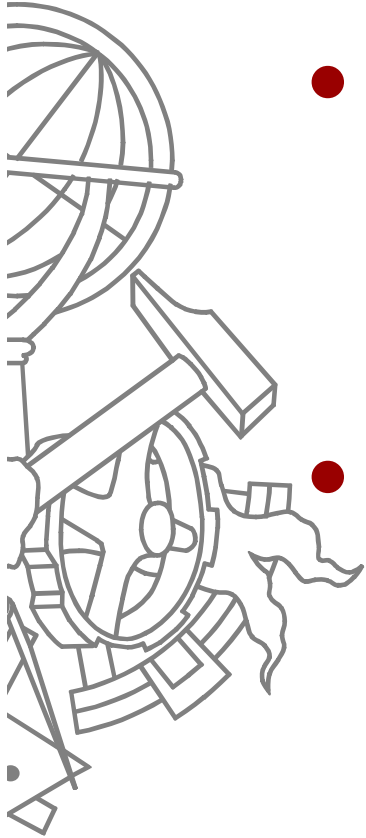
```
#include <whateverYouNeed.h>
main()
{
    InitializeAWindowPlease();
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glClear (GL_COLOR_BUFFER_BIT);
    glMatrixMode (GL_PROJECTION)
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
    glColor3f (1.0, 1.0, 1.0);
    glBegin (GL_POLYGON);
        glVertex3f (0.25, 0.25, 0.0);
        glVertex3f (0.75, 0.25, 0.0);
        glVertex3f (0.75, 0.75, 0.0);
        glVertex3f (0.25, 0.75, 0.0);
    glEnd();
    glFlush();
    UpdateTheWindowAndCheckForEvents();
}
```

} Limpar ecrã

} Definir sist.
coordenadas

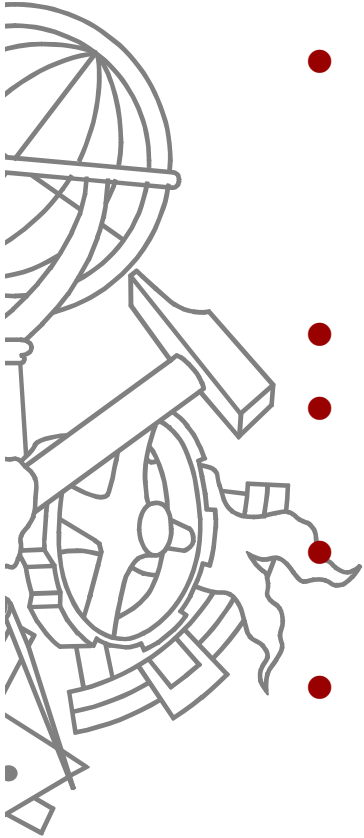
} desenhar

Limpar ecrã



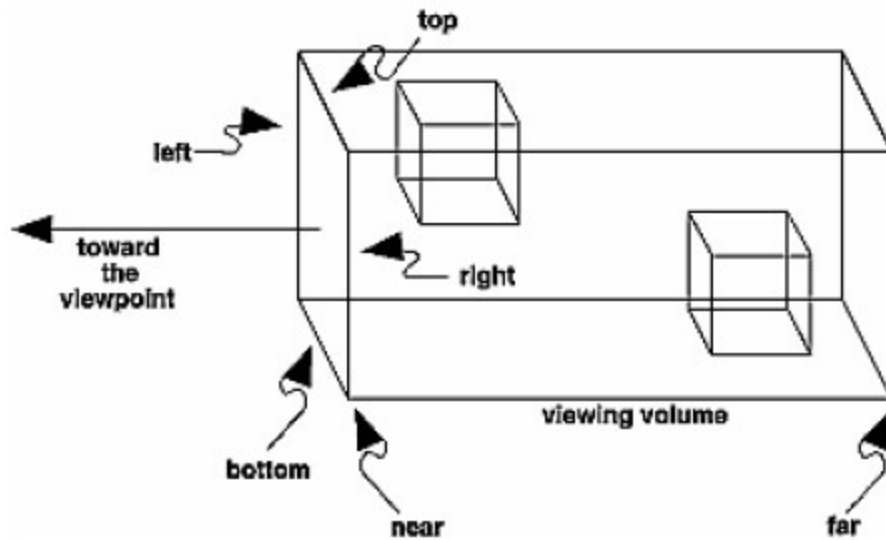
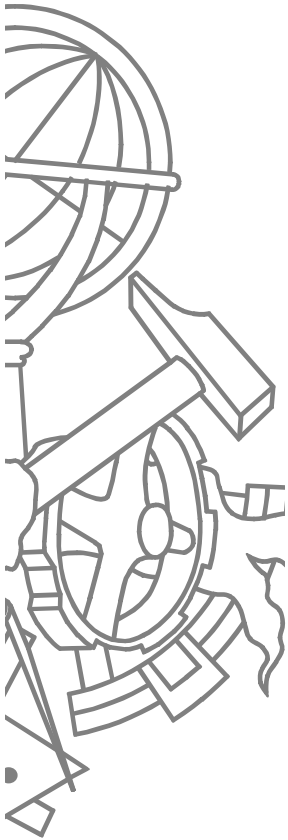
- **`glClearColor(red, green, blue, alpha)`**
 - Define qual a cor utilizada como “fundo” para limpar o ecrã
- **`glClear(GL_COLOR_BUFFER_BIT)`**
 - Limpa um ou mais *buffers* do OpenGL
 - `GL_COLOR_BUFFER_BIT` = *buffer* de cor
 - Será utilizado futuramente para limpar o *buffer* de profundidade em 3D

Sistema de coordenadas

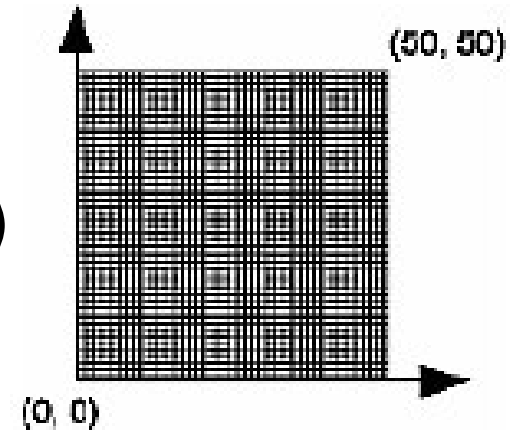


- `glViewport(x, y, width, height)`
 - Define a área de desenho do OpenGL dentro da área total da janela
 - Por omissão ocupa toda a janela
- `glMatrixMode(GL_PROJECTION)`
- `glLoadIdentity()`
 - Limpa a matriz de projecção
- `glOrtho(left, right, bottom, top, zNear, zFar)`
- `gluOrtho2D(left, right, bottom, top)`
 - Define uma projecção ortográfica sobre a cena modelada, criando um sistema de coordenadas

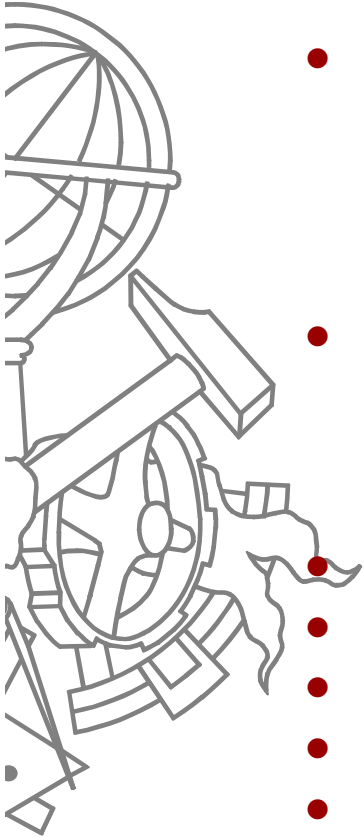
Sistema de coordenadas



- `gluOrtho2D(0.0, 50.0, 0.0, 50.0)`



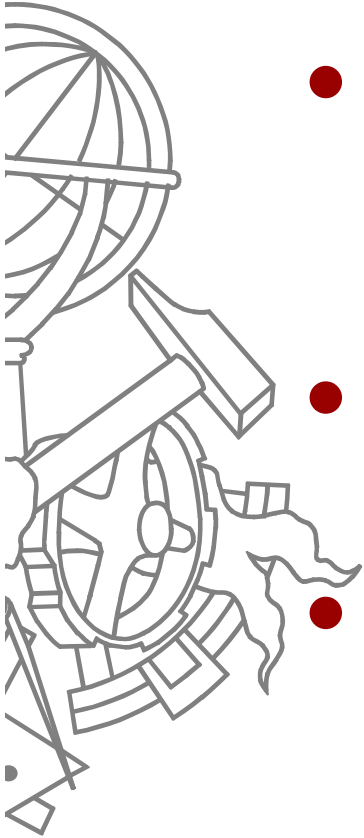
Cor



- RGBA
 - glColor{3|4}
 - red + green + blue
 - Alfa : usado para *blending* e transparências (falaremos futuramente)
 - [0.0 , 1.0]
- Index (*color map*)
 - Utiliza *color map* do sistema gestor de janelas
 - Um índice para o mapa em vez dos componentes de cor

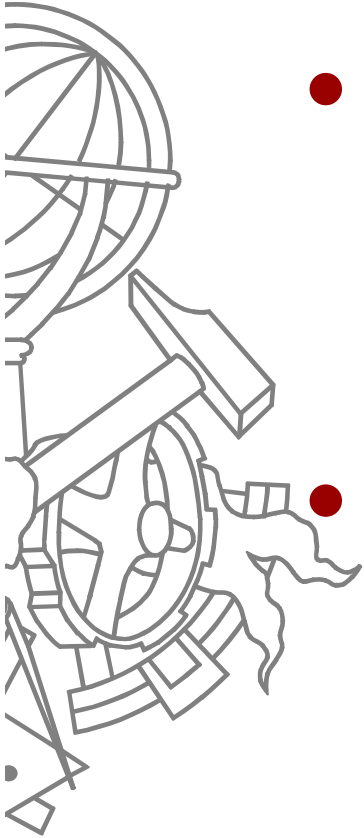
```
• glColor3f(0.0, 0.0, 0.0);    /* black    */
• glColor3f(1.0, 0.0, 0.0);    /* red      */
• glColor3f(0.0, 1.0, 0.0);    /* green    */
• glColor3f(1.0, 1.0, 0.0);    /* yellow   */
• glColor3f(0.0, 0.0, 1.0);    /* blue     */
• glColor3f(1.0, 0.0, 1.0);    /* magenta  */
• glColor3f(0.0, 1.0, 1.0);    /* cyan     */
• glColor3f(1.0, 1.0, 1.0);    /* white    */
```

Desenhar



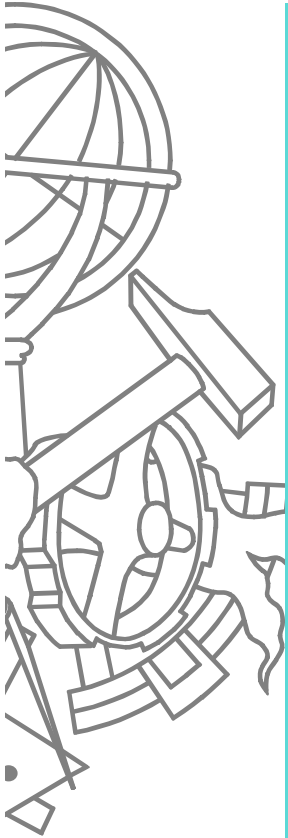
- **glBegin (*mode*) / glEnd**
 - Define um objecto da cena
 - Neste caso um polígono (GL_POLYGON)
- **glVertex**
 - Define um vértice do objecto a desenhar
- **glFlush**
 - Força o *pipeline* do OpenGL a terminar o processamento e desenhar os pixels no ecrã

OpenGL e o sistema gestor de janelas



- OpenGL é independente do sistema gestor de janelas
 - Utilizar a API do sistema gestor de janelas
- Ou
 - Utilizar GLUT

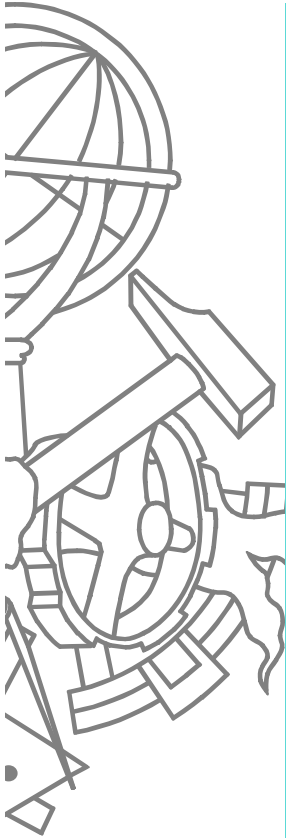
Esqueleto programa GLUT



```
#include <GL\glut.h>
void init();
void display();

void main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (250, 250);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("hello");
    init ();
    glutDisplayFunc (display);
    glutMainLoop ();
}
```

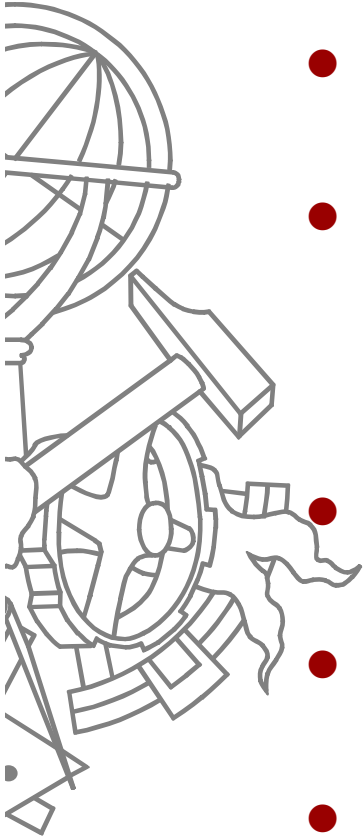
Esqueleto programa GLUT



```
void init()
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}

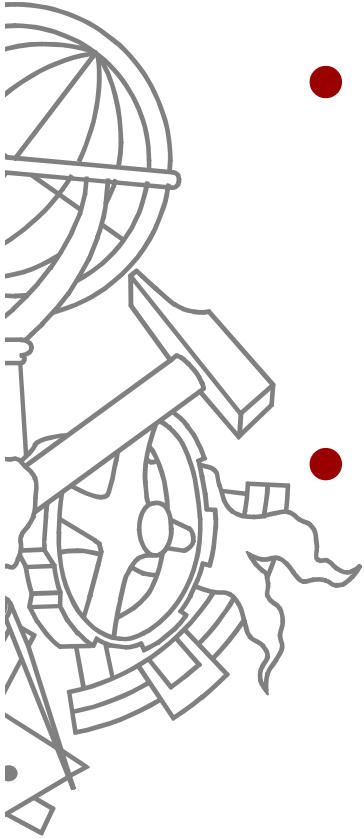
void display()
{
    glClear (GL_COLOR_BUFFER_BIT);
    glColor3f (1.0, 1.0, 1.0);
    glBegin (GL_POLYGON);
        glVertex3f (0.25, 0.25, 0.0);
        glVertex3f (0.75, 0.25, 0.0);
        glVertex3f (0.75, 0.75, 0.0);
        glVertex3f (0.25, 0.75, 0.0);
    glEnd();
    glFlush ();
}
```

Inicialização



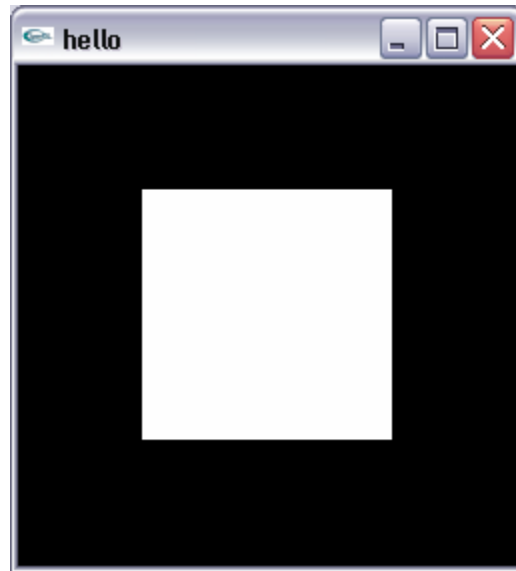
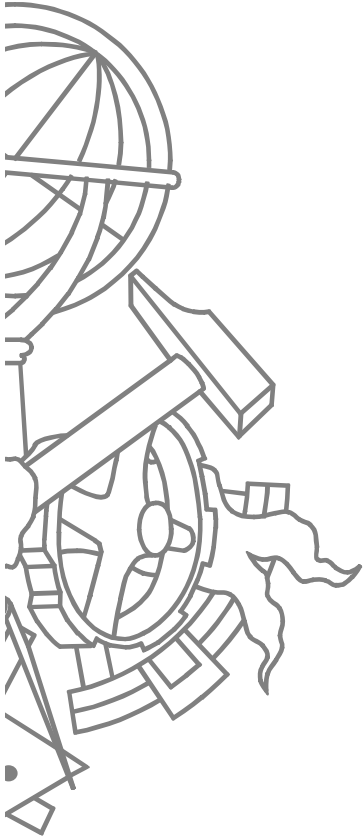
- **glutInit** (*argc*, *argv*)
 - Inicializa a biblioteca GLUT
- **glutInitDisplayMode** (*mode*)
 - Indica qual o modo de funcionamento a usar
 - GLUT_SINGLE
 - GLUT_RGB
- **glutInitWindowSize** (*width*, *height*)
 - Indica tamanho inicial da janela
- **glutInitWindowPos** (*x*, *y*)
 - Indica posição inicial da janela
- **glutCreateWindow** (*titulo*)
 - Cria a janela da aplicação

Callback e ciclo principal



- **glutDisplayFunc** (*callback*)
 - Indica qual a função a invocar sempre que for necessário desenhar o conteúdo da janela
- **glutMainLoop**
 - Processa os eventos do sistema gestor de janelas (rato, teclado, ...) e invoca as *callbacks* registadas

Demo



Exemplos

