Módulo 1

Sistemas Gráficos e Interação

Instituto Superior de Engenharia do Porto

Filipe Pacheco

ffp@isep.ipp.pt

Introdução ao OpenGL

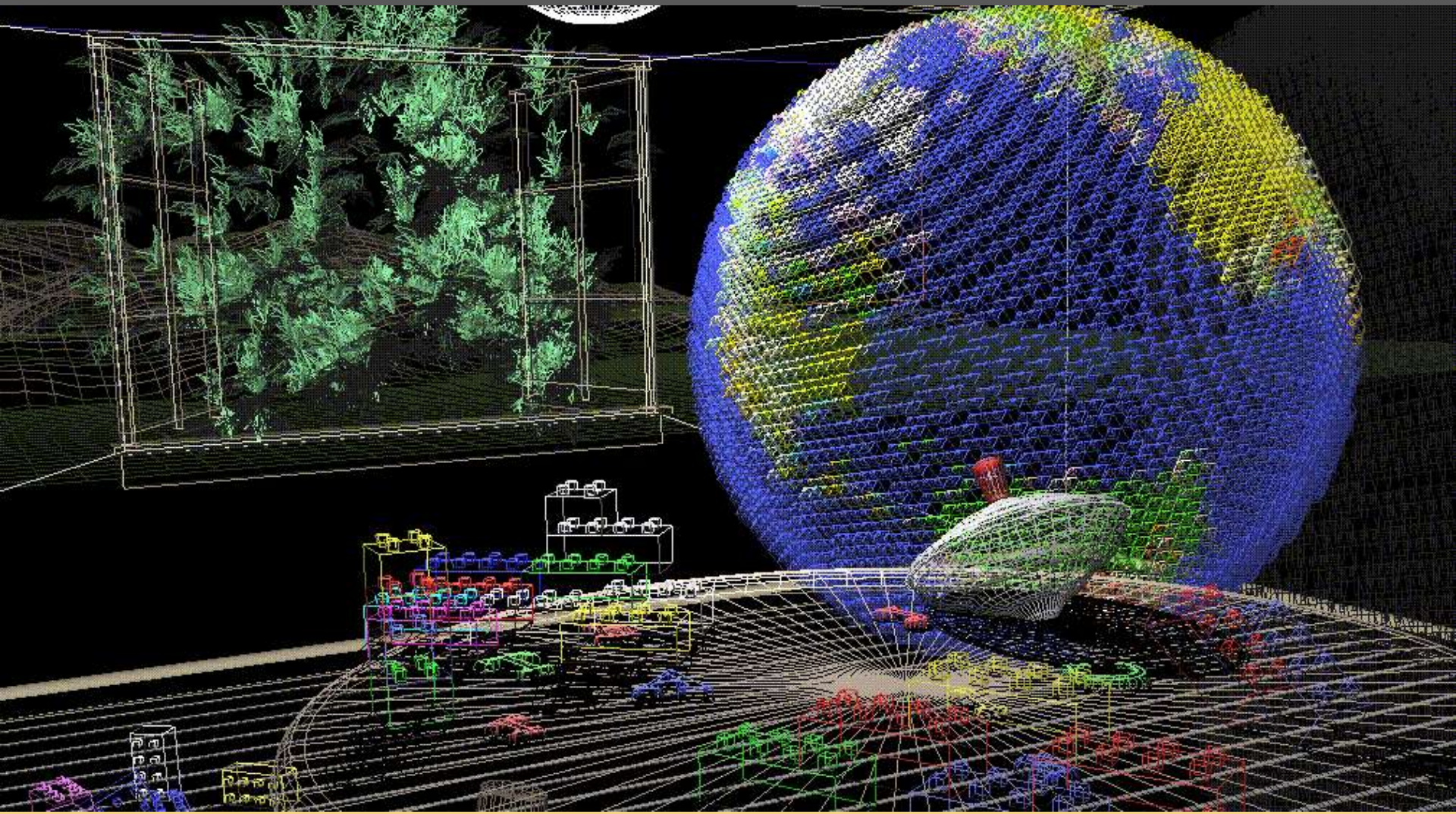
Conteúdo

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

OpenGL

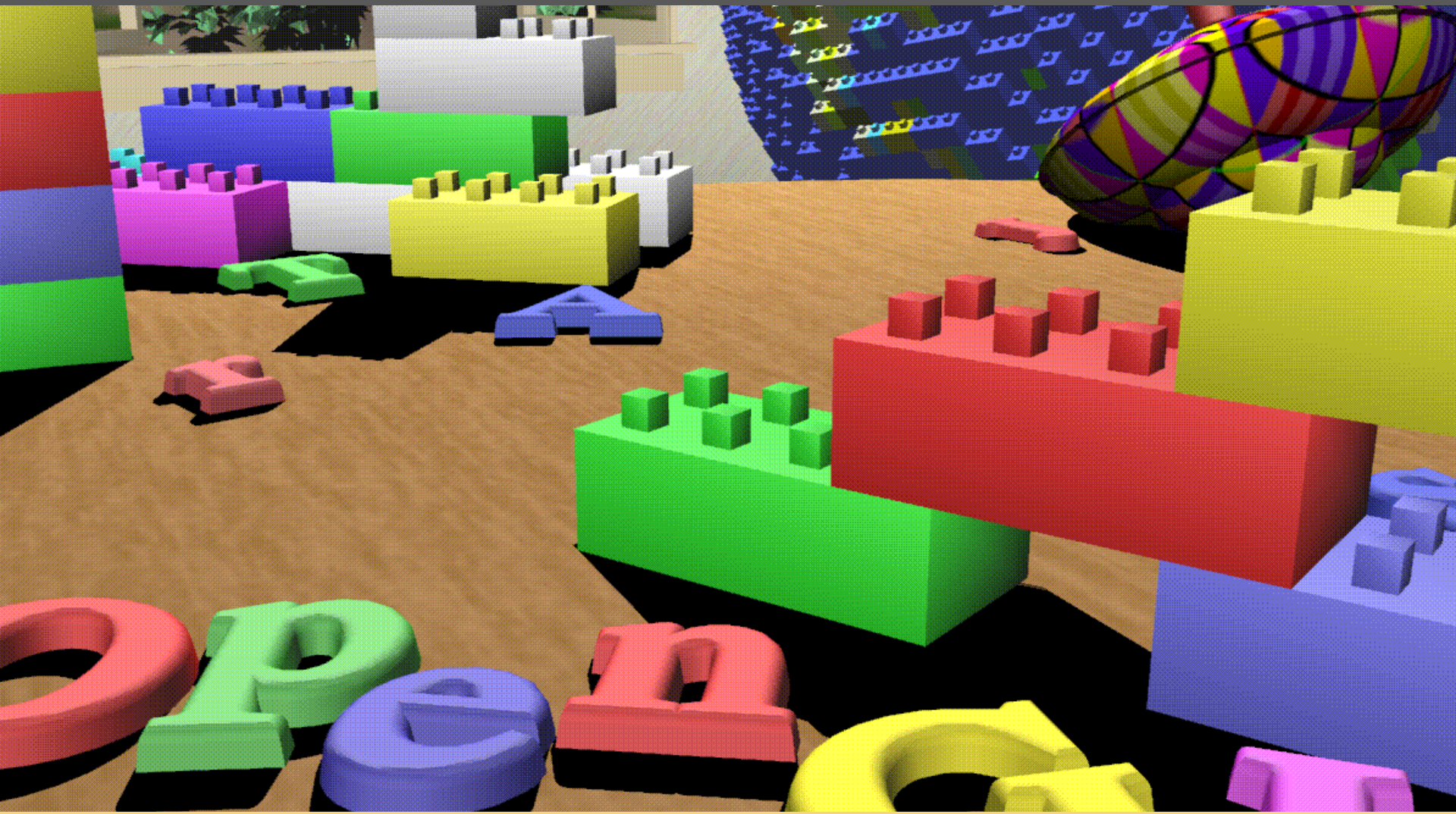
- **Open G**raphics **L**ibrary
- Generic graphics API 2D/3D
  - Only contains simple primitives
  - Complex models built on these primitives
- Independent of the operating system
- Independent of the window management system

Uma cena: wireframe



Sombras e iluminação

Pormenor



Efeitos ambiente

“Bits and bytes”

- OpenGL
- GLU – OpenGL Utility library
- GLUT – OpenGL Utility Toolkit
- GLX / WGL / CGL – OpenGL for X-Windows/OpenGL for Windows/OpenGL for Mac OS X
- Java bindings
- GlAx – OpenGL ActiveX
- CsGL – OpenGL for C#
- WebGL – OpenGL for HTML 5

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

- Function name prefix
  - gl ➔ OpenGL
  - glu ➔ *OpenGL* Utility Library
  - glut ➔ *OpenGL* Utility Toolkit
  - glaux ➔ *OpenGL* Auxiliary Library (deprecated)
- { gl | glu | glut } FUNC [{ 1 | 2 | 3 | 4 } TYPE [v]]
  - TYPE := b | d | f | i | s | ub | ui | us
- Examples:
  - **gl**Color**3b**, **gl**Color**3f**, **gl**Color**4d**, **gl**Color**4dv**
  - **gl**LoadMatrix**d**, **gl**LoadMatrix**f**
  - **glu**LookAt
  - **glut**SwapBuffers

Esqueleto de programa

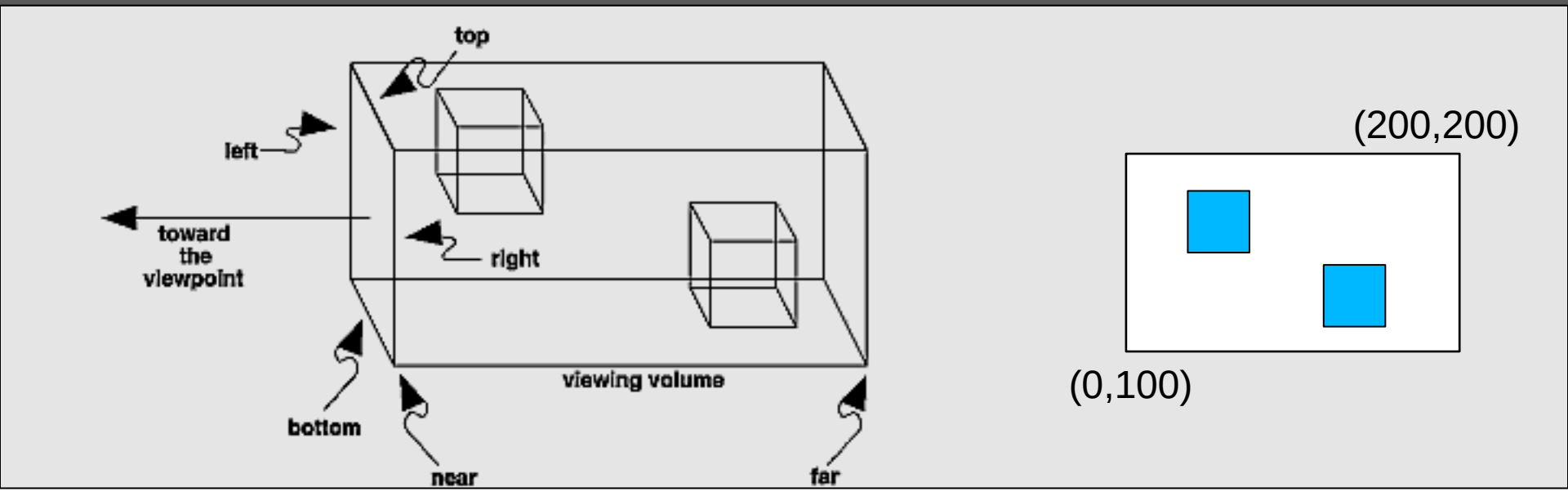
```
#include <whateverYouNeed.h>
int main()
{
    InitializeAWindowPlease();
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
    glColor3f(1.0, 1.0, 1.0);
    glBegin(GL_POLYGON);
        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();
    return 0;
}
```

Clear display

Define the coordinate system

Draw

Sistema de coordenadas



- gluOrtho2D(0.0, 200.0, 100.0, 200.0)

left

right

bottom

top

Default
near=1.0
far=-1.0

Cor

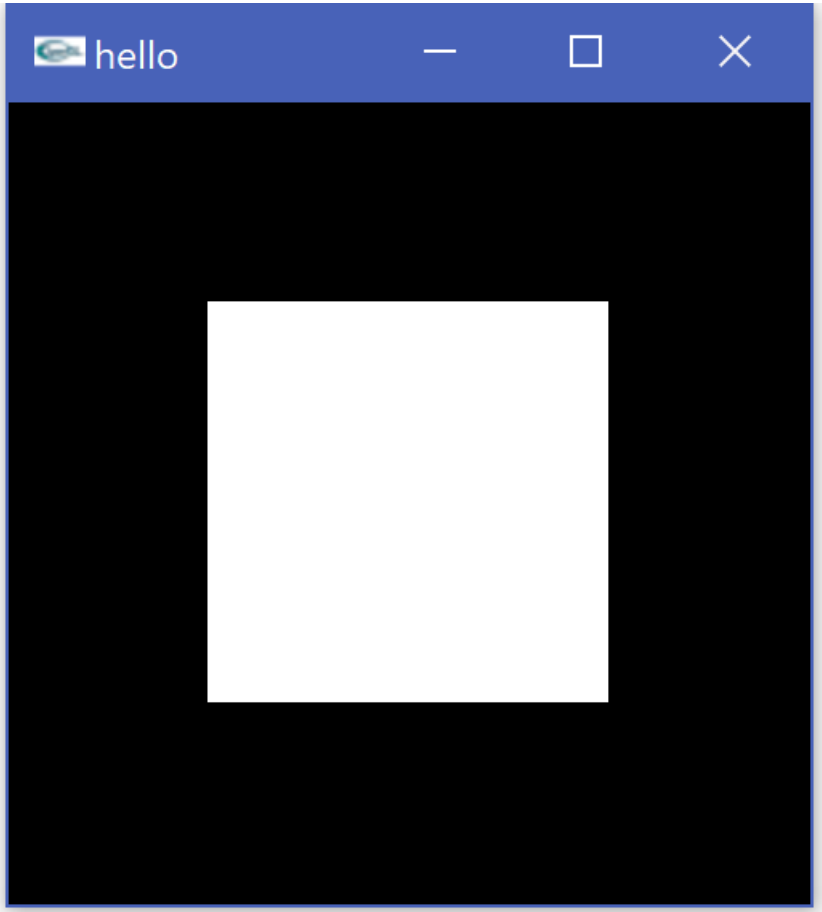
- **RGBA**
  - glColor{3|4}
  - red + green + blue [+ alfa]
  - alfa : *blending* & transparency (talk later)
  - [0.0 , 1.0]
- **Index** (*color map*)
  - Uses *color map* from the window system
  - Not used in SGRAI
- **Exemplos**

```
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 na object
  - example: poligon (GL_POLYGON)
- glVertex...()
  - Define a vertex of the object
- glFlush()
  - Forces the OpenGL pipeline to complete all the drawing processing

Demo



Exemplos