



UFMA

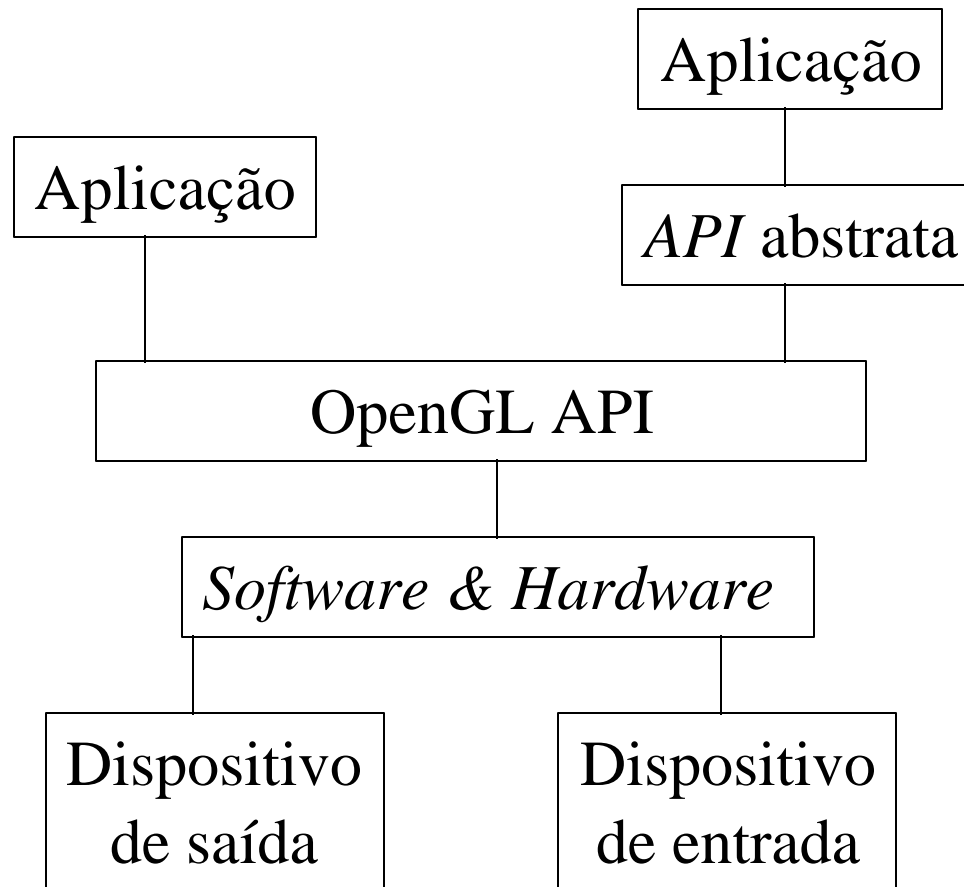
Introdução à Computação Gráfica
DEINF-UFMA
Prof. Anselmo Paiva

**Departamento
de
Informática**

OpenGL

OpenGL: o que é?

- API
 - Interface para programador de aplicação



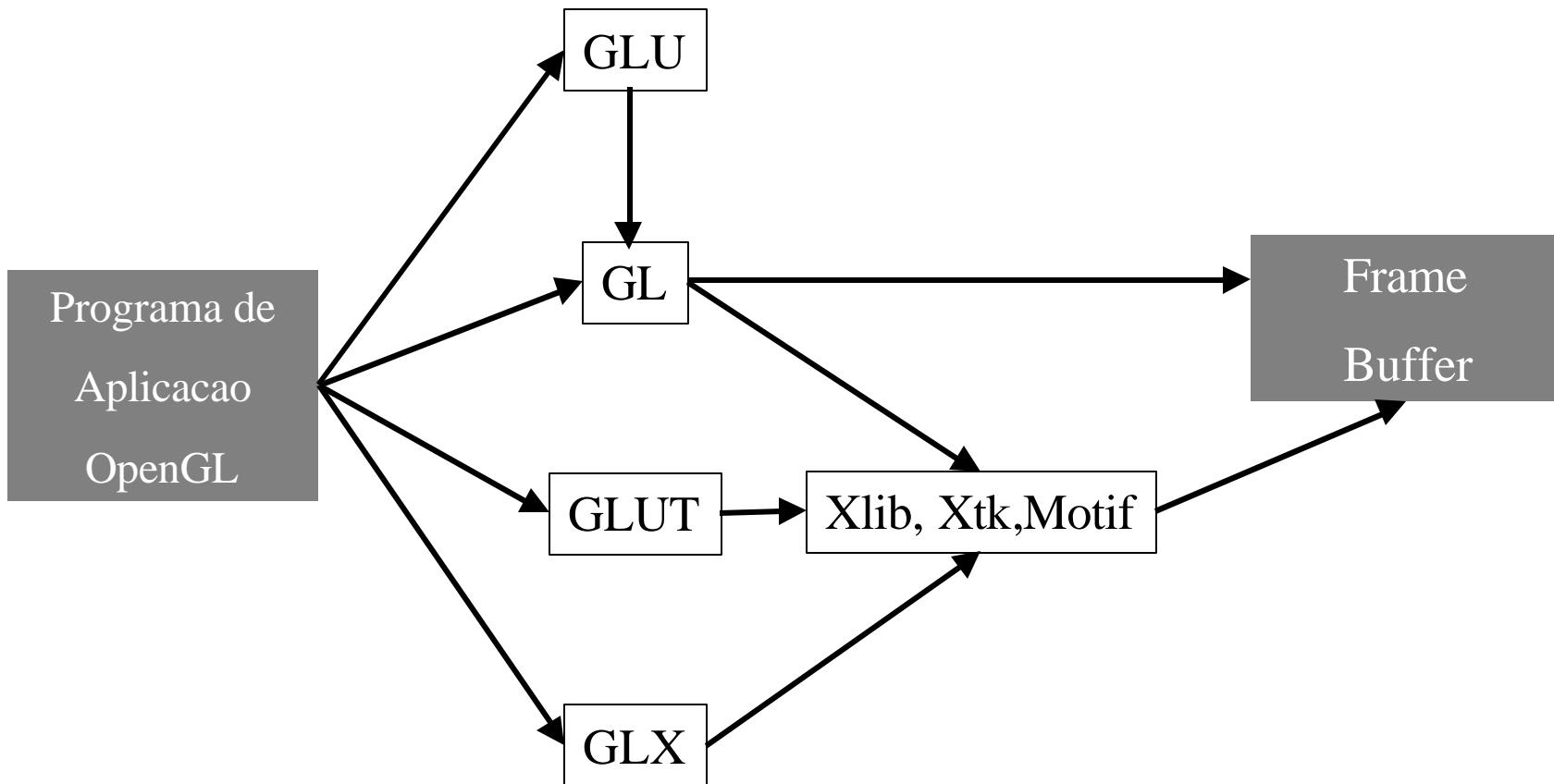
Por que OpenGL?

- primitivas geométricas e imagens
- arquitetura bem definida
- relativamente simples
- boa performance (sw & hw)
- bem documentado
- independente de sistemas de janelas
- padrão
 - disponível em diversas plataformas

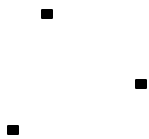
O que faz o OpenGL

- Desenhar objetos: desenha pontos, linhas and poligonos.
Using these
- Controlar a visualizacao dos objetos: possui um conjunto de transformacoes para visualizacao e modelagem
- Aplicar iluminacao: permite a manipulacao de varios tipos de fontes luminosas
- Especificar modelos de iluminacao
- Melhoramentos na Imagem: antialias, *blend*, *fog*
- Mapeamento de texturas:
- Animacoes:
- Implementa *double buffering*

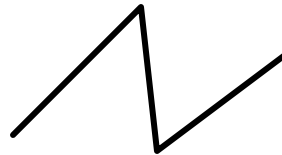
Organização da Biblioteca (XWindow)



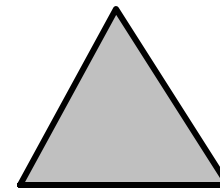
Primitivas geométricas básicas



Ponto



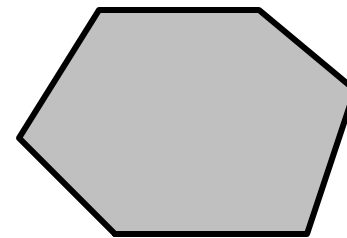
Linha



Triângulo



Quadrado



Polígono (convexo)

Objetos 3D



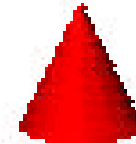
Polyhedra



Sphere

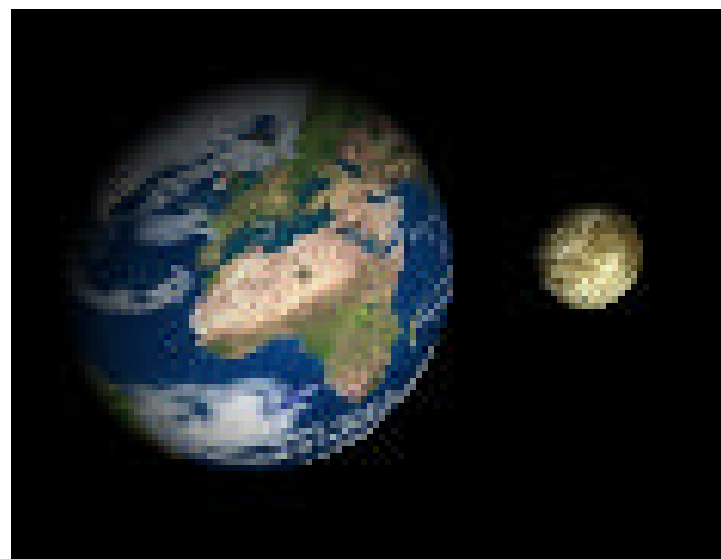
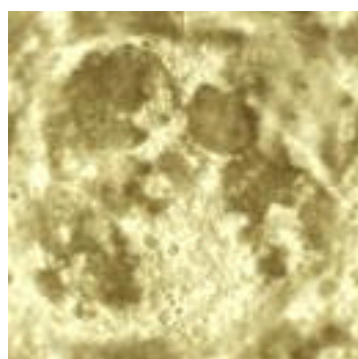
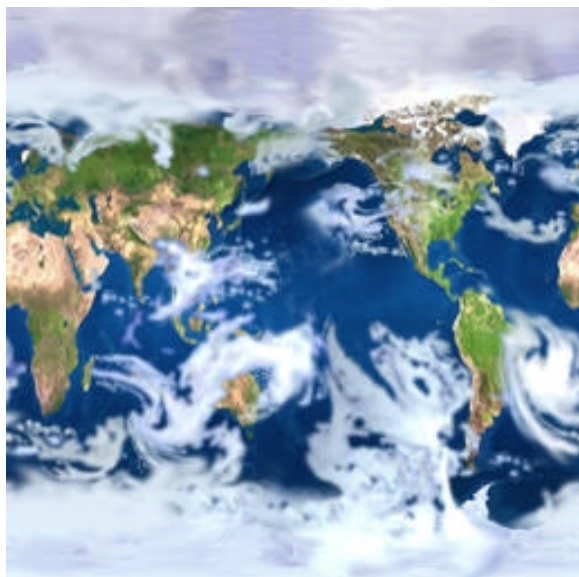


Bezier Surfaces

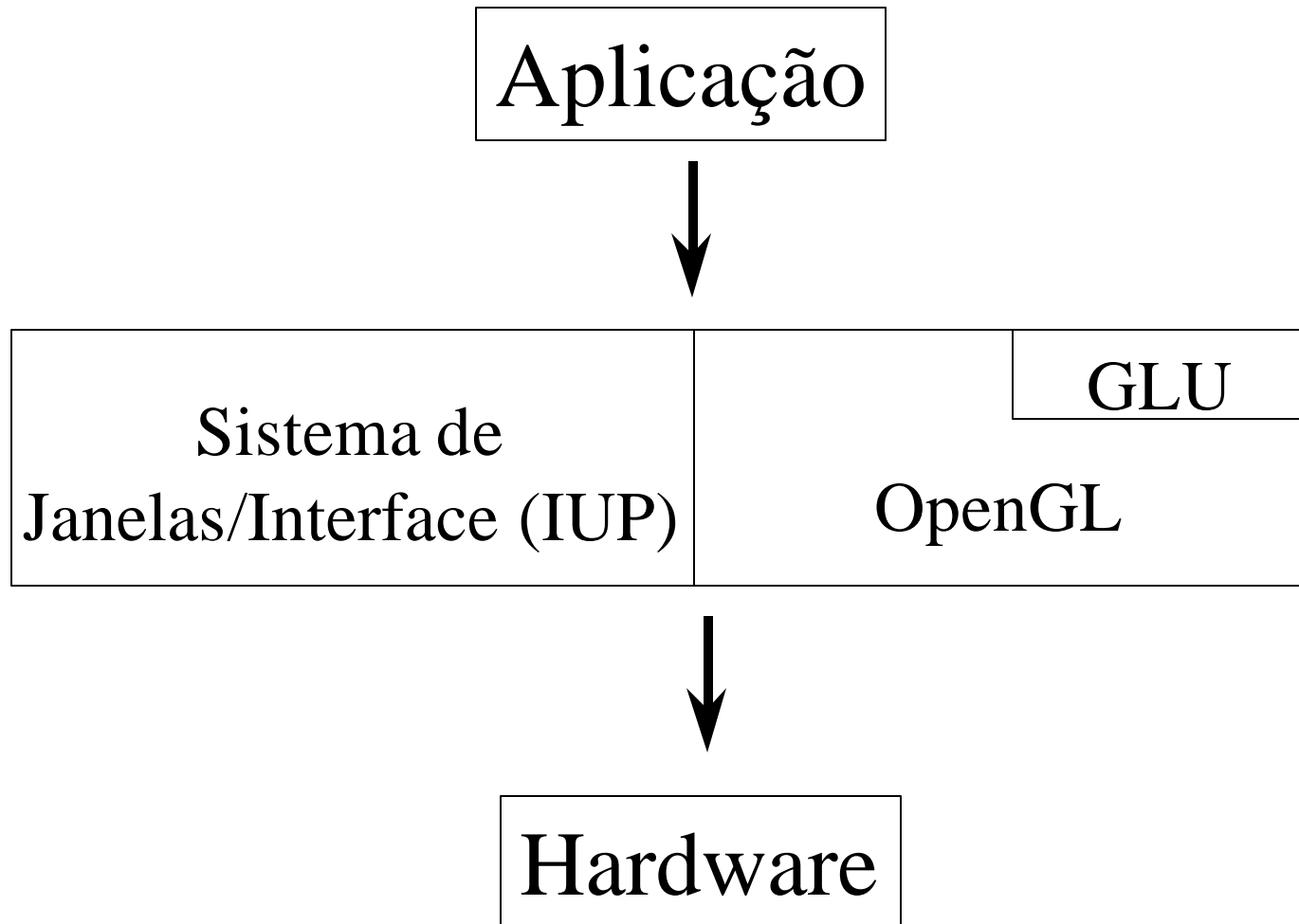


Quadric

Imagem e Textura



Aplicação típica



Programa simples (usando GLUT)

```
#ifdef _WIN32
#include <windows.h>
#endif
#include "GL/gl.h"
#include "GL/glu.h"
#include "GL/glut.h"

int main (int argc, char* argv[])
{
    /* openg GLUT */
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (250, 250);

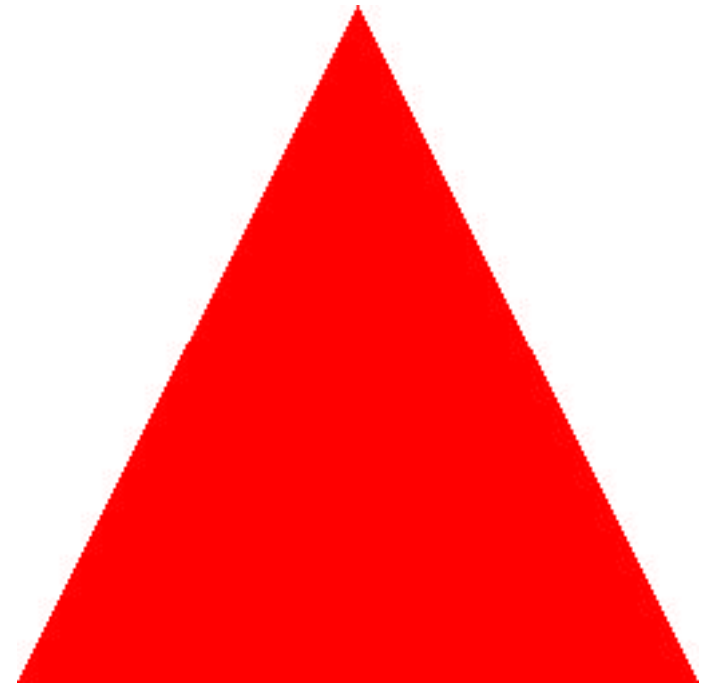
    /* create window */
    glutCreateWindow ("simple");
    glutDisplayFunc(display);

    /* interact ... */
    glutMainLoop();
    return 0;
}
```

```
void display (void)
{
  /* clear window */
  glClearColor(1,1,1,1);
  glClear(GL_COLOR_BUFFER_BIT);

  /* draw red triangle */
  glColor3d(1,0,0);
  glBegin(GL_TRIANGLES);
  glVertex2d(-1,-1);
  glVertex2d(1,-1);
  glVertex2d(0,1);
  glEnd();

  /* update screen */
  glFlush();
}
```



OpenGL: máquina de estado

- Trabalha com o conceito de valor corrente
 - Iluminação
 - Shading
 - Textura
 - etc.

`glEnable/glDisable`

`glBegin(tipo_de_prim);`

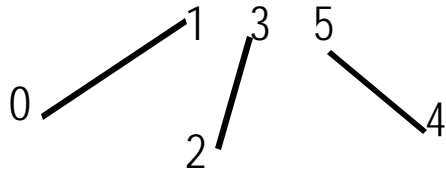
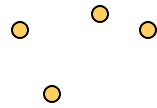
...define atributo de vértice

...define vértice

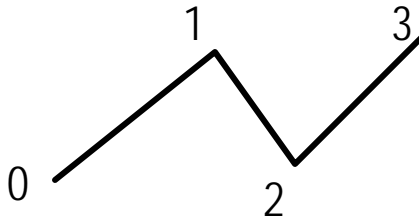
`glEnd();`

Tipos de primitivas

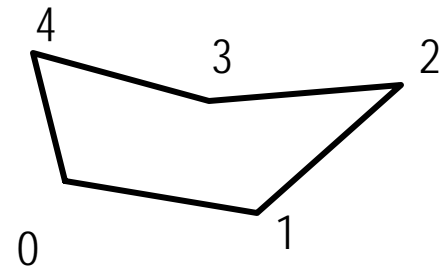
GL_POINTS



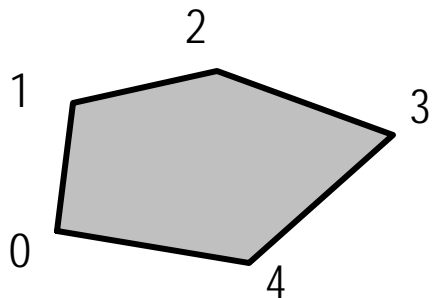
GL_LINES



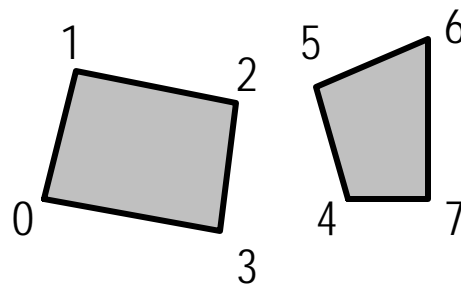
GL_LINE_STRIP



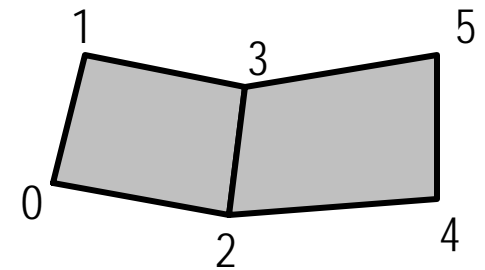
GL_LINE_LOOP



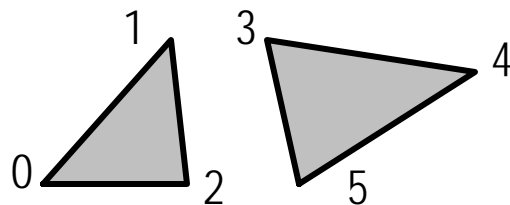
GL_POLYGON



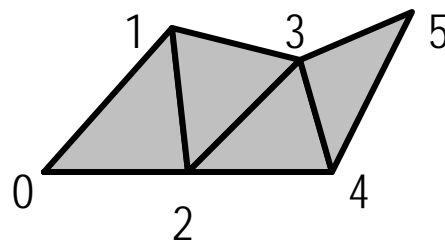
GL_QUADS



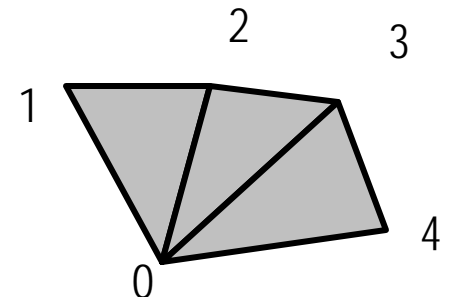
GL_QUAD_STRIP



GL_TRIANGLES



GL_TRIANGLE_STRIP



GL_TRIANGLE_FAN

`glVertex{tam}{tipo}{vetor} (...);`

exemplo:

```
GLdouble pos[ ] = {0.4,9.0,2.0};  
glVertex3dv(pos);
```

ou

```
glVertex3d(0.4,9.0,2.0);
```

Especificação de atributos: Cor

- Modelo de cor

- RGB

- `glColor3d(red,green,blue);`

- *Color index*

- Paleta previamente definida

- `lupGLPalette (handle, index, red, green, blue);`

- ...

- `glIndexi(index);`

Projeção

- 2D
 - retângulo de clipping
 - Exemplo:

```
glMatrixMode(GL_PROJECTION)
glLoadIdentity();
gluOrtho2D(left, right, bottom, top)
glMatrixMode(GL_MODELVIEW)
```

Referências

- “The Red Book”
OpenGL: Programming Guide
Release 1.1
M. Woo, J. Neider, T. Davis
- Web sites
The official OpenGL web page
<http://www.opengl.org>
SGI's OpenGL WWW Center
<http://www.sgi.com/Technology/OpenGL>
Gateway to OpenGL
http://reality.sgi.com/mjk_asd/opengl-links.html