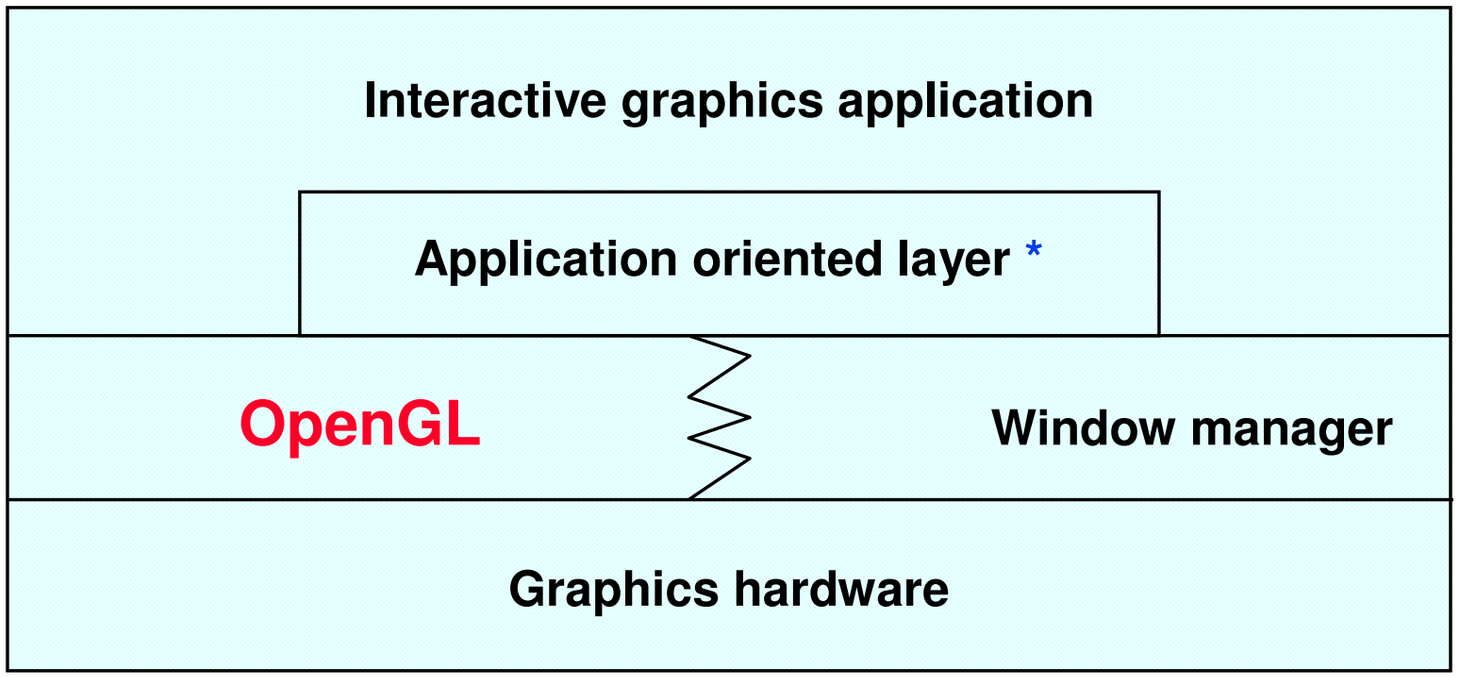
Introdução ao OpenGL

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Fernando Birra

The OpenGL graphics system

Layer model:



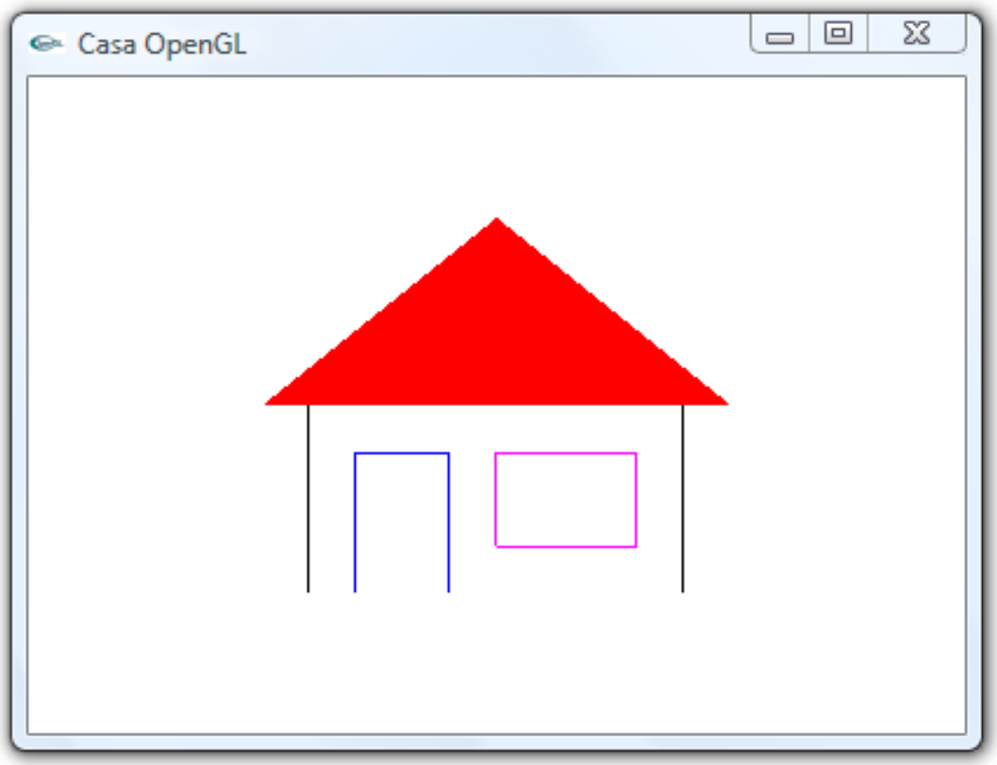
* e.g.

GLUT
JOGL
LWJGL
Open Inventor

M.Próspero

```c
// Casa.c
// Author: M.Próspero

#include <GL/glut.h>

void Desenhar (void)
{
    glMatrixMode (GL_MODELVIEW);
    glLoadIdentity ();
    glClear (GL_COLOR_BUFFER_BIT);

    glColor3f (0.0, 0.0, 0.0);  // paredes:
        glBegin (GL_LINES);
        glVertex2f (120.0,-220.0);
        glVertex2f (120.0,-140.0);
        glVertex2f (280.0,-220.0);
        glVertex2f (280.0,-140.0);
        glEnd ();

    glColor3f (0.0, 0.0, 1.0);  // porta:
        glBegin (GL_LINE_STRIP);
        glVertex2f (140.0, -220.0);
        glVertex2f (140.0, -160.0);
        glVertex2f (180.0, -160.0);
        glVertex2f (180.0, -220.0);
        glEnd ();

    glColor3f (1.0, 0.0, 1.0);  // janela:
        glBegin (GL_LINE_LOOP);
        glVertex2i (200, -160);
        glVertex2i (260, -160);
        glVertex2i (260, -200);
        glVertex2i (200, -200);
        glEnd ();

    glColor3f (1.0, 0.0, 0.0);  // telhado:
        glBegin (GL_TRIANGLES);
        glVertex2f (100.0, -140.0);
        glVertex2f (200.0, -60.0);
        glVertex2f (300.0, -140.0);
        glEnd ();

    glFlush ();

}
```

```c
void Enquadrar (GLsizei w, GLsizei h)
{
    glViewport (0, 0, w, h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (0.0, 400.0, -280.0, 0.0);
}

int main (int argc, char** argv)
{
    glutInit (&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition (0, 0);
    glutInitWindowSize (400, 280);
    glutCreateWindow ("Casa OpenGL");
    glClearColor (1.0, 1.0, 1.0, 0.0);
    glutReshapeFunc (Enquadrar);
    glutDisplayFunc (Desenhar);
    glutMainLoop ();
    return (0);
}
```



M.Próspero

void **glBegin**(GLenum *mode*);

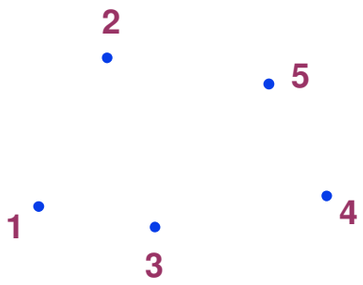
Marks the beginning of a vertex-data list that describes a geometric primitive. The type of primitive is indicated by *mode*, which can be any of the values shown in Table 2-2.

Value	Meaning
GL_POINTS	individual points
GL_LINES	pairs of vertices interpreted as individual line segments
GL_LINE_STRIP	series of connected line segments
GL_LINE_LOOP	same as above, with a segment added between last and first vertices
GL_TRIANGLES	triples of vertices interpreted as triangles
GL_TRIANGLE_STRIP	linked strip of triangles
GL_TRIANGLE_FAN	linked fan of triangles
GL_QUADS	quadruples of vertices interpreted as four-sided polygons
GL_QUAD_STRIP	linked strip of quadrilaterals
GL_POLYGON	boundary of a simple, convex polygon

Table 2-2 Geometric Primitive Names and Meanings

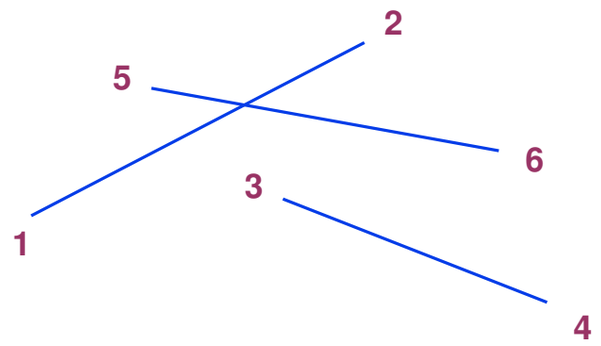
void **glEnd**(void);

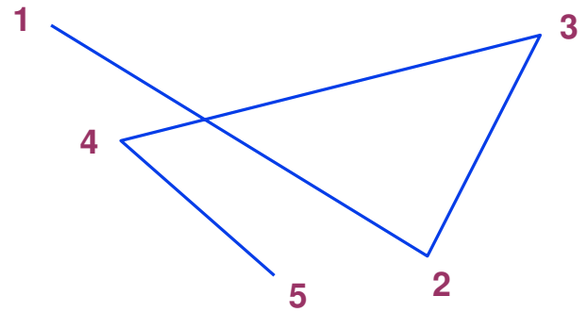
Marks the end of a vertex-data list.



GL_POINTS

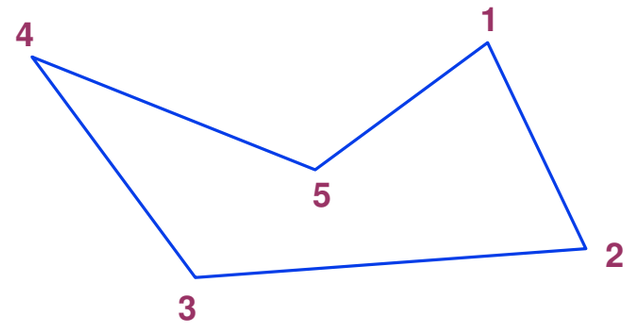
GL_LINES



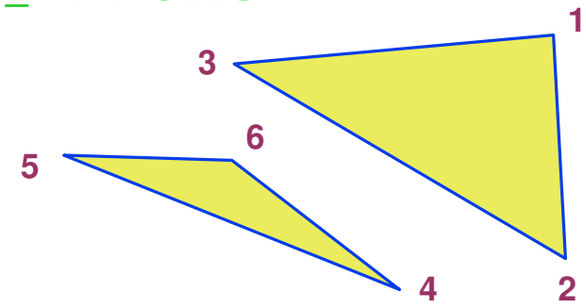


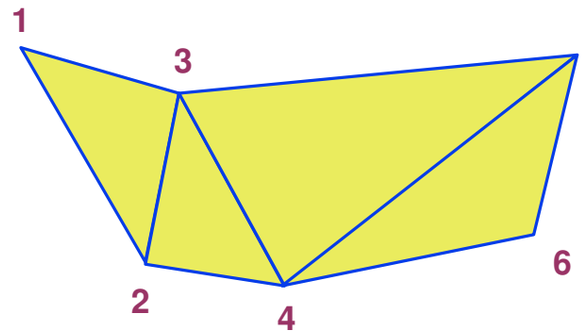
GL_LINE_STRIP

GL_LINE_LOOP



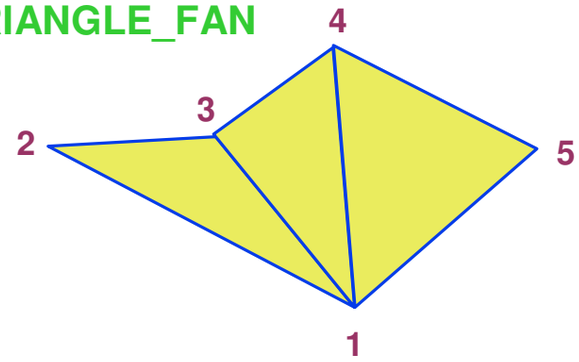
GL_TRIANGLES



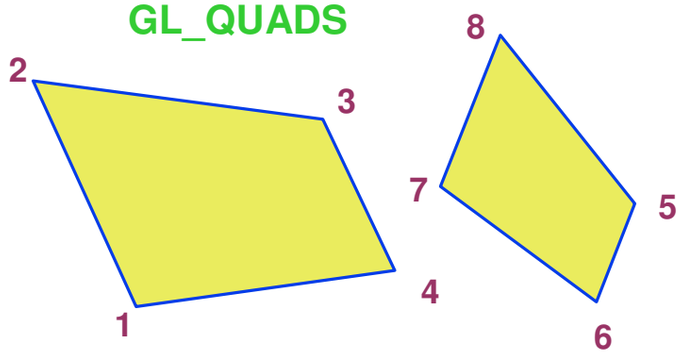


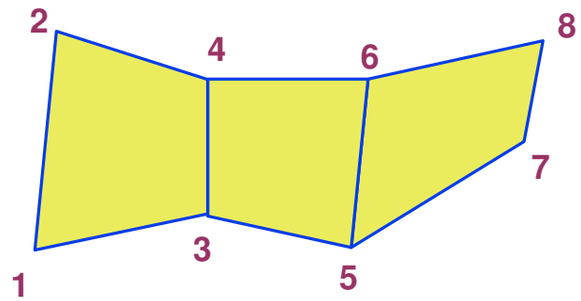
GL_TRIANGLE_STRIP

GL_TRIANGLE_FAN



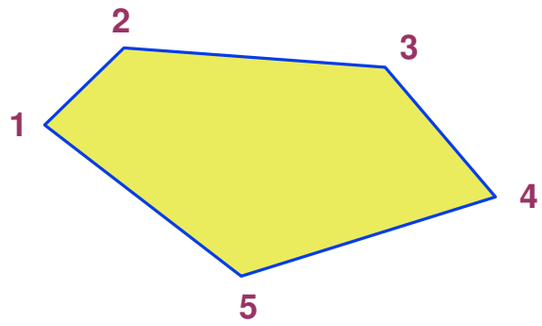
GL_QUADS



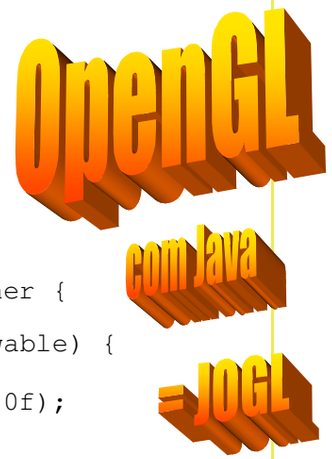


GL_QUAD_STRIP

GL_POLYGON

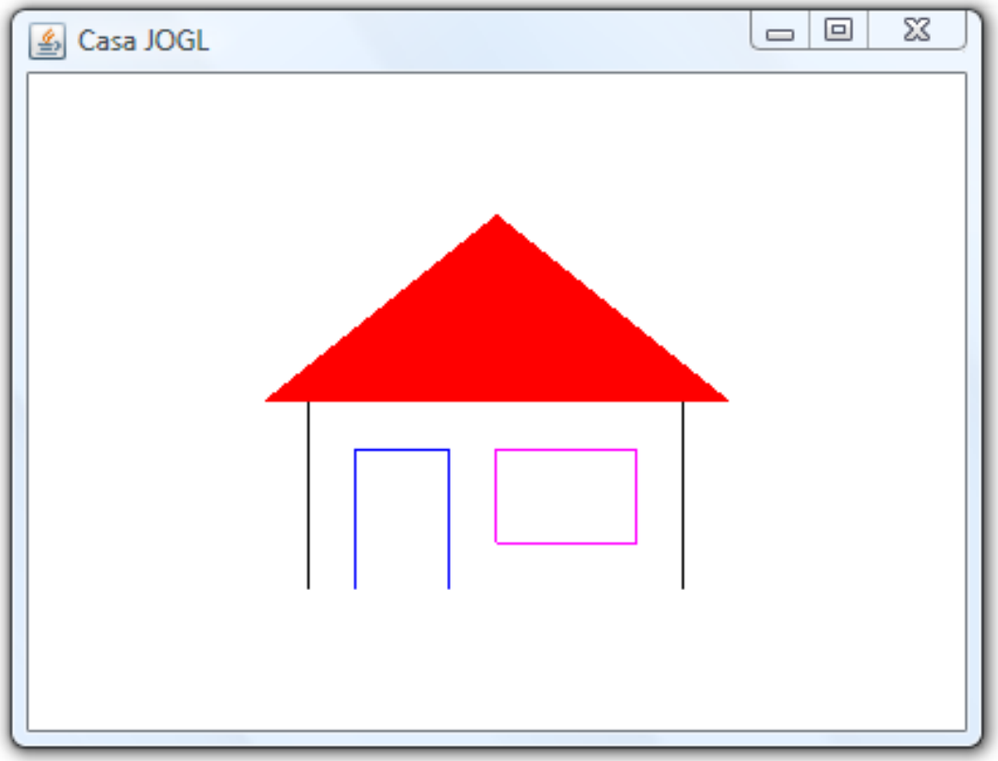


M.Próspero




```
import java.awt.Frame;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.media.opengl.GLEventListener;
import javax.media.opengl.GL;
import javax.media.opengl.GLAutoDrawable;
import javax.media.opengl.GLCanvas;
import javax.media.opengl.glu.GLU;
/**
 * @author M. Próspero
 */
public class Casa implements GLEventListener {

    public void init(GLAutoDrawable gLDrawable) {
      GL gl = gLDrawable.getGL();
      gl.glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
    }

    public void display(GLAutoDrawable gLDrawable) {
      GL gl = gLDrawable.getGL();
      gl.glClear(GL.GL_COLOR_BUFFER_BIT);
      gl.glMatrixMode(GL.GL_MODELVIEW);
      gl.glLoadIdentity();

      gl.glColor3d(0.0, 0.0, 0.0);  // paredes:
      gl.glBegin(GL.GL_LINES);
      gl.glVertex2d(120.0,-220.0);
      gl.glVertex2d(120.0,-140.0);
      gl.glVertex2d(280.0,-220.0);
      gl.glVertex2d(280.0,-140.0);
      gl.glEnd();

      gl.glColor3d(0.0, 0.0, 1.0);  // porta:
      gl.glBegin(GL.GL_LINE_STRIP);
      gl.glVertex2d(140.0, -220.0);
      gl.glVertex2d(140.0, -160.0);
      gl.glVertex2d(180.0, -160.0);
      gl.glVertex2d(180.0, -220.0);
      gl.glEnd();

      gl.glColor3d(1.0, 0.0, 1.0);  // janela:
      gl.glBegin(GL.GL_LINE_LOOP);
      gl.glVertex2d(200.0, -160.0);
      gl.glVertex2d(260.0, -160.0);
      gl.glVertex2d(260.0, -200.0);
      gl.glVertex2d(200.0, -200.0);
      gl.glEnd();

      gl.glColor3d(1.0, 0.0, 0.0);  // telhado:
      gl.glBegin(GL.GL_TRIANGLES);
      gl.glVertex2d(100.0, -140.0);
      gl.glVertex2d(200.0, -60.0);
      gl.glVertex2d(300.0, -140.0);
      gl.glEnd();
    }
```

```
    public void reshape(GLAutoDrawable gLDrawable, int x, int y,
                        int width, int height) {
      GL gl = gLDrawable.getGL();
      GLU glu = new GLU();
      gl.glViewport(0, 0, width, height);
      gl.glMatrixMode(GL.GL_PROJECTION);
      gl.glLoadIdentity();
      glu.gluOrtho2D(0.0, 400.0, -280.0, 0.0);
    }

    public void displayChanged(GLAutoDrawable gLDrawable,
                    boolean modeChanged, boolean deviceChanged) {}

    public static void main(String[] args) {
      Frame frame = new Frame("Casa JOGL");
      GLCanvas canvas = new GLCanvas();
      canvas.addGLEventListener(new Casa());
      canvas.setSize(400, 280);
      frame.add(canvas);
      frame.pack();
      frame.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
          System.exit(0);
        }
      });
      frame.setVisible(true);
    }
}
```



M.Próspero

The OpenGL Sintax

Prefix	Meaning
gl	OpenGL command
GL_	OpenGL defined constant

Suffix	Meaning
2...	Two arguments of type ...
3...	Three arguments of type ...
4...	Four arguments of type ...

Suffix	Data Type	OpenGL Type	C-language Type
b	8-bit integer	GLbyte	signed char
s	16-bit integer	GLshort	short
i	32-bit integer	GLint, GLsizei	long
f	32-bit floating-point	GLfloat, GLclampf	float
d	64-bit floating-point	GLdouble, GLclampd	double
ub	8-bit unsigned integer	GLubyte, GLboolean	unsigned char
us	16-bit unsigned integer	GLushort	unsigned short
ui	32-bit unsigned integer	GLuint, GLenum, GLbitfield	unsigned long
v	pointer to a vector		

M.Próspero