Graphics&Geometric Computing Lab
@USTC

OpenGL Programming
- A Brief Introduction

Ligang Liu
Graphics&Geometric Computing Lab
USTC
http://staff.ustc.edu.cn/~lgliu

What is OpenGL

- Open Graphics Library
- Graphics rendering API
  - Graphics software library
  - Software interface to graphics hardware

Benefits of using OpenGL

- Windows system independent
- Operating system independent
- Standard for graphics programming
- Available on various platforms

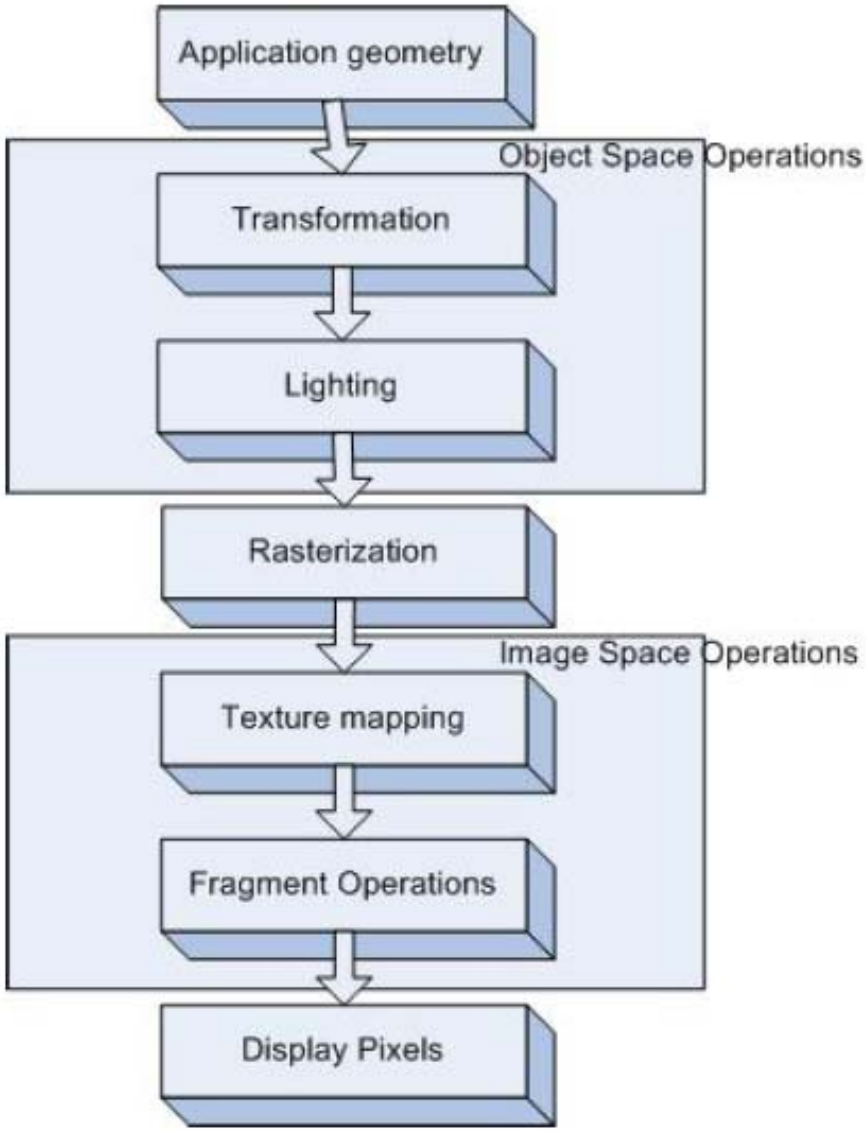
OpenGL as a Rendering Engine

- Geometric primitives
  - Points, lines, polygons
- Image primitives
  - Images and bitmaps
  - Separate pipeline for images and geometry
  - Texture mapping
- State dependent rendering
  - Color, materials, light sources etc.

OpenGL related APIs

- AGL, GLX, WGL
  - Glue between OpenGL and windowing systems for various platform
- GLU (OpenGL Utility Library)
  - Many modeling features : quadric surfaces, NURBS, tessellators
- GLUT(OpenGL Utility Toolkit)
  - Windowing and user interface API. It provides handy shape primitives (torus, teapot, cube) too.

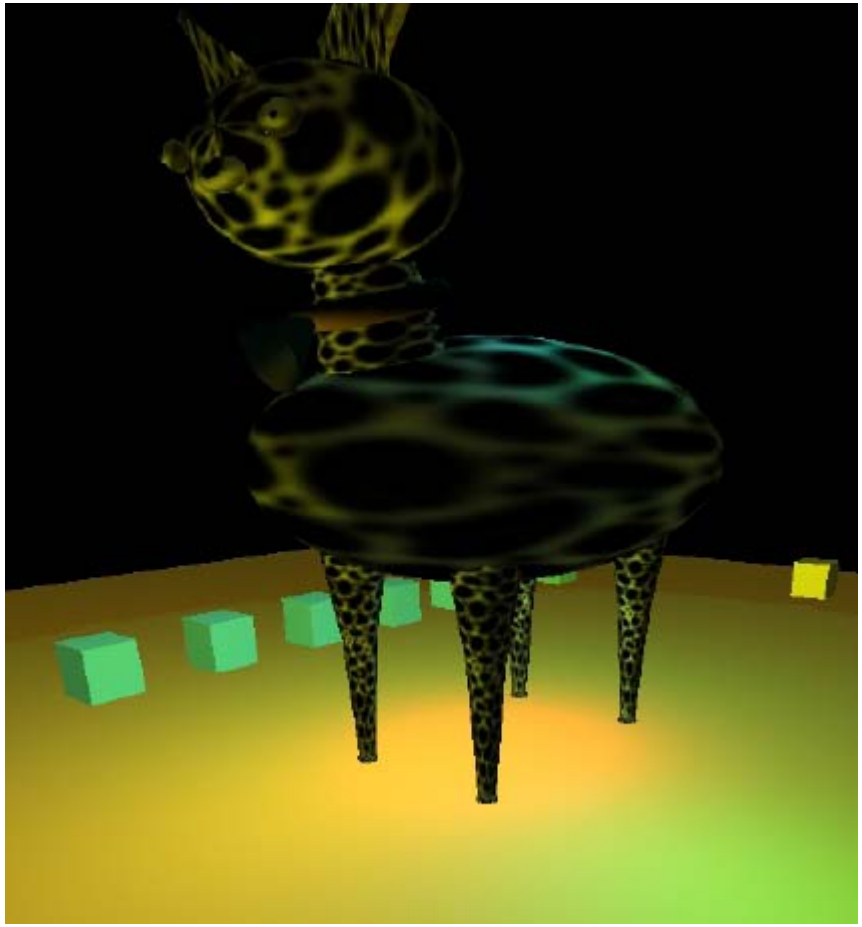
OpenGL Rendering Pipeline



Setting up the OpenGL Environment

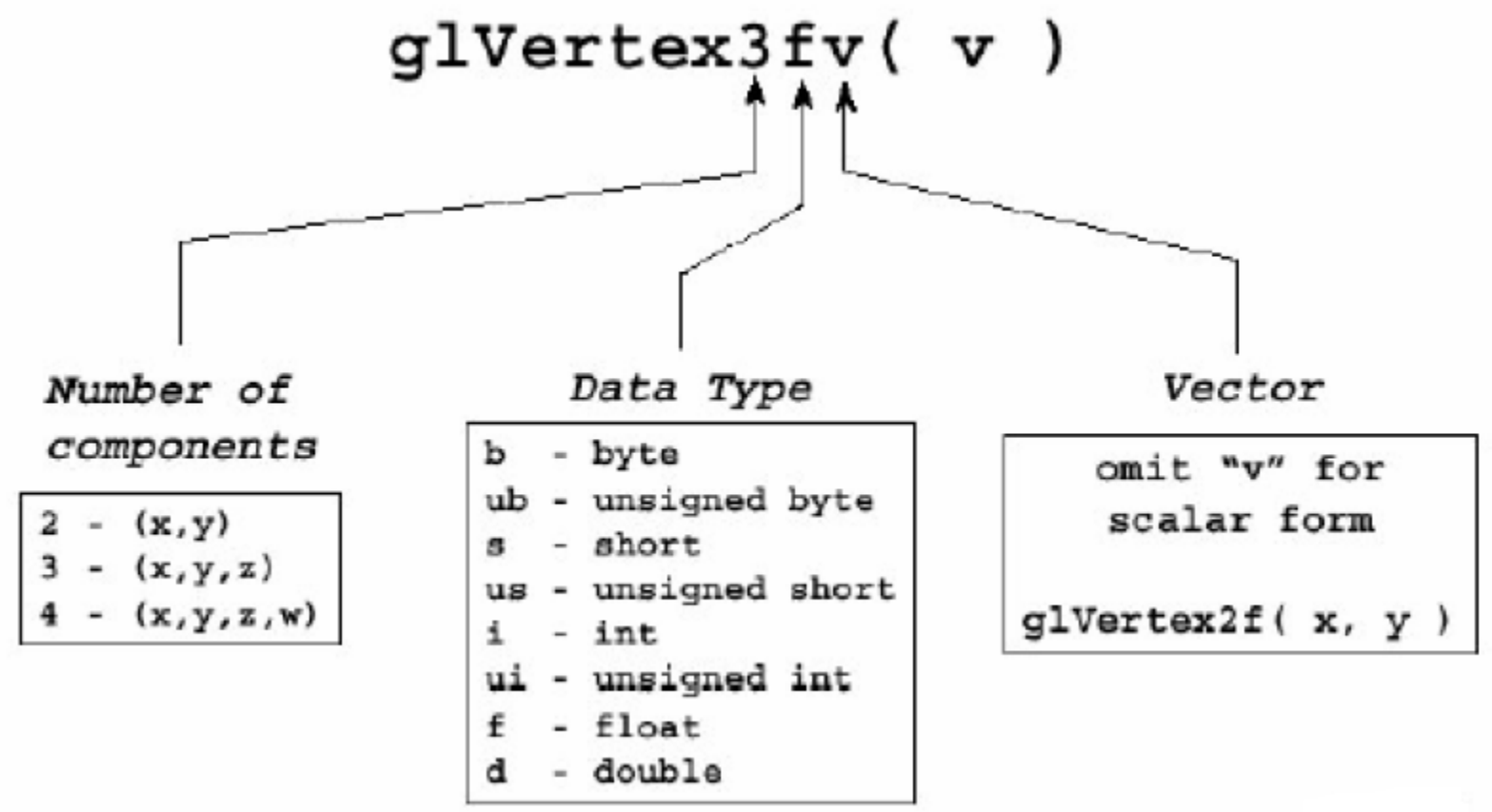
- Header files
  - #include <GL/gl.h>
  - #include <GL/glu.h>
  - #include <GL/glut.h>
- Libraries
  - GL, GLU, GLUT etc
- Built in types (to support multiple platforms)
  - GLfloat, GLint, GLdouble, GLenum, etc.

Sample Rendering



Basic of Drawing in OpenGL

OpenGL Command Formats



OpenGL Command Syntax

- All command names begin with gl
  - Ex.: glVertex3f(0.0, 1.0, 1.0);
- Constant names are in all uppercase
  - Ex.: GL_COLOR_BUFFER_BIT
- Data types begin with GL
  - Ex.: GLfloatonevertex[3];
- Most commands end in two characters that determine the number of expected arguments and data type
  - Ex.: glVertex3f(…) => 3 GLfloatarguments

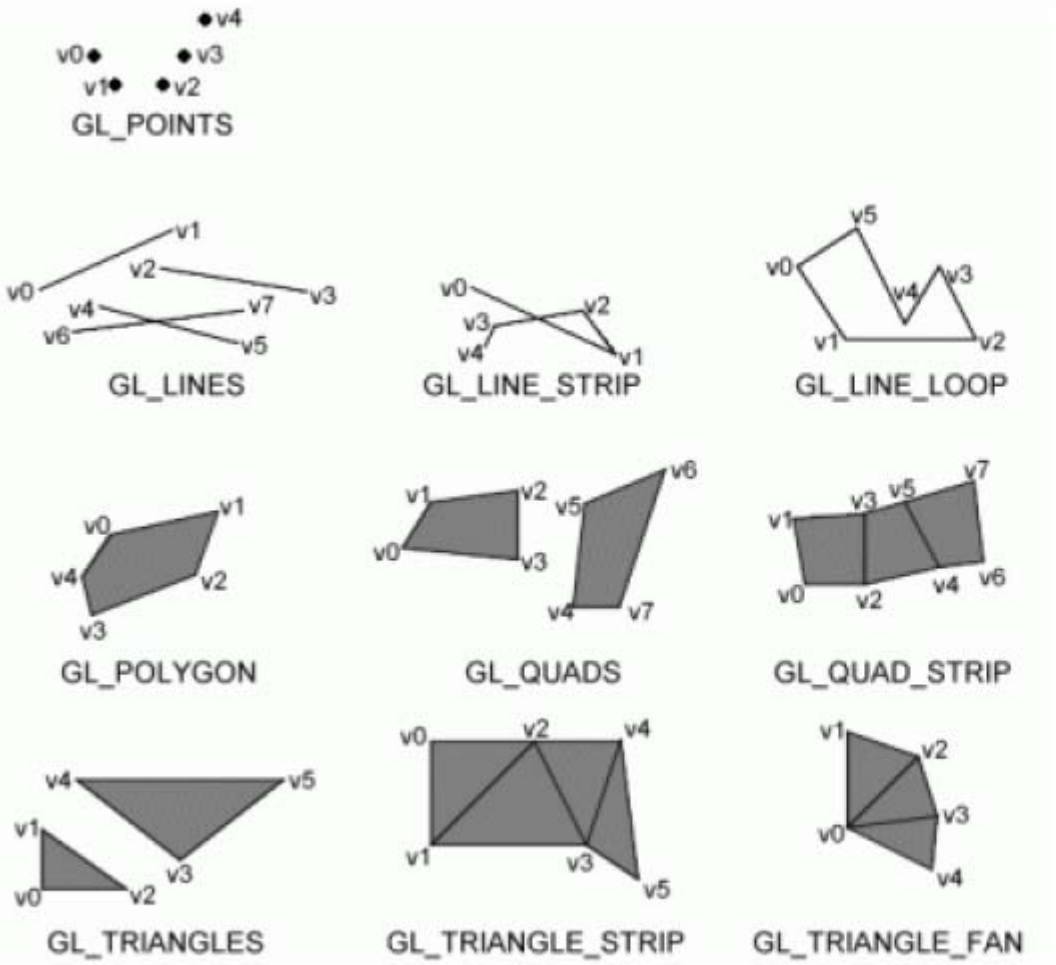
Specifying Vertex

- glVertex{234}{sfid}[v](Type coords);
- Examples
  - glVertex2f(1.0, 2.0)
  - glVertex3i(10, 10 ,20)
  - Glfloatlist[3] = {10, 10, 20}
  - glVertex3iv(list)

Specifying a Color

- glColor{3,4}{bifd…}[v](Typecoords)
  - glColor3f(1.0, 0.0, 0.0) Red
  - glColor3f(0.0, 1.0, 0.0) Green
  - glColor3f(0.0, 0.0, 1.0) Blue
  - glColor3f(1.0, 1.0, 0.0) Yellow

OpenGL Geometric Primitives



GL drawing command:
`glBegin(`***primitive_type***`);`
…
`glEnd();`

where ***primitive_type*** is one of:

1. **GL_POINTS**
2. **GL_LINES**
3. **GL_LINE_STRIP**
4. **GL_LINE_LOOP**
5. **GL_TRIANGLE**
6. **GL_QUADS**
7. **GL_POLYGON**
8. **GL_TRIANGLE_STRIP**
9. **GL_TRIANLE_LINE**
10. **GL_QUAD_STRIP**

Specifying a Square

```
glBegin( GL_QUADS );
        glVertex3f( 0.0, -1.0, 0.0 );
        glVertex3f( 1.0, 0.0, 0.0 );
        glVertex3f( 0.0, 1.0, 0.0 );
        glVertex3f( -1.0, 0.0, 0.0 );
glEnd();
```

Coloring the Square

```
glColor3f(1.0, 1.0, 0.0);
glBegin( GL_QUADS );
        glVertex3f( 0.0, -1.0, 0.0 );
        glVertex3f( 1.0, 0.0, 0.0 );
        glVertex3f( 0.0, 1.0, 0.0 );
        glVertex3f( -1.0, 0.0, 0.0 );
glEnd();
```

OpenGL as a State Machine

- OpenGL is a state machine. Its put it into various states (or modes) while in operation.
- The states remain in effect until changed. E.g.
  - Current color, current line pattern, etc.
    - What color to use –glColor()
      - All subsequent vertices will be assigned that color
    - What rendering mode (wireframe or solid)
      - All subsequent polygons will be rendered that way
- Boolean states can be set with glEnable(), glDisable()
- glGet*() can be used to query any of the thecurrent state variables.

More…

- Model transformation
- View transformation
- Projection transformation
- Lighting
- Texture mapping
- …

Resources

- http://www.opengl.org
- Siggraph courses
- OpenGL red book
- OpenGL blue book
- So many books on OpenGL
- So many websites on OpenGL

About DirectX

Q&A