






**An Interactive Introduction to
OpenGL Programming**

Dave Shreiner
Ed Angel
Vicki Shreiner




What You'll See Today

- General OpenGL Introduction
- Rendering Primitives



OpenGL and GLUT Overview

Vicki Shreiner




OpenGL and GLUT Overview

- What is OpenGL & what can it do for me?
- OpenGL in windowing systems
- Why GLUT
- A GLUT program template




What Is OpenGL?

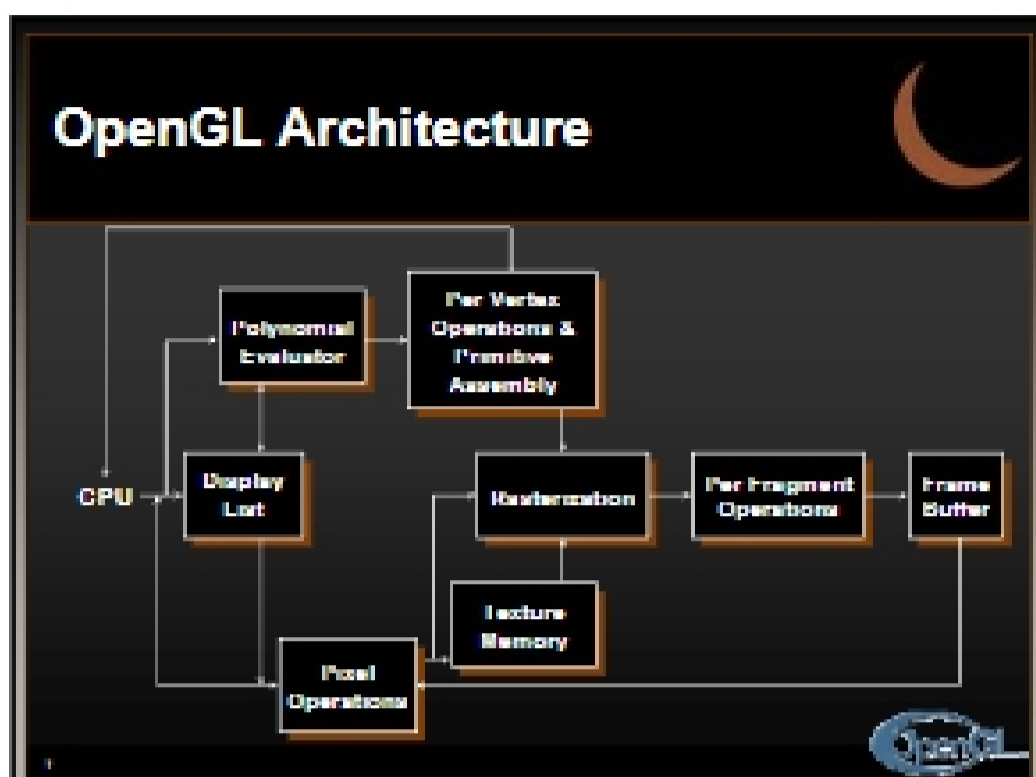
- A State Machine!



What Is OpenGL?

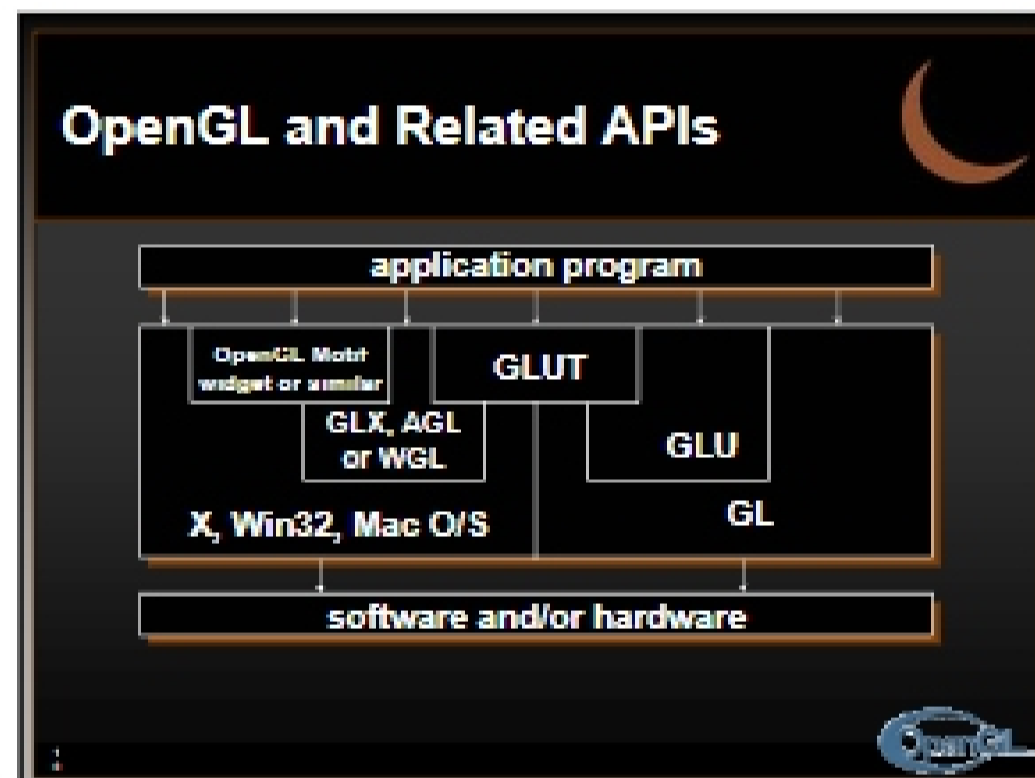
- Graphics rendering API
 - high-quality color images composed of geometric and image primitives
 - window system independent
 - operating system independent





- ### OpenGL as a Renderer
- **Geometric primitives**
 - points, lines and polygons
 - **Image Primitives**
 - images and bitmaps
 - separate pipeline for images and geometry
 - linked through texture mapping
 - **Rendering depends on state**
 - normals, colors, materials, light sources, etc.

- ### Related APIs
- **AGL, GLX, WGL**
 - glue between OpenGL and windowing systems
 - **GLU (OpenGL Utility Library)**
 - part of OpenGL
 - NURBS, tessellators, quadric shapes, etc.
 - **GLUT (OpenGL Utility Toolkit)**
 - portable windowing API
 - not officially part of OpenGL



- ### Preliminaries
- **Headers Files**
 - #include <GL/gl.h>
 - #include <GL/glu.h>
 - #include <GL/glut.h>
 - **Libraries**
 - **Enumerated Types**
 - OpenGL defines numerous types for compatibility
 - GLfloat, GLint, GLenum, etc.

- ### GLUT Basics
- **Application Structure**
 - Configure and open window
 - Initialize OpenGL state
 - Register input callback functions
 - render
 - resize
 - input: keyboard, mouse, etc.
 - Enter event processing loop

Sample Program

```
void main( int argc, char** argv )
{
    int mode = GLUT_RGB|GLUT_DOUBLE;
    glutInitDisplayMode( mode );
    glutCreateWindow( argv[0] );
    init();
    glutDisplayFunc( display );
    glutReshapeFunc( reshape );
    glutKeyboardFunc( key );
    glutIdleFunc( idle );
    glutMainLoop();
}
```

OpenGL Initialization

- Set up whatever state you're going to use

```
void init( void )
{
    glClearColor( 0.0, 0.0, 0.0, 1.0 );
    glClearDepth( 1.0 );

    glEnable( GL_LIGHT0 );
    glEnable( GL_LIGHTING );
    glEnable( GL_DEPTH_TEST );
}
```

GLUT Callback Functions

- Routine to call when something happens
 - window resize or redraw
 - user input
 - animation
- “Register” callbacks with GLUT

```
glutDisplayFunc( display );
glutIdleFunc( idle );
glutKeyboardFunc( keyboard );
```

Rendering Callback

- Do all of your drawing here

```
glutDisplayFunc( display );
```

```
void display( void )
{
    glClear( GL_COLOR_BUFFER_BIT );
    glBegin( GL_TRIANGLE_STRIP );
    glVertex3fv( v[0] );
    glVertex3fv( v[1] );
    glVertex3fv( v[2] );
    glVertex3fv( v[3] );
    glEnd();
    glutSwapBuffers();
}
```

Idle Callbacks

- Use for animation and continuous update

```
glutIdleFunc( idle );
```

```
void idle( void )
{
    t += dt;
    glutPostRedisplay();
}
```

User Input Callbacks

- Process user input

```
glutKeyboardFunc( keyboard );
```

```
void keyboard( char key, int x, int y )
{
    switch( key ) {
        case 'q' : case 'Q' :
            exit( EXIT_SUCCESS );
            break;

        case 'r' : case 'R' :
            rotate = GL_TRUE;
            break;
    }
}
```