OpenGL’s State Machine

All rendering attributes are encapsulated in the OpenGL State
- rendering styles
- shading
- lighting
- texture mapping

Manipulating OpenGL State

Appearance is controlled by current state
for each (primitive to render)
update OpenGL state
render primitive

Manipulating vertex attributes is most common way to manipulate state
- glColor*() / glIndex*()
- glNormal*()
- glTexCoord*()

Controlling current state

Setting State
- glPointSize( size );
- glLineStipple( repeat, pattern );
- glShadeModel( GL_SMOOTH );

Enabling Features
- glEnable( GL_LIGHTING );
- glDisable( GL_TEXTURE_2D );

An Interactive Introduction to OpenGL Programming

Dave Shreiner
Ed Angel
Vicki Shreiner

What You’ll See Today

General OpenGL Introduction
Rendering Primitives
Rendering Modes
Lighting
Texture Mapping
Additional Rendering Attributes
Imaging

Goals for Today

Demonstrate enough OpenGL to write an interactive graphics program with
- custom modeled 3D objects or imagery
- lighting
- texture mapping

Introduce advanced topics for future investigation
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:
- colors, materials, light sources, etc.

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

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

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 );
```
```c
void display( void )
{
    glClear( GL_COLOR_BUFFER_BIT );
    glBegin( GL_TRIANGLES );
    glVertex3f( v[0] );
    glVertex3f( v[1] );
    glVertex3f( v[2] );
    glEnd();
    glutSwapBuffers();
}
```
Idle Callbacks

Use for animation and continuous update

```c
glutIdleFunc( idle );
```

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

User Input Callbacks

Process user input

```c
glutKeyboardFunc( keyboard );
```

```c
void keyboard( unsigned char key, int x, int y )
{
  switch( key )
  {
    case 'q' : case 'Q' : exit( EXIT_SUCCESS );
    case 'r' : case 'R' :
      rotate = GL_TRUE;
      glutPostRedisplay();
      break;
  }
}
```