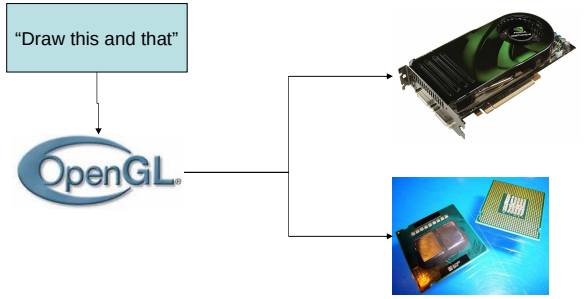
Intro to…

Steven An

CS4620 2008 Fall

What is it?

- API Standard for 3D rasterization
  - Isolates you from underlying implementation



Brief History

- Developed by SGI, for CAD, modeling
- Adopted by gaming, GPUs (Quake)
- Main competitor: Direct3D (Microsoft)

Conceptual Architecture

- State machine
  - You send commands, it operates/remembers

Source: http://www.java-tips.org/other-api-tips/jogl/3d-shapes-nehe-tutorial-jogl-port.html

Usage Overview

- Create OpenGL Context
- Set it up
  - Viewport size, features, etc.
- Every frame, send draw commands
  - Adjust its state as needed

What to Draw

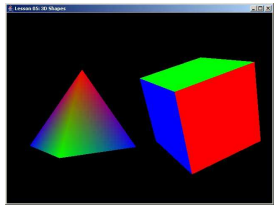
- Send triangle vertices:
  - gl.glBegin(GL.GL_TRIANGLES);
  - gl.glVertex3f(0.0f, 1.0f, 0.0f);
  - …
  - gl.glEnd();
- …or as a buffer in memory

How to Draw: Transformations

- Can set a transformation to be applied to subsequent draw-commands
  - glRotate*, glTranslate*, glScale*, glLoadIdentity, etc. functions affect it
  - IMPORTANT: These are post-multiplies!
- Also, maintains a stack for you!
  - glPushMatrix – copies the top, pushes on
  - glPopMatrix – removes top
  - Top matrix always active
- Well-suited for scene hierarchies
  - Depth-first traversal
  - Pattern: push, transform, draw recursively, pop

How to Draw: Material parameters

- Needed for basic lighting. Ex:
  - gl.glMaterialfv(GL.GL_FRONT, GL.GL_DIFFUSE, { 1, 0, 0, 1 }, 0);
  - glColor also
- Other parameters: Specular, ambient, shininess, normal



Example

p’ = TRp

```
gl.glLoadIdentity();
gl.glTranslatef(-1.5f, 0.0f, -6.0f);
gl.glRotatef(rtri, 0.0f, 1.0f, 0.0f);
gl.glBegin(GL.GL_TRIANGLES);         // Drawing Using Triangles
gl.glColor3f(1.0f, 0.0f, 0.0f);      // Red
gl.glVertex3f(0.0f, 1.0f, 0.0f);     // Top Of Triangle (Front)
gl.glColor3f(0.0f, 1.0f, 0.0f);      // Green
gl.glVertex3f(-1.0f, -1.0f, 1.0f);   // Left Of Triangle (Front)
gl.glColor3f(0.0f, 0.0f, 1.0f);      // Blue
gl.glVertex3f(1.0f, -1.0f, 1.0f);    // Right Of Triangle (Front)
gl.glColor3f(1.0f, 0.0f, 0.0f);      // Red
gl.glVertex3f(0.0f, 1.0f, 0.0f);     // Top Of Triangle (Right)
gl.glColor3f(0.0f, 0.0f, 1.0f);      // Blue
gl.glVertex3f(1.0f, -1.0f, 1.0f);    // Left Of Triangle (Right)
gl.glColor3f(0.0f, 1.0f, 0.0f);      // Green
gl.glVertex3f(1.0f, -1.0f, -1.0f);   // Right Of Triangle (Right)
gl.glColor3f(1.0f, 0.0f, 0.0f);      // Red
gl.glVertex3f(0.0f, 1.0f, 0.0f);     // Top Of Triangle (Back)
gl.glColor3f(0.0f, 1.0f, 0.0f);      // Green
gl.glVertex3f(1.0f, -1.0f, -1.0f);   // Left Of Triangle (Back)
gl.glColor3f(0.0f, 0.0f, 1.0f);      // Blue
gl.glVertex3f(-1.0f, -1.0f, -1.0f);  // Right Of Triangle (Back)
gl.glColor3f(1.0f, 0.0f, 0.0f);      // Red
gl.glVertex3f(0.0f, 1.0f, 0.0f);     // Top Of Triangle (Left)
gl.glColor3f(0.0f, 0.0f, 1.0f);      // Blue
gl.glVertex3f(-1.0f, -1.0f, -1.0f);  // Left Of Triangle (Left)
gl.glColor3f(0.0f, 1.0f, 0.0f);      // Green
gl.glVertex3f(-1.0f, -1.0f, 1.0f);   // Right Of Triangle (Left)
gl.glEnd();                          // Finished Drawing The Triangle
```

Conclusion & Useful Resources

- Great for prototyping, learning
- Tons of other features: shaders, etc.
- http://www.opengl.org/sdk/docs/man/
- JOGL specific: http://www.java-tips.org/other-api-tips/jogl/3d-shapes-nehe-tutorial-jogl-port.html