OpenGL Help Session

CS248 Fall 2006

Zhengyun Zhang

References

- OpenGL Programming Guide (Red Book)
  - start here
- OpenGL Reference Guide (Blue Book)
  - use this to look up individual functions
- OpenGL Shading Language (Orange Book)
  - gets you started with GLSL

OpenGL is a state machine

- States like projection matrix, current vertex color, etc.
- We can change the states by using GL function calls like glPushMatrix.
- The state is used when we are drawing primitives.

OpenGL needs to be connected to the windowing system

- OpenGL by itself does not talk to the windowing system/ manager by itself.
- Need a toolkit to tell the windowing system that we need an OpenGL window.
- Examples:
  - GLUT (used in Project 2)
  - SDL (recommended for Project 3)
  - wxWidgets, Qt (full fledged widget toolkits, probably overkill for a game)

OpenGL function suffixes

- OpenGL functions that take different types of arguments while providing the same functionality will often have a suffix to denote which type of function they are:
  - glVertex2i - input is 2 integers
  - glVertex3fv - input is 3 floats in an array
  - glVertex3f - input is 3 floats

OpenGL function suffixes

- Extensions to the OpenGL base system often have their own suffixes. For example:
  - glCreateProgramObjectARB
    (to create a shader program using an ARB extension)
- Use GLEW (OpenGL Extension Wrangler) for easy access to extensions.

Other OpenGL Hints

- OpenGL uses a right-handed coordinate system.
- Light positions are not sent through the Model-View matrix.
- Projection matrix should only be used for camera position, etc. It has a shorter stack than the Model-View matrix.
- New matrix is multiplied on the right. Latest matrix to be multiplied on is the *first* operation to be performed on the vertex locations.

Basic OpenGL Game Flowchart

1. Load up an OpenGL window using a toolkit to talk to the windowing system
2. Set up projection matrices, shading properties, etc. Load textures, etc.
3. Event loop:
    1. Check for any events or user input and process them
    2. Redraw the OpenGL scene as necessary
    3. Wait a small amount of time.

Sample OpenGL Program

- Based on the example SDL application Andrew wrote that's linked off the project 3 handout: http://graphics.stanford.edu/courses/cs248-06/SDLDemo.zip
- I'm going to demonstrate a Python version, but the overall structure of the program should be the same.