
Lecture 16

OpenGL

What do you need (on a PC)?

http://www.smartcg.com/tech/cg/courses/CS480/misc/OpenGL_PC

gl.h

glu.h

glut.h

glu32.lib

glut32.lib

opengl32.lib

glu32.dll [part of XP, in C:\WINDOWS\system32]

glut32.dll

opengl32.dll [part of XP, in C:\WINDOWS\system32]

GL, GLU, GLUT

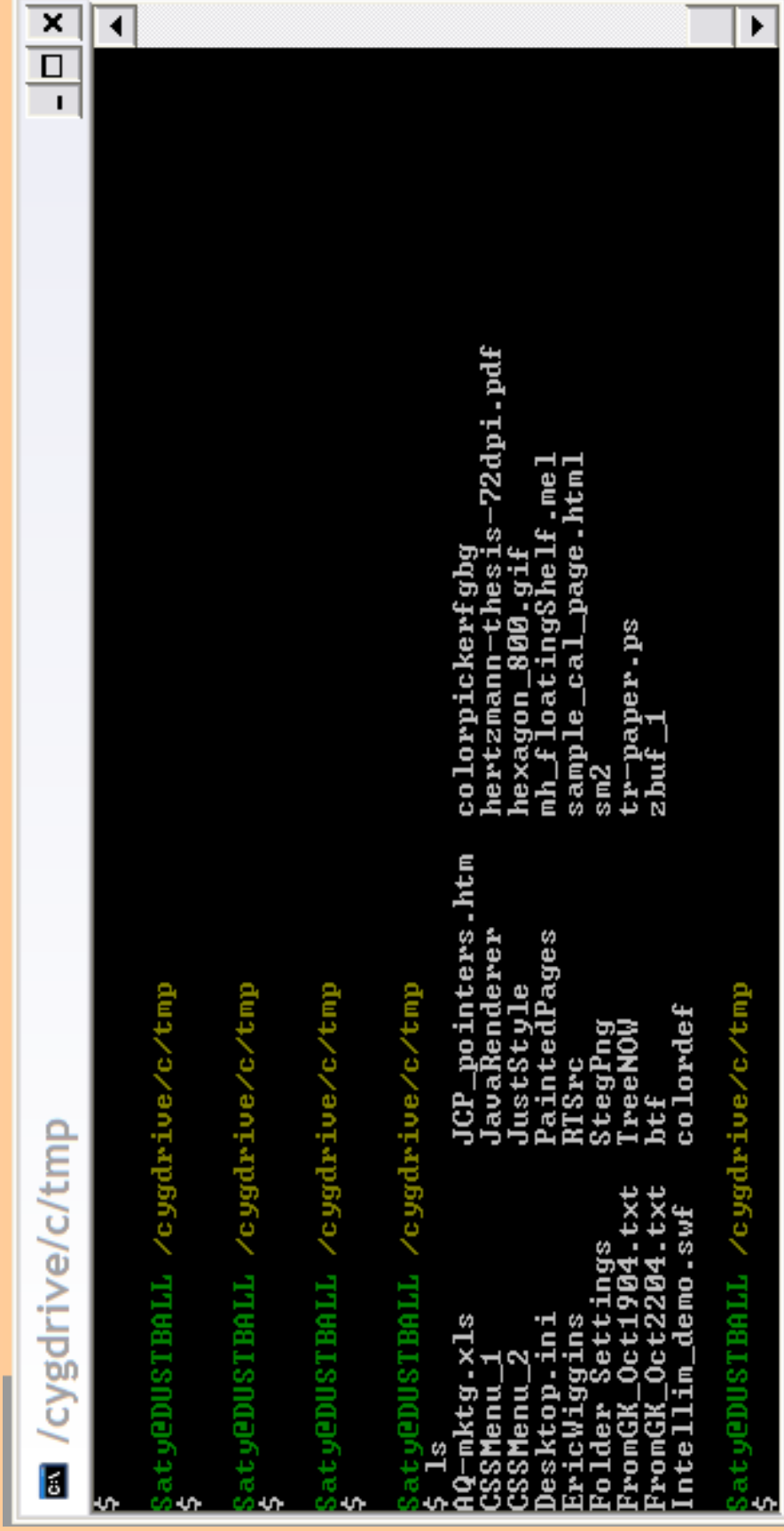
GL: Graphics library calls.

GLU: OpenGL Utility Library. This is a set of functions to create texture mipmaps from a base image, map coordinates between screen and object space, and draw quadric surfaces and NURBS.

GLUT: GLUT is the OpenGL Utility Toolkit, a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. GLUT makes it considerably easier to learn about and explore OpenGL Programming.

Unix (bash shell) on a PC

Cygwin: www.cygwin.com















```
cyg /cygdrive/c/tmp
$
$ Saty@DUSTIBALL /cygdrive/c/tmp
$
$ Saty@DUSTIBALL /cygdrive/c/tmp
$
$ Saty@DUSTIBALL /cygdrive/c/tmp
$
$ Saty@DUSTIBALL /cygdrive/c/tmp
$ ls
AQ-mktg.xls          JCP_pointers.htm
CSSMenu_1            JavaRenderer
CSSMenu_2            JustStyle
Desktop.ini          PaintedPages
EricWiggins          RTSrc
Folder Settings     StegPng
FromGK_Oct1904.txt  TreeNOW
FromGK_Oct2204.txt  btf
Intelliim_demo.swf  colordef
$ Saty@DUSTIBALL /cygdrive/c/tmp
$
```

The screenshot shows a terminal window with a white title bar containing the text "cyg /cygdrive/c/tmp". The terminal background is black with white text. The user has entered several commands, including the 'ls' command, which lists various files and folders in the current directory. The files listed include AQ-mktg.xls, CSSMenu_1, CSSMenu_2, Desktop.ini, EricWiggins, Folder Settings, FromGK_Oct1904.txt, FromGK_Oct2204.txt, Intelliim_demo.swf, JCP_pointers.htm, JavaRenderer, JustStyle, PaintedPages, RTSrc, StegPng, TreeNOW, btf, colordef, colorpickerfgyb, hertzmann-thesis-72dpi.pdf, hexagon_800.gif, mh_floatingShelf.mel, sample_cal_page.html, sm2, tr-paper.ps, and zbuf_1. The prompt '\$' is visible at the beginning and end of each line.

What else do you need?

source code (.c program)

makefile

 glu32.dll	137 KB	Application Exte...	7/29/1996 12:09...
 glut32.dll	166 KB	Application Exte...	8/18/1998 4:25 ...
 opengl32.dll	670 KB	Application Exte...	8/18/2001 4:00 ...
 h	67 KB	C Header file	5/22/1996 6:52 ...
 h	18 KB	C Header file	5/17/1996 3:35 ...
 h	21 KB	C Header file	8/18/1998 4:24 ...
 c	1 KB	C Source file	10/31/2004 11:2...
 c	1 KB	C Source file	10/31/2004 10:5...
 makefile	1 KB	File	10/31/2004 11:3...
 glu32.lib	49 KB	LIB File	6/12/1996 3:37 ...
 glut32.lib	79 KB	LIB File	8/18/1998 4:25 ...
 opengl32.lib	329 KB	LIB File	7/16/1996 3:33 ...

makefile

Here's how you'd compile simple.c to get simple.exe:

make simple

INCLPATH = -I .

LIBPATH = -L .

Win XP comes pre-installed with the latest

version of glu32.dll and opengl32.dll. So

if you get errors about missing glu calls,

do the following:

a. hide glu32.dll, glu32.lib, opengl32.lib

and open32.lib in the current dir. (create

a subdir. and move them there)

b. use this alternate version of LIBS, which

will use the dll files in WINDOWS\system32:

LIBS = -lglut32 -lglu32 -lopengl32

LIBS = -lglut32 -lglu32 opengl32.lib

%.%.c

gcc \$(INCLPATH) \$(LIBPATH) \$^ \$(LIBS) -o \$@

simple.c [compile using 'make simple']

```
#include <stdio.h>
#include "glut.h"

void draw()
{
    glClearColor(0,0,0,0);
    glClear(GL_COLOR_BUFFER_BIT);

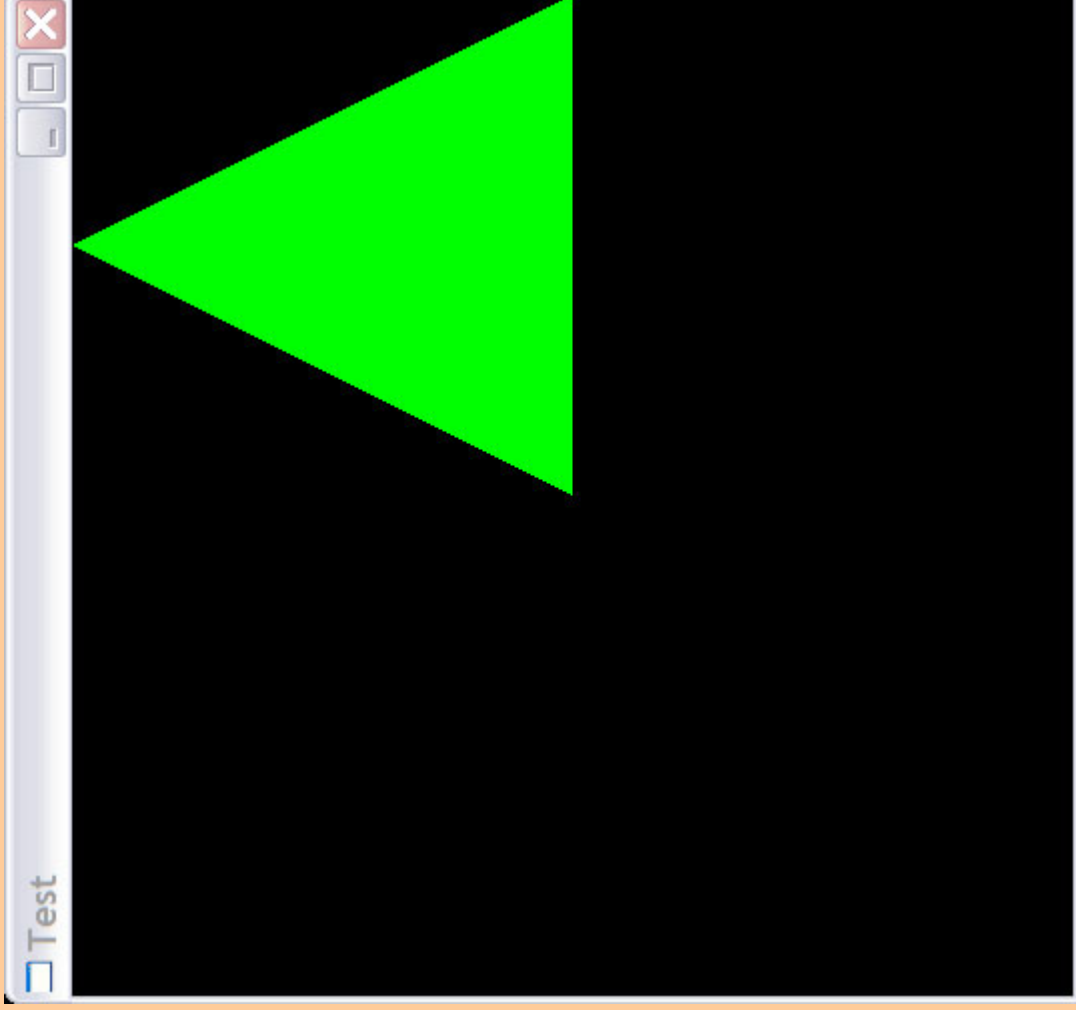
    glColor3f(0, 1, 0);
    glBegin(GL_TRIANGLES);

        glVertex2f(0,0);
        glVertex2f(1,0);
        glVertex2f(0.5, 1);

    glEnd();
    glutSwapBuffers();
}
```

```
int main(int argc, char **argv)
{
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
    glutInitWindowSize(500,500);
    glutCreateWindow("Test");
    glutDisplayFunc(draw);
    glutMainLoop();
}
```

simple.exe

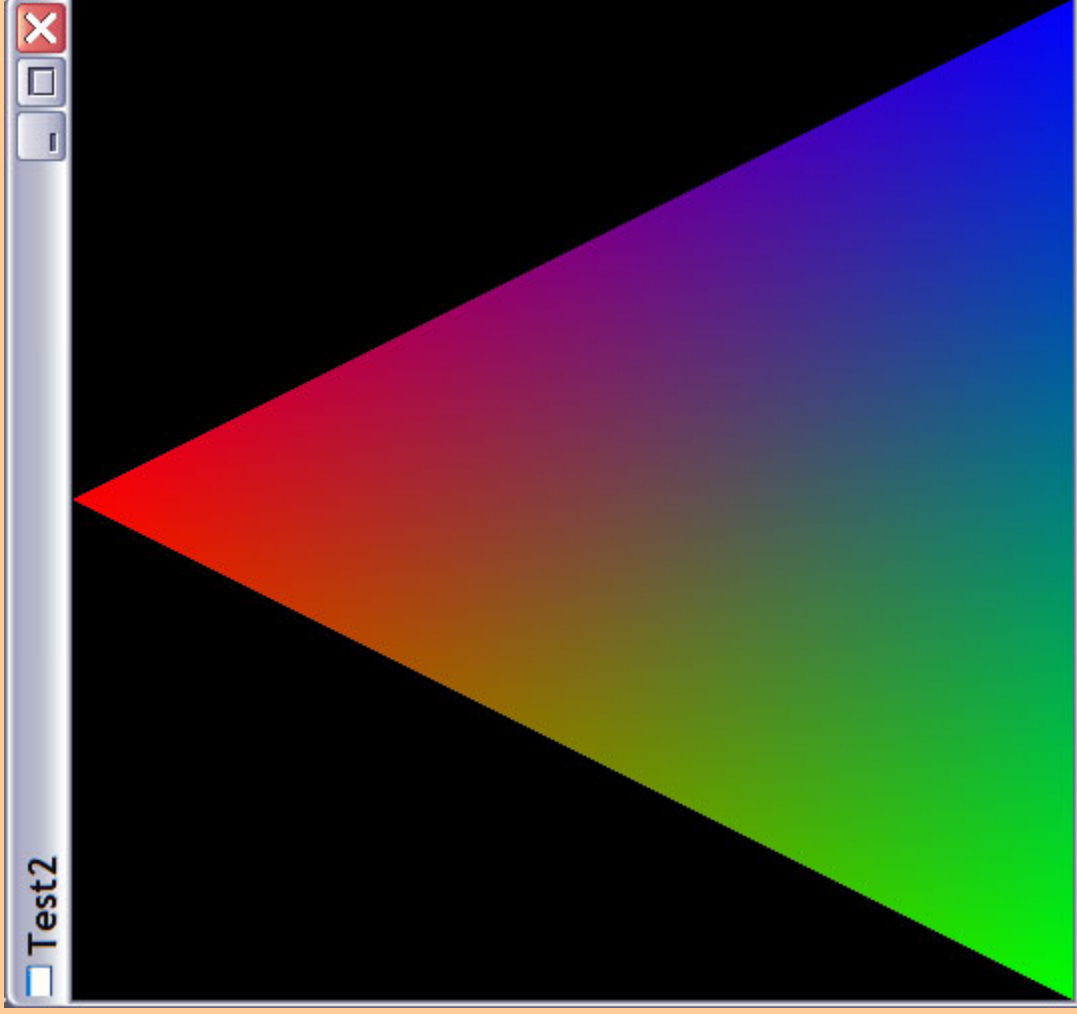


simple2.c

```
#include <stdio.h>
#include "glut.h"
void draw()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    glColor3f(1.0f, 0.0f, 0.0f);
    glVertex2i(0, 1);
    glColor3f(0.0f, 1.0f, 0.0f);
    glVertex2i(-1, -1);
    glColor3f(0.0f, 0.0f, 1.0f);
    glVertex2i(1, -1);
    glEnd();
    // glFlush();
    glutSwapBuffers();
}

int main(int argc, char **argv)
{
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Test2");
    glutDisplayFunc(draw);
    glutMainLoop();
}
```

simple2.exe



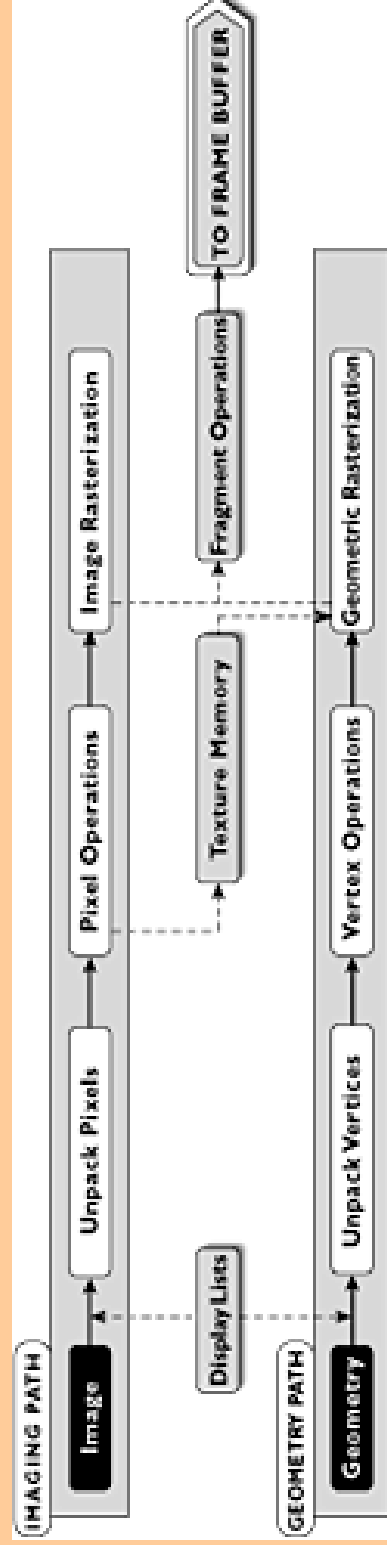
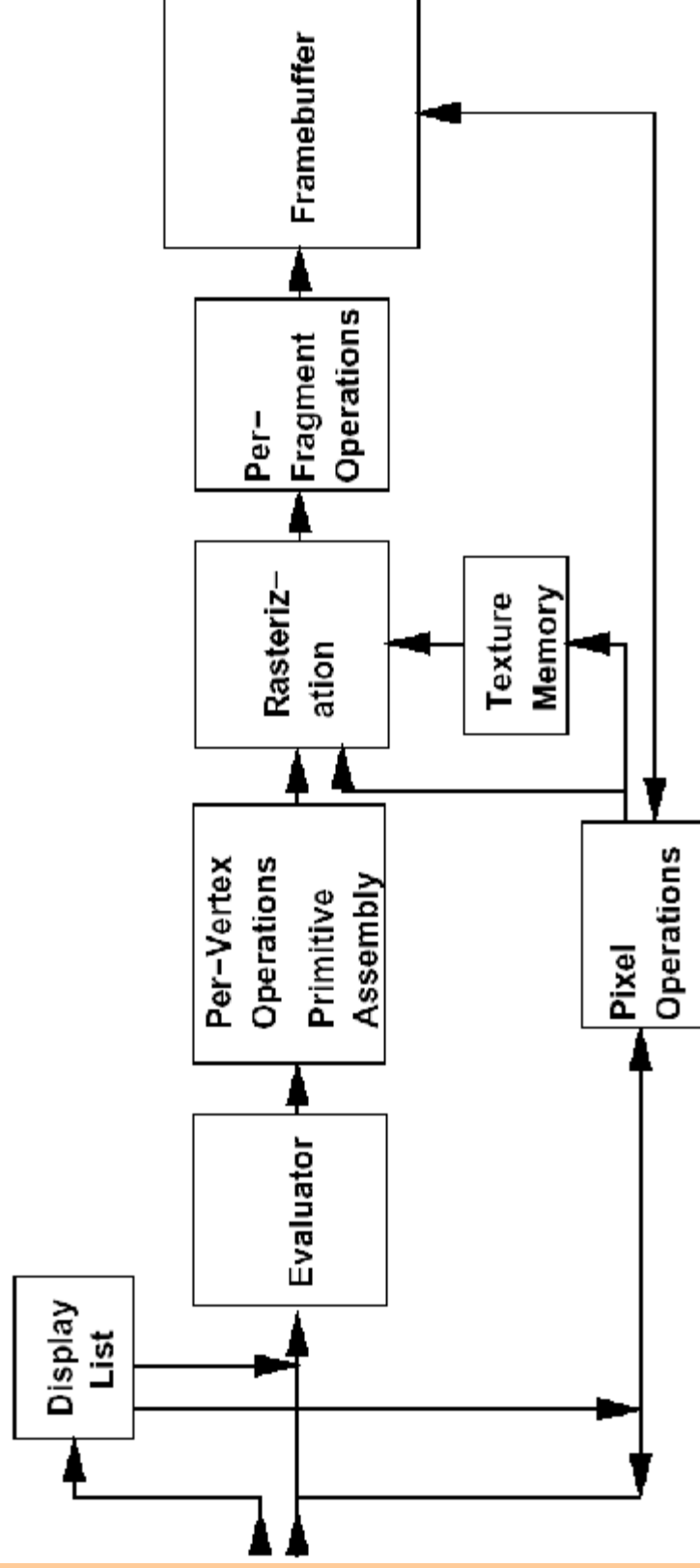
What is OpenGL?

The OpenGL® API is the premier environment for developing portable, interactive 2D and 3D graphics applications. A low-level, vendor-neutral software interface, the OpenGL API has often been called the "assembler language" of computer graphics. In addition to providing enormous flexibility and functionality, OpenGL applications enjoy the broadest platform accessibility in the industry. Applications in markets such as CAD, content creation, energy, entertainment, game development, manufacturing, medical, and VRML have benefited from the breadth of platform accessibility and depth of functionality of the OpenGL API. Since SGI introduced the OpenGL API in 1992, it has grown into the industry's leading cross-platform 2D and 3D graphics API and its presence continues to grow every day.

Benefits of the API

- ▶ **standard across platforms**
- ▶ **hardware implementations!**

OpenGL pipeline



Feature set (highlights)

Accumulation buffer
Alpha blending
Anti-aliasing
Color-index mode
Display list
Double buffering
Feedback
Gouraud shading
Immediate mode
Materials lighting and shading

Pixel operations
Polynomial evaluators
Primitives
RGBA mode
Selection and picking
Stencil planes
Texture mapping
Three-dimensional texturing
Transformation
Z-buffering

Function call categories

OpenGL

Primitives
Vertex arrays
Coordinate transformations
Coloring and lighting
Clipping
Rasterization
Pixel operations
Textures (and fog)
Framebuffer operations
Evaluators
Selection and feedback
Display lists
Modes and execution
State queries

GLU

Coordinate transformations
Manipulating images for texturing
Polygon tessellation
Quadric objects
Rendering spheres, cylinders, disks
NURBS curves, surfaces
Describing errors

GLUT

Beginning event processing
Initialization
Window mgmt.
Overlay mgmt.
Menu mgmt.
Callback registration
Color index colormap mgmt.
State retrieval
Font rendering
Geometric object rendering

Resources

OpenGL consortium: www.opengl.org

OpenGL Programming Guide [“red book”]

OpenGL Reference Manual [“blue book”]

SIGGRAPH Course Notes (eg. 1996, '97, '98, 2004)

Hearn and Baker: Computer Graphics with OpenGL, 3rd Ed.

Angel: Interactive Computer Graphics: A Top-Down Approach
Using OpenGL, 3rd Ed.

<http://www.opengl.org/resources/code>

<http://www.xmission.com/~nate/opengl.html>