CS/EECE 433 Fall, 2006
E. Angel

Assignment One: Graphics Performance

The two purposes of this assignment are to get you started programming in OpenGL and to learn something about real vs. theoretical performance of graphics systems.

Your program should generate a display with a known number of primitives. If you use something like the sphere in section 2.4.3 you can easily modify the program to switch among polygons, quads, quad strips, triangles, and triangle strips. By increasing the number of spheres and their size (and location) you can control the number of primitives and the number of pixels generated. By using glPolygonMode, your can switch between drawing edges and filling the interiors of polygons. By switching among these options and using some sort of timer, you should be able to determine the performance of the system that you are using.

You should turn in a report on your results. It should answer the following questions:

How many polygons per second can you render unfilled and filled?
Does the size of the polygons matter?
What is the effect of the context switches when you try to assign a different color to each vertex?
What is the difference between using quads, quadstrips, triangles, and triangle strips vs. polygons?
What is the effect of putting multiple quads or triangles between a glBegin and a glEnd instead of using a glBegin and glEnd for each one?
Is there any difference if, for a fixed number of primitives, all them are in the clipping volume?
Does the form of glVertex matter, e.g. glVertex3f, glVertex3fv, glVertex3i?
How close can you come to the advertised performance of your graphics card?

Notes: There are a number of ways to compute elapsed time. One is to use some method provided by your operating system. Another is to use glutGet(GLUT_ELAPSED_TIME) which returns the time in milliseconds since glutInit or the first invocation of the function.

Most graphics cards allow the option of coupling the frame rate of the display and the redisplay of the of graphics. In linux, this is sometimes done by setting an environment variable such as VSYNC_TO_BLANK. On Windows systems, you can often set this option through the display control panel. On the Mac, you can set it through a preferences menu that that you can pull down when running an OpenGL program. For animated displays, you usually want this option on since most simple programs will generate their output in less than a frame time. However, for this assignment you probably do not want it on since the graphics card will most likely be sitting idle most of the time which will lead to misleading results.

If you use the sphere for your primitives, can you avoid having the execution time dominated by the computation of sines and cosines? Hint: Use a table.

Note: Since we have not discussed shading in OpenGL, the best you will be able to do is display polygons either in a solid color or let OpenGL interpolate the colors assigned to each vertex.

You should also turn in a copy of your code and be prepared to demo your program.

Due: Tuesday, September 19.