CS5500 Computer Graphics
Graphics Pipeline (Part 1 of 3)

Assigned: April 24, 2006

Due: 23:59 May 4, 2006 (10% penalty for each day late)

Implement the front end of the 3D graphics pipeline, which can handle:

(1) Lines in 3D world space

(2) Translation, rotation, and scaling

(3) Viewing transformation and projection

The input file to your program looks like the following:

```
eye: xEye, yEye, zEye
lookat: xLook, yLook, zLook
up: xUp, yUp, zUp
Perspective: vFOV, near, far
display: width, height
translate: x, y, z        <--- Optional
rotate: angle, xAxis, yAxis. zAxis <--- Optional
Line: x1, y1, z1, x2, y2, z2
Line: x1, y1, z1, x2, y2, z2
...
```

TASKS

(a) Find or build a library that can handle vector and matrix operations such as multiplication, dot product, and cross product.

(b) Set up your viewing matrix using the eye, lookat, and up vectors.

(c) Set up your projection matrix using the vFOV, near, and far parameters. The aspect ratio may be obtained from the window width and height.

(d) Integrate the above modules so that you may transform the 3D lines to the 2D screen space. Then apply the Bresenham's line drawing algorithm to display them.

(e) Note that the only OpenGL functions you are allowed to use are those for drawing pixels. To draw the pixels on the screen, you may simply draw them as 2D points using OpenGL, such as:

```
glVertex2f(2.0*x/width-1.0, 2.0*y/height-1)
```

There are many other approaches which you're free to explore. For examples, you may use the OpenGL bitmap or pixel map operations. You may also draw the pixels using Microsoft Windows GDI functions. Using those approaches will be certainly more efficient than drawing the pixels as 3D OpenGL points. However, the efficiency of the pixel drawing will not affect your score in this assignment.

(f) Optionally, you may allow modeling transformation (such as rotation, translation, and scaling) to the 3D lines.

Submit your program source files online through the course webpage.