CMSI 371

COMPUTER GRAPHICS

Spring 2007

Assignment 0405

Time to apply the projection/transformation mathematics that you've seen so far.

Not For Submission

1. Read Angel Chapter 5, particularly Sections 5.1 to 5.5 and Sections 5.8 to 5.10, for additional details on the mathematics behind projection and camera positioning.
2. Read Angel Chapter 7, Sections 7.1 to 7.7.
3. Read red book Chapter 3 and Appendices E–F for additional details.

For Submission

Implement a simple 3D wireframe viewing library using the mathematics that has been covered so far. This library should have the following:

1. Separate model-view and projection matrices, with one of these serving as a "current matrix" on which transform functions operate, similar to how OpenGL specifies the *matrix mode*.
2. Your own implementations of the OpenGL *glOrtho*, *glFrustum*, *glTranslate*, *glRotate*, *glScale*, *gluLookAt*, and *glViewport* functions. These functions should all affect the current matrix.
3. A *drawLine3D(x1, y1, z1, x2, y2, z2)* function that draws the given line segment under the current model-view, projection, and viewport transformations.

Show off your library by providing a demonstration program that displays one or more 3D wireframe objects in a window.

Implementation Platforms

To implement your library, you'll need a 2D drawing environment; choose one of the following:

- A Java implementation using Swing and *paintComponent*(*Graphics*)
- A C implementation using OpenGL *only* for clearing the display, setting colors, and drawing 2D points (i.e., you may only use *glVertex2** for drawing, and you may *not* use the OpenGL versions of the functions that you're implementing)
- A JavaScript implementation using the *canvas* tag within a Web browser

Commit your code to */homework/cmsi371/mygl*. Tag everything as *hw-0405*.

Extra Credit

Implement clipping in your library; to demonstrate your implementation, limit the viewport in your demonstration program so that it is smaller than the containing window.