CMSI 371

COMPUTER GRAPHICS

Spring 2008

Assignment 0410

This assignment seeks to exercise your transformation and matrix skills. And yes, for better or for worse, it also tries to remind you that work on your project should be proceeding concurrently.

Not for Submission

- The material discussed in class to this point corresponds to Chapter 4 in Angel, Sections 4.6–4.11. Appendices B and C are also useful for covering the pure mathematics in detail. Up next: Chapter 5.
- Appendices E and F in the red book also provide additional material, pertaining specifically of course to how OpenGL is implemented.

For Submission

There are three parts to this assignment; they individually should not take too much time, and two weeks should be enough for all of them.

Project Progress Report

Commit a plain text file called *progress.txt* in your */projects/cmsi371* directories. In that file, include dated entries stating what you've done about your graphics project as of that date. The file may have as many entries as you wish, up to April 10. Feel free to also commit this file as frequently as necessary (just like the rest of your work!).

Programming

Write a Java method that performs matrix multiplication for matrices of arbitrary size, as well as a suite of JUnit tests for verifying that method's correctness. Implement the method within a class called `math.Matrices`, with this signature:

```
public static double[][] multiply(
    double[][] m, double[][] n);
```

Assume a [row][column] indexing scheme, and throw an `IllegalArgumentException` if the matrix arguments cannot be multiplied to each other.

Commit your code under */homework/cmsi371/matrices*, with `math.Matrices` residing under *src/main/java* and its test suite residing under *src/test/java* within the *matrices* directory.

Written Exercises

Answer the following questions on hardcopy:

1. Write out the 2D version of the *instance transformation* described in Angel pages 208–209. Use s_x and s_y to represent the scaling parameters, θ to represent the angle of rotation, and t_x and t_y to represent the amount by which to translate.
2. Show, by calculating the resulting matrices, whether or not the following sequences of 3D transformations commute:
   a. Rotation by θ about the x axis and uniform scaling (i.e., a single s scale factor).
   b. Translation by $<t_x, t_y, t_z>$ and rotation by θ about the z axis.
   c. Two rotations about the y axis, by θ_1 and θ_2.
3. Suppose you're writing a 3D chess game, with the chessboard centered on the origin, aligned with the xz-plane, and having squares that are 10 × 10 in size. Chess pieces are drawn with their bases centered on the origin, and the overall scene can be rotated around the y axis for custom viewing. If (f, r) represents a piece's square, with (0, 0) representing the "lower left" square of the board and (7,7) representing the "upper right" of that board, and θ represents the custom viewing angle, derive the transformation that needs to be applied to a piece so that it will be drawn at the correct location in world coordinates. For example, a piece located at square (2, 1) with a viewing angle of 90° should have its origin end up at (25, 0, 15).

Extra Credit

You'll get extra credit if you write up the answers to the written exercises (including intermediate expressions, if necessary) in LaTeX, and commit the *.tex* source file under */homework/cmsi371/matrices* (i.e., on the same level as the *src* directory).