Assignment 0215

The goal of this assignment is to transfer the mathematical concepts that we have discussed so far into practice as actual code. You can also pick up additional skills as well, depending on the option that you select: either (1) you learn how to port Java classes to C++, or (2) you learn how to interact with OpenGL from Java, via JOGL.

Not for Submission

1. Read Chapter 4 in Angel, and if you need more background, consult Appendices B and C.
2. If you haven’t done so yet, visit Nate Robins’s OpenGL tutorial site:
   http://www.xmission.com/~nate/tutors.html — download the tutorial source, build them, and run them. At this point, his tutorials are more connected to the material at hand.

For Submission

Same routine as before — hardcopy submission, with e-mail to assist me in showing you off.

The overall task is to extend our sample Fireworks program to 3D. This task breaks down into the following specifics:

1. First, make an implementation decision — you can do the assigned work in one of two ways. Pick the one that interests you the most, or perhaps the one that is most similar to the environment within which you will do your research project and paper:
   a. Port the entire program to C++, invoking OpenGL as before.
   b. Keep the program in Java, using JOGL as the bridge to OpenGL. Today’s handout and some links on the course Web site should have the information you’ll need.
2. Based on your decision in (1), create a Vector3D library and refactor the code to use this library where applicable. In particular, Spark movement and acceleration should use vectors; you may find a few more spots where vectors can be used as well.
3. Finally, we’re in 3D now, so you’ll also need to adjust your display code to OpenGL (whether in C++ or in Java through JOGL). Note that in JOGL’s implementation does not have a totally complete GLUT bridge (it’s still a work in progress), so just use what’s available.

Extra Credit

Performing the following task in the time allotted will count as an additional homework credit.

4. Port the Spark drawing code completely to use OpenGL transforms. The current code already uses translation; there are alternative ways that use scale and rotation to draw each spark (and its tail).