

# CMSI 371

## COMPUTER GRAPHICS

Spring 2009

### Assignment 0319

Since this assignment has spring break in between, we're a little more ambitious here. The purpose is two-pronged; one part forces some movement in your graphics projects, while the other is a preview of upcoming class material.

#### Not for Submission

As mentioned previously, Sections 4.1–4.2 and Appendix B in the Angel book, as well as Appendix E in the red book, round out the material covered in class so far.

#### For Submission

##### Graphics Project Design and Prototype

Make a final decision on your project, sketch up an initial design, and put together a simple prototype reflecting this design. Specifically:

1. Modify and commit a new version of your README file, so that it is now a detailed description of your chosen graphics project.
2. Submit hardcopy UML use case and structural diagrams of your project. If you used an electronic tool for making these diagrams, commit your files under a *doc/* subdirectory within your directory of the *gallery/* source tree.
3. Commit some kind of prototype code into the *gallery/* source tree. At a *bare* minimum, you should at least have some data structures, functions, and stubs that indicate where your code will go. Of course, do as much work as you like; the point of this assignment is to get you substantially started on the work.

##### OpenGL Reverse Engineering

Remember the teapot midterm question? Well, a form of this code has been committed under *homework/cmsi371/fakegl*. This program replicates the teapot drawing sequence, but instead of drawing teapots on the screen, it just prints the final location and scale of each teapot. Fill out the fake OpenGL functions so that they work like the real thing, and commit your modifications to CVS.

#### Extra Credit

Figure out and implement some form of collision detection *between* balls in the incomplete *battleballs* code that you already have. Submit the following artifacts to get the extra credit:

1. Hardcopy notes (handwritten is fine, LaTeX is better) showing the mathematics behind your collision detection algorithm.
2. The modified *battleballs* code that implements this algorithm, committed to CVS.

Feel free to e-mail me over the break about any of these deliverables, in case you have questions or are somehow stuck. Have fun!