CMSC 23700 **Introduction to Computer Graphics** **Project 0**
Fall 2003 **September 30**

A simple OpenGL viewer

Due: Friday, October 10 at 9pm

1 Summary

For this project, you will have to implement a simple viewer in **C** using OpenGL and the GLUT library. Chapters 1–6 of the "*OpenGL Primer*" cover the aspects of the OpenGL and GLUT that are relevant to this project.

The goals of this project are to get your feet wet with simple graphics programming and to give you some quick feedback on the course submission and grading policies.

2 Description

Your task in this project is to implement an interactive viewer of a simple 3D animation. Initially, the viewer is located at $(0, 0, -4)$ (this is called the *camera* or *eye* position) and is looking towards the positive Z-axis and the animation is located at the origin. Your program is responsible for displaying the animation and for allowing the user to vary the camera position using the keyboard.

2.1 The animation

Your viewer displays an animated spiked ball, which is constructed from a sphere and six cones (use the functions `glutSolidSphere` and `glutSolidCone` from the GLUT library). In the ball's coordinate system, the center of the ball is at the origin and its radius is 0.5. There are six spikes aligned with the positive and negative axis (we call these X^+, X^-, Y^+, Y^-, Z^+, and Z^-). The spikes are cones with a base diameter of 0.5 and a height h_d (where d is the direction of the spike). The ball is animated in two ways: it rotates and the sizes of the spikes oscillate. The ball rotates once per minute in a counterclockwise direction around the negative Y-axis (*i.e.*, looking down on the $X - Z$ plane); we use θ to denote the angle of rotation in degrees. We define these motions as a function of time t (measured in seconds) as follows:[1]

$$\begin{aligned}
\theta &= 6t \pmod{360} \quad \text{(in degrees)}\\
h_{X^+} &= 0.6 + 0.4\sin(2\pi(\frac{t}{5} + 1))\\
h_{X^-} &= 0.6 + 0.4\cos(2\pi(\frac{t}{7} + 2))\\
h_{Y^+} &= 0.6 + 0.4\sin(2\pi(\frac{t}{7} + 4))\\
h_{Y^-} &= 0.6 + 0.4\cos(2\pi(\frac{t}{3} + 1))\\
h_{Z^+} &= 0.6 + 0.4\sin(2\pi(\frac{t}{3} + 2))\\
h_{Z^-} &= 0.6 + 0.4\cos(2\pi(\frac{t}{5} + 3))
\end{aligned}$$

For example, at time $t = 2$, we have the following values:

$$\begin{aligned}
\theta &= \text{12 degrees}\\
h_{X^+} &= 0.6 + 0.4\sin(2\pi(\frac{2}{5} + 1))\\
&\approx 0.8351\\
h_{X^-} &= 0.6 + 0.4\cos(2\pi(\frac{2}{7} + 2))\\
&\approx 0.5110\\
h_{Y^+} &= 0.6 + 0.4\sin(2\pi(\frac{2}{7} + 4))\\
&\approx 0.9900\\
h_{Y^-} &= 0.6 + 0.4\cos(2\pi(\frac{2}{3} + 1))\\
&= 0.4000\\
h_{Z^+} &= 0.6 + 0.4\sin(2\pi(\frac{2}{3} + 2))\\
&\approx 0.2536\\
h_{Z^-} &= 0.6 + 0.4\cos(2\pi(\frac{2}{5} + 3))\\
&\approx 0.2764
\end{aligned}$$

In your animation, the sphere and each spike should be given a different color (you choose).

You should experiment with different lighting settings to understand how they affect the displayed image. For your submitted version use the combination of ambient light and a directional light shining from the viewer's upper left and use smooth shading.

2.2 User interface

Your viewer should support the following keyboard commands:

space	toggle the animation on/off
`+`	move the camera 0.05 units toward the "look-at" point.
`-`	move the camera 0.05 units away from the "look-at" point.
`q`	quit the viewer

[1] Use the `gettimeofday` system call to get the value of t.

The camera should not get closer than $z = -2$ (note that the spike can grow to a height of 1 from the center of the sphere). Your application should also handle resizing the viewport.

2.3 Further work

For extra credit, try adding more features to your viewer. Possibilities include wireframe mode, changable lighting models, and more camera movements.

3 Submission

We will set up a CVS repository for each student on the Computer Science server. This repository will be seeded with CVS modules for each of the projects. For this project, the initial module is named `project-0` and contains a makefile. You should use this repository to hold the source for your project. We will collect the projects at 9pm on Friday October 10th from the repositories, so make sure that you have committed your final version before then.

You will also be expected to "demo" your code during your Lab session in Week 2. While your project may not be finished by then, we hope that you will have something to show.