CSE168: Rendering Algorithms
Assignment 0
Due Tuesday April 13 at 6:00pm

Toshiya Hachisuka

This assignment will familiarize you with the provided code base and prepare you for the remaining assignments.

Miro Overview

Miro is the base code we will be using for the class. Miro provides an OpenGL view of a scene with basic functionality to position the camera and raytrace an image. You can move the camera around in the scene using the A, S, D, W, Q, and Z keys. Dragging the mouse allows you to rotate the camera. Pressing the 'r' key raytraces the current view. Pressing 'g' switches back into OpenGL mode. You can save a snapshot of the current view by pressing the 'i' key. This generates a .ppm file in your current working directory called miro_xxxx.ppm where xxxx is the time the render was taken. You should use this feature to create your submitted images.

Task 1: Setup Your Development Environment: 10%

If you are using Linux or Mac OS X, make sure you are able to compile and run the code using the provided Makefiles. In order to compile the code on Mac OS X, you need to change one line in the Makedefs file. Open up the file in a text editor and modify the indicated line.

If you are using Windows, create and setup a Visual C++ project for the provided base code. You may want to check out http://csf11.acs.uwosh.edu/cs371/visualstudio/ for step-by-step instructions on how to create a Visual C++ project with GLUT.

After you are confident your project is working properly, go through the base code to get an understanding of program flow. This will be the code

base you will be using for the remainder of the quarter so become familiar with it.

Task 2: Render the Spiral Scene: 30%

In main.cpp there is a `makeSpiralScene` function. Render an image using this scene. The provided function only uses one color for all spheres. Modify the colors in the scene to make for an interesting image. Be creative. As an example, your rendered image could look something like the first image in Figure 1:

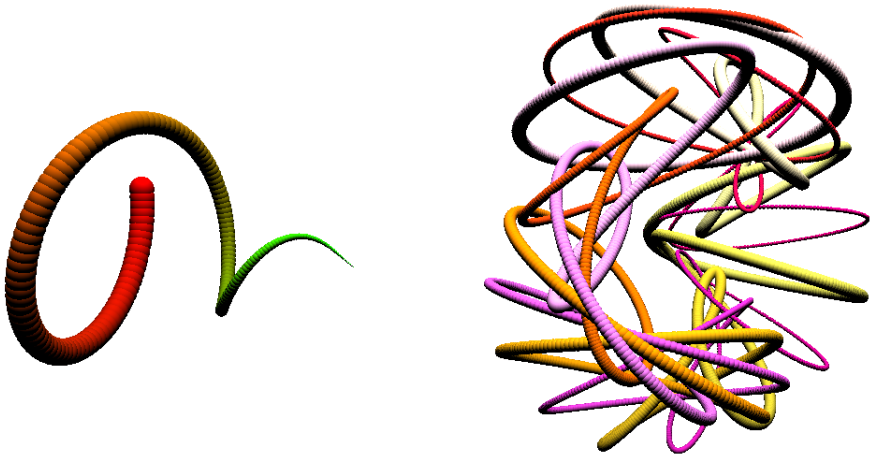

Figure 1: Example results for Tasks 2 and 3.

Task 3: Create and Render a Spirograph 50%

Use the provided spiral scene as reference, create a new function, `makeSpirographScene`, in `main.cpp` which creates a Spirograph using spheres with changing colors.
The equations of Spirograph are:

$$
\begin{aligned}
cx(A, B, t) &= A * \cos(B * t) \\
cy(A, B, t) &= A * \sin(B * t) \\
x &= cx(AX, BX, t) + cy(AZ, BZ, t) \\
y &= cy(AX, BX, t) + cx(AY, BY, t) \\
z &= cx(AZ, BZ, t) + cy(AY, BY, t),
\end{aligned}
$$

where AX, BX, AY, BY, AZ, and BZ are constants that define the shape of Spirograph and t is the parameter in $[0, 2\pi]$. For more information about this function, check out
http://maxwelldemon.com/2010/01/14/spirographs-and-the-third-dimension/.

Play with the constants and color scheme to come up with an interesting image. Your final image could look something like the second image in Figure 1:

Submitting Your Work 10%

You need to show your work to the TA by the due day during lab hours. We encourage you to submit (show) your work earlier than the due day. If you wait until the due day, you will be facing a long waiting line for grading - it is not recommended.

You should also submit your rendered images from Task 2 and 3 above at 512^2 resolution. Convert all your images to PNG format before submission (do not send them as .ppm files), and package up your images into a zip file. If your archive is larger than 500KB, you might be doing something wrong.

Your archive and images should be named according to your last and first name. For instance, John Smith should submit `smith_john.zip` which contains `smith_john1.png` and `smith_john2.png`. Include the text "CSE168 Assignment 0" in the subject of your email.

Email your archive to cse168-turnin@graphics.ucsd.edu.