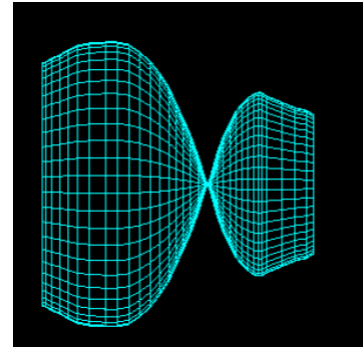


Graphics Programming Project 3 Report

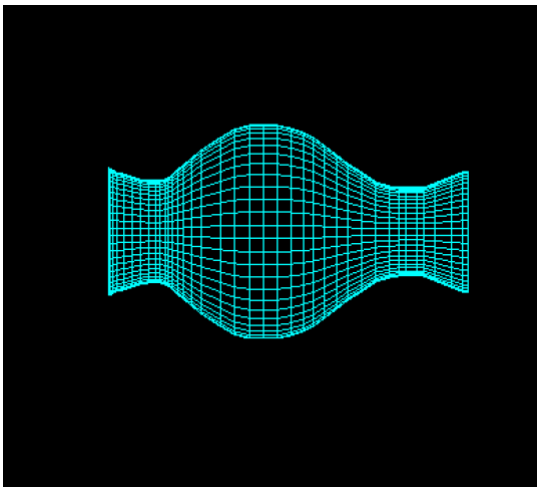
Luke Miller
010785453

Problem Statement:

The goal of the programming assignment is to create a formula-based object of rotation, display said polygon surface using OpenGL, and finally create a hand drawn object of rotation. This is to give us a first hands on with 3D objects in OpenGL. The inputs of the program are the user draws a line with their mouse and can rotate the image on the x y and z axis. There are few places of error handling that keep the user from being able to draw when drawing outside of the window. For example, if the user draws off the window the lines will continue just outside the view of the user.



Design:



When I designed my program, I first decided to use the project 'objects3.cpp' as a baseline. Then I added the callback functions I needed for the keyboard input, mouse input, and motion input. Then I decided what global variables I'd need to be able to manipulate and redraw new shapes, so I have one global vector: to hold what the user is drawing. I also have global constants to hold a few flags and counters and the properties of the window. Then I have two global Surfaces that hold the original shape, and the new one to be drawn by the user. I then have some supporting functions for taking the surface of

the drawn object and rotate it around the x-axis. There are a handful of functions taken from the 'object3.cpp' file but most are not in use at the moment.

Implementation:

To get started I looked for widgets that would be useful to the project in the class's source folder. Then I copied the some of the default code from one of the 'objects3.cpp' files for the callbacks and adapted them for the shortcuts I needed. After that I decided to pretty much get at it with how I draw and display the input trying to just get it to redraw a vector data set so I knew it would work when converting the drawn line into a Surface object. The hardest part was trying to figure out the correct boundaries for segmenting up the drawn line. I ended up not being able to figure out the correct ones because sometimes my line of rotation stops short and starts too tall.

Testing:

After each part was programmed, I checked to make sure it was drawing the surfaces correctly. The drawing function is where I ran into most of my problems for it

was drawing the drawn surface wrong and making the last segment of the surface be $(0,0,0)$. After I the bug was not where I had originally thought it was, I found it to be an error in one of my for loops drawing the line in the center. This program is relatively simple but there were some special cases to test for like trying to draw really long lines or crossing the x-axis while drawing. I tried throwing every weird instance at the program and it seems to handle it well with only two bugs to report. There is a final rotation around the x-axis that doesn't happen while drawing that makes it look like the surface isn't complete, and sometimes when drawing longer lines the line doesn't always draw long enough.

Conclusions:

In conclusion, this program was mostly a success. Everything works almost as expected. I am proud of the program and I believe this was a great first step for working with 3D objects in the class. Knowing what I know now I wouldn't have wasted time thinking there weren't areas of code without bugs, because that's were my main bug was. If it weren't for that the assignment would have taken me less than 3.5 hours. Three and a half hours with debugging isn't wholly that bad and I am proud of how it turned out.

