

Aaron Templeton

0734119

HW 3 – 3D Graphics

CS 4600

Calculating the Normals:

I understood that the Indices in the program were the polygons and that I need to calculate the points to get the vertices. I used the points to index into the vertices array and calculate the normal. I also understood from lectures that the cross product was going to have to be used to calculate the normals and that all of the lengths will need to be normalized. After several attempts and moving around some bits of code, I had a teapot that was close to having the correct normals, but, it was too dark around the edges.



After Wednesday's lecture I realized that I hadn't normalized the values twice, as our professor suggested. After normalizing again, I had the correct shading.



I also did a lot of online research to better understand how to calculate the normal of the vertices.

Here are the sites I read:

<https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/shading-normals>

<http://www.opengl-tutorial.org/beginners-tutorials/tutorial-8-basic-shading/>

they helped me gain a better understanding of the normal and the basics of shading a 3d object.

Spinning the Teapot:

I knew that using the time with the trigonometric functions in the matrix would allow it to rotate. I used the rotation matrices and the correct trig functions to get it to rotate. After playing with the values in the matrix, I figured I only needed to use the following matrix indexes: 0,2,5,8,10,14,15. These same indexes are used in the checkerboard matrix as well, which I saw while looking over the assignment.

Dolly Zoom:

The dolly zoom is the most confusing part of the assignment to me. I did so much online research but there was not anything that was extremely helpful. The most helpful resource was:

http://www.mathematik.uni-marburg.de/~thormae/lectures/graphics1/graphics_6_1_eng_web.html#1

I knew that the FOV and distance had to change together in opposite directions to mimic the vertigo effect, but I couldn't figure out how to change it over time. I asked a classmate who recommended that I think about it using trig functions. After thinking for minute, I decided to try and use the sine function with the time and calculate the new FOV and distance. It works, but its not perfect. It sometimes leaves the FOV, but I can't figure out how to fix that part.

Orthographic projection:

I found a good online resource for the orthographic projection:

<http://www.learnopengles.com/tag/orthographic-projection/> it helped me understand what I needed to put into the glOrtho function. I played with the numbers a bit until I got it to work.

Youtube video:

<https://youtu.be/-SIZtuISRJE>