Week 6 Report (7/14/16)

Getting Started With JOGL

Java OpenGL is very similar to regular OpenGL Many of the functions are named the same and do what they are needed. Setting up a build environment required downloading all the needed jar files and also its respective natives file. The OpenGL profiles initially presented a problem because my computer was only compatible with GL4 and GL2 profiles initially.

Problems Encountered

I was not able to fully translate the sample C program over. Many of the functions in OpenGL C worked differently than they do in JOGL. The most troubled function was *glTexImage2D* which generates the texture image. This function required a buffer which works differently in Java than in C. In C, a 3D array is automatically converted to 1D but for Java we needed to start with a 1D array, represent it as a 3D array and passing it back in the function argument as a 1D buffer. This shouldn’t be too hard to implement with the example online.

Conclusion

My biggest priority is adding my original Java code to JOGL This will be done by embedding JOGL into a JPanel. Once I can successfully do that, I can start to use JOGL’s drawing capabilities to correct represent the floating point scale for my grid. I want to get domain normalization correct because it would then correct my code’s problem with multiple elements interacting dysfunctional. OpenGL takes care of drawing differently that Java’s graphic library in that it takes care of the pixel point of where to draw. So the benefits is that you don’t have to worry about pixel location while implementation.