I really enjoyed working on this project. Unfortunately, we were unable to get a working program for the presentation, however the overall code is syntactically correct and has been a great learning experience my group and I. Due to some hardware issues with some of my group however, a lot of the coding was handled by a couple of members of the group and greatly affected the delivery of a completely working piece of code. Overall, I think I personally learned a lot about one of my favorite coding languages, not only that you can recreate tools that you need on the fly if that specific language doesn’t completely support it, such as the Vector3d class that was implemented in the VectorPy.py file. Most of the materials that I used to update my knowledge of python were inspired by members of stack overflow who pointed out different tips and tricks of how to implement some of the same functions you’d find in C++ and translate them into python. Python Crash Course was also a very good reference for me as it had a lot of easy to read examples about syntax and class creation. Creation of the VectorPy file itself was a learning experience as well, the definition of the init functions for every class was surprisingly easy to implement in Python, if only just a little peculiar from what I’ve learned in C++. These peculiarities come mostly in part from the init function, but also are due to the specific spacing needed when creating classes, functions, if and elif statements and of course variables themselves. I didn’t have much trouble with the classes bar some syntactic issues, but the main issue was testing when only two members of our group could successfully compile the code. Another issue our group was having at first was communication and sharing of written code. We eventually all downloaded a collaborative application called “Flock”, a free application which allows the user to join chat channels and post information as files or as just small snippets of code or text and other features that assist with teams of people working on documents together. Once we had that in place to share our work, we actually got a few things done. I didn’t realize that OpenGL was the same no matter what the language, however I did run into an issue where some of the GLUT keywords such as color, I wasn’t able to use floating point values and instead had to substitute in values from 0 – 255. The next issue I had was using a Node class instead of actual nodes that are available natively in the C++ language. Once we had a class in place that was situated in the TargetPy.py file, I was able to begin translating the pointers. This is where I believe I ran into the biggest issue. Once the pointers were implemented, I began testing which showed that the MainPy.py file was compiling correctly with no errors. Once I added the GLUT operations in the main file and then running the file from my terminal, every time the program ran it would break and prompt apple to show me an error that python had quit unexpectedly. I’m not exactly sure what is causing this error, as the code ran and compiled fine even with the multiple files that were being referenced to and from each other, as I figured that would be the main issue when integrating all of these files together. This turned out to not be the case, and as such I am still trying to get this code to work regardless. After some more research I found that the error is a “segmentation fault” error, which occurs sometimes when a daemon thread continues running without being checked. I do not know how to effectively debug this error however, as much of the documentation of it online is fragmented. Altogether, I really enjoyed working on this project, and I hope to continue doing more interesting projects in this class as the semester ends.