Ryan Miller

CS 499 Milestone Three Narrative

This artifact is an OpenGL program that renders a scene of 3D objects, originally created for the CS 330 Computational Graphics and Visualization course. Prior to enhancement this artifact contained functions that literally defined the position of the points and triangles needed to draw cylinders. This involved specifying the coordinate locations for each and every point in the 3D shape line by line. This is both an inefficient time-consuming process. Additionally, if you wanted to upscale the polygonal resolution of these shapes, you would have to do so manually by redefining and adding all necessary points. At large scales this could take hours, so the appropriate solution to enhance this artifact was to define an algorithms that would dynamically generate the shape, so that by changing a couple of values at the top of the function, the shape could easily be upscaled.

I decided to include this artifact in my ePortfolio under the Algorithms and Data Structures category because it demonstrates my ability to apply algorithmic solutions and manage data structures in a practical context, improving the overall quality and performance of the program. The enhancement resulted in better looking and hi poly shapes that helped better communicate the representation of each object in the scene. In order to complete the enhancement, I had to develop algorithms for dynamic shape generation, efficiently manage and store the generated vertices, and demonstrate an understanding of more advanced OpenGL concepts. I conducted a code review in which I documented my plan to enhance the original shape generation, I also ensured that the new artifact is well commented.

The process of enhancing and modifying the artifact was interesting, mostly because I haven’t worked with OpenGL at all since completing the original artifact for the CS 330 course. I had to relearn several concepts and had some trouble initially with getting the right dependencies properly pathed as I had lost the original dependency files. One of the biggest challenges I faced came as a direct result of having little to no OpenGL practice in the last year and was also largely due to poor commenting on the original source code. This issue involved not being able to see the properly generated shapes after writing the generation algorithms. The problem was related to a separate part of the program where the shapes are actually drawn after the points are defined. There are several different drawing methods, and I apparently was still trying to use the method that is specific to literally defined points (draw by arrays). Once I switched to the appropriate method (draw by elements), the shapes generated perfectly. This project provided excellent practice in algorithm design, data structure management, and graphics programming.