Artifact One Narrative:

The artifact one was used for both the Software design and engineering and algorithms and data structures. This Artifact was created during the CS330 Computational Graphics and Visualization course. The Original project is a program that takes vertex points and generates objects in a scene. This project is the beginning of what can be grown into a game engine for rending 3D objects and environment.

For the Category of Software Design and Engineering, this project originally had all the functions and information in a single file. Though this is not necessarily an issue for some smaller programs, this program is meant to grow so that it can in the future house all the functions necessary to generate a larger environment. By doing the design and the layout of the program it allows for some of the processes that would later need to be removed, to already be removed. This also allows for the main source file to be cleaned up. This allows there to be less code in the main file, making it easier to read through during the review process. Though this would mean there is more files to review. They will be smaller in size and easier to follow through with out getting lost or spending an excessive amount of time scrolling through a program looking for a specific function.

For the Category of Algorithms and Data Structures, this artifact had a lot of basic Algorithms and Data structures. Cleaning up some of the already designed in structures and creating a separate structure to house the shader information allowed for the main file to be cleaned up. The enhancements for this category come from the cleaning of the main mesh structures so that they functioned easier and for creation the Shader and Texture programs that took information form the basic shader text file and from texture images and setting them to data structures that were retained in the program and able to be called and placed on the objects in the scene. In the original version of the program the shader information was directly stored in the main function. In the updated version the shader is kept separately and called via the shader program and parsed into a Vertex object and a Fragment object. These objects are then used to communicate with the GPU so that it knows how to render the scene. By having the shader kept outside of the program as a whole and only parsing it in when ran allows for the shader text file to be changed out if it gets corrupted or needs to be updated.

Through out the enhancement of the project there were many issues and some changes that had to be reverted to the original because of conflicts with the code. Originally, I planned to pull all the basic functions for creating the virtual arrays and their buffers from the main class file, but after some conversion issues these functions had to get reverted back to their original. As each enhancement was started, I ended up having to do a bit of research over time I found there are many ways to complete the processes and not all of them work well with each other.