CS 499 Enhancement Two Narrative Algorithm and Data Structures

Southern New Hampshire University

Trey Patten

**Briefly describe the artifact. What is it? When was it created?**

The artifact that was chosen is from CS-330 Comp Graphic and Visualization. The object of the course was to take a 2D image (Fig 1) and make a 3D representation of it. The artifact was supposed to be the final project that was created using Microsoft Visual Studio (MVS). The language that I used to during this course was C++. The artifact that I am working with was created in April 2023.

**Justify the inclusion of the artifact in your ePortfolio. Why did you select this item?**

The reason why I chose this artifact is because I did not finish the final assignment completely. After the course I realized that OpenGL is widely used in industries such as gaming, computer-aided design, scientific visualization, and simulations. Working on an OpenGL project allowed me to gain practical experience. I currently work for a DoD program, and they have an office of developers who design visual objects for the DoD. I would like to hopefully seek a job opportunity with that department to showcase what I can bring to the table.

**What specific components of the artifact showcase your skills and abilities in algorithms and data structure?**

For this Enhancement, I wanted to showcase that the project combines algorithms and data structures to manipulate the 3D objects, apply lighting models, and efficiently manage resources within the OpenGL graphics pipeline. These components showcase the use of algorithms and data structures in the context of computer graphics and real-time rendering. The following components of the artifact were updated to show that the code

**Shader Logic**: The fragment shader (surfaceFragmentShaderSource) includes algorithms for lighting calculations, including ambient, diffuse, and specular lighting. These calculations involve vector normalization, dot products, and mathematical operations, illustrating algorithmic approaches to lighting computations.

**OpenGL Object Handling:** Throughout the code, there are various OpenGL objects, such as vertex array objects (VAOs), vertex buffer objects (VBOs), and textures, are created, bound, and managed. These objects are essential for efficient rendering and represent data structures within the OpenGL context.

A screenshot of a computer program

Description automatically generated

**Math Libraries**: The use of math libraries like GLM for vector and matrix operations is itself an example of leveraging existing algorithms and data structures to simplify complex mathematical operations common in computer graphics.

These and other additional components highlight some of the algorithms and data structures in various facets of programming, from data management to rendering optimization and error handling.

**How was the artifact improved?**

The artifact improvement for this category was to align it with Course Outcome #3: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices (data structures and algorithms).

* A camera function named UProcessInput was added to allow input from a user via keyboard keys using the GLFW library**.** The primary data structure used in this code is the gCamera object, which is an instance of the Camera class. This camera object holds information about the camera's position, orientation, and movement parameters.
* The algorithm for processing input involves the following steps:

1. "Escape" key (GLFW\_KEY\_ESCAPE) to close the GLFW window.
2. "W", "S", "A", "D" keys (GLFW\_KEY\_W, GLFW\_KEY\_S, GLFW\_KEY\_A, GLFW\_KEY\_D) for camera movement in various directions.
3. "E" and "Q" keys (GLFW\_KEY\_E, GLFW\_KEY\_Q) for moving the camera up and down.
4. "P" key (GLFW\_KEY\_P) to reset the camera to a predefined "home" position.
5. "O" key (GLFW\_KEY\_O) to set the camera to a different predefined position and orientation.

**A screen shot of a computer program

Description automatically generated**

**Did you meet the course objectives you planned to meet with this enhancement in Module One?**

For this category, I planned that the data structure for the images was going to increase. But after working on the camera functionality, I noticed that camera actually involved more algorithms than I thought. I thought that, algorithms were a procedure for solving a mathematical problem or to solve certain operations. Taking Input from a user via keyboard, and then running it through a series of commands and libraries to get a result, is an algorithm. So, no I didn’t me the enhancement I originally planned for, I believe I got one better…. After the third try from the Prof of course.

**Do you have any updates to your outcome-coverage plans?**

None at this time, I need to apply what the Prof provides in the feedback moving forward.

**Reflect on the process of enhancing and/or modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?**

I continue to gain a deeper understanding of OpenGL and computer graphics, highlighting the importance of trial and error and learning from resources such as documentation and online forums. What I found challenging during this enhancement was that implementing algorithms and data structures correctly can be tricky. Small errors in code can lead to incorrect results or unexpected behavior, making debugging a crucial skill.