Gagujas, Michael

CS499 – Computer Science Capstone

July 28, 2024

Milestone Three: Algorithms and Data Structures

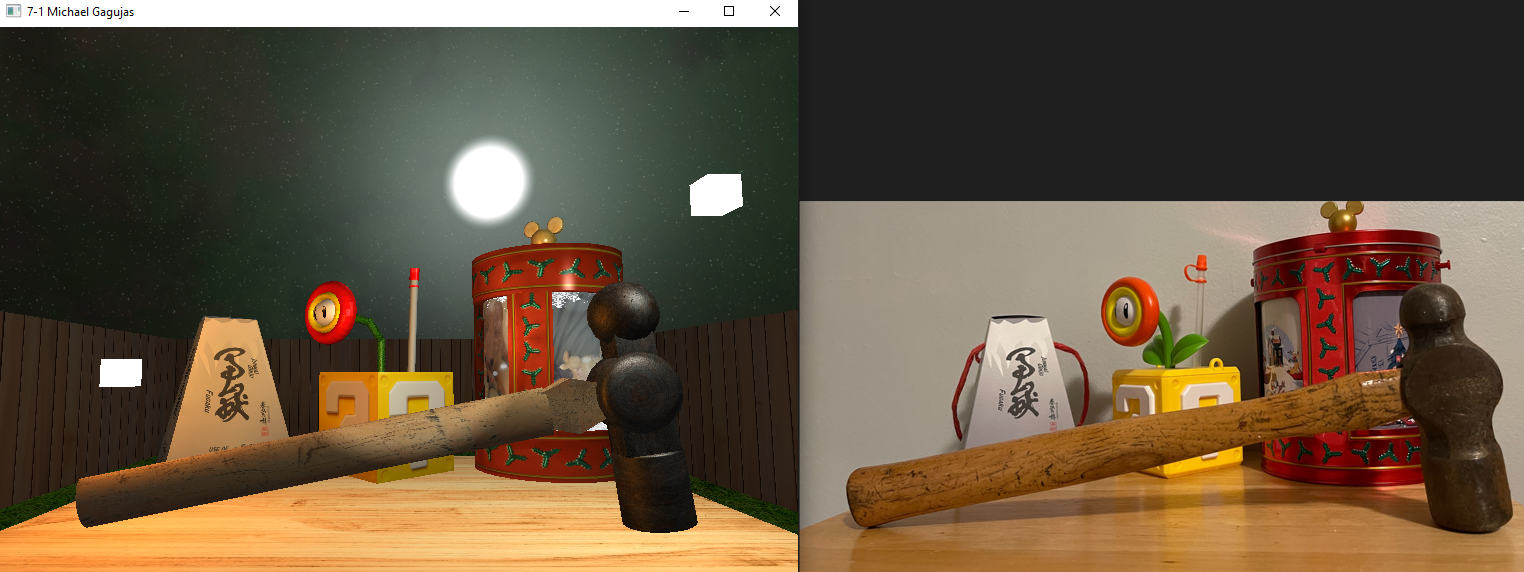
***Algorithms and Data Structures***

Enhancement 1 – Implement Discrete Level of Detail

Enhancement 2 – Improve Mesh Transformation Process

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

The artifact that I’ve selected to demonstrate my understanding of the algorithms and data structures category is my OpenGL Project from CS330 Computational Graphics and Visualizations.

 This artifact involves recreating a 2D image as a 3D scene, and it effectively applies the use of textures and the Phong lighting model to create a realistic and visually appealing representation of a photo. It was completed in March 2024 during the 24EW3 term as my final project, but I’ve recently updated it by adding software engineering and design enhancements like modularizing the code into reusable light classes that demonstrate my capability to implement polymorphism and encapsulation in parent and child classes. It also saw enhancements with utilizing configuration files to better manage magic numbers and adding complexity to the project by adding moving objects in the scene. In general, this project demonstrates my proficiency in generating accurate representations of three-dimensional objects and creating interactive graphics applications that respond to input devices.

1. Justify the inclusion of the artifact in your ePortfolio. Why did you select this item? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?

I’ve included my OpenGL project as an artifact in my ePortfolio because it has a strong potential to exemplify a majority of the five course outcomes, demonstrates my ability to apply theoretical concepts in a practical setting, and contains topics that I’ve been wanting to experiment with. I selected this project because I enjoyed creating it, but also because I have a lot of ideas and concepts that I want to apply to it to make it better. The specific components of the artifact that showcase my skills and abilities in software development are the meshes class for applying algorithms to build and manipulate 3D shapes that represent a real-world object, the texture class for handling image files and demonstrating understanding of textures coordinates, and the shader class for demonstrating my ability to understand the graphics pipeline and how to manipulate it.

Overall, the project was improved in multiple ways that aligned with the algorithms and data structures category. My first enhancement involved implementing discrete level of detail where in which depending on the cameras distance from an object, it uses either a low polygon mesh for further objects or a high polygon mesh for close objects where higher detail is important. Discrete level of detail uses algorithms involving camera distance and making different versions of meshes to increase performance. By using lower polygon models for further objects, I can significantly reduce the number of polygons that the GPU needs to process, which leads to improved rendering and higher frame rates. As can be seen from the video, I have a very basic implementation at the moment where I only convert the sphere mesh to change models based on its distance from the camera. To emphasize the Discrete Level of Detial in the video, I used a more noticeable polygon difference as can be seen when the camera moves back and forth within the threshold range that I set when selecting an appropriate polygon version. I had issues implementing a more efficient method regarding applying arrays inside the struct of my GLMesh, which I will further elaborate on in question four.

For my second algorithms and data structures enhancement, I made the MeshCreator class more efficient by setting repeated arguments as private data members that are initialized once and then reused throughout the class. This made my code more readable and maintainable while also improving its performance since it added unnecessary overhead when repeatedly initializing arguments. This enhancement also enhanced the encapsulation of the MeshCreator class as the private data members made it so that only the methods of the class could access and modify them, which protected the integrity of the data and made the class easier to use correctly. Overall, I have a functional implementation of Discrete Level of Detail that I am currently working on with regards to optimizing it and applying it to all shape models.

1. Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

I feel I did meet the course objectives as I was able to successfully implement discrete level of detail, but I also feel that I met too many pitfalls and should have been able to complete more. Luckily, I was able to construct a working method and can now dedicate time towards optimizing it and applying it to my entire project. Once discrete level of detail is applied to the entire scene with entireties of objects, it will be an innovative skill and technique that accomplishes goals like improving performance and visuals that strongly contribute to the overall realism of a project. In addition, this milestone helped me meet objectives like solving a given problem using algorithmic principles and computer science practices appropriate to its solution by using the matrix translation to calculate the proximity of an object with the coordinates of the camera to determine which level of detail model should be rendered.

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

Although I didn’t implement hash maps and linked lists like I initially planned, I learned more about data structures and algorithms by thinking through various solutions and weighing their benefits and disadvantages towards a project’s specific needs. After doing more research, I ultimately decided on using arrays for handling the different polygon models because they are contiguous in memory and better for frequent and quick access. A linked list would be better if I had frequent insertions and deletions for the models with different levels of details, but Discrete Level of Detail (DLOD) commonly creates all of the meshes at once and then only needs to access them, which is a design more suited towards arrays. In general, accessing elements in a linked list is much slower than an array since the worst-case scenario requires traversing through the entire list to find an element while arrays can easily access an element in constant time if they know the index of each respective high or low polygon version of a mesh.

With regards to challenges I experienced, my initial approach with modifying the GLMesh struct in MeshCreator.h to include an array consisting of a low polygon and high polygon version of a mesh had numerous refactors that caused other areas of code to break. I, ultimately want my code to be as DRY (Do Not Repeat Yourself) as possible, so I’m currently working on conducting further debugging to determine where the mistake or correction is needed. Overall, I have a working implementation of Discrete Level of Detail that can be applied to the entire project, but I now want to focus on making it more efficient and in line with coding best practices.