CS499

Sheel Patel

September 22, 2024

Algorithms and Data Structures Narrative

The artifact I’ve chosen for my Algorithms and Data Structures category is my 3D rendering of a 2D scene. The goal of the project was to select a 2D scene and recreate it in 3D using OpenGL. The scene I chose is from the Super Nintendo game Yoshi’s Island. This was the final project for the CS-330: Computational Graphics and Visualization course. I took this course during the summer of 2024 at Southern New Hampshire University.

I chose this artifact because it was one of the projects I most enjoyed during my time in the Computer Science program. I also chose it because I work on GPUs so I found this project especially interesting. Although I took this course recently, I have still though of way I can improve on the original code already. For my enhancement, I have chosen to convert some of the previous code I had written for creating objects into separate functions that can be used to repeatedly create objects. I’ve also added additional objects and created functions for adding them into the scene. As you’ll see in the original scene vs the enhanced scene, there are many objects in the scene which are repetitive. As such it is ideal to create a function which can be used to create those objects. Understanding how and when to do is very important to my development as a software developer. Creating functions for repetitive code means our code will be cleaner, more efficient, and take less time to develop or adjust.

**Source Scene:**



**Original Scene:**

A cartoon of a green and brown object

Description automatically generated

**Enhanced Scene:**

A video game screen with a brick wall and green objects

Description automatically generated with medium confidence

**Enhanced Code:**

A computer screen shot of a program code

Description automatically generated

A screenshot of a computer program

Description automatically generated

In comparing the examples above its obvious that a significant number of additions were added to the scene to bring it to be closer to the scene it is attempting to recreate. In the enhanced scene there are several small objects that it would be tedious to recreate and position constantly. It also makes it difficult to read through the code and identify which object you are looking at. Therefore, it is extremely beneficial to try to condense as much of the code as possible by turning any code that is repetitive into a function which can be used to create the objects. In the coding example provided of the enhanced code, the code for creating the platforms in the scene has been turned into its own function which is passed an x, y, and z location to determine the position of the platform. The second code example shows how that function, among others, are called in the scene rendering function. The cleaned-up code allows for changes to platform textures, sizing, and other values to be changed in the function to adjust all the platforms. This leads to a more streamlined code that is easier to read and work in.

The changes I have made apply to a couple of the course outcomes. My use of a function to reduce repetitive code and make it more streamlined and easier to read achieves the developing of a solution that solves a problem by using computer science practices. Additionally, by implementing these functions in a variety of way such as the coins taking in how many columns and rows of coins there should be or the tree creation taking in the zrotation to be able to angle the trees on the hills. This demonstrates my ability to use well-founded and innovative techniques to implement computer solutions.

As I went through the process of enhancing my scene render to more of the scene by using functions to reduce repetitive code, I faced some challenges and learned some lessons. I learned how important it is to reduce repetitive code as I tried to look through my old code. The old code had so many lines that it made it difficult to keep track of what I was looking at and working on. I also had some difficulty with creating the function for creating coins as I needed to leave every other space empty. As I worked on determining what functions should be created and how, my main consideration was what objects were used multiple times and how to easily keep complex shapes together. With this in mind, I began to create my various functions. Getting the coin creation correct took a little bit of trial and error, but it was far easier than creating each individual coin on its own. I also had to make different functions for the different types of mountains in the scene because angling the trees was a bit tricky. Overall, creating the different functions and fleshing out the scene was enjoyable as it cleaned up the original code, made it easier to read, and further in line with best practices.