The concept of 4D within OpenGL

J Y HOOD
10/09/24

The following you're are about to read is the project I wish to make to improve my skills in the coding field after a long break. I want to get back into the swing of coding, I found redoing the basics to be pretty boring and wished to challenge myself just enough to get said confidence to possibly start a new project involving game development somehow, it should be meantied while I do plan to use OpenGL https://en.wikipedia.org/wiki/OpenGL

I hope to return to the game engine of unity and perhaps do the odd thing with GL in the future. Iv'e already started the project however after using Github https://en.wikipedia.org/wiki/GitHub Trying to switch between my main PC and my old laptop
(where all the libraries for GL were created)

Cloning the project back to my PC broke all the current files which is why I'm taking the opportunity to write this document explaining the process, perhaps thinking of it more like a diary of sorts.

Currently the project sits at I would roughly 12 to 15% Now despite the implication the title the project is only 4D *in concept* the application will not run an actual 4D effect rather an 3D model of a tesseract https://en.wikipedia.org/wiki/Tesseract

I'll proceed this doc with the code explanation and while I use ChatGPT and other resources to confirm things, I'll be still writing this in my own words! This is an experiment within itself so I can remember more concepts and context. I'm so sick of not being able to remember these things so I'm taking the last route I can think of. This may take up more time but if done right I'll always have a resource to fall back on.

Now the project is simple

A window pop up that the user can control left and right (the x axis) the user will be shown

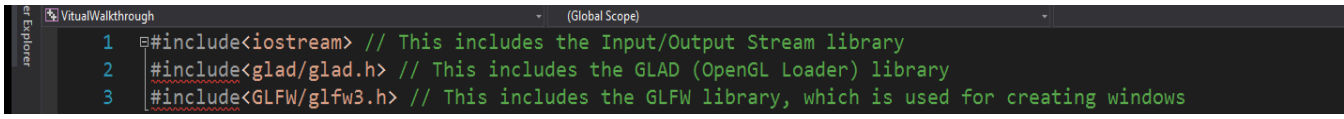
A 2D plane

A 3D cube

A 4D tesseract (as mentioned above)

Like a virtual library it shows an example of 2D to 4D but also OpenGL at it's best.

1. The libraries .cpp



I start with the use of *iostream* this is really used in pretty much every c++ console program as it allows the basics such as input and output to the main console.

“glad/glad” is OpenGL main library of source (none of the syntax would work without it)
Styax https://en.wikipedia.org/wiki/Syntax_(programming_languages)

“GLFW/glfw3.h” is a version of the GLFW library, this is pretty including everything with the application windows (not to be confused with OpenGL which is the graphics library)

The main function is written “int main” which of course will hold everything needed the context of the project

“glfwInit()” is declared as a function, think of it as an on button for my tools within GLFW I’m about to use (aka anything related to the graphics)

Next you’ll see three different types of “glfwWindowHint” these all have slight differences
Major
Minor
And Core Profile