**Project 2 - Transformations**

|  |  |
| --- | --- |
| **Name:** | Mukunth Balaramachandran Srinivasan |
| **uNID:** | u1467270 |

**What’s implemented?**

*All requirements implemented.* Vertices of an object are displayed with functionalities to rotate, zoom, and re-compile shaders. The object is also centered on the viewport by finding the centers of its bounding box along the x and y axes.

|  |  |
| --- | --- |
|  |  |
| Fig 1. Rotate the object using the left mouse button (click and drag) | |

|  |  |
| --- | --- |
|  |  |
| Fig 2. Zoom the object using the right mouse button (click and drag) | |

|  |  |
| --- | --- |
| A screen shot of a teapot  Description automatically generated | A teapot with dots  Description automatically generated |
| Fig 3. Re-compile shaders during runtime by pressing F6 (change in the fragment shader) | |

**What could not be implemented?**

*The optional requirement:*

* *Pressing the P key switches between perspective and orthogonal transformation.*
* *When using orthogonal transformation, use one over camera distance as a uniform scale factor.*

I tried normalizing the perspective projection matrix using the member function Normalize() from cyMatrix.h (Matrix4 class). The object gets distorted in orthogonal view since I’m preserving the object's size whenever the window is resized.

**Additional functionalities**

**Window resizing:**

I’ve also implemented a resize function and mapped it to the glutReshapeFunc() callback. Whenever the window is resized, the viewport size is changed, and the object’s size is preserved by adjusting the field-of-view (FOV) and the aspect ratio.

**How to use implementation?**

g++ main.cpp -o main -lfreeglut -lglu32 -lopengl32 -lglew32

This command will generate the output file “main” (“main.exe” in Windows) in the working directory. This command includes the GLEW 32-bit linker. I didn’t use an IDE and had all the libraries and headers globally installed, so I didn’t have to use -I and -L tags to specify paths to headers and DLLs. The folder structure for the headers in include is as follows:

-> include

-> GL / all FreeGLUT and GLEW headers

-> cyCodeBase / all cyCodeBase headers

**OS and Compiler**

|  |  |
| --- | --- |
| *Operating System* | Windows 11 (x64) |
| *Compiler* | g++ |

**External libraries and additional requirements**

Apart from FreeGLUT, GLEW and cyCodeBase have been used for this implementation.