report_weatherly

March 23, 2023

1 Report - HW3

1.0.1 Chad Weatherly

Problem

For homework 3, the problem was to implement orthographic and camera projections into the Gz class.

Method

For my method, I didn't add anything new to GzMatrix, as I did the work in the Gz class.

Within Gz, several updated methods were added to introduce matrix transformations, with the main functionality coming from these 3 tranformations (details were gathered from the notes, handouts, and OpenGL documentation):

- `lookAt`, which creates a transformation matrix that will take a 3D coordinate in R^3 and transform it such that the origin is at the eye/camera, there is a reference point at which the camera is pointing to, and an up direction given by a vector.
- `orthorgaphic`, which creates a parallel projection from a 3D space to a 2D space.
- `perspective`, which creates a projection from the 2D shape to the viewport.

The idea was that given the matrix `transMatrix`, a private attribute of `Gz`, this matrix would be a composite of all projection/transformation matrics, s.t.:

$M_{trans} = M_{perspective} M_{orthographic} M_{lookAt}$

As this would be the order that new points would be transformed. The idea was that, given a reference point in 3D, that point could be transformed using M_{trans} to the new coordinate system. As each triangle was extracted from Tris.txt, the 3 points would be extracted, converted to homogenous coordinates (by adding a w-value of 1), transformed, then converted back to cartesian coordinates by dividing by w.

Implementation

This class and all of its methods were coded in C++, with the initial declaration of the class and its methods/attributes in the Gz.h file, then implemented in Gz.cpp.

After the coding was done, compiling was done through the terminal, as I am coding on MacOS with an M2 max chip on VS code. Currently, VS code struggles to compile and link files, but this was circumvented using the terminal. From the HW1 directory, the below code was used to compile the program to an executable object, hw1.exe:

```
clang++ src/main.cpp src/Gz.cpp src/GzImage.cpp src/GzFrameBuffer.cpp src/GzMatrix -o hw3
```

For using G++:

```
g++ src/main.cpp src/Gz.cpp src/GzImage.cpp src/GzFrameBuffer.cpp src/GzMatrix -o hw3
```

As I am new to C++, I was not able to create a makefile to compile, but the above code should work fine on any machine. Also, the hw3.exe file is included in this directory, and can also be run using ./hw3 after compiling.

Results

While I was able to compile and run the program, I was unable to get successful images to save. I found that some part of the `perspective` transformation was causing the `transMatrix` to have infinity or -infinity values. I was unable with the time given to debug and find the issue. Also, the homework instructions and notes from class were difficult to understand, so I'm still not even sure if my methods were right either.

```
[ ]:
```