Matthew Goodman

My Matrix4f is fully feature complete and is capable of multiplication, inverse, perspective matrices, and transformation matrices.

I’m using java and Vulkan so I’m not much sure what use Mat4Stack would have. That said I did put together a quick Matrix4fStack that works as described and uses all of my Matrix math.

My Sphere class creates a sphere using parametric coordinates starting with the top and bottom and then creating vertical rings going around. I seperated your division parameter into horizontal divisions and vertical divisions for more control. I am using an index buffer to render them but I’m not using triangle strips. I’m not sure if that counts for half EXTRA CREDIT. It’s a bit weird to use triangle strips in Vulkan so I’m not sure if that would be the fastest way anyways.

I’m using GLFW to capture mouse motions when the left button is held down. I store the values and use quaternions to recreate the view matrix for EXTRA CREDIT.

I am rendering the entire cube of spheres. I am using several for loops to generate the materials and update the descriptor sets.

I created a basic phong shader from the given skeleton code. Ambient and emissive lighting work perfectly. Diffuse and Specular are having very weird issues. I’ve double checked the math and it looks correct. I think the issue is either with the normals or the view matrix. Either way, it still renders, it just looks a little off.

The PointLight is also rotatable with WASD/IJKL. Using Quaternions to rotate them.

I should have completed all of the parts on the rubric. The only thing missing is that either my lighting shader is off or my view matrix is off. Everything else is accounted for. As for Extra Credit, I did use quaternions for my rotations and I used am using index buffers for rendering (although no triangle\_strips).

All code has been pushed to my git: <https://github.com/MasterMatthew/EngineDev22>

A compiled version: <https://github.com/MasterMatthew/EngineDev22/releases>