CS4052: Computer Graphics – Assignment 4

Matt Donnelly – 11350561

Abstract

For this assignment we were asked to implement a number of features. Firstly, we needed to write an interactive camera allowing the user to move forward, backwards, left, and right and allow for additional degrees of freedom of rotation. Secondly, we needed to adjust our program to render more than one mesh which demonstrates a parent-child hierarchical animation. Finally, we were required to implement Phong Illumination inside our shaders with one model rotating to show correct normal transformation.

Solution

To start this assignment I began by implementing a camera class. The camera class keeps track of its position and direction represented as vectors which change based off its movements. Using these vectors the view matrix for the world can be calculated to pass to the shader. To move forward I simply add the direction vector to the position and to strafe I add the cross product of the up direction and forward direction. For rotation I decided to use quaternions to prevent gimbal lock so all 3 axises can be potentially rotated. When the user moves the mouse I create a quaternion representing the new orientation of the camera and then set the forward direction to be equal to the orientation converted into a matrix multiplied with the forward direction.

Following this I created a mesh in blender representing the ground to render in my program alongside the Super Mario Block I used in the previous assignment. Both these are the stored in separate Vertex Arrays Objects with their own buffered positon coordinates, texture coordinates and normals, To implement a model demonstrating a parent-child hierarchical animation, I decided render two blocks stacked on top of each other moving left to right with the top block rotating. This is done by first creating a model matrix to position the bottom block, I then translate this model to position the top block above it and apply a rotation transformation to it as well.

Finally, for implementing Phong lighting I calculate the intensity of the ambient, diffusion and spectral lighting and then multiply it with the value read in from the texture. In order for the intensity to be calculated the eye position, surface normal position and view matrix must be passed to the fragment shader.

Screenshot