Vincent Loignon

Tyler Cifelli

Lighting

Gouraud:

For Gouraud shading, we based our algorithm off the one the book uses to teach you the different shading components and the one it uses for Phong in the lighting chapter. The vertex shader does all of the lighting calculations (eye vector, light vector, reflection vector, etc) and color calculations (ambient, diffuse and specular), not including anything regarding the texture color. It then passes the specular and diffuse information, along with the TexCoord to the pixel shader for final alteration. The pixel shader (based on values passed to the shader) will either sample the texture for a color or use the color the object is if it was told not to use the texture and use this to perform the last alteration to the color calculations. The shader also incorporates the ability to disable both specular and diffuse components through a Boolean value passed to the shader from the material. This prevents all color calculations for the component if it is disabled rather than just passing a 0 value to the shader and running the calculations anyways. The result of the shader is an accurate gouraud lighting effect on the object.

Phong:

Our phong method of shading works a bit differently than Gouraud. The struct that we use for Phong has different data types in order to fulfill what phong needs to communicate between the vertex and pixel shaders. OutputVS\_PHONG has a position, normal, world position, and texture coordinate. Inside the Phong vertex shader calculates all four of these variables using variables passed into the shader, and hands the output off to the pixel shader for coloring, unlike Gouraud which does both in the vertex. The Phong pixel shader takes this output and does a few more altering calculations to make sure everything transferred well. This includes renormalizing, calculating eye vector, and the reflection vector. Finally, the pixel shader calculates color with the three components (specular, ambient, diffuse) and omits the ones that were told to be omitted with information passed into the shader (if we toggle off diffuse and specular, these values will remain 0.0f). Then, if a texture is used, we incorporate the rgb values into the diffuse component, and add all of the components together to produce the correct color.