Vincent Loignon

Tyler Cifelli

Assignment 4

Implementation Details

We based our implementation off of the examples given by Frank Luna. Rather than separating the code into multiple techniques or files, we decided to incorporate into one file and use Booleans to determine when certain operations are to be performed.

Vertex Shader:

We start by taking the vertex input (position, normal, tangent, binormal, texcoord) and preparing them for the pixel shader. We create a TBN matrix, transpose it and transpose the eye vector and the light vector into Tangent space. We also prepare our normal, world position, position and texcoord as per usual.

Pixel Shader:

We re-normalize the input to avoid weird problems from interpolation. We sample our texture to get its color for later use. We sample the normal map and prepare the normal from it for later use. We calculate our reflection vector for the light to prepare for the lighting calculations later; the normal for this are determined by whether or not gUseNormalMap is true or false (true means to use the normal map’s normal and false means to use the vertex’s normal). We calculate the reflection vector for the environment map; once again the normal is determined by whether or not we are using the normal map as well as what value the normal strength is set to. We use this to sample the environment map for later use. We now calculate the coefficients t and s which are used for specular and diffuse respectively. Before actually using s, we determine whether or not to modify it and modify as needed based off of gUseNormalMap. This will properly blend the normal map with the regular normals based off a normal map strength value. We then calculate all our specular, diffuse, ambient and alpha values based off the material, light and blend values. If we are supposed to use our texture (gUseTexture is true), we now apply the texture’s color to the ambient, diffuse and alpha. We now also check to see if the environment map should be used. If so, we apply it properly to the specular component. Lastly we combine the ambient, diffuse and specular components to create our final color.