Andrew Esposito & Nikolai Alexander-Himlan

4/25/2015

EGP-300-52

Assignment 4 – Normal Mapping & Reflections

**Main purpose of the exercise:**

* Enhanced knowledge of shader rendering
* Get familiar with more advanced technique (reflection and normal map)
* Get familiar with D3DXMESH class data structure.
* Support Texture and the advance shading of models.

**Controls:**

* Mouse – move camera
* Mouse Wheel – zoom in/out
* E - Toggle Ambient Rendering
* D - Toggle Diffuse Rendering
* C - Toggle Specular Rendering
* R - Toggle Reflectivity
* T - Toggle Texture Rendering
* N - Toggle Normal Map Rendering
* -/+ - Adjust the strength of the Reflectivity
* A/S - Adjust the strength of the Normal Map
* [/] - Adjust the strength of the Specularity
* 1-7 - Set the strength of the Specularity
* G - Set the material of the objects to Gourad shading
* P - Set the material of the objects to Phong shading
* I - Set the material of the objects to Advanced Phong shading
* O - Iterate through each model
* F1-F10 - Switch between models

**Extra Information:**

* Our texture file is stored under /Assets
* Our shaders are stored under /Shaders

**Known Problems:**

* DirectX 9.0 primitives do not appear to have enough vertices to allow proper uv coordinate wrapping, causing some of the problems with texturing on the shapes.
* Normal mapping will disappear if reflection & specular lighting are disabled.
* Half of the object reflections reflect the background on the opposite side of the shape.

**Lighting Equation:**

***The following steps make up our program’s lighting equation:***

1. Computing the vector from the vertex to the eye position in world space by normalizing the eye position subtracted by the vertex position.
2. Get the data from the normal map.
3. Compute the reflection vector, environment color, reflection value, and reflection blending.
4. Get the dot matrix of the light’s reflection vector and the vector from the vertex to the eye position, and put it to the power of the value that represents the strength of the specular lighting to determine how much specular light is visible to the viewer.
5. Determine the diffuse light intensity that strikes the vertex by determining the dot matrix of the normal and light direction according to the world view.
6. Compute the ambient, diffuse and specular lighting by multiplying the color of the material and lighting for each different method material separately. For specular lighting, multiply the resulting product by the specular light the viewer can see. For diffuse lighting, multiply the resulting product by the diffuse light intensity that strikes the vertex.
7. Combine the reflection values into the ambient and diffuse lighting values.
8. Initialize the texture color & alpha values.
9. Combine the ambient lighting into the diffuse lighting & modify the alpha value by the diffuse.
10. Modify the pixel color using the texture color and the diffuse value.
11. If diffuse and ambient lighting are both disabled, reflections will need to be reapplied.
12. Add specular lighting, if applicable, to the pixel color.
13. Lastly, copy the alpha value for diffuse lighting and return the pixel color.