Assignment 3 - Material Render & Shaders

Due Mar 2 by 11:59pm **Points** 100 **Submitting** a file upload

Exercise 3 – Material & Shaders - Color, Lighting and Texture

Please read all guidelines and post any questions you might have as soon as possible so the entire class can learn the answer.

I recommend starting right away as the exercise is challenging and requires team work.

General Guidelines:

1. Submission should be made via the DropBox in order to easily grade it.
2. Any theory should be made in Word/PDF and sketches can be added as jpg attached to the document
3. Submission of this exercise is **in couples**
4. Partners should implement the project together with full knowledge of the entire code.

Submission Guidelines:

1. The project should be submitted as a zip file with the exercise and the last name of the students – for example '**Exercise3_Solo_Skywalker.zip**'.
2. The submitted zip file should contain**only files needed to build the project and run it**. In other words, the project files, the sources and headers and the resource files (textures, models…) - no intermediate files or directories (such as sdf, obj and others) should be included – review the zip files I created for the book projects as an example.
3. A working release version of your application (.exe file) should be included in the zip with all dlls – the application should be able to run without need to compile or change.

Submission Date:

1. Submission to the drop box should be made by **Monday, March 2st 2015 at midnight**

Overview

-

This exercise is based on the previous project – additional h and cpp file for a base material class were added to the skeleton project as a reference that you need to download.

Reference for how to connect the shader, pass parameters and texture can be found in the course text book. I advise you to go briefly over it during the implementation.

In this exercise you need to connect your procedural generation (keep the classes themselves) of the sphere, cylinder and cone and replace them with the data generated by Direct3D.

To do that you need to use the following DirectX 9c functions:

- D3DXCreateBox
- D3DXCreateSphere
- D3DXCreateCylinder
- Replace your cone creation as well (how do you create a cone using the above functions?)
- Bonus points for adding Tea Pot and Torus!

Goals of the exercise:

- Move to shader rendering and basic shader system (goodbye to the fixed pipeline)
- Learn about D3DXMESH class data structure.

- Implement and connect a basic material
- Implement both **Gouraud** and **Phong** shading - **each partner is required to implement one technique (vertex and pixel shader)** for your existing primitives
- Support Texture and the generic lighting model.

Implementation Specifications

1. Download, enhance and implement the **BaseMaterial** class provided in the new skeleton project as follows:
    1. The BaseMaterial class is the base class that represents a model material (colors, reflectiveness, texture…)
    2. The class should hold a pointer to the effect file and connect to the effect parameters
    3. During render the class will pass all required data to the shader.
    4. The class pointer should be held by the **BaseObject3D** class so that each object can have its own material with possible different parameter
    5. During its render, the base model will call the material with all needed data (such as matrices, viewer location, light…) so that the material will set it to the shader prior the render of the model data.
    6. Add the following attributes to each Vertex (you need to enhance the vertex declaration accordingly – review the book for examples):
        1. i. Texture coordinates:
        2. ii. Normal:

1. Write a short one page documentation (Word or PDF) that describes your lighting equation.

1. Create an effect file that will implement the Phong shading model as explained in class and the book – **the shader must contains comments explaining each component!**

UI Controls

1. Camera control – we will be now be moving to 'examine mode' camera:
    1. **a. Model should be placed at the (0,0,0) of the world**
    2. Mouse movements should control rotation of camera around the model
    3. Mouse wheel should control radius / distance from the model

1. Lighting equation should work perfectly **with and without** texture.
2. Switch between Solid render and Wireframe by pressing '**W**'
3. Switch with/without texture by pressing '**T**'
4. Switch between Objects by pressing '**O**'
5. Switch Specular on/off with '**S**'
6. Switch Diffuse on/off with '**D**'

Grading

- **[15%] Exe code runs properly**
- [10%] Clean code standards
- [10%] Camera / model mouse control
- [5%] Wireframe / Solid mode
- [30%] Gouraud / Phong shaders technique
- [10%] Specular
- [10%] Diffuse + ambient
- [10%] Texture mapping
- [5%] Bonus points for extra cool!