Assignment 6: A lit and textured interactive virtual world

Out: 15th March 2019 **Due: 26th March 2019 at 11:59pm**

In this assignment you will extend your program from Assignment 5 by adding lights and textures.

Look at the provided humanoid-lights-textures.xml for examples of different aspects of this assignment.

Lighting basics (30 points)

A light can be attached to any type of node (in the coordinate system of the node). **Please use the Light class from the Lighting example we discussed in the lectures.**

1. Modify your scene graph classes so that any node can store multiple lights. Add functions to add a light to a node so that it can be used by the XML parser. Modify the parser so that it can parse lights, as specified in the example xml file.
2. The material of an object can be specified as shown in the XML file. Your XML reader should already be able to parse all material tags.
   Note: The light uses some of the same tags as the material (e.g. ambient, diffuse, specular). Thus the parsing of these tags depends on whether you are specifying material or light properties.
3. Incorporate the lighting shaders from the example in class in your code. The scene graph will now (automatically) keep track of all the new shader variables.
4. Write a function that will descend the scene graph, convert all the lights into the view coordinate system and return them in a list. Be sure to incorporate the animation transforms if any.
5. Modify the scene graph's draw function so that draws with the light.
6. Drawing a leaf should now pass material properties instead of simply a "color".

Think about how you will implement 4-6 above. This functionality should be suitably divided among the scene graph and the renderer classes. Remember that they have been designed so that the scene graph and nodes are independent of the library used to render them (i.e. JOGL), while all the JOGL-specific code is confined to the renderers.

Also remember that all lights apply to all objects.

I highly recommend that you prepare a small, simple scene with 1-2 objects to test your program.

Spot lights (10 points)

The lighting shaders currently do not support spot lights. Add support for spotlights. This involves two things: changing the XML parser so that it supports additional tags for spot direction and angle for a light, and changing the shader so that it takes these parameters and uses them correctly.

Texture mapping (30 points)

An image can be specified in the XML file using the <image> tag. Each image tag specifies a unique name. All such images are converted into TextureImage objects (look at the TextureImage class from the example in lecture). Then, an object can use a texture using the "texture" attribute. Look at the provided XML file for an example of this.

1. Add code so that it enables the appropriate texture object, and uses it.

Note: The code in the shaders that does texture mapping will now expect a texture bound to every object. For those objects that are not textured, this will produce an incorrect result. In order to avoid this, create a 1x1 "white" texture image, and include it in your scene XML file. Then, either change the parser, or the actual drawing so that if there is no

texture bound to the object, it uses this white texture instead. A white texture essentially means “no texture”, since the shader is blending the texture color with the shading color.

Add a single texture to your small scene graph to test this part.

Part 2: Your scene (20 points)

Enhance the scene graph model that you created in Assignment 4 in the following ways:

1. Instead of “color”, each part of your model should now have material.
2. At least two different texture images should be used in your scene. They should be clearly visible in the rendering (i.e. don’t texture a part that is so tiny in the rendering that it is difficult to see the texture). Be sure to scale the texture appropriately so that it looks realistic (e.g. a brick texture where the bricks look unrealistically big is not acceptable).
3. Your scene should contain at least two stationary lights, and at least one light that is tied with a moving model. At least one light should be a spotlight, and its effect should be clearly visible in the rendering.

You are free to use images that you obtained from other sources, assuming it is legally allowed. HOWEVER if you do not own them, you must cite their source in a separate README file clearly mentioning the source of the image (to the extent that we are able to access it in the same way as you). Failure to do this will result in a grade penalty!

Part 4: Document (10 points)

Submit a user manual for this program as a PDF file that showcases its features. This should include screen shots of what it can do, and any features that it supports (including keyboard control of the camera). This is your chance to show us what you have achieved, so make sure you capture good screen shots!

What to submit

Submit the project folder, other accompanying files and the PDF document as a zipped file (make sure you include the README file with citations, keyboard control and a summary of what you could and could not complete).