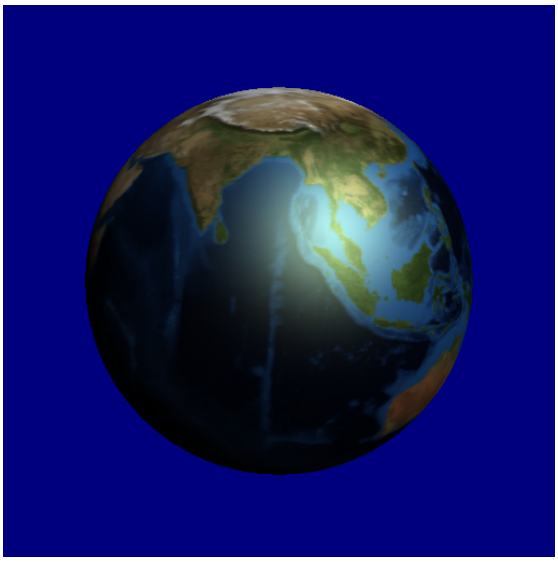

Figure 1: The world!

INF3320 - Obligatory Exercise 3

Lighting and texturing with GLSL

Bart Siwek, revised by André Brodtkorb

Revision:

We will pickup where the last compulsory exercise left us. Since we have the ability to inspect the objects at various angles lets make something that is worth looking at - Earth for example.

The Application

The goal of this application is to setup texturing, use the loaded texture with GLSL to create a textured sphere lit using Phong shading. Specifically you will need to:

1. Load, compile and link a shader program.
2. Create a Phong shader that will compute per-pixel lighting with respect to a single light in the scene.
3. Assign a value to the uniform variable representing the specular color of the sphere.
4. Load the texture using GLI library, setup its parameters and generate mipmaps.
5. Assign the texture to a sampler uniform variable in the shader and use it to sample a color that will be used for diffuse and ambient components.

DETAILS AND HINTS

This exercise uses a new library - GLI (OpenGL Image). This is a header only library authored by the same person as GLM. This library much like GLM, requires no compilation and is easy to install.

Try to tackle each task separately. First make sure you can load and execute the simplest shader (we suggest using a shader that returns a constant color). For you convenience and to keep the boiler-plate code low you are given a `Shader` class that should make this task much easier.

Next work on the Phong shading using constant values for material diffuse, specular, ambient colors, as well as specular exponent. The Phong shader in its full form should do per-pixel lighting with respect to the only light in the scene (treat it as a point light). It should include attenuation from the light source and use a half-vector for specular computations.

The next task to tackle is assigning a value to the `SpecularColor` uniform variable. You will need to obtain the variable ID from the shader and bind a four dimensional vector value to it. We will use this value as follows - the RGB components of the value will be used directly in the shading equation for specular color. The A(lpha) value will be used as a specular exponent in the shading equation, thus you will need to scale it from $[0, 1]$ range to a more reasonable one (we suggest $[0, 256]$).

The second to last task is to load and setup a texture in OpenGL. This is where the GLI library and especially the $gli :: createTexture2D$ function will aid you greatly. After the texture has been created bind it, set its wrapping (we suggest repeating it in both directions) and interpolation parameters (we suggest linear and mipmap-linear). You are given the `EarthDiffuse.dds` file that contains the earth texture we used.

A final task is to use the texture. You will need to bind it, assign the texture unit id to the sampler uniform variable of the shader and the sample the texture in the shader. The color you get should be used as a diffuse and ambient material color.

Last hint - the `Shader` class contains a lot of methods that aid the debugging of shader compilation, liking, uniform and attribute access. Use them to your advantage.

HANDING IN THE EXERCISE

The assignment is individual and everyone shall create their own program. If you choose to use code or derivations of code that is not your own, its source and author shall be explicitly cited. We will normally accept some degree of unoriginal work, but we may require that parts of the unoriginal code must be rewritten.

We reserve the right to do an oral examination of the student for each submitted exercise. The failing or passing of this examination will decide whether you pass this exercise, and this decision is final.

The program should preferably be a C++ program, and the visualisation shall be done using OpenGL. The choice of windowing toolkit is the student's. Document your efforts by commenting the code.

All source code as well as a Makefile shall be included. The Makefile should have at least a target to build the program as well as a clean-target that deletes all object-files as well as the executable. Assume that your user name is `foo` (exchange "foo" with your user name). Put all files in a directory called `foo-3`.
Handing in the assignment is performed by creating a private repository at either GitHub (sign up as student for five private repositories on `https://github.com/edu`) or Bitbucket, and then sending an email to `mariuek@ifi.uio.no` with the clone URL and a short description detailing the files involved in your hand-in. Make sure the program compiles and runs cleanly on the computers at IFI, and that it fulfils the requirements outlined in the evaluation guidelines.

Good luck!