Name: Joonas Nissinen

Student number: [redacted]

Assignment: Exercise 1 - The Standford Bunny model in OpenGL

1. Compiling the program

The program has been built in a Windows environment using Microsoft Visual Studio 2010. The compiling of the program requires OpenGL 3.2 (or later). The required libraries by the program are the standard C/C++ libraries, freeglut and glew.

The .zip package contains all the required source code files, including the shaders and all textures used by the program. The package also contains the precompiled .exe program that was compiled in Windows 7 64-bit environment using Microsoft Visual Studio 2010 and all the Visual Studio 2010 project files.

If you're using a Windows environment you can either try executing the .exe file or open the given Visual Studio 2010 project files. When in Visual Studio 2010, if everything went right you should be able to compile and run the project.

If you're in a linux environment you need to manually compile the project using either gcc or g++. The compile command I have used requires that you have the freeglut and glew libraries correctly installed in your system.

Compile using: gcc -lGLEW -lglut -lGLU bunny.cpp plyparser.cpp utils.cpp rgbparser.cpp

Run using.......: ./a.out

I had some problems with the linux machine due to some OpenGL extension problem. Those problems, however could be fixed by changing the state of the GL_EXT_direct_state_access flag in the bunny.hh file between 0 and 1. The compile problems should look like missing or redefinitions of the glProgramUniform -functions declared in the bunny.hh. Another possible problem is the naming of GL.h in the bunny.hh, which on some Linux systems appears to be gl.h.

2. Program features

The following represents the features present in the program from the feature requirements tables in the exercise description.

2.1 Mandatory requirements

Requirement	Status
Perusasiat, kuten z-puskuri ja kaksoispuskurointi on toteutettu oikein	Done
Käytetään piirtämisessä vertex array -objekteja, eikä OpenGL 1 -tyylisiä käskyjä	Done
Käytetään shadereita, jotka ladataan tiedostoista, käännetään, linkitetään onnistuneesti	Done
Kirjoitetaan vertex shader ja fragment shader -koodi GLSL:n versiolla 1.5	Done
Käyttöohjeet eli miten ohjelmassa navigoidaan, miten se lopetetaan, jne	Done

2.2 Basic requirements

Requirement	Status
Ladataan 3D-objekti sisään tiedostosta	Done
Ladataan tekstuuri tiedostosta	Done
Päivitetään viewport, jos ikkunan koko muuttuu	Done
Toteutetaan suuntavalo (directional light) shadereilla	Done
Shaderit toteuttavat Phong- tai Blinn-Phong-sävytyksen (per fragment, ei vain per vertex)	Done
Shaderit toteuttavat teksturoinnin ja moduloinnin Phong-mallin kanssa	Done
Tekstuuri on mapattu jotenkin järkevästi objektiin; helppo koordinaattiakselin suuntainen projektio riittää	Done
Toteutetaan interaktiivinen käyttöliittymä, jossa näppäimistön (2 p) ja hiiren (2 p) avulla voi liikkua piirretyn objektin ympärillä (2 p) ja skaalata objektin kokoa (2 p)	Done

2.3 Extra requirements

Requirement	Status
Toon shader –versio Phong-sävytyksestä	Done

Requirement	Status
Vertex shaderilla toteutettu objektin animointi; esim. sykkivä tai värähtelevä Stanford Bunny, eikä juokseva tms., mikä vaatisi itse mallin animointia	Done
Proseduraalinen tekstuuri ja sen mappaus objektiin	Done
Bump mapping/ Normal mapping	Done
Pistevalon toteutus siten, että intensiteetti on kääntäen verrannollinen etäisyyden neliöön	Done
Spotlightin toteutus	Done
Käytetään useampaa kuin yhtä tekstuuria 3D-mallin eri osissa[1]	Maybe

[1] The bump mapping uses two texture files applied to the same model, so this is a bit iffy. Also this would be trivial since my model uses cube mapping I could just change the texture of one face of the cube, which would map different texture on some parts of the bunny. However, without any u,v -mapping information in the model file though this is silly, so I do not feature this in the program. I tried and created my own u,v mapping using sphere mapping and cube mapping, but there were problems because of the vertex data. I can show you the results and the code if you wish.

2.4 Other features

Requirement	Status
Rautalankamalli objektista	Done
Heijastuskarttaversio teksturoinnista	Done
Muuttuva proseduraalinen teskturointi	Done

3. Using the program

Once the program is successfully started and if there are no suprise errors present the following commands should be at your disposal through your keyboard and mouse. You can exit the program either through the escape (ESC) button or by closing the window from the X in the window's corner.

3.1 Keyboard functions

- UP: Translate the camera up in the world
- DOWN: Translate the camera down in the world
- LEFT: Translate the camera left in the world
- RIGHT: Translate the camera left in the world

- E: Enable the effects applied to the object (a soft of compromise between throbbing and wavy effect)
- T: Switch textures (is not enabled in the procedural texturing)
- S: Switch shader program (4 different shader programs available in the program)
- W: Enable the wireframe model of the object

- +: Scale the object up
- -: Scale the object down

- ESC: Close the program

3.2 Mouse functions

- Scroll wheel up: zoom in
- Scroll wheel down: zoom out

- Press left & move: rotate the object
- Press right & move: rotate the camera

4. Other notes about the program

The fourth shader program representing the bump/normal -mapping uses a sort of reflection mapping in for the texture. Thats why the textures appear different on the object. If you look at the shader code for the shader your can disable this by removing the comments of one line in the beginning of the shader's main program.

The third shader program representing the procedural texturing on the object changes the texturing as a function of time. This is a feature not a bug. The texture mapped is a small portion of the Julia fractal.

Thank you for your understanding in my case of illness. I am sorry that the submission was late, but

I believe it was worth it. I am currently still getting better. In case you have any problems or questions, please email me to joonas.nissinen@aalto.fi.