CIS 636 Introduction to Computer Graphics
CIS 736 Computer Graphics

Spring 2011

Homework 4 (Machine Problem)

Lighting Effects and Surface Detail

Assigned: ~~Mon 28 Feb 2011~~Sat 05 Mar 2011
Due: Fri 18 Mar 2011 (before midnight)

The purpose of this homework is to help you practice applying maps and textures to the polygon meshes you learned to load in Machine Problem 2.

This homework is worth a total of 20 points (2%).

Upload an electronic copy of the assignment in PDF form (converted from your word processor, or scanned) to your K-State Online (KSOL) drop box before the due date and time.

References

636 only

Mappings

OpenGL cube mapping: http://bit.ly/eJEdAM
Transparency using alpha blending in OpenGL: http://bit.ly/hs82Za
Transparency example: http://bit.ly/gr6oGW
Shadow mapping in OpenGL – tutorial by Paul Baker: http://bit.ly/e1LA2N

File I/O

OpenGL.org documentation on loading JPEG files: http://bit.ly/fJbwJn

Lighting

Emissive surface material in OpenGL: http://bit.ly/hhPn8u

636 & 736

Vertex Buffer Objects (VBOs) in OpenGL

NeHe tutorial #45, 2004: http://bit.ly/gTDOql
Ozone3D tutorial, 2006 – 2007: http://bit.ly/gXvmic
Brian Hall's tutorial (simple, Windows-based), 2006: http://bit.ly/hj4kJT

Direct3D

DirectXers Wiki Tutorial 1: http://bit.ly/cvCYb3
Toymaker FVF tutorial: http://www.toymaker.info/Games/html/fvf.html
Craig Andera's *DirectX* tutorials
    03. Rendering basics: http://bit.ly/cVCWLr
    04. Vertex Buffer creation: http://bit.ly/8YNzDT

05. Vertex Buffer rendering: http://bit.ly/9kam4o
07. 3-D rendering: http://bit.ly/bHPad9
08. Lighting (and shading): http://bit.ly/ck1XOm

GameDev Vertex Buffer tutorial: http://www.gamedev.net/reference/articles/article1946.asp
Toymaker VB tutorial: http://www.toymaker.info/Games/html/buffers.html
Tutorialized DrawPrimitive() for rendering VBs: http://bit.ly/atM1D0

Shaders

Cartoon shader in GLSL: http://bit.ly/fZYAfH

736 only

Mappings

Stencil test in OpenGL: http://bit.ly/ewsqSu
Bump mapping in OpenGL – tutorial by Paul Baker: http://bit.ly/fun4a5

Textures

Procedural textures in OpenGL: http://bit.ly/hvT0v5
Procedural noise texture example, with code: http://bit.ly/gMWsoj
Underwater caustics in OpenGL: http://bit.ly/grYFgL

636 Problems

First, load the `.geo` file for the house. You may use your own solution to MP2 or the given code (crediting your source properly per the citation instructions).

For your solutions, you may use OpenGL *or* Direct3D. Specify which, and turn in your source code, screen shots of the results, and a `README.txt` file containing compilation instructions.

1. **(25%) Cube Mapping (Reflection/Environment Mapping).** Write a program in C/C++ (or extend your MP2 solution) to apply a simple cube reflection map to it. Use the six images from http://bit.ly/eJEdAM as the environment.

2. **(25%) Alpha Blending-Based Transparency Mapping.** Follow the OpenGL.org instructions and those given in Lecture 16 to make the house transparent, and place one instance from it in front of another. Both should be shaded with a (different) solid color.

3. **(50%) Shadow Mapping.** Follow the instructions from Paul Baker's tutorial on *PaulsProjects.net* (http://bit.ly/e1LA2N) to apply a shadow map to the house.

Extra credit (25%). Do one of the following:

a) **(Emissive light sources)** Make your house glow green in the dark by making its surface emissive. See the "emissive material" tutorial and sample code under "Lighting" above.
b) **(Non-photorealistic rendering: toon shader)** Apply the cartoon shader given above to the house. You may use HLSL if you prefer.

(You may do both parts for fun, but only 25% extra credit will be given.)

736 Problems

First, load the `.geo` file for the house. You may use your own solution to MP2 or the given code (crediting your source properly per the citation instructions).

1. **(25%) Stencil Buffer Shadows.** Write a program in C/C++ to cast a shadow from the house using a stencil test. Follow the documentation given in Mark Kilgard's tutorial above.

2. **(50%) Procedural Texture.** Use Direct3D for this problem. Apply a procedural texture to the house. You may use a noise texture, modifying the one given above, or look up and implement your own (*e.g.*, wood grain, *etc.*).

3. **(25%) Bump Mapping.** Follow the instructions from Paul Baker's tutorial on *PaulsProjects.net* (http://bit.ly/fun4a5) to apply a bump map to the house. Use the normal map given in his `.zip` archive.

Extra credit (25%). Do one of the following:

a) **(Caustics)** Send your house to Atlantis! Apply underwater caustics as shown in the tutorial under "Textures" above to make it look as if it is at the bottom of the sea.
b) **(Non-photorealistic rendering: toon shader)** Apply the cartoon shader given above to the house.

(You may do both parts for fun, but only 25% extra credit will be given.)

Class Participation (required):

Post your final *revised* draft proposal, as specified in class, to KSOL by Sat 12 Feb 2011.
Consider entering the First Annual K-State Computer Graphics Logo Design Contest (instructions **now** posted in KSOL).