Concordia University

COMP 371 --- Fall 2012

Computer Graphics

Assignment 3

Theory part due November 23, 2012 (no late submissions allowed)

Programming part due November 30, 2012 (no late submissions allowed)

PURPOSE: This assignment will give you the opportunity to learn more about surface illumination, 3D object modeling, and texture mapping. This assignment contains both a **theoretical** question, which you don't need to implement, and a **practical** question, which requires you to write simple OpenGL programs.

SUBMISSION: The assignments must be done individually. On-line submission is required – a hard copy will not be read or evaluated.

PREPARATION: Parts of this assignment may require knowledge of basic concepts from chapters 10 through 19 of your textbook.

Question #1: (20%) [Theoretical Question]

Consider toroidal surface of revolution formed when an ellipse in the xy plane with semi-major and semi-minor axes r_x and r_y is first shifted r units along the x-axis and then revolved about the y-axis.

a) Derive an implicit equation of the torus in the Cartesian coordinates.
b) Calculate the normal vector to the surface at point P=(x,y,z) on the surface of the torus.
c) Derive parametric representation of the torus.
d) Assume Phong illumination model: $I = I_a k_a + f_{att} I_l [k_d \max(0, N \cdot L) + k_s (\max(0, R \cdot V))^{n_s}]$, where

I_a = 50, $I_l = 100$, k_a =0.2, k_d =0.3, k_s =0.4, $n_s = 3$, $f_{att} = \frac{1}{\| P - l \|^2}$, l=(0,1,1), v=(1,0,0) and *l* and *v* are light source position and viewer position, respectively. Calculate the value of illumination at point $P = (\sqrt{2}, \sqrt{3}/2, \sqrt{2})$ on the torus with the radii r_x =2, r_y =1 and r=3.

e) This question is about environment texture mapping. We are interested in texturing the object with the reflection of the environment on the surface of shiny object, e.g. the reflection of the moon landscape in the astronaut face mask, or reflection in mirrored sun glasses of Boss Godfrey character in 1960s movie *Cool Hand Luke*. As the object moves the reflection texture varies (flows over). In OpenGL glTexGen texture function with parameter GL_SPHERE_MAP generates texture coordinates for environmental mapping for vertex *V* on the object using the following formulas

$$s = r_x / m + \frac{1}{2}, t = r_y / m + \frac{1}{2} \quad (1)$$

where $r = (r_x, r_y, r_z)^T$ is the vector reflected from the object at *V* and calculated from

$$r = u - 2nn^T u$$

n is the current unit normal to the object surface at *V* after transformation to eye coordinates,
u is unit vector from the origin to the object vertex *V* (in eye coordinates) and

$$m = 2\sqrt{r_x^2 + r_y^2 + r_z^2}\ .$$

We assume that the environmental texture is mapped on the unit sphere placed far away from the viewer. Derive equations (1).

Question #2: (80%) [Practical Question]

Description: *This assignment is an extension of the OpenGL programming practical problems developed by you in your earlier assignments. OpenGL includes functions that will allow you to easily program a number of advanced 3D rendering concepts some of which will be covered in the class/tutorials. The extension has been chosen so that you will be able to understand most of these concepts.*

Problem Statement:

This assignment aims at incorporating lighting, texture and some advanced effects into the program that you have built so far. Extend your assignment #2 to include the following functionalities:

R1) Lighting:

- Light 1: Create a spotlight (white) positioned at the front of the motorcycle facing backwards and slightly downward at an angle somewhere between 20 and 30 degrees (illuminating the ground partially in front of the motorcycle). Provide controls to toggle the light between high beam/low beam modes. The high beam/low beam toggle should not only change the spotlight direction but also its spot cut-off angle. Generally, the high beam position is brighter and farther-reaching.
- Light 2: Create a moving light source (blue). The light source should move in a circular motion (not spinning on one spot) above the motorcycle and trailer about an axis passing vertically along the center of gravity of the motorcycle, with the light facing downwards. Choose a spot cut-off angle according to your convenience. Use a toggle to start/stop the motion of the light.
- Light 3: Create another spotlight (crimson red) located at the third person camera and directed along its view direction. This light source moves with the third person camera. Choose a spot cut-off angle according to your convenience.
- Note the colors in the brackets are suggested light colors. You can pick your own colors, but remember to provide different light colors to distinguish between them. Suggested keyboard controls are mentioned in a later section.

R2) Texture Mapping

- Use suitable textures for motorcycle's lamp, gas tank, seat, exhaust pipe, engine, trailer, or other parts of the body. At the least three different textures should be used for texture mapping. You'll find a collection of downloadable textures at **http://www.cgtextures.com/**, but you're certainly welcome to find some elsewhere or even create your own if you wish (e.g., using any Photo or Image Editor).

R3) Material Properties

- Set the material properties of for motorcycle gas tank and mudguards. Toggle between a metallic shine and rusty appearance. If you are adventurous, try adding some chrome detailing to other parts of the motorcycle like wheels rims, spokes, exhaust pipe and chain.

R4) BONUS (15%). Advanced effects – ANY ONE OF THE FOUR BELOW

- Motion Blur: Produce motion blur (moving objects look blurred) effect for the rotating wheels and the rotating chain. There are many ways of doing this. The Accumulation buffer, Alpha Blending, etc., are some of the simple techniques of implementing it.
- Smoke: When smoke is enabled, simulate fire and smoke coming out of the exhaust pipe of the motorcycle. Use controls to turn the smoke on/off.
- Environment texture mapping: Texture the shiny gas tank with the reflection of the environment to create the effect similar to, e.g., the reflection of the moon landscape in the astronaut face mask. Add sufficient ambient light to view the objects around.
- Mirror reflection: Add two mirrors attached to the handlebars. Create mirror reflections at the mirrors. Use multipass technique involving stencil buffer.

Suggested keyboard commands to control the view:

- Toggle using l/L enabling and disabling of Lighting.
- Use F1 to turn on/off light 1.
- Use F2 to turn on/off light 2.
- Use F3 to turn on/off light 3.
- Toggle the material property of the cannon barrel from rusty matt (diffuse only) finish to glossy chrome (shiny/specular) finish using F4.
- Motion of Light 2 should be started/halted by toggling key '2'.
- The high beam/low beam mode of Light 1 should be controlled using h/H.
- If you are implementing smoke use g/G to control it.

[Note – The above commands are in addition to the ones given in assignment #1 and #2. Make suitable assumptions wherever needed.]

What to hand in for Assignment 3

Questions 1: (20%) (Theoretical Question)
- Submit your answers to question 1 **electronically** (as a MS Word or PDF file, for example, called **TheoryYourIDnumber.doc** or **TheoryYourIDnumber.pdf)**: **https://eas.encs.concordia.ca/eas/authentication.jsp**
Submit this file as "Theory Assignment 3".

Question 2 (80%) (Practical Question)
Submission Deliverables (Only Electronic submissions accepted) :

1. Submit a well-commented OpenGL source program including data files, if any, along with the Makefile or Project/Workspace files, and any other instructions for compiling/building/running your program. The source codes must be submitted **electronically** in a **zip format** with all the required files (example: **YourIDnumber.zip**): **https://eas.encs.concordia.ca/eas/authentication.jsp**
   Submit this zip file as "Programming Assignment 3".
2. Demonstrate your working program from an **exe file dated on or before November 30, 2012** to the lab instructor assigned to evaluate this assignment, during the week of Dec. 3, 2012.
3. A brief write-up about your program, highlighting the implementation of lighting and texture, usage instructions and special features that you would like us to consider during the evaluation.

EVALUATION CRITERIA for Question 2 of this assignment

1. Only working programs will get credit. If your program does not run, we will not debug it.
2. Breakup:
   - R1: 45% (equally divided among the requirements listed in item R1 above)
   - R2: 30%
   - R3: 15%
   - R4: 15% (bonus)
   - Source program structure and readability: 10%
     - General aesthetics and write-up: 5%

Note on marking: For those requirements where there is a possible variation in fulfilling the requirement, just barely meeting the requirement will not result in a full mark. For example, if your model only barely has the lighting listed in R1, expect only partial marks. Models where there is an obvious extra effort made to make the graphics more appealing may expect full or nearly full marks. The esthetic/artistic aspect of graphics is also important, not just the mechanical aspects.