

# CPSC 314

## Assignment 3

Due Fri Dec 1, 2006, 4pm

Answer the questions in the spaces provided on the question sheets. If you run out of space for an answer, use separate pages and staple them to your assignment.

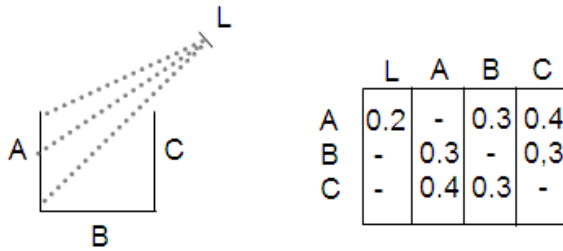
Name: \_\_\_\_\_

Student Number: \_\_\_\_\_

Question 1	/ 6
Question 2	/ 8
Question 3	/ 4
Question 4	/ 30
TOTAL	/ 48

## 1. (6 points) Radiosity

The following figure illustrates a light source  $L$  that illuminates surface  $A$  directly, and therefore also indirectly illuminates surfaces  $B$  and  $C$ . Assume that surfaces  $A$ ,  $B$ , and  $C$  are diffuse, and  $\rho_A = 0.8$ ,  $\rho_B = 0.8$ ,  $\rho_C = 0.8$ . The table below gives the form factors of the energy transfers between the various surfaces.



Solve for  $I_A$ ,  $I_B$ ,  $I_C$ , the amount of light leaving surfaces  $A$ ,  $B$ , and  $C$  if the light source  $L$  emits a total of  $500W$  of light. Give the amounts of light absorbed by each of surfaces  $A$ ,  $B$ , and  $C$ . Give the amount of light that gets lost directly into space from  $L$ . Give the amount of light that gets lost into space after leaving each of surfaces  $A$ ,  $B$ , and  $C$ .

## 2. Raytracing

- (a) (4 points) You are given a ray equation  $P(t) = P_0 + Vt$  and an axis-aligned bounding box (AABB) given by  $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ . Write the pseudocode for a function  $intersect(P_0, V, x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$  that returns *true* if the ray intersects the bounding box.

- (b) (4 points) Estimate the cost, in cycles, of the ray-AABB intersection test, and compare that with the cost of a ray-sphere intersection test. Assume that a floating point addition takes 2 cycles, a floating point multiplication takes 4 cycles, and that a division takes 22 cycles. Compare this with the cost of doing a ray-sphere intersection test. Suppose that you are considering to use either an AABB or a sphere as a bounding object for the raytracing of a complex statue. Which test would you recommend using and why?

## 3. (4 points) Parametric Curves

Determine the basis matrix for a parametric cubic curve defined by  $P_0 = P(0)$ ,  $P_{0.5} = P(0.5)$ ,  $P_1 = P(1)$ ,  $T_1 = P'(1)$ . Do not bother with numerically inverting any matrices.

#### 4. (30 points) Graphics Demo

Implement a scene, animation, game, or tutorial of your choice in OpenGL. You may work in groups of two if you like. A list of ideas is suggested below. Your final mark will be based in part on a short demo given to a TA or the instructor. You can use any of the OpenGL demo programs provided during this course as a starting point if you like, although you should document this in your README file.

If you need help regarding how to implement any particular features, do not hesitate to ask the instructor or the TAs. Be sure to develop your project in testable stages. The best demos will be glorified forever in the 314 Hall of Fame!

##### **Surfaces of Revolution or Swept Surfaces**

An object like a wine glass is a surface of revolution. Build an application which lets you draw a curve using the mouse, and then this curve is turned into a 3D object by rotating it around a given axis, or by sweeping some other fixed cross-sectional shape along the curve.

##### **Tutorial**

Implement a tutorial for part of the course that you think would benefit from a demonstration or tutorial. One idea is to generate a graphics 3D visualization of the various rays that are produced during ray tracing a simple scene. Another is to produce a graphical demonstration of Bezier curves or Bezier surface patches. You might animate the control points in a procedural fashion and show the resulting animated shape of the curve or surface. One last idea is to graphically illustrate, in 3D, the various vectors involved in performing local lighting computations. As the user moves the mouse, the vectors  $N$ ,  $R$ ,  $L$ ,  $V$  are all drawn for the given surface point.

##### **Ray Tracer**

Build a simple ray tracer. Begin with simple scenes. A single reflective sphere that sits on a checkerboard terrain is a good test case. Work towards more complex scenes as time allows.

##### **Fractal Terrain**

Interesting mountain-like terrains are easily generated using a recursive procedure and random numbers. Talk to the instructor or your TAs if you're interested in this.

##### **Particle System**

Use simple physics ( $F=ma$ ) to implement a fireworks simulation.

##### **Another demo of your choosing**

Create an OpenGL scene or animation that illustrates something that you have a personal interest in. Add something interesting to one of the previous assignments.

##### **Driving Game**

Create a world with roads, or perhaps rough terrain without roads. Implement a mouse or keyboard interface for steering your virtual car around in this world. Ideas for optional

features could include collision detection, night-driving mode, having the car leave tracks on the pavement or sand, providing control over the camera point-of-view (first person, third person, etc.), adding other autonomous cars in the world, etc. Be creative and add your own features.

### **Research Report**

Investigate some area of interest to you in further depth. How is computer graphics used to support medical imaging? What are the social implications of being able to easily create videos of 3D scenes that are indistinguishable from reality? Should the goal of computer graphics be to continually achieve better realism, or should it be to achieve new visual styles that would be impossible with conventional film? How has computer graphics influenced the design of objects around us? Include proper citations of reference material.

### **Hand-in Instructions**

You do not have to hand in any printed code. You need only do one submission for a group of two. Create a README file that includes the names and login ID of the group members of your project, and any information you would like to pass on the marker.

Create a folder called 'assn3' under your cs314 directory and put all the source files, your makefile, and your README file there. Also include any images that are used as texture maps. Do not use further sub-directories.

The assignment should be handed in with the exact command:

```
handin cs314 assn3
```

The projects will be marked in part based on a short demo of your project to the instructor or the TA during designated time slots on Dec 4, 8, and 11. Further details on specific times will be announced.