Assignment 5: 3D Viewing and Lighting

ITCS 6120/8120 Computer Graphics
Fall 2012
Revision: 11/26/2012 4:28 PM
Professor Zachary Wartell

Due Date: Wednesday, December 5 2012 11:59PM.

Associated Material:

- Textbook Edition #3: Chapter 5,6,7
- Textbook Edition #4: Chapter 10, Chapter 17

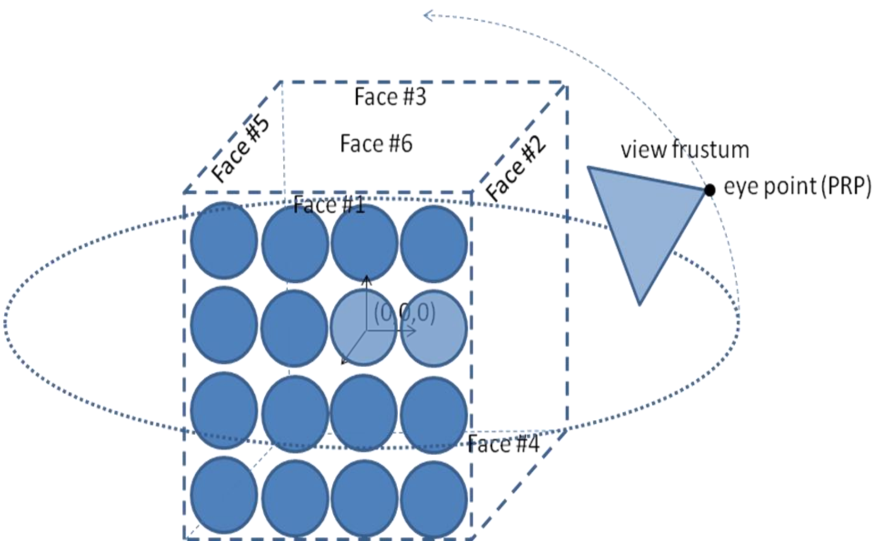
  OR

- OpenGL Programming Guide (Redbook)
  - Chapter 3 Viewing
  - Chapter 5 Lighting
  - Chapter 7 Display Lists

Overview:

The purpose of this assignment is to render a 3D scene in OpenGL consisting of a number of 3D spheres with varying material reflection properties and to implement a simple camera control interaction.

The scene will consists of 96 spheres layout out on the six faces of a cube (the cube itself is not rendered). The camera will look toward the center of the cube and the user can rotate the view up and down and side to side.



1) The UI should allow the view to be rotated around the scene. The view should always look at a fixed point in the center of the scene. Make the fixed point (0,0,0) in world space. The UI can rotate the view up/down and left/right.

- Key's 'a' and 'd' should rotate left/right
- Key's 's' and 'w' rotate up and down.
- EXTRA CREDIT:
    i. (10 points) Moving the mouse upward with the left button pressed should rotate up and down. Moving the mouse sideways with the left button pressed should rotate left/right.
    ii. (5 points) Use the mouse motion with the right button pressed to zoom in and out by moving the camera closer/farther from the look-at point.

2) The scene should contain 4 OpenGL point source lights plus some ambient lighting.

- keys '1' through '4' should turn on and off each of the 4 lights.
- The position of each light should be at some fixed distance along an invisible line from the center of the cube through a corner of the cube. Each light should be positioned on a line through a different corner. You choose the distances along the line; you choose the colors of the lights.

3) The 3D scene should contain 96 spheres. They will be arranged on the faces of an invisible cube with each face containing 16 spheres arranged in 4 x 4 grid. All spheres will be the sam

Formatted: Bullets and Numbering

4) e size. They will vary in their material properties. Some of these properties are set in the requirements below. You can choose the remaining aspects of the

material properties.

- Face #1 (“front”) – Face #1 of 4 x 4 spheres should contain no specular reflection and no diffuse reflection, but only ambient reflection. The ambient reflection coefficient should vary from 0 to 1 as you move across the 4 columns of spheres. The each row should differ in what common base color the spheres in that row are. You choose the colors.
- Face #2 (“right”) – Face #2 of 4 x 4 spheres should contain no specular reflection, no ambient reflection and varying diffuse reflection. The diffuse reflection coefficient should vary from 0 to 1 as you move across the 4 columns of spheres. The each set of spheres in a given row share a common base color but different rows have a different base color. You choose the colors.
- Face #3 (“back”) - Face #3 of 4 x 4 spheres should contain no specular reflection, varying ambient reflection and varying diffuse reflection. The diffuse reflection coefficient should vary from 0 to 1 as you move across the 4 <u>columns</u> of spheres. The ambient reflection coefficient should vary from 0 to 1 as you move up the 4 <u>rows</u> of spheres. <u>All</u> 16 spheres should use the same common base color.
- Face #4 (“bottom”) - Face #4 of 4 x 4 spheres should contain a fixed amount of ambient reflection and fixed amount of diffuse reflection and a varying amount of specular reflection. The specular reflection coefficient should vary from 0 to 1 as you move across the 4 <u>columns</u> of spheres. The specular exponent coefficient should vary from 0 to 1 as you move up across different <u>rows</u> of spheres. <u>All</u> spheres should use the same common base color. (You choose a color). Choose a fixed ambient and diffuse reflection amount so that the specular highlights and there variations are visible.
- Face #5 (“left”) – Face #5 of 4 x 4 spheres should contain a fixed amount of ambient reflection and fixed amount of diffuse reflection and no specular reflection and varying emission coefficients. Pick whatever colors you want. Make at least 4 of the spheres emit their own light. Make one or more of the spheres emission coefficient change in color or intensity continuously.
- Face #6 (“top”) – Face #6 of 4 x 4 spheres should varying in material properties in any way you want (color, diffuse, specular, emission, etc.)

Skeleton Code:

svn copy:

```
https://cci-subv.uncc.edu/svn/zwartell-public/ITCS
x120/trunk/public/OpenGLTrainer v2/Tools and Examples/Cube Skeleton
I
```

to your Project 5 directory. Verify that this skeleton code for Project 5 compiles and executes. (Make sure your Third Party Libraries/OpenGLTrainer is based on "OpenGLTrainer v2").

Details:

Setting up 3D Viewing

Chapter 3 of the OpenGL Programming Guide covers OpenGL 3D viewing and transformations. Chapter 3's cube.c example is a good starting point. This cube.c code discussed in Chapter 3 was copied and re-organized into the Cube Skeleton I example mentioned above.

> **Suggestion**: Use the gluLookAt function. To implement the view rotation the `target point` of the gluLookAt function's view specification is constant while the `eye point` location should move over the surface of a sphere. An easy way to do this is to use the parametric equation for the surface of a sphere (see textbook Appendix A-9) to map two angle values to the x,y,z coordinates of a point on a sphere. The eye point is assigned the x,y,z coordinates computed from the two angle values which are controlled by the user interface—one angle is controlled by the up/down key and the other is controlled by the left/right key.

Add hidden surface elimination

- 3rd Edition textbook, Section 9-14 or 4th Edition textbook 16-14 covers the basics of setting up hidden surface elimination. Alternatively see Chapter 10 of the OpenGL Programming Guide.
  - Enable OpenGL depthbuffering. (see glEnable, GL_DEPTH_BUFFER and GL_DEPTH_BUFFER_BIT; also note that glutInitDisplayMode must be told to allocate the depthbuffer for a GLUT window).
- enable backface culling

Formatted: Bullets and Numbering

Spheres:

- use the gluNewQuadric and gluSphere function to generate the spheres.
- use display lists to store the generated OpenGL geometry (vertex/normals/etc) on the GPU. (Chapter 7 "Displays Lists" of Redbook). This can make a substantial increase in framerate.

Suggested Implementation Sequence:

1. First render 1 sphere without lighting using some fixed color using the gluLookAt for camera setup.
2. Next render and position 16 spheres on just one face of the cube. (Still without lighting using some fixed color).
3. Implement the key-based camera rotation.
4. Create and position one GL light and enable OpenGL lighting
5. Set up OGL materials for the spheres of the one face.
6. Create the other GL lights and added code to turn them on and off
7. Update your code to use Display Lists for the spheres.
8. Set up OGL materials for the spheres for the other 5 faces.
9. REPEAT 7.

Grading:

1. 30% - camera manipulation and rendering off at least one set of unlit spheres.
2. 60% - (split 6 ways, 10% per face) Spheres of each face are rendering with lighting and described in the assignment requirements.