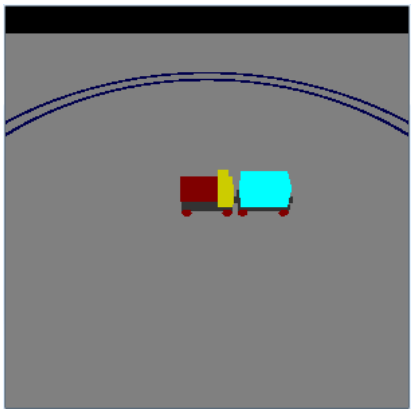
COSC363 Computer Graphics
Lab03: Illumination

Aim:

This lab provides an introduction to different aspects of the OpenGL illumination model. In this lab, you will learn to set up light sources, adjust material properties and transform the light position.

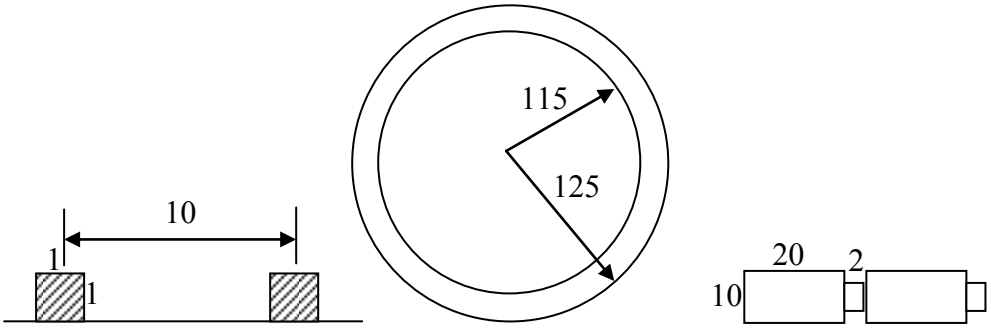
I. Rail.cpp:

1. The program displays a scene consisting of a circular rail track, and the models of a locomotive and a wagon (Fig. (a)). The program includes functions to create these models using a combination of GLUT/GLU objects.



(a)

2. Let us consider some geometrical characteristics of the rail objects created by the program. The rail track has a unit width and a unit height (Fig. (b)). The inner radius of the track is 115 units, and the outer radius 125 (Fig. (c)). The base dimensions of both the engine and the wagons are 20 units x 10 units (Fig. (d)), with a connector length of 2 units.



(b) (c) (d)

3. Please go through the slides [5]-5 to [5]-9 to get an overview of OpenGL functions used for setting up light sources in a scene.

4. Introduce scene illumination by enabling lighting, selecting the light source GL_LIGHT0, and defining light's properties inside the `initialize()` function:

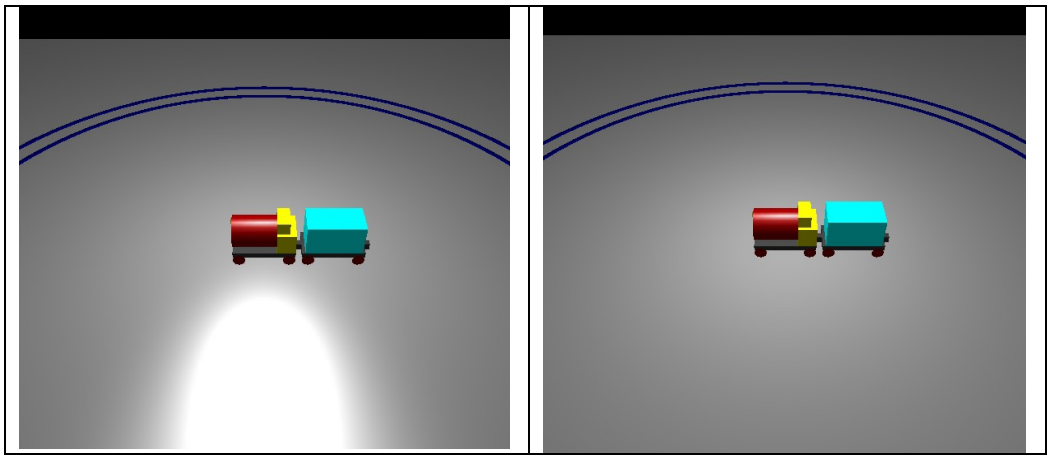
```
float grey[4] = {0.2, 0.2, 0.2, 1.};  //light's ambient color
float white[4] = {1.0, 1.0, 1.0, 1.}; // diffuse and spec color
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);
glLightfv(GL_LIGHT0, GL_AMBIENT, grey);
glLightfv(GL_LIGHT0, GL_DIFFUSE, white);
glLightfv(GL_LIGHT0, GL_SPECULAR, white);
```

   Position the light at the point (0, 50, 0) by adding the statement `glLightfv(GL_LIGHT0, GL_POSITION, lgt_pos);` inside the display() function. Run the program.

5. You will notice that when lighting is enabled, OpenGL ignores previously defined colour values specified using glColor4f(..). Assigning 3-component material properties for each object (as shown on slide [5]-7) could often lead to cumbersome and complex code. One way to overcome this problem is to first define a common specular material property for all surfaces using the function call `glMaterialfv(GL_FRONT, GL_SPECULAR, white);` along with other initialization code. Also assign a shininess factor of 50 for specular highlights. The ambient and diffuse material properties of each object can be made to track the colour values already defined for that object using glColor4f(..). This particular mode of using material colour is enabled by the following statements at the initialization stage:

```
glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
glEnable(GL_COLOR_MATERIAL);
```

   You should now get a display as shown in Fig. (e). The specular component of the floor plane causes a large highlight to appear in the middle of the display window. Inside the `floor()` method, disable specular reflections from the floor by setting its specular material component to black. Reset this component to white after drawing the floor (Fig. (f)).

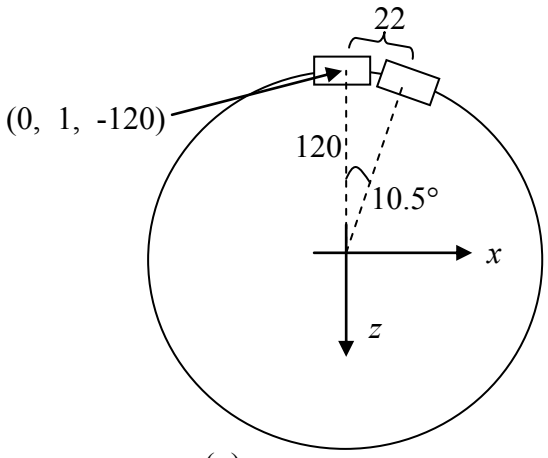


(e) (f)

6. Lighting requires surface normal definitions for each primitive. All GLUT objects have surface normal vectors predefined at each vertex. Go through the

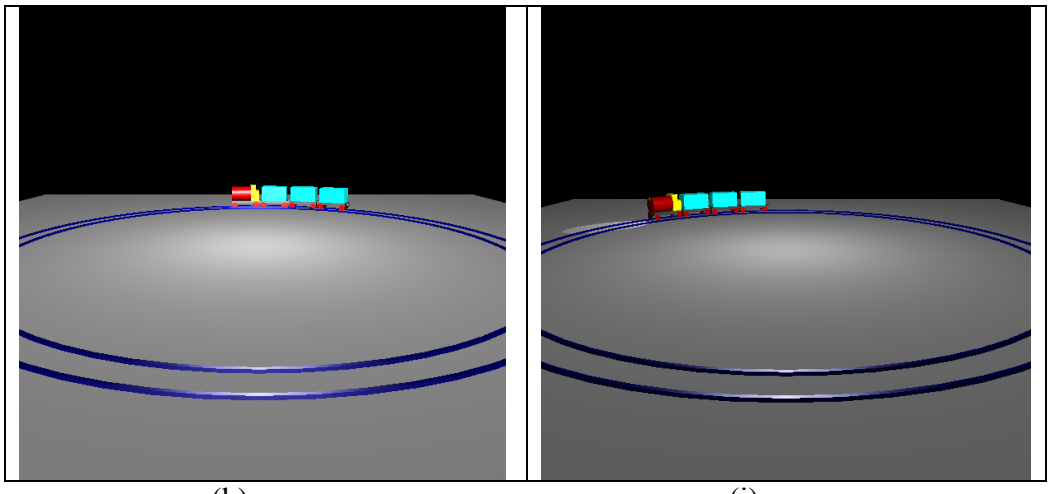
code for the functions `floor()` and `track()` and note the definitions of normal vectors. Also note that the floor plane is tessellated into a large number of small squares.

7. Now, let us place the train on its tracks! Move the rail engine from its current position at the origin to the position (0, 1, –120). The *y*-value 1 corresponds to the track height, and the *z* value 120 corresponds to the mean radius of the tracks (see Fig. (g)). To get the correct orientation of the first wagon on the circular track, you will need to first translate it from origin to (0, 1, –120) and then rotate it about the *y*-axis by 10.5 degrees (positive or negative?). (How was the separation angle 10.5 degs computed?)



(g)

8. Add a few more wagons to the train. Change the camera position to (0, 50, 200) (Fig. (h)). Use a timer call back to move the entire train along the track by applying a continuous rotation about the *y*-axis. Adjust the timer delay and angle step size to get a smooth animation. Note that the program uses double-buffered animation to eliminate screen flicker. This is done by specifying GLUT_DOUBLE in `glutInitDisplayMode()` function, and by calling `glutSwapBuffers()` after drawing the scene.



(h) (i)

9. Create a new light source GL_LIGHT1 with the same parameters as GL_LIGHT0. Convert it to a spotlight by adding the following statements:

```
glLightf(GL_LIGHT1, GL_SPOT_CUTOFF, 30.0);
glLightf(GL_LIGHT1, GL_SPOT_EXPONENT,0.01);
```

   Place the spotlight at (-10, 14, 0). This corresponds to the headlight position on the locomotive. Specify its direction such that it is oriented towards the floor (Fig. (i)). All transformations applied to the locomotive must also be applied to the spotlight, so that it moves as if attached to the engine.

10. Further improvements:

    - Create a train station using GLUT objects and place it in the scene.
    - Update the camera's look direction continuously to track the moving train. You could also try to update the field of view according to the distance of the train from the camera.

II. Quiz-03

The quiz will remain open until **5pm, 28-Mar-2014**.

A quiz can be attempted only once. A question within a quiz may be attempted multiple times. However, a fraction of the marks will be deducted from the second attempt of each question.