SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1712-Graphics and Multimedia Lab

Programming Assignment 10

Creating a 3D Scene in C++ using OpenGL

Name: Jayannthan P T *Dept: CSE 'A'* *Roll No.: 205001049*

Write a C++ program using Opengl to draw atleast 2 3D objects. Apply lighting and texture and render the scene.

OpenGL Functions to use:

glShadeModel()

glMaterialfv()

glLightfv()

glEnable()

glGenTextures()

glTexEnvf()

glBindTexture()

glTexParameteri()

glTexCoord2f()

Source code:

```
#include <GL/glut.h>
#include <Windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
GLfloat black[] = {0.0, 0.0, 0.0, 1.0};
GLfloat white[] = {1.0, 1.0, 1.0, 1.0};
GLfloat direction[] = {1.0, 1.0, 1.0, 0.0};
float teapot_rotate = 0.2, teapot_rotate_direction = 1, teapot_posx = -0.5, teapot_posy =
1.0, teapot_xplace = 0, teapot_yplace = 0;
float teaspoon_posx = 0.75, teaspoon_posy = 2.5, teaspoon_yplace = 0;
float sugar1_posx = 0.65, sugar1_posy = 2.5, sugar2_posx = 0.8, sugar2_posy = 2.75,
sugar1_yplace = 0, sugar2_yplace = 0;
float teacup_rotate = 0;
void display()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);
    glPushMatrix();
    glEnable(GL_TEXTURE_2D);
    glDisable(GL_TEXTURE_2D);
    glPushMatrix();
    GLfloat teacup_color[] = {0.482, 1, 0.161, 0.0};
    GLfloat teacup_mat_shininess[] = {100};
    glMaterialfv(GL_FRONT, GL_DIFFUSE, teacup_color);
    glMaterialfv(GL_FRONT, GL_SHININESS, teacup_mat_shininess);
    glTranslatef(0.75, -0.25, 0.0);
    glRotatef(teacup_rotate, 0, 1, 0);
    glutSolidTeacup(1.0);
    glPopMatrix();
    glPushMatrix();
    GLfloat teapot_color[] = {0.486, 0.212, 0.871, 0.0};
    GLfloat teapot_mat_shininess[] = {100};
    glMaterialfv(GL_FRONT, GL_DIFFUSE, teapot_color);
    glMaterialfv(GL_FRONT, GL_SHININESS, teapot_mat_shininess);
    glTranslatef(teapot_posx, teapot_posy, 0.0);
    glRotatef(teapot_rotate, 0, 0, 1);
    glutSolidTeapot(0.75);
    glPopMatrix();
    GLfloat sugar_color[] = {1, 1, 1, 0.0};
    GLfloat sugar_mat_shininess[] = {50};
    glPushMatrix();
    glMaterialfv(GL_FRONT, GL_DIFFUSE, sugar_color);
    glMaterialfv(GL_FRONT, GL_SHININESS, sugar_mat_shininess);
    glTranslatef(sugar1_posx, sugar1_posy, 0.0);
    glRotatef(-45.0, 0, 0, 1);
    glutSolidCube(0.1);
    glPopMatrix();
    glPushMatrix();
    glMaterialfv(GL_FRONT, GL_DIFFUSE, sugar_color);
    glMaterialfv(GL_FRONT, GL_SHININESS, sugar_mat_shininess);
    glTranslatef(sugar2_posx, sugar2_posy, 0.0);
```

```
    glRotatef(45.0, 0, 0, 1);
    glutSolidCube(0.1);
    glPopMatrix();
    glPushMatrix();
    GLfloat teaspoon_color[] = {0.2, 0.2, 0.2, 0.0};
    GLfloat teaspoon_mat_shininess[] = {100};
    glMaterialfv(GL_FRONT, GL_DIFFUSE, teaspoon_color);
    glMaterialfv(GL_FRONT, GL_SHININESS, teaspoon_mat_shininess);
    glTranslatef(teaspoon_posx, teaspoon_posy, 0.0);
    glRotatef(135, 0, 1, 0);
    glRotatef(-60, 1, 0, 0);
    glutSolidTeaspoon(1.25);
    glPopMatrix();
    if (teapot_rotate_direction == 1 && teapot_rotate > -45.0)
        teapot_rotate -= 0.5;
    if (teapot_rotate_direction == 1 && teapot_rotate <= -45.0)
        teapot_rotate_direction = -1;
    if (teapot_rotate_direction == -1 && teapot_rotate < 0)
        teapot_rotate += 0.5;
    if (teapot_rotate_direction == -1 && teapot_rotate >= 0)
        teapot_rotate_direction = 0;
    teacup_rotate -= 0.2;
    if (teapot_rotate_direction == 0)
    {
        if (teapot_posx > -1.25 && teapot_xplace == 0)
            teapot_posx -= 0.05;
        if (teapot_posx <= -1.25)
            teapot_xplace = 1;
        if (teapot_posy > 0 && teapot_yplace == 0)
            teapot_posy -= 0.05;
        if (teapot_posy <= -1)
            teapot_yplace = 1;
    }
    if (teapot_rotate_direction == 0)
    {
        if (sugar1_posy > -0.5 && sugar1_yplace == 0)
            sugar1_posy -= 0.05;
        if (sugar1_posy <= -0.5)
            sugar1_yplace = 1;
        if (sugar2_posy > -0.5 && sugar2_yplace == 0)
            sugar2_posy -= 0.05;
        if (sugar2_posy <= -0.5)
            sugar2_yplace = 1;
    }
    if (sugar1_yplace == 1 && sugar2_yplace == 1)
    {
        if (teaspoon_posy > -0.25 && teaspoon_yplace == 0)
            teaspoon_posy -= 0.05;
        if (teaspoon_posy <= -0.5)
            teaspoon_yplace = 1;
    }
    glutSwapBuffers();
}
void reshape(GLint w, GLint h)
```

```
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    GLfloat aspect = GLfloat(w) / GLfloat(h);
    glLoadIdentity();
    glOrtho(-2.5, 2.5, -2.5 / aspect, 2.5 / aspect, -10.0, 10.0);
}
void init()
{
    glClearColor(1, 1, 1, 1);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, white);
    glMaterialfv(GL_FRONT, GL_SPECULAR, white);
    glMaterialf(GL_FRONT, GL_SHININESS, 30);
    glLightfv(GL_LIGHT0, GL_AMBIENT, black);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, white);
    glLightfv(GL_LIGHT0, GL_SPECULAR, white);
    glLightfv(GL_LIGHT0, GL_POSITION, direction);
    glEnable(GL_LIGHTING);
    light glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
    depth glShadeModel(GL_FLAT);
    glEnable(GL_TEXTURE_2D);
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
    glTexImage2D(GL_TEXTURE_2D, 0, 3, 2, 2, 0, GL_RGB, GL_UNSIGNED_BYTE, texture);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glRotatef(20.0, 1.0, 0.0, 0.0);
}
void sceneDemo(int v)
{
    glutPostRedisplay();
    glutTimerFunc(1000 / 24, sceneDemo, 0);
}
int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowPosition(80, 80);
    glutInitWindowSize(800, 600);
    glutCreateWindow("Exercise 10");
    glutReshapeFunc(reshape);
    glutDisplayFunc(display);
    glutTimerFunc(1000, sceneDemo, 0);
    init();
    glutMainLoop();
}
```

Output:

Frames: