Department of Computer Science and Engineering

Shivanirudh S G, 185001146, Semester VII

7 October 2021

UCS1712 - Graphics and Multimedia Lab

Exercise 10: 3-D Scene in C++ using OpenGL

Aim:

Write a C++ program using Opengl to draw atleast four 3D objects. Apply lighting and texture and render the scene. Apply transformations to create a simple 3D animation.

Code:

```
#ifndef LOPENGL_H
#define LOPENGL_H

#include <GL/freeglut.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <math.h>
#include <stdio.h>
#include<iostream>
#include<vector>
#include<ctime>
#include <unistd.h>
using namespace std;

#endif
```

```
#ifndef LUTIL_H
#define LUTIL_H

#include "Headers.h"

//Screen Constants
const int SCREEN_WIDTH = 640;
const int SCREEN_HEIGHT = 480;
const int SCREEN_FPS = 60;
const int POINT_SIZE=2;

int INC = 1;

void initGL();

void render(int state);

void keyboardKeys(unsigned char key, int x, int y);

void drawAxes();


#endif
```

```
#include "Signatures.h"

void initGL(){
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glShadeModel(GL_SMOOTH);
    GLfloat light_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };
    GLfloat light_position[] = { 1, 0, 1, 0 };
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glLightfv(GL_LIGHT0, GL_POSITION, light_position);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glEnable(GL_DEPTH_TEST);
}


void render(int state){
    if (state == 0)
        INC = 1;
    else if (state == 10)
        INC = -1;
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(0.0, 1.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    glMatrixMode(GL_MODELVIEW);
    // Cube
    glPushMatrix();
    GLfloat cube_color[] = { 0.85, 0.078, 0.23, 1.0 };
    glMaterialfv(GL_FRONT, GL_DIFFUSE, cube_color);
    glScalef(0.4*state, 0.35*state, 1.0);
    glTranslatef(0.4, -1.0, 0.0);
    glutSolidCube(0.8);
    glPopMatrix();
    // Teapot
    glPushMatrix();
```

```
    GLfloat teapot_color[] = { 0,1,0, 0.0 };
    GLfloat mat_shininess[] = { 100 };
    glMaterialfv(GL_FRONT, GL_DIFFUSE, teapot_color);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    glTranslatef(3.0, 2, 0.0);
    glRotatef(-0.2 * state, 0, 0, 1);
    glutSolidIcosahedron();
    glPopMatrix();
    // Ramp
    glPushMatrix();
    GLfloat ramp_color[] = { 0, 0.74, 1, 1.0 };
    glMaterialfv(GL_FRONT, GL_DIFFUSE, ramp_color);
    glTranslatef(-1.2, -0.2*state, 0);
    glutSolidTorus(0.3, 0.7, 10, 10);
    glPopMatrix();

    glPushMatrix();
    GLfloat ball_color[] = { 0.29, 0, 0.50, 1.0 };
    glMaterialfv(GL_FRONT, GL_DIFFUSE, ball_color);
    glRotatef(-0.1 * state, 0, 0, 1);
    glTranslatef(-2.5 - 0.25*state, -2, 0);
    glutSolidRhombicDodecahedron();
    glPopMatrix();
    glutSwapBuffers();
    glutTimerFunc(1000 / 30, render, state + INC);
}
```

```
#include "Helpers.h"

void reshape(int w, int h);
void runMainLoop();


int main(int argc, char *args[])
{

    glutInit( &argc, args );

    glutInitContextVersion( 2, 1 );

    glutInitDisplayMode( GLUT_SINGLE|GLUT_RGB );
    glutInitWindowSize( SCREEN_WIDTH, SCREEN_HEIGHT );
    glutCreateWindow( "OpenGL" );

    initGL();
    glutDisplayFunc(runMainLoop);
    glutReshapeFunc(reshape);

    glutMainLoop();

    return 0;
}

void reshape(int w, int h) {
    glViewport(0, 0, (GLsizei)w, (GLsizei)h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(75, 1, 1, 20);
```

```
32      glMatrixMode(GL_MODELVIEW);
33  }
34  void runMainLoop() {
35
36      glutTimerFunc(1000 / 60, render, 0);
37  }
```

Output: