**UCS1712 – GRAPHICS AND MULTIMEDIA LAB**

**Ex. No.10 Creating a 3D Scene in C++ using OpenGL**

**Date:** 4/10/21 **Name:** Srinath S

**Class:** CSE-C  **Roll:** 185001205

**Question:**

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.

[Use built-in transformation functions]

OpenGL Functions to use: glShadeModel() glMaterialfv() glLightfv() glEnable() glGenTextures() glTexEnvf() glBindTexture() glTexParameteri() glTexCoord2f()

**Code:**

#include <GL/glut.h>

#include <GL/glu.h>

#include <stdlib.h>

#include <stdio.h>

int increment = 1;

void initialize(void)

{

glClearColor(1.0, 1.0, 1.0, 0.0);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_DEPTH\_TEST);

}

void draw\_objects(int state)

{

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

if (state == 0) increment = 1;

else if (state == 10) increment = -1;

glLoadIdentity();

gluLookAt(0.0, 1.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

glMatrixMode(GL\_MODELVIEW);

glPushMatrix();

GLfloat ball\_color[] = {

0.59,

0.1,

0.55,

1.0

};

double scale = 0.15;

glTranslatef(-1, 2.5 + scale \* state, 0);

/\*glutSolidSphere(0.5, 10, 10);\*/

/\*glutSolidCube(1.0);\*/

/\*glutSolidTeapot(0.7);\*/

glutSolidTorus(2.0, 4.0, 5, 10);

glPopMatrix();

glutSwapBuffers();

glutTimerFunc(1000 / 60, draw\_objects, state + increment);

}

void threedanim() {

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glutTimerFunc(1000 / 60, draw\_objects, 0);

}

void reshape(int width, int height)

{

glViewport(0, 0, (GLsizei)width, (GLsizei)height);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(75, 1, 1, 20);

glMatrixMode(GL\_MODELVIEW);

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutCreateWindow("3D Animations");

initialize();

glutDisplayFunc(threedanim);

glutReshapeFunc(reshape);

glutMainLoop();

return 0;

}

**Output:**



