**Lab Exercise 10: Creating a 3D Scene in C++ using OpenGL**

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]

CODE:

#include<gl/glut.h>

#include<iostream>

#include<vector>

using namespace std;

GLfloat alpha = 300, theta = 300, gamma = 300;

int index=0;

bool rev=false;

vector<vector<GLfloat>> coords(8, vector<GLfloat>(3));

GLfloat light\_position[] = { 1.0, 1.0, 1.0, 0.0 };

void init(void)

{

GLfloat mat\_specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat mat\_shininess[] = { 50.0 };

glClearColor(0.0, 0.0, 0.0, 0.0);

glShadeModel(GL\_SMOOTH);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, mat\_specular);

glMaterialfv(GL\_FRONT, GL\_SHININESS, mat\_shininess);

glLightfv(GL\_LIGHT0, GL\_POSITION, light\_position);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

glEnable(GL\_DEPTH\_TEST);

}

void display(void)

{

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

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(100, 1, 0.1, 10000);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glRotatef(30, 0, 1, 0);

gluLookAt(gamma, alpha, theta, -500, 0, -500, 0, 1, 0);

glutSolidCube(100);

gluLookAt(gamma, alpha, theta, 500, 0, -500, 0, 1, 0);

glutSolidTeapot(100);

gluLookAt(gamma, alpha, theta, -500, 0, 500, 0, 1, 0);

glutSolidSphere(100,20,20);

gluLookAt(gamma, alpha, theta, 500, 0, 500, 0, 1, 0);

glutSolidTorus(50,100,20,20);

gluLookAt(gamma, alpha, theta, 1000, 0, 0, 0, 1, 0);

glFlush();

}

void timer(int v)

{

if (!rev) {

alpha += 1;

theta += 1;

gamma += 1;

if (alpha == 500) rev = true;

}

else {

alpha -= 1;

theta -= 1;

gamma -= 1;

if (alpha == 200) rev = false;

}

glutPostRedisplay();

glutTimerFunc(10, timer, v);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB | GLUT\_DEPTH);

glutInitWindowSize(1000, 1000);

glutCreateWindow("3D Scene");

init();

glutDisplayFunc(display);

glutTimerFunc(10,timer,0);

glutMainLoop();

return 0;

}

OUTPUT:





