**UCS 1712 – GRAPHICS AND MULTIMEDIA LAB**

**ASSIGNMENT – 10**

**VISHAL N**

**185001198**

**28.10.2021 CSEC**

**1. CREATE 3D SCENE:**

#pragma warning(disable : 4996)

#include <GL/glut.h>

#include <GL/glu.h>

#include <stdlib.h>

#include <stdio.h>

int INC = 1;

void initialize(void) {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glShadeModel(GL\_SMOOTH);

    GLfloat light\_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };

    GLfloat light\_position[] = { 0, 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);

}

GLuint LoadTexture(const char\* filename) {

    GLuint texture;

    int width, height;

    unsigned char\* data;

    FILE\* file;

    file = fopen(filename, "rb");

    if (file == NULL) return 0;

    width = 474;

    height = 395;

    data = (unsigned char\*)malloc(width \* height \* 3);

    fread(data, width \* height \* 3, 1, file);

    fclose(file);

    for (int i = 0; i < width \* height; ++i) {

        int index = i \* 3;

        unsigned char B, R;

        B = data[index];

        R = data[index + 2];

        data[index] = R;

        data[index + 2] = B;

    }

    glGenTextures(1, &texture);

    glBindTexture(GL\_TEXTURE\_2D, texture);

    glTexEnvf(GL\_TEXTURE\_ENV, GL\_TEXTURE\_ENV\_MODE, GL\_MODULATE);

    glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER,

        GL\_LINEAR\_MIPMAP\_NEAREST);

    glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER,

        GL\_LINEAR);

    glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_S, GL\_REPEAT);

    glTexParameterf(GL\_TEXTURE\_2D, GL\_TEXTURE\_WRAP\_T, GL\_REPEAT);

    gluBuild2DMipmaps(GL\_TEXTURE\_2D, 3, width, height, GL\_RGB,

        GL\_UNSIGNED\_BYTE, data);

    free(data);

    return texture;

}

void drawScene(int state) {

    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);

    glPushMatrix();

    GLfloat cube\_color[] = { 0.26, 0.46, 0.7, 1.0 };

    glMaterialfv(GL\_FRONT, GL\_DIFFUSE, cube\_color);

    glScalef(4, 1.5, 1.0);

    glTranslatef(0.4, -1.0, 0.0);

    glutSolidCube(1.0);

    glPopMatrix();

    glPushMatrix();

    glEnable(GL\_TEXTURE\_2D);

    GLfloat teapot\_color[] = { 0.9, 0.2, 0.9, 0.0 };

    GLfloat mat\_shininess[] = { 10 };

    glMaterialfv(GL\_FRONT, GL\_DIFFUSE, teapot\_color);

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

    glScalef(2, 2, 2);

    glTranslatef(1.1, 0.25, 0.0);

    glutSolidTeapot(0.7);

    glDisable(GL\_TEXTURE\_2D);

    glPopMatrix();

    glPushMatrix();

    GLfloat ramp\_color[] = { 0.8, 0.34, 0.19, 1.0 };

    mat\_shininess[0] = 100;

    glMaterialfv(GL\_FRONT, GL\_DIFFUSE, ramp\_color);

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

    glRotatef(0, 0, 1, 1);

    glTranslatef(0.0, -2.4, 0);

    glScalef(10.0, 0.2, 1.9);

    glutSolidCube(1.0);

    glPopMatrix();

    glPushMatrix();

    GLfloat ball\_color[] = { 0.3, 0.8, 0.2, 0.1 };

    glMaterialfv(GL\_FRONT, GL\_DIFFUSE, ball\_color);

    glRotatef(-0.1, 0, 0, 1);

    glTranslatef(-2.5 - 0.25, -2, 0);

    glutSolidSphere(0.5, 10, 10);

    glPopMatrix();

    glutSwapBuffers();

    glutTimerFunc(1000 / 60, drawScene, state + INC);

}

void reshape(int w, int h) {

    glViewport(0, 0, (GLsizei)w, (GLsizei)h);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluPerspective(75, 1, 1, 20);

    glMatrixMode(GL\_MODELVIEW);

}

void sceneDemo() {

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

    glutTimerFunc(1000 / 60, drawScene, 0);

}

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

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

    glutInitWindowSize(500, 500);

    glutCreateWindow("Create 3D Scene");

    initialize();

    glutDisplayFunc(sceneDemo);

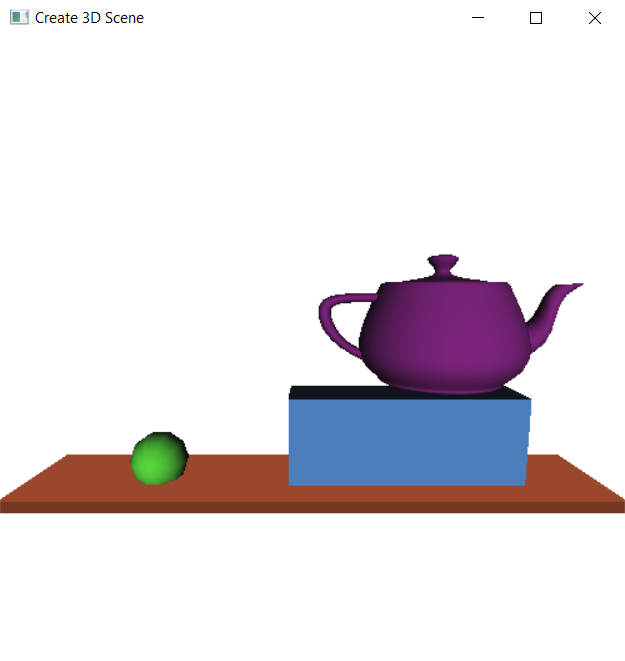
    glutReshapeFunc(reshape);

    glutMainLoop();

    return 0;

}

**OUTPUTS:**

****