Experiment Name: Lighting in Open GL

Introduction:This opengl project make lighting.

Code:

#include <iostream>

#include <stdlib.h>

#include<windows.h>

#ifdef \_\_APPLE\_\_

#include <OpenGL/OpenGL.h>

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

using namespace std;

//Called when a key is pressed

void handleKeypress(unsigned char key, int x, int y) {

switch (key) {

case 27: //Escape key

exit(0);

}

}

//Initializes 3D rendering

void initRendering() {

glEnable(GL\_DEPTH\_TEST);

glEnable(GL\_COLOR\_MATERIAL);

glEnable(GL\_LIGHTING); //Enable lighting

//you can have upto 8 lighting

glEnable(GL\_LIGHT0); //Enable light #0

glEnable(GL\_LIGHT1); //Enable light #1

glEnable(GL\_NORMALIZE); //Automatically normalize normals

//glShadeModel(GL\_SMOOTH); //Enable smooth shading

}

//Called when the window is resized

void handleResize(int w, int h) {

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluPerspective(45.0, (double)w / (double)h, 1.0, 200.0);

}

float \_angle = -70.0f;

//Draws the 3D scene

void drawScene() {

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

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

glTranslatef(0.0f, 0.0f, -8.0f);

GLfloat ambientColor[] = {0.2f, 0.2f, 0.2f, 1.0f}; //Color (0.2, 0.2, 0.2) and intensity //can be greater than 1 so not like color

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, ambientColor);

//Add positioned light

GLfloat lightColor0[] = {0.5f, 0.5f, 0.5f, 1.0f}; //Color (0.5, 0.5, 0.5)

GLfloat lightPos0[] = {4.0f, 0.0f, 8.0f, 1.0f}; //Positioned at (4, 0, 8)

glLightfv(GL\_LIGHT0, GL\_DIFFUSE, lightColor0);

glLightfv(GL\_LIGHT0, GL\_POSITION, lightPos0);

//Add directed light

GLfloat lightColor1[] = {0.5f, 0.2f, 0.2f, 1.0f}; //Color (0.5, 0.2, 0.2)

//Coming from the direction (-1, 0.5, 0.5)

// 0 because direced light source

GLfloat lightPos1[] = {-1.0f, 0.5f, 0.5f, 0.0f};

glLightfv(GL\_LIGHT1, GL\_DIFFUSE, lightColor1);

glLightfv(GL\_LIGHT1, GL\_POSITION, lightPos1);

glRotatef(\_angle, 1.0f, 0.0f, 0.0f);

glColor3f(1.0f, 1.0f, 0.0f);

glBegin(GL\_QUADS);

glNormal3f(0.0f, 0.0f, 1.0f);

//glNormal3f(-1.0f, 0.0f, 1.0f);

glVertex3f(-1.5f, -1.0f, 1.5f);

//glNormal3f(1.0f, 0.0f, 1.0f);

glVertex3f(1.5f, -1.0f, 1.5f);

//glNormal3f(1.0f, 0.0f, 1.0f);

glVertex3f(1.5f, 1.0f, 1.5f);

//glNormal3f(-1.0f, 0.0f, 1.0f);

glVertex3f(-1.5f, 1.0f, 1.5f);

//Right

glNormal3f(1.0f, 0.0f, 0.0f);

//glNormal3f(1.0f, 0.0f, -1.0f);

glVertex3f(1.5f, -1.0f, -1.5f);

//glNormal3f(1.0f, 0.0f, -1.0f);

glVertex3f(1.5f, 1.0f, -1.5f);

//glNormal3f(1.0f, 0.0f, 1.0f);

glVertex3f(1.5f, 1.0f, 1.5f);

//glNormal3f(1.0f, 0.0f, 1.0f);

glVertex3f(1.5f, -1.0f, 1.5f);

//Back

glNormal3f(0.0f, 0.0f, -1.0f);

//glNormal3f(-1.0f, 0.0f, -1.0f);

glVertex3f(-1.5f, -1.0f, -1.5f);

//glNormal3f(-1.0f, 0.0f, -1.0f);

glVertex3f(-1.5f, 1.0f, -1.5f);

//glNormal3f(1.0f, 0.0f, -1.0f);

glVertex3f(1.5f, 1.0f, -1.5f);

//glNormal3f(1.0f, 0.0f, -1.0f);

glVertex3f(1.5f, -1.0f, -1.5f);

//Left

glNormal3f(-1.0f, 0.0f, 0.0f);

//glNormal3f(-1.0f, 0.0f, -1.0f);

glVertex3f(-1.5f, -1.0f, -1.5f);

//glNormal3f(-1.0f, 0.0f, 1.0f);

glVertex3f(-1.5f, -1.0f, 1.5f);

//glNormal3f(-1.0f, 0.0f, 1.0f);

glVertex3f(-1.5f, 1.0f, 1.5f);

//glNormal3f(-1.0f, 0.0f, -1.0f);

glVertex3f(-1.5f, 1.0f, -1.5f);

glEnd();

glutSwapBuffers();

}

void update(int value) {

\_angle += 1.5f;

if (\_angle > 360) {

\_angle -= 360;

}

glutPostRedisplay();

glutTimerFunc(25, update, 0);

}

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

//Initialize GLUT

glutInit(&argc, argv);

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

glutInitWindowSize(400, 400);

//Create the window

glutCreateWindow("Lighting ");

initRendering();

//Set handler functions

glutDisplayFunc(drawScene);

glutKeyboardFunc(handleKeypress);

glutReshapeFunc(handleResize);

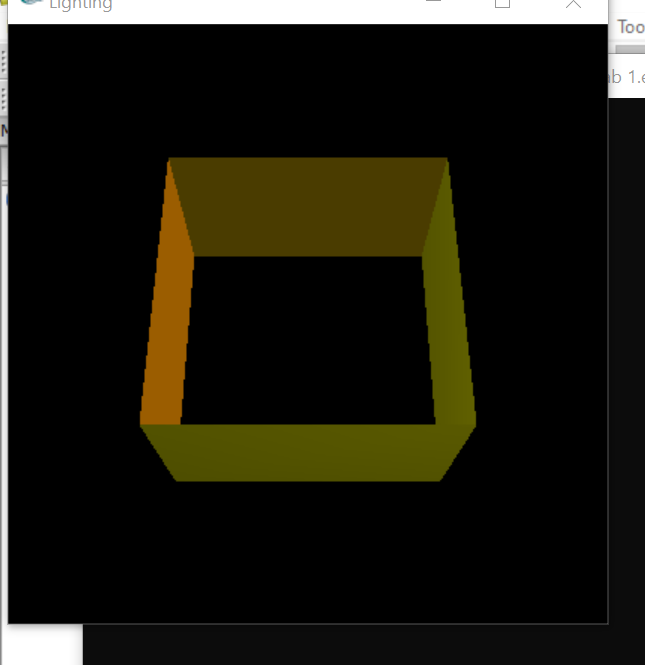
glutTimerFunc(25, update, 0); //Add a timer

glutMainLoop();

return 0;

}

Output:



Conculationthis site I use some function glutInit (&argc, argv).the window size using the function glutInitWindowSize(400,400).Set the starting position for the window using the function glutInitWindowPosition (0, 0).Initialize the window and the title using the function glutCreateWindow(“lighting’’). Clear the screen using the function glClear(GL\_COLOR\_BUFFER\_BIT). //you can have upto 8 lighting Enable light #0 to 8 glEnable(GL\_LIGHTING); Enable lighting. glEnable(GL\_NORMALIZE);.for automatic lighting system.