ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | И. М. Лозоватский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Работа с трехмерными объектами |
| по курсу: КОМПЬЮТЕРНАЯ ГРАФИКА |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134к |  |  |  | Костяков Н.А. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Цель работы**

Создать приложение, выводящее объемный объект средствами Open GL

**Ход выполнения**

**Листинг**

#define FREEGLUT\_STATIC

#include <GL/glut.h>

#include <math.h>

void draw();

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

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

glutInit(&argc, argv);

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

glutInitWindowSize(400, 400);

glutInitWindowPosition(500, 200);

glutCreateWindow("Shape");

glClearColor(0, 0, 0, 0);

glutKeyboardFunc(key);

glutDisplayFunc(draw);

glutMainLoop();

return 0;

}

void draw() {

glClear(GL\_COLOR\_BUFFER\_BIT);

GLfloat cubeVertexArray[8][3];

GLfloat cubeColorArray[8][3];

GLubyte cubeIndexArray[6][4];

//вершины

cubeVertexArray[0][0] = -0.5f;

cubeVertexArray[0][1] = - 0.5f;

cubeVertexArray[0][2] = 0.5f;

cubeVertexArray[1][0] = - 0.5f;

cubeVertexArray[1][1] = 0.5f ;

cubeVertexArray[1][2] = 0.5f;

cubeVertexArray[2][0] = 0.5f;

cubeVertexArray[2][1] = 0.5f;

cubeVertexArray[2][2] = 0.5f;

cubeVertexArray[3][0] = 0.5f;

cubeVertexArray[3][1] =- 0.5f;

cubeVertexArray[3][2] = 0.5f;

cubeVertexArray[4][0] = - 0.5f;

cubeVertexArray[4][1] = - 0.5f;

cubeVertexArray[4][2] = - 0.5f;

cubeVertexArray[5][0] = - 0.5f;

cubeVertexArray[5][1] = 0.5f;

cubeVertexArray[5][2] = - 0.5f;

cubeVertexArray[6][0] = 0.5f;

cubeVertexArray[6][1] = 0.5f;

cubeVertexArray[6][2] = - 0.5f;

cubeVertexArray[7][0] =0.5f;

cubeVertexArray[7][1] = - 0.5f;

cubeVertexArray[7][2] = - 0.5f;

//цвет вершины

cubeColorArray[0][0] = 0.0;

cubeColorArray[0][1] = 0.0;

cubeColorArray[0][2] = 1.0;

cubeColorArray[1][0] = 0.6;

cubeColorArray[1][1] = 0.98;

cubeColorArray[1][2] = 0.6;

cubeColorArray[2][0] = 1.0;

cubeColorArray[2][1] = 0.84;

cubeColorArray[2][2] = 0.8;

cubeColorArray[3][0] = 0.8;

cubeColorArray[3][1] = 0.36;

cubeColorArray[3][2] = 0.36;

cubeColorArray[4][0] = 1.0;

cubeColorArray[4][1] = 0.27;

cubeColorArray[4][2] = 0.0;

cubeColorArray[5][0] = 0.82;

cubeColorArray[5][1] = 0.13;

cubeColorArray[5][2] = 0.56;

cubeColorArray[6][0] = 0.54;

cubeColorArray[6][1] = 0.17;

cubeColorArray[6][2] = 0.89;

cubeColorArray[7][0] = 0.0;

cubeColorArray[7][1] = 1.0;

cubeColorArray[7][2] = 1.0;

//грани

cubeIndexArray[0][0] = 0;

cubeIndexArray[0][1] = 3;

cubeIndexArray[0][2] = 2;

cubeIndexArray[0][3] = 1;

cubeIndexArray[1][0] = 0;

cubeIndexArray[1][1] = 1;

cubeIndexArray[1][2] = 5;

cubeIndexArray[1][3] = 4;

cubeIndexArray[2][0] = 7;

cubeIndexArray[2][1] = 4;

cubeIndexArray[2][2] = 5;

cubeIndexArray[2][3] = 6;

cubeIndexArray[3][0] = 3;

cubeIndexArray[3][1] = 7;

cubeIndexArray[3][2] = 6;

cubeIndexArray[3][3] = 2;

cubeIndexArray[4][0] = 1;

cubeIndexArray[4][1] = 2;

cubeIndexArray[4][2] = 6;

cubeIndexArray[4][3] = 5;

cubeIndexArray[5][0] = 0;

cubeIndexArray[5][1] = 4;

cubeIndexArray[5][2] = 7;

cubeIndexArray[5][3] = 3;

glEnableClientState(GL\_VERTEX\_ARRAY);

glEnableClientState(GL\_COLOR\_ARRAY);

glVertexPointer(3, GL\_FLOAT, 0, cubeVertexArray);

glColorPointer(3, GL\_FLOAT, 0, cubeColorArray);

glDrawElements(GL\_QUADS, 24, GL\_UNSIGNED\_BYTE, cubeIndexArray);

glutSwapBuffers();

glutPostRedisplay();

}

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

switch (key) {

case 'w': {

glRotatef(3, 1, 0.0f, 0.0f);

break;

}

case 's': {

glRotatef(-3, 1, 0.0f, 0.0f);

break;

}

case 'a': {

glRotatef(3, 0, 1, 0.0f);

break;

}

case 'd': {

glRotatef(-3, 0, 1, 0.0f);

break;

}

case 'e': {

glRotatef(-3, 0, 0, 1);

break;

}

case 'q': {

glRotatef(3, 0, 0, 1);

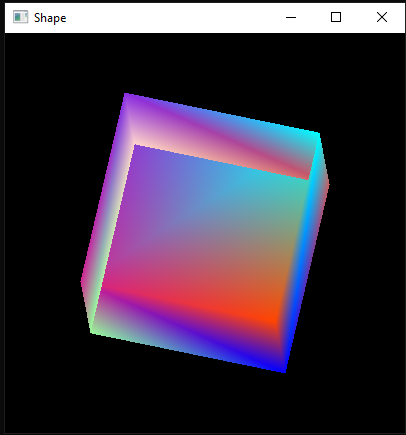
break;

}

}

}

**Результат работы программы**



**Вывод**

Я изучил основы работы с библиотекой opengl и вывел куб