ГУАП

КАФЕДРА №41

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| доцент, кодн.тех.наук  должность, уч. степень, звание |  | подпись, дата |  | Щёкин  инициалы, фамилия |

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №

ОПРЕДЕЛЕНИЕ ПОПАДАНИЯ ТОЧКИ В ОБЛАСТЬ

по курсу: КОМПЬЮТЕРНАЯ ГРАФИКА

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | | 4333K |  |  |  | Герасимов С.А. |
|  | номер группы | |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург

2023

**Задание:**

Лабораторная работа №2

Работа с трехмерными объектами

Создать приложение, выводящее объемный объект средствами Open GL.

Среда программирования – Visual C++ или QT Creator c MinGW и Open GL

**Текст программы:**

#include <GLFW/glfw3.h>

#include <unistd.h>

#include <GLUT/glut.h>

#include <OpenGL/gl.h>

#include <math.h>

typedef struct \_POINTFLOAT {

float x;

float y;

float z;

} POINTFLOAT, \*PPOINTFLOAT;

void showObj(){

float colors[] = {

0, 0, 0,

0, 0, 1,

0, 1, 0,

0, 1, 1,

1, 0, 0,

1, 0, 1

};

POINTFLOAT verties[] = {

{0, 0, 2},

{1, 1, 0},

{1, -1, 0},

{-1, -1, 0},

{-1, 1, 0},

{1, 1, 0},

};

glPushMatrix();

glVertexPointer(3, GL\_FLOAT, 0, &verties);

glColorPointer(3, GL\_FLOAT, 0, &colors);

glEnableClientState(GL\_VERTEX\_ARRAY);

glEnableClientState(GL\_COLOR\_ARRAY);

//glColor3f(0.5f, 0.5f, 0.5f);

glDrawArrays(GL\_TRIANGLE\_FAN, 0, 6);

glDisableClientState(GL\_COLOR\_ARRAY);

glDisableClientState(GL\_VERTEX\_ARRAY);

glPopMatrix();

}

int main(void)

{

float alpha = 0.0f;

int height = 700;

int weith = 700;

GLFWwindow\* window;

/\* Initialize the library \*/

if (!glfwInit())

return -1;

/\* Create a windowed mode window and its OpenGL context \*/

window = glfwCreateWindow(height, weith, "Hello World", NULL, NULL);

if (!window)

{

glfwTerminate();

return -1;

}

/\* Make the window's context current \*/

glfwMakeContextCurrent(window);

glLoadIdentity();

//glOrtho(-2, 2, -2, 2, -1, 1); // ортоганальная проекция

glFrustum(-1, 1, -1, 1, 2, 80); // перспективная проекция

glTranslatef(0, -1, -5);

glRotated(-90, 1, 0, 0);

/\* Loop until the user closes the window \*/

while (!glfwWindowShouldClose(window))

{

glClearColor(0.7f, 1.0f, 0.7f, 0.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

showObj();

glRotatef(1, 0, 0, 1);

glfwPollEvents();

glfwSwapBuffers(window);

usleep(1);

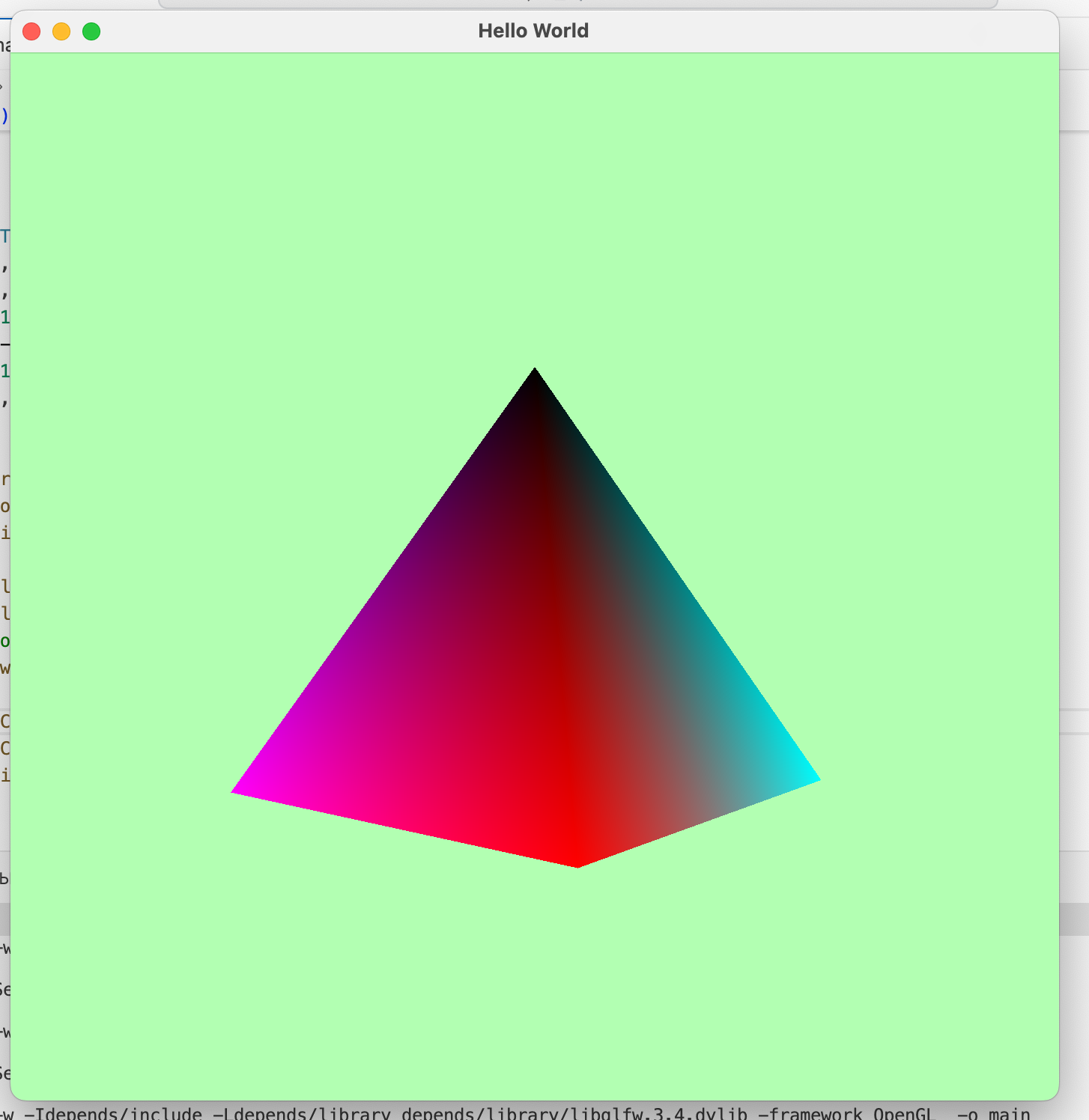
}

glfwTerminate();

return 0;

}

**Снимок экрана:**

****