МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

кандидат технических наук, доцент С.В.Щекин должность, уч. степень, звание подпись, дата инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

№2

**«Работа с источником света и свойствами материала поверхности»**

по дисциплине: КОМПЬЮТЕРНАЯ ГРАФИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. Z7431 20.12.2019 М.Д.Семочкин

подпись, дата инициалы, фамилия

Студ. билет 2014/1054

Санкт-Петербург

2019

**Содержание**

1. Титульный лист
2. Содержание
3. Задание
4. Используемые инструменты разработки
5. Описание программы
6. Текст программы
7. Результат работы программы
8. Вывод
9. Список использованной литературы

**Задание**

Включить источник света, задать отражающие свойства поверхностей: диффузное, зеркальное отражение, цвет поверхности.

Вывести несколько объемных объектов. Каждый из них должен иметь различные свойства поверхностей (доминирующее зеркальное или диффузное отражение, различный цвет внешних поверхностей).

**Используемые инструменты разработки**

IDE – Microsoft Visual C++ 2010 Express

Библиотека для работы с openGL - freeglut 3.0.0 MinGW

ОС – Windows 7 x64

**Описание программы**

Написанная программа рисует три сферы, каждая из которых имеет различные параметры:

1. Только диффузное отражение, синий цвет
2. Диффузное и доминирующее зеркальное отражение, зеленый цвет
3. Диффузное и зеркальное отражение, зеленый цвет

**Текст программы**

// Программа рисует три сферы с разными параметрами поверхности

#include <stdlib.h>

#include <GL/freeglut.h>

void init(void) {

GLfloat ambient[] = { 0.0, 0.0, 0.0, 1.0 };

GLfloat diffuse[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat specular[] = { 1.0, 1.0, 1.0, 1.0 };

GLfloat position[] = { 0.0, 3.0, 2.0, 0.0 };

GLfloat lmodel\_ambient[] = { 0.4, 0.4, 0.4, 1.0 };

GLfloat local\_view[] = { 0.0 };

glClearColor(5.0, 5.0, 5.0, 0.0);

glEnable(GL\_DEPTH\_TEST);

glShadeModel(GL\_SMOOTH);

glLightfv(GL\_LIGHT0, GL\_AMBIENT, ambient);

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

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

glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, lmodel\_ambient);

glLightModelfv(GL\_LIGHT\_MODEL\_LOCAL\_VIEWER, local\_view);

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT0);

}

void display(void) {

GLfloat no\_mat[] = { 0.0, 0.0, 0.0, 1.0 };

GLfloat mat\_ambient[] = { 0.7, 0.7, 0.7, 1.0 };

GLfloat mat\_ambient\_color[] = { 0.1, 0.8, 0.2, 1.0 };

GLfloat mat\_diffuse[] = { 0.1, 0.9, 0.8, 1.0 };

GLfloat mat\_diffuse\_2[] = { 0.1, 0.2, 0.8, 1.0 };

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

GLfloat no\_shininess[] = { 0.0 };

GLfloat low\_shininess[] = { 5.0 };

GLfloat high\_shininess[] = { 100.0 };

GLfloat mat\_emission[] = {0.3, 0.2, 0.2, 0.0};

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

// Только диффузное отражение

glPushMatrix();

glTranslatef (-3.75, 3.0, 0.0);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, no\_mat);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse\_2);

glMaterialfv(GL\_FRONT, GL\_SPECULAR, no\_mat);

glMaterialfv(GL\_FRONT, GL\_SHININESS, no\_shininess);

glMaterialfv(GL\_FRONT, GL\_EMISSION, no\_mat);

glutSolidSphere(1.0, 16, 16);

glPopMatrix();

// Доминирующее зеркальное отражение

glPushMatrix();

glTranslatef (-1.25, 3.0, 0.0);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, no\_mat);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);

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

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

glMaterialfv(GL\_FRONT, GL\_EMISSION, no\_mat);

glutSolidSphere(1.0, 16, 16);

glPopMatrix();

// Диффузное и зеркальное отражение

glPushMatrix();

glTranslatef (1.25, 3.0, 0.0);

glMaterialfv(GL\_FRONT, GL\_AMBIENT, no\_mat);

glMaterialfv(GL\_FRONT, GL\_DIFFUSE, mat\_diffuse);

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

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

glMaterialfv(GL\_FRONT, GL\_EMISSION, no\_mat);

glutSolidSphere(1.0, 16, 16);

glPopMatrix();

glFlush();

}

void reshape(int w, int h) {

glViewport(0, 0, w, h);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho (-6.0, 6.0, -3.0\*((GLfloat)h\*2)/(GLfloat)w, 3.0\*((GLfloat)h\*2)/(GLfloat)w, -10.0, 10.0);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

}

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

glutInit(&argc, argv);

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

glutInitWindowSize (600, 600);

glutCreateWindow(argv[0]);

init();

glutReshapeFunc(reshape);

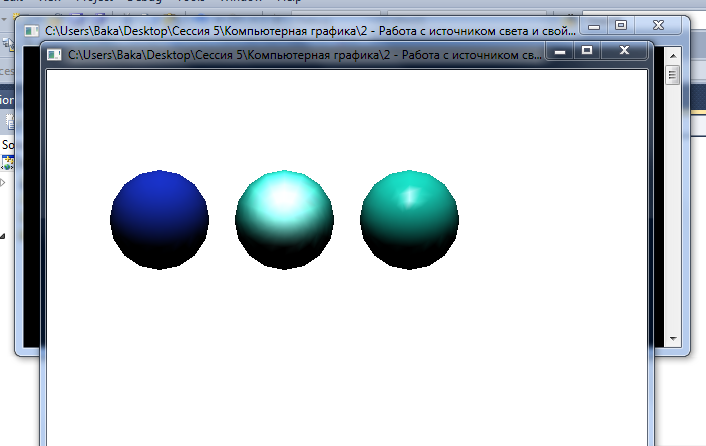
glutDisplayFunc(display);

glutMainLoop();

return 0;

}

**Результат работы программы**



**Вывод**

Во время выполнения лабораторной работы были получены навыки использования источника освещения, использования различных типов поверхности объемных фигур с помощью OpenGL при написании программ на C++.

**Список использованной литературы**

1. Эйнджел Э. Интерактивная компьютерная графика. Вводный курс на базе
2. OpenGL - М.: Вильямс, 2001. - 592 с.
3. Хирн, Бейкер. Компьютерная графика и стандарт Open GL – М.: Вильямс, 2005, 1168 c.
4. Роджерс Д. Алгоритмические основы машинной графики- М.: Мир, 1989. – 512 с.
5. Земсков Ю.В. Qt 4 на примерах. - СПб.: БХВ - Петербург, 2008. - 608 с. : рис. + эл. опт. диск (CD-ROМ)