МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ПРОГРАММИРОВАНИЯ

КАФЕДРА компьютерных технологий и программной инженерии

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  |  |  | Н.А. Солоевьева |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| «Работа с источником света и свойствами материала поверхности.» |
| по курсу: Компьютерная графика |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4932 |  |  |  | А.И.Белов |
|  |  |  | подпись, дата |  | инициалы, фамилия |

**Цель работы:** 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).

**Листинг:**

**#include <Windows.h>**

**#include <glut.h>**

**#include <ctime>**

**#include <iostream>**

**using namespace std;**

**void draw(void)**

**{**

**glShadeModel(GL\_SMOOTH);**

**GLfloat front\_color[] = { 0,1,0,1 };**

**GLfloat back\_color[] = { 0,0,1,1 };**

**glBegin(GL\_TRIANGLE\_STRIP);**

**glColor3ub(200, 200, 200);**

**float MatDiffuse1[] = { 1.0f, 0.0f, 0.0f, 0.0f };**

**float MatSpecular1[] = { 0.0f, 0.0f, 0.0f, 0.0f };**

**float MatShininess1 = 0.0f;**

**float MatEmission1[] = { 1.0f, 1.0f, 1.0f, 1.0f };**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_DIFFUSE, MatDiffuse1); //рассеянный свет**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SPECULAR, MatSpecular1); //отраженный свет**

**glMaterialf(GL\_FRONT\_AND\_BACK, GL\_SHININESS, MatShininess1); //степень отражаемого света**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SHININESS, MatEmission1); //излучаемый свет**

**glVertex3f(-3.0f, 0.0f, 0.0f);**

**glVertex3f(-1.5f, -1.5f, 3.0f);**

**glVertex3f(0.0f, 0.0f, 0.0f);**

**glColor3ub(200, 0, 0);**

**glVertex3f(0.0f, 0.0f, 0.0f);**

**glVertex3f(-1.5f, -1.5f, 3.0f);**

**glVertex3f(0.0f, -3.0f, 0.0f);**

**glVertex3f(0.0f, -3.0f, 0.0f);**

**glVertex3f(-1.5f, -1.5f, 3.0f);**

**glVertex3f(-3.0f, -3.0f, 0.0f);**

**glVertex3f(-3.0f, -3.0f, 0.0f);**

**glVertex3f(-1.5f, -1.5f, 3.0f);**

**glVertex3f(-3.0f, 0.0f, 0.0f);**

**glEnd();**

**glBegin(GL\_TRIANGLE\_STRIP);**

**float MatDiffuse2[] = { 0.0f, 1.0f, 0.0f, 1.0f };**

**float MatSpecular2[] = { 1.0f, 1.0f,1.0f, 1.0f };**

**float MatShininess2 = 0.0f;**

**float MatEmission2[] = { 0.0f, 0.0f, 0.0f, 0.0f };**

**glColor3ub(200, 200, 200);**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_DIFFUSE, MatDiffuse2);**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SPECULAR, MatSpecular2);**

**glMaterialf(GL\_FRONT\_AND\_BACK, GL\_SHININESS, MatShininess2);**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SHININESS, MatEmission2);**

**glVertex3f(-6.0f, -3.0f, 0.0f);**

**glVertex3f(-4.5f, -4.5f, 3.0f);**

**glVertex3f(-3.0f, -3.0f, 0.0f);**

**glVertex3f(-3.0f, -3.0f, 0.0f);**

**glVertex3f(-4.5f, -4.5f, 3.0f);**

**glVertex3f(-3.0f, -6.0f, 0.0f);**

**glVertex3f(-3.0f, -6.0f, 0.0f);**

**glVertex3f(-4.5f, -4.5f, 3.0f);**

**glVertex3f(-6.0f, -6.0f, 0.0f);**

**glVertex3f(-6.0f, -6.0f, 0.0f);**

**glVertex3f(-4.5f, -4.5f, 3.0f);**

**glVertex3f(-6.0f, -3.0f, 0.0f);**

**glEnd();**

**glBegin(GL\_TRIANGLE\_STRIP);**

**float MatDiffuse3[] = { 0.0f, 0.0f, 1.0f, 0.0f };**

**float MatSpecular3[] = { 0.0f, 0.0f, 0.0f, 0.0f };**

**float MatShininess3 = 0;**

**float MatEmission3[] = { 0.0f, 0.0f, 0.0f, 0.0f };**

**glColor3ub(200, 200, 200);**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_DIFFUSE, MatDiffuse3);**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SPECULAR, MatSpecular3);**

**glMaterialf(GL\_FRONT\_AND\_BACK, GL\_SHININESS, MatShininess3);**

**glMaterialfv(GL\_FRONT\_AND\_BACK, GL\_SHININESS, MatEmission3);**

**glVertex3f(-9.0f, -6.0f, 0.0f);**

**glVertex3f(-7.5f, -7.5f, 3.0f);**

**glVertex3f(-6.0f, -6.0f, 0.0f);**

**glVertex3f(-6.0f, -6.0f, 0.0f);**

**glVertex3f(-7.5f, -7.5f, 3.0f);**

**glVertex3f(-6.0f, -9.0f, 0.0f);**

**glVertex3f(-6.0f, -9.0f, 0.0f);**

**glVertex3f(-7.5f, -7.5f, 3.0f);**

**glVertex3f(-9.0f, -9.0f, 0.0f);**

**glVertex3f(-9.0f, -9.0f, 0.0f);**

**glVertex3f(-7.5f, -7.5f, 3.0f);**

**glVertex3f(-9.0f, -6.0f, 0.0f);**

**glEnd();**

**}**

**void init\_l() {**

**float light0\_diffuse[] = { 1.0, 1.0, 1.0, 1.0 };// устанавливаем диффузный цвет света**

**GLfloat light0\_position[] = { 1.0, 1.0, 4.0, 1.0 }; // позиция источника света**

**GLfloat light0\_spot\_direction[] = { 0.3, 0.0, -1.0 };**

**glClearColor(0.0, 0.0, 0.0, 0.0);**

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

**glShadeModel(GL\_SMOOTH);**

**glColor3f(0.0, 0.0, 0.0);**

**glDisable(GL\_COLOR\_MATERIAL);**

**glEnable(GL\_LIGHTING);**

**glLightModelf(GL\_LIGHT\_MODEL\_AMBIENT, GL\_TRUE);**

**glEnable(GL\_LIGHT0); // разрешаем использовать light0**

**glLightfv(GL\_LIGHT0, GL\_DIFFUSE, light0\_diffuse); // устанавливаем источнику света light0 диффузный свет, который указали ранее**

**glLightfv(GL\_LIGHT0, GL\_POSITION, light0\_position); // устанавливаем положение источника света, указанное ранее**

**glLightfv(GL\_LIGHT0, GL\_SPECULAR, light0\_diffuse);**

**glLightf(GL\_LIGHT0, GL\_SPOT\_CUTOFF, 30);**

**glLightfv(GL\_LIGHT0, GL\_SPOT\_DIRECTION, light0\_spot\_direction);**

**glLightf(GL\_LIGHT0, GL\_SPOT\_EXPONENT, 15.0);**

**}**

**GLfloat spin = 0.0;**

**GLfloat speed = 0.005;**

**void RenderScene(void) {**

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

**glEnable(GL\_DEPTH\_TEST);**

**glLoadIdentity();**

**init\_l();**

**glEnable(GL\_NORMALIZE);**

**gluLookAt(16.0, 16.0, 18.0, 1.0, 1.0, 4.0, 3.0, 3.0, 3.0);**

**glRotatef(spin, 0, 0, 0.01);**

**draw();**

**spin += speed;**

**glutSwapBuffers();**

**}**

**void Systemthing(int w, int h) {**

**float ratio = w \* 1.0 / h;**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

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

**gluPerspective(45.0f, ratio, 0.1f, 100.0f);**

**glMatrixMode(GL\_MODELVIEW);**

**}**

**int main(int argc, char\* argv[])**

**{**

**glutInit(&argc, argv);**

**srand(time(NULL));**

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

**glutInitWindowSize(800, 680);**

**glutInitWindowPosition(200, 170);**

**glutCreateWindow("lab3");**

**glutDisplayFunc(RenderScene);**

**glutReshapeFunc(Systemthing);**

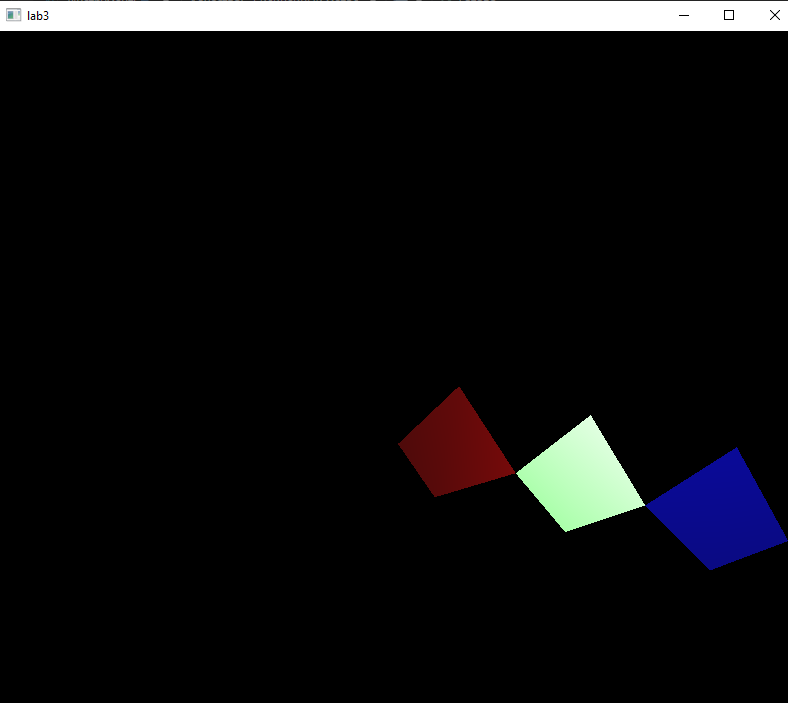
**glutIdleFunc(RenderScene);**

**glutMainLoop();**

**return 0;**

**}**

**Пример работы:**

****

**Вывод:** в результате проделанной работы мы включили источник света, задать отражающие свойства поверхностей: диффузное, зеркальное отражение, цвет поверхности и вывели несколько объемных объектов.