

ГУАП

КАФЕДРА № 43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

Щёкин С. В.

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Работа с источником света и свойствами источников и
материалов.

по курсу: Компьютерная графика

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

подпись, дата

Бобрович Н. С.

инициалы, фамилия

Санкт-Петербург 2022

Задание:

Включить 3 источника света, задать отражающие свойства поверхностей, положение и цвет источников света.

Рекомендуется цвет поверхностей сделать одинаковым, а отражающие свойства – разными. Вывести несколько объемных объектов. Источники света должны иметь различный цвет. Необходимо организовать вращение сцены, управляемое с клавиатуры (сцена с объектами объектами вращается, источники неподвижны).

Листинг программы:

```

1 //Включить 3 источника света, задать отражающие свойства поверхностей,
2 //положение и цвет источников света.Рекомендуется цвет поверхностей
3 //сделать одинаковым, а отражающие свойства – разными.
4 //Вывести несколько объемных объектов.Источники света должны иметь
5 //различный цвет.Необходимо организовать вращение сцены, управляемое с
6 //клавиатуры(сцена с объектами объекты вращается, источники неподвижны)
7
8 #define ESCAPE '\033'
9
10 #include <glut.h>
11
12 #include <iostream>
13
14 #include<math.h>
15
16 GLint WinWid = 800, WinHei = 800;
17
18 double rotate_y = 0;
19
20 double rotate_x = 0;
21
22 void Timer(int)
23 {
24
25
26     glutPostRedisplay();
27     glutTimerFunc(60, Timer, 0);
28 }
29
30 /* Функция вызывается при изменении размеров окна */
31 void Reshape(GLint w, GLint h)
32 {
33
34
35     WinWid = w;
36     WinHei = h;
37
38     glViewport(0, 0, w, h);
39     glClearColor(0.1, 0.1, 0.1, 1.0);
40     glMatrixMode(GL_PROJECTION);
41     glLoadIdentity();
42     glOrtho(-WinWid / 2, WinWid / 2, -WinHei / 2, WinHei / 2, -600.0, 600.0);
43     glMatrixMode(GL_MODELVIEW);
44 }
45
46
47 void Keyboard(int key, int x, int y) { //перемещение сцены с помощью кл
48

```

```

49     if (key == GLUT_KEY_RIGHT)
50
51         rotate_y += 5;
52
53     else if (key == GLUT_KEY_LEFT)
54
55         rotate_y -= 5;
56
57     else if (key == GLUT_KEY_UP)
58
59         rotate_x += 5;
60
61     else if (key == GLUT_KEY_DOWN)
62
63         rotate_x -= 5;
64
65     glutPostRedisplay();
66
67 }
68
69 float ambient[] = { 0.4, 0.4, 0.4, 1.0 }; //цвет фонового излучения ис
70 float diffuse[] = { 0.0, 1.0, 0.0, 1.0 }; //цвет рассеянного излучения и
71 float diffuse2[] = { 1.0, 0.0, 0.0, 1.0 };
72 float diffuse3[] = { 0.0, 0.0, 1.0, 1.0 };
73
74 float spec[] = { 1.0, 0.0, 0.0, 1.0 }; //цвет зеркального излучения ис
75 float spec2[] = { 0.0, 1.0, 0.0, 1.0 };
76 float spec3[] = { 1.0, 1.0, 0.0, 1.0 };
77 float specref[] = { 1.0, 1.0, 1.0, 1.0 };
78
79 float lpos1[] = { WinWid / 2 - 100, WinHei / 2 + 100, 10.0, 1.0 }; //на
80 float lpos2[] = { -WinWid / 2 + 100, WinHei / 2 - 100, 10.0, 1.0 };
81 float lpos3[] = { 0, -WinHei / 2 + 100, 10.0, 1.0 };
82 float black[] = { 0.0, 0.0, 0.0, 1.0 };
83
84 void Draw() {
85
86     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
87
88     glEnable(GL_DEPTH_TEST);
89     glEnable(GL_LIGHTING); // включаем света
90     glEnable(GL_LIGHT0); // включаем источник света 0
91     glEnable(GL_LIGHT1); // включаем источник света 1
92     glEnable(GL_LIGHT2); // включаем источник света 2
93     glEnable(GL_COLOR_MATERIAL);
94
95     glLightfv(GL_LIGHT0, GL_AMBIENT, ambient); //переменные для 0 источ
96     glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);

```



```

97     glLightfv(GL_LIGHT0, GL_POSITION, lpos1);
98     glLightfv(GL_LIGHT0, GL_SPECULAR, spec);
99
100     glLightfv(GL_LIGHT1, GL_AMBIENT, ambient); //переменные для 1 ист
101     glLightfv(GL_LIGHT1, GL_DIFFUSE, diffuse2);
102     glLightfv(GL_LIGHT1, GL_POSITION, lpos2);
103     glLightfv(GL_LIGHT1, GL_SPECULAR, spec2);
104
105     glLightfv(GL_LIGHT2, GL_AMBIENT, ambient); //переменные для 2 источ
106     glLightfv(GL_LIGHT2, GL_DIFFUSE, diffuse3);
107     glLightfv(GL_LIGHT2, GL_POSITION, lpos3);
108     glLightfv(GL_LIGHT2, GL_SPECULAR, spec3);
109
110
111     // сфера
112     glColor3f(0.5, 0.5, 0.1);
113     glPushMatrix(); //записываем матрицу в стек
114     glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, specref); //задаем пара
115     glMateriali(GL_FRONT_AND_BACK, GL_SHININESS, 20);
116     glColorMaterial(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE);
117     glRotatef(rotate_y, 0, 1, 0); //поворот объекта по y
118     glRotatef(rotate_x, 1, 0, 0);
119     glTranslatef(-200, 0, 0);
120     glScalef(100, 100, 100);
121     glutSolidSphere(0.8, 40, 40); // сфера
122     glPopMatrix(); //восстанавливаем матрицу из стека
123
124     // куб
125
126     glPushMatrix(); //записываем матрицу в стек
127     glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, specref); //задаем пар
128     glMateriali(GL_FRONT_AND_BACK, GL_SHININESS, 70);
129     glColorMaterial(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE);
130     glRotatef(rotate_y, 0.0, 1.0, 0.0); //поворот объекта по x
131     glRotatef(rotate_x, 1, 0, 0);
132     glColor3f(0.1, 0.5, 0.1);
133     glScalef(100, 100, 100);
134     glutSolidCube(1);
135     glPopMatrix(); //восстанавливаем матрицу из стека
136
137     // чайник
138     glColor3f(0.1, 0.5, 0.1);
139     glPushMatrix();
140     glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, specref); //задаем пара
141     glMateriali(GL_FRONT_AND_BACK, GL_SHININESS, -10);
142     glColorMaterial(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE);
143     glRotatef(rotate_y, 0.0, 1.0, 0.0); //поворот объекта по y
144     glRotatef(rotate_x, 1, 0, 0); //поворот объекта по x

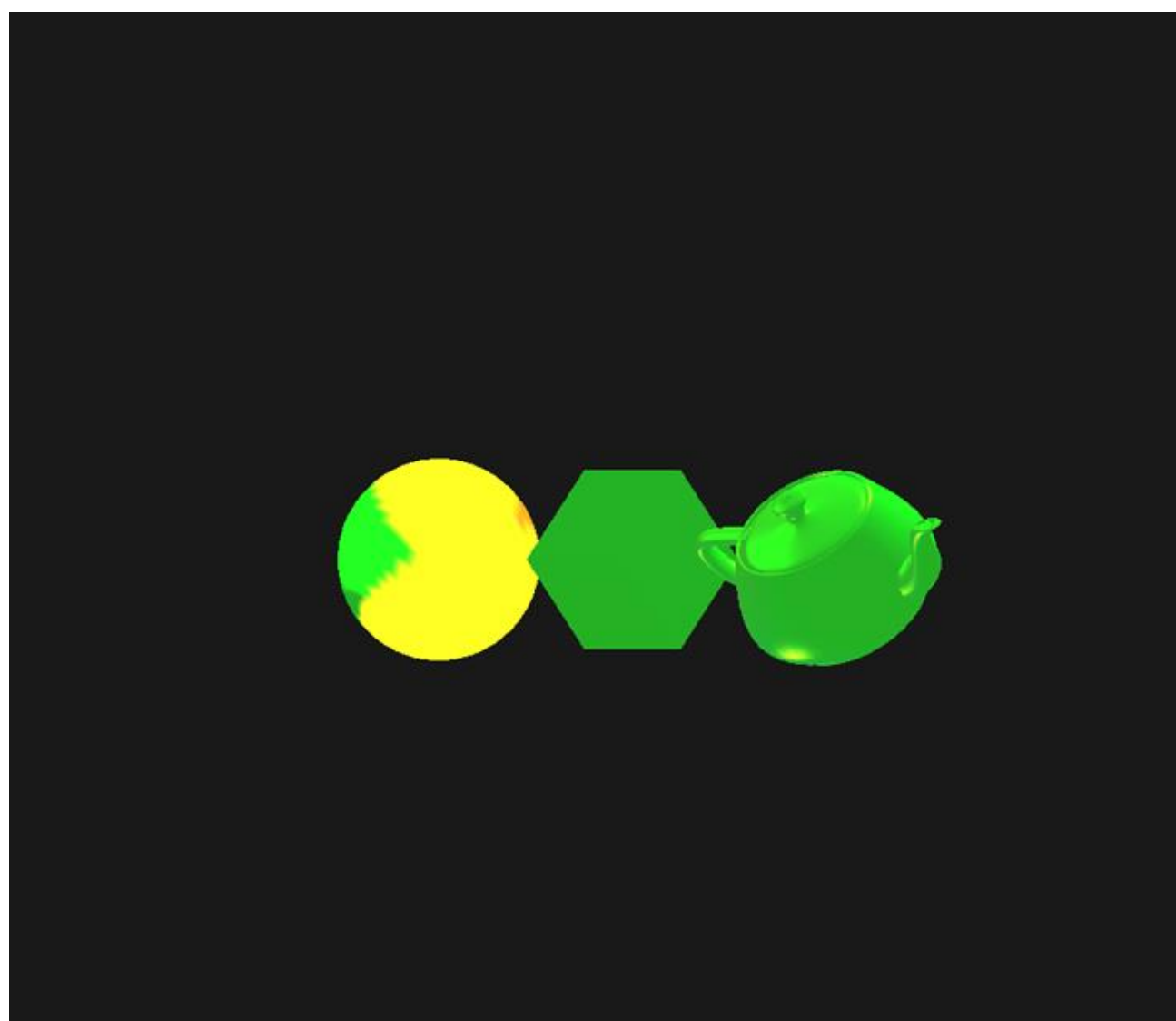
```

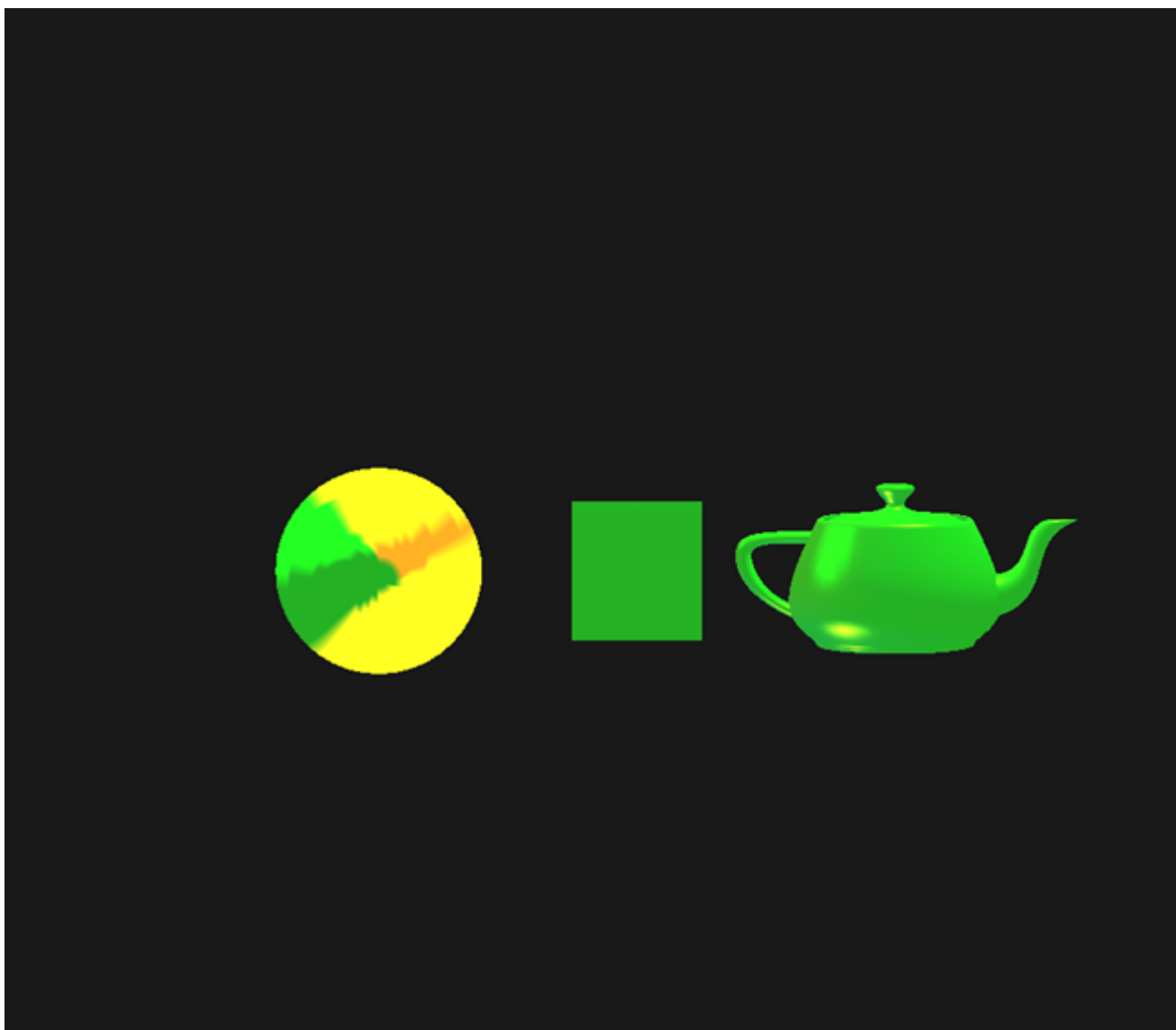
```

145     glTranslatef(200, 0, 0);
146     glScalef(55, 55, 55);
147     glutSolidTeapot(1.5);
148     glPopMatrix();//восстанавливаем матрицу из стека
149
150     glutSwapBuffers();
151
152 }
153
154 int main(int argc, char** argv)
155 {
156
157
158     glutInit(&argc, argv);
159     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA);
160     glutInitWindowPosition(100, 100);
161     glutInitWindowSize(WinWid, WinHei);
162     glutCreateWindow("Лаба4");
163
164     glutDisplayFunc(Draw);//отрисовка сцены
165     glutReshapeFunc(Reshape);
166
167     glutTimerFunc(60, Timer, 0);
168     glutSpecialFunc(Keyboard);
169
170     glutMainLoop();
171
172     return 0;
173 }

```

Результат работы:





Выводы:

В результате выполнения работы были получены навыки работы с источником света и свойствами источников и материалов.