

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 буфером глубины, прозрачностью
по курсу: Компьютерная графика

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

подпись, дата

Бобрович Н. С.

инициалы, фамилия

Санкт-Петербург 2022

Задание:

Вывести несколько пересекающихся объектов, как минимум один из них должен быть прозрачным. Включить 3 источника света, задать свойства поверхностей и источников. Организовать раздельное вращение источников света вокруг неподвижной сцены, управляемое с клавиатуры или мышью.

Листинг программы:

```

1  #include <glut.h>
2  #include <iostream>
3  #include <math.h>
4
5  GLint WinWid = 800, WinHei = 800;
6  int Angle = 0.0;
7  int Angle1 = 0.0;
8  int Angle2 = 0.0;
9  int Angle3 = 0.0;
10 float Scale = 1.0;
11
12 float ambient[] = { 0.4, 0.4, 0.4, 1.0 };
13 float diffuse[] = { 0.0, 1.0, 0.0, 1.0 };
14 float diffuse2[] = { 1.0, 0.0, 0.0, 1.0 };
15 float diffuse3[] = { 0.0, 0.0, 1.0, 1.0 };
16 float spec[] = { 1.0, 0.0, 0.0, 1.0 };
17 float spec2[] = { 0.0, 1.0, 0.0, 1.0 };
18 float spec3[] = { 1.0, 1.0, 0.0, 1.0 };
19 float specref[] = { 1.0, 1.0, 1.0, 1.0 };
20 float lpos1[] = { WinWid / 2 - 100, WinHei / 2 + 100, 10.0, 1.0 };
21 float lpos2[] = { -WinWid / 2 + 100, WinHei / 2 - 100, 10.0, 1.0 };
22 float lpos3[] = { 0, -WinHei / 2 + 100, 10.0, 1.0 };
23 float black[] = { 0.0, 0.0, 0.0, 1.0 };
24
25 void Timer(int)
26 {
27     glutPostRedisplay();
28     glutTimerFunc(1, Timer, 0);
29 }
30
31 void Keyboard(unsigned char key, int x, int y)
32 {
33     switch (key)
34     {
35     case 'd': Angle++;
36         break;
37     case 'a': Angle--;
38         break;
39     case 's': Angle1++;
40         break;
41     case 'w': Angle1--;
42         break;
43     case 'q': Angle2++;
44         break;
45     case 'e': Angle2--;
46         break;
47     }
48 }

```

```

49
50 void Reshape(GLint w, GLint h)
51 {
52     WinWid = w;
53     WinHei = h;
54
55     glViewport(0, 0, w, h);
56     glClearColor(0.3, 0.7, 0.4, 0.9);
57     glMatrixMode(GL_PROJECTION);
58     glLoadIdentity();
59     glOrtho(-WinWid / 2, WinWid / 2, -WinHei / 2, WinHei / 2, -600.0, 600.0);
60     glMatrixMode(GL_MODELVIEW);
61
62 }
63
64 void Draw() {
65     glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
66     glPushMatrix();
67     glRotatef(Angle3, 0.1, 0.1, 0.0);
68     glEnable(GL_DEPTH_TEST);
69     glEnable(GL_LIGHTING);
70     glEnable(GL_LIGHT0);
71     glEnable(GL_LIGHT1);
72     glEnable(GL_LIGHT2);
73     glEnable(GL_COLOR_MATERIAL);
74
75     glLightfv(GL_LIGHT0, GL_AMBIENT, ambient);
76     glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse);
77     glLightfv(GL_LIGHT0, GL_POSITION, lpos1);
78     glLightfv(GL_LIGHT0, GL_SPECULAR, spec);
79
80     glLightfv(GL_LIGHT1, GL_AMBIENT, ambient);
81     glLightfv(GL_LIGHT1, GL_DIFFUSE, diffuse2);
82     glLightfv(GL_LIGHT1, GL_POSITION, lpos2);
83     glLightfv(GL_LIGHT1, GL_SPECULAR, spec2);
84
85     glLightfv(GL_LIGHT2, GL_AMBIENT, ambient);
86     glLightfv(GL_LIGHT2, GL_DIFFUSE, diffuse3);
87     glLightfv(GL_LIGHT2, GL_POSITION, lpos3);
88     glLightfv(GL_LIGHT2, GL_SPECULAR, spec3);
89
90     Angle3 += 1;
91
92     glPopMatrix();
93
94     //top
95
96     glPushMatrix();

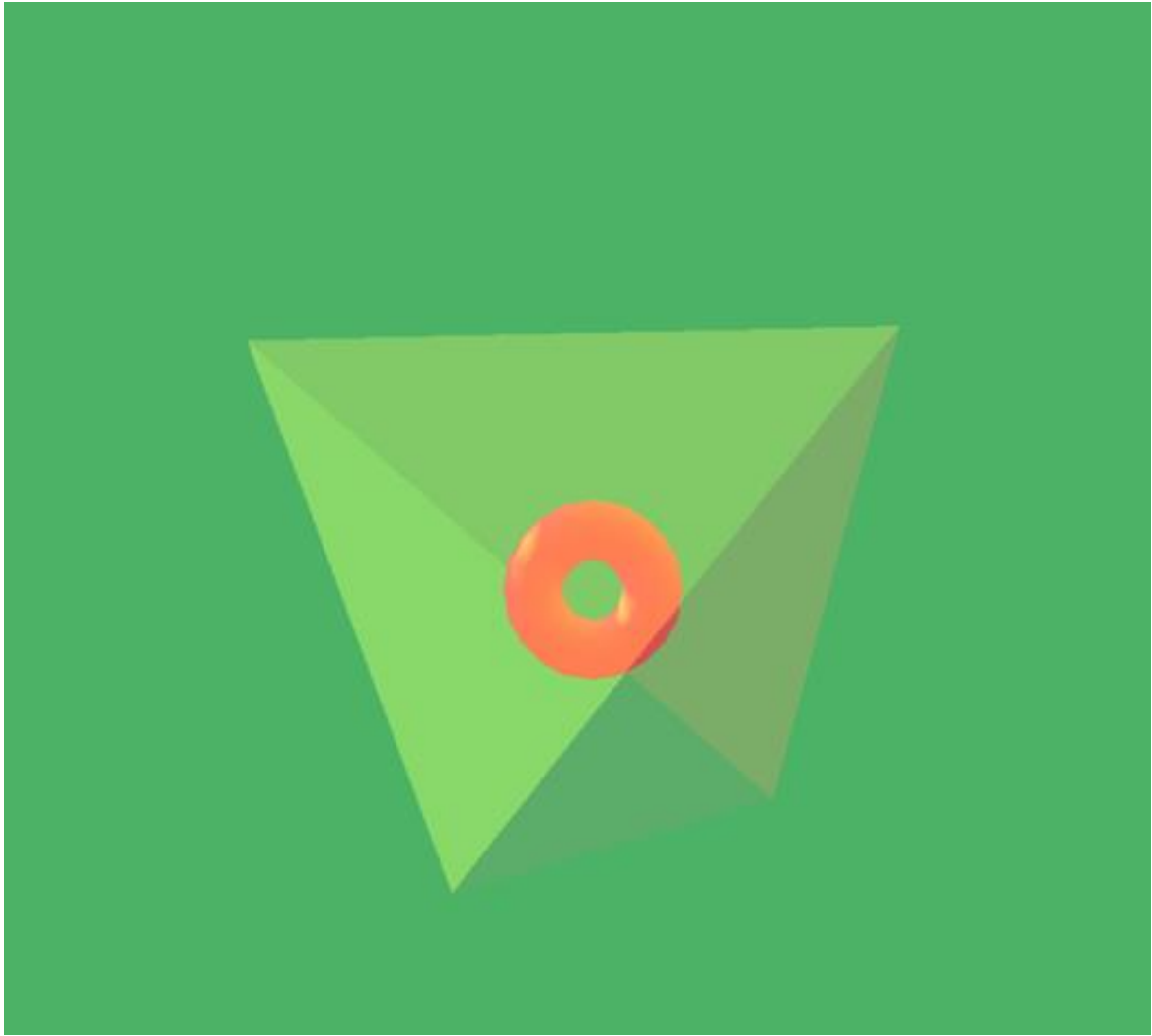
```

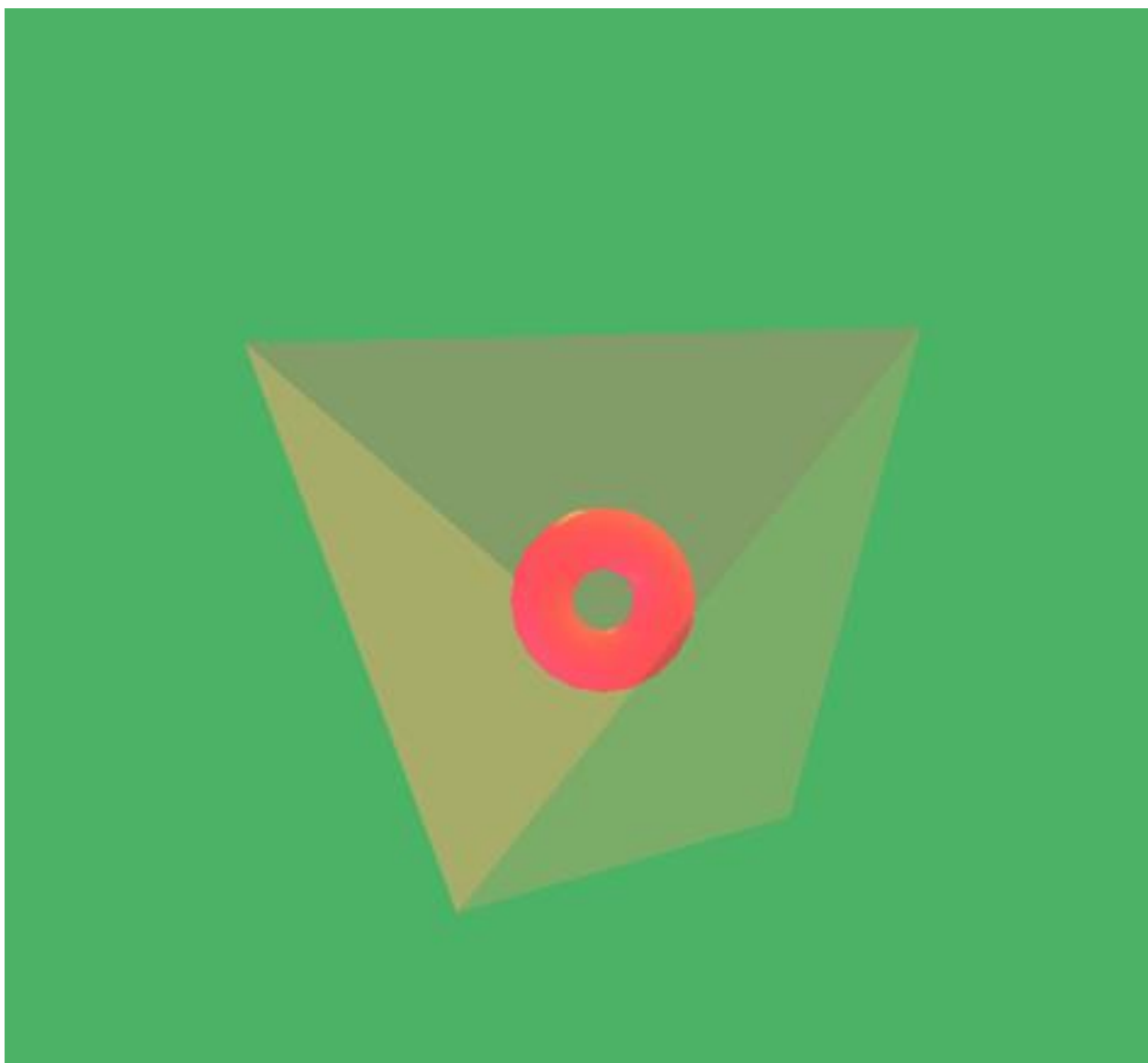
```

97     glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, specref);
98     glMateriali(GL_FRONT_AND_BACK, GL_SHININESS, 70);
99     glColorMaterial(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE);
100    glRotatef(Angle, 0.0, 1.0, 0.0);
101    glRotatef(Angle1, 1, 0, 0);
102    glRotatef(Angle2, 0, 0, 1);
103    glColor3f(1, 0.2, 0.2);
104    glutSolidTorus(20, 40, 10, 20);
105    glPopMatrix();
106
107    //тетраэдр
108
109    glPushMatrix();
110    glEnable(GL_BLEND);
111    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
112    glColor4f(0.3, 0.3, 0.3, 0.3);
113    glMaterialfv(GL_FRONT_AND_BACK, GL_SPECULAR, specref);
114    glMateriali(GL_FRONT_AND_BACK, GL_SHININESS, 0.9);
115    glColorMaterial(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE);
116    glRotatef(Angle, 0.0, 1.0, 0.0);
117    glRotatef(Angle1, 1, 0, 0);
118    glRotatef(Angle2, 0, 0, 1);
119    glScalef(300, 300, 300);
120    glutSolidTetrahedron();
121    glPopMatrix();
122    glutSwapBuffers();
123 }
124
125 int main(int argc, char** argv)
126 {
127     glutInit(&argc, argv);
128     glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA | GLUT_DEPTH);
129     glutInitWindowPosition(100, 100);
130     glutInitWindowSize(WinWid, WinHei);
131     glutCreateWindow("Лаба 5");
132     glutDisplayFunc(Draw);
133     glutReshapeFunc(Reshape);
134     glutTimerFunc(1, Timer, 0);
135     glutKeyboardFunc(Keyboard);
136     glutMainLoop();
137
138     return 0;
139 }
140
141

```

Результат работы:





Выводы:

В результате выполнения работы были получены навыки работы с буфером глубины, прозрачностью.