**计算机图形学实验报告**

**实 验 五**

**烟台理工学院**

班级： 信2121-1

姓名： 张瀛煜

学号： 202105721124

**一、实验内容 1**

1. 实验内容及要求

1. 在彩色立方体程序中，任选参数（与默认值不同）定义虚拟照相机。

2. 在内容1的基础上分别使用投影：

a. 正交投影

b. 透视投影

2. 实验程序

1. //
2. //  main.c
3. //  基于投影的阴影
4. //
5. //  Created by 張瀛煜 on 2022/9/28.
6. //

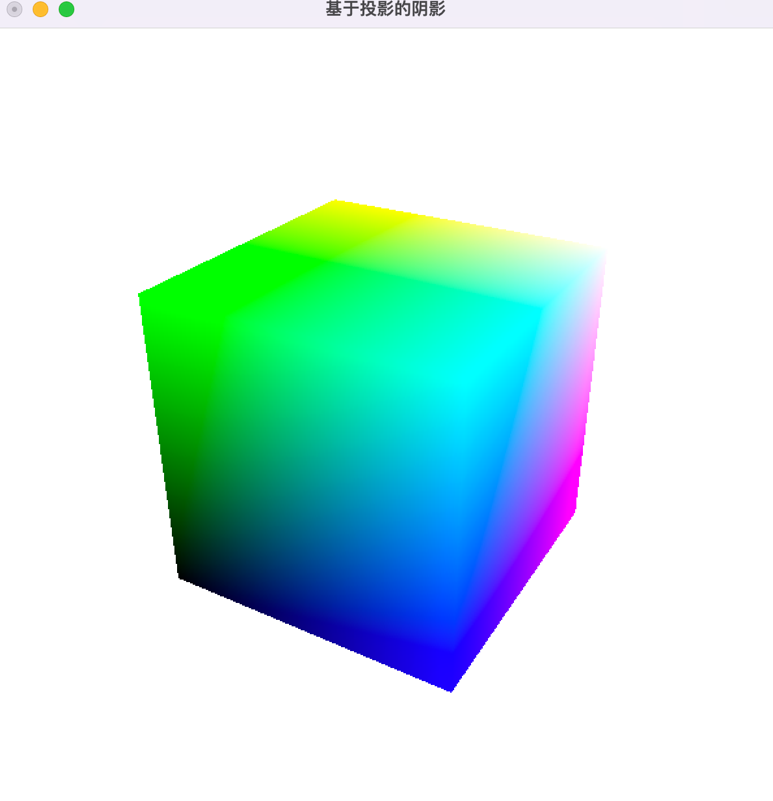
9. #include <stdio.h>
10. #include <GLUT/GLUT.h>
11. #include <math.h>
13. **const** GLubyte mindex[] =
14. {0,3,2,1,
15. 2,3,7,6,
16. 0,4,7,3,
17. 1,2,6,5,
18. 4,5,6,7,
19. 0,1,5,4};
21. **const** **float** vertex[] = { -1.0, -1.0, -1.0, 1.0, -1.0, -1.0, 1.0, 1.0, -1.0, -1.0, 1.0, -1.0,-1.0, -1.0, 1.0, 1.0, -1.0, 1.0, 1.0, 1.0, 1.0, -1.0, 1.0, 1.0 };
22. **const** **float** color[] = { 0, 0, 0, 1.0, 0, 0, 1.0, 1.0, 0, 0, 1.0, 0,0, 0, 1.0, 1.0, 0, 1.0, 1.0, 1.0, 1.0, 0, 1.0, 1.0 };
23. **float** light[] = { 0, 0, 0, 0 };
24. **float** theta[] = { 0, 0, 0, 0 };
25. **int** axis;
26. GLfloat m[16];
28. **void** myDisplay(){
29. glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);    //清除颜色缓存和深度缓存
30. glLoadIdentity();
32. glTranslated(0, 0, -6);
33. glRotatef(30, 1.0, 0.0, 0.0);
34. glRotatef(60, 0.0, 1.0, 0.0);
35. glRotatef(0, 0.0, 0.0, 1.0);
37. glDrawElements(GL\_QUADS, 24, GL\_UNSIGNED\_BYTE, mindex);      //绘制正方体

40. glBegin(GL\_POLYGON);
41. glVertex3f(-0.5, 0.5, -0.5);
42. glVertex3f(-0.5, 0.5, 0.5);
43. glVertex3f(0.5, 0.5, 0.5);
44. glVertex3f(0.5, 0.5, -0.5);
45. glEnd();
46. glPushMatrix(); //保存当前矩阵状态
47. glTranslatef(light[0], light[1],light[2]); //平移回原位置
48. glMultMatrixf(m); //透视投影
49. glTranslatef(-light[0], -light[1],-light[2]); //把光源移动到原点
50. glColor3f(0.0,0.0,0.0);
51. glBegin(GL\_POLYGON);
52. glVertex3f(-0.5, 0.5, -0.5);
53. glVertex3f(-0.5, 0.5, 0.5);
54. glVertex3f(0.5, 0.5, 0.5);
55. glVertex3f(0.5, 0.5, -0.5);
56. glEnd();
57. glPopMatrix();//恢复矩阵状态
58. glutSwapBuffers();  //清除颜色缓存和深度缓存
59. }

62. **void** Reshape(**int** w, **int** h){//两个参数：窗口被移动后大小
63. glViewport(0, 0, w, h);
64. glMatrixMode(GL\_PROJECTION);
65. glLoadIdentity();
66. gluPerspective(45.0f, (GLfloat)w / (GLfloat)h, 0.1f, 100.0f);
67. glMatrixMode(GL\_MODELVIEW);
68. glLoadIdentity();
69. }
70. **void** initWindow(**int** argc, **char** \*argv[], **int** width, **int** height, **char** \*title){
71. glutInit(&argc, argv);
72. glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGB|GLUT\_DEPTH);       //使用双缓存模式和深度缓存
73. glutInitWindowSize(width, height);       //指定窗口大小
74. glutCreateWindow(title);
76. glClearColor(1, 1, 1, 0);
77. glShadeModel(GL\_SMOOTH);
78. glEnable(GL\_DEPTH\_TEST);//激活深度测试
80. glEnableClientState(GL\_COLOR\_ARRAY);
81. glEnableClientState(GL\_VERTEX\_ARRAY);
82. glVertexPointer(3, GL\_FLOAT, 0, vertex);
83. glColorPointer(3, GL\_FLOAT, 0, color);
85. }
87. **int** main(**int** argc, **char** \*argv[]){
88. **for**(**int** i=0;i<16;i++)
89. m[i]=0.0;
90. m[0]=m[5]=m[10]=1.0;
91. m[7]=-1.0/light[1];
92. initWindow(argc, argv, 600, 600, "基于投影的阴影");
93. glutDisplayFunc(myDisplay);
94. glutReshapeFunc(Reshape);
95. // glutIdleFunc(spinCube);
96. //    glutMouseFunc(mouse);
97. //    glutKeyboardFunc(keyboard);
98. glutMainLoop();
99. **return** 0;
100. }

3. 运行结果

正交投影



透视投影

