**计算机图形学实验报告**

**实 验 六**

**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. 调式程序sphere.c至运行成功。

2. 在sphere.c的基础上分别进行如下更改：

a. 修改点光源位置

b. 修改点光源环境光反射、漫反射和镜面反射的分量值

c. 修改球体的材质，增加自发光属性

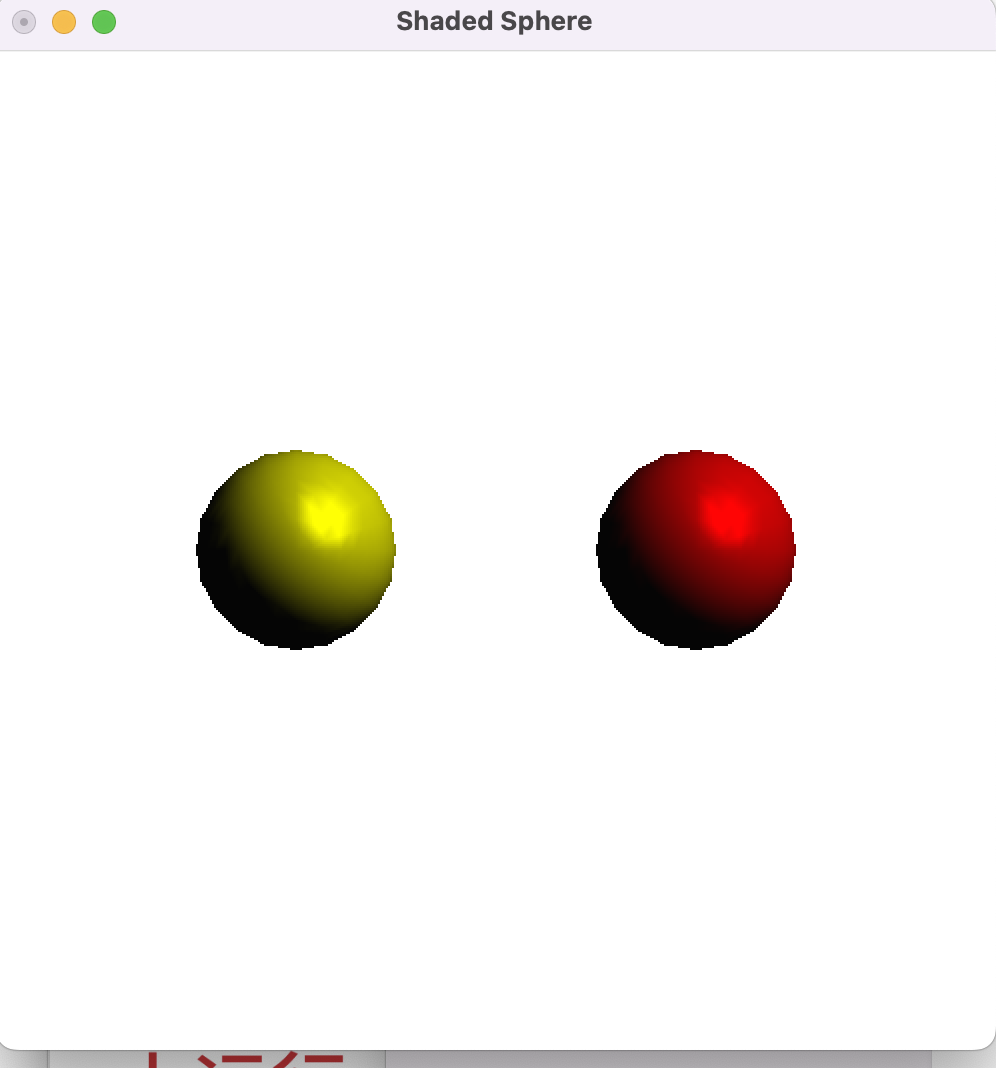
d. 追加一个新的光源（选做）

2. 实验程序

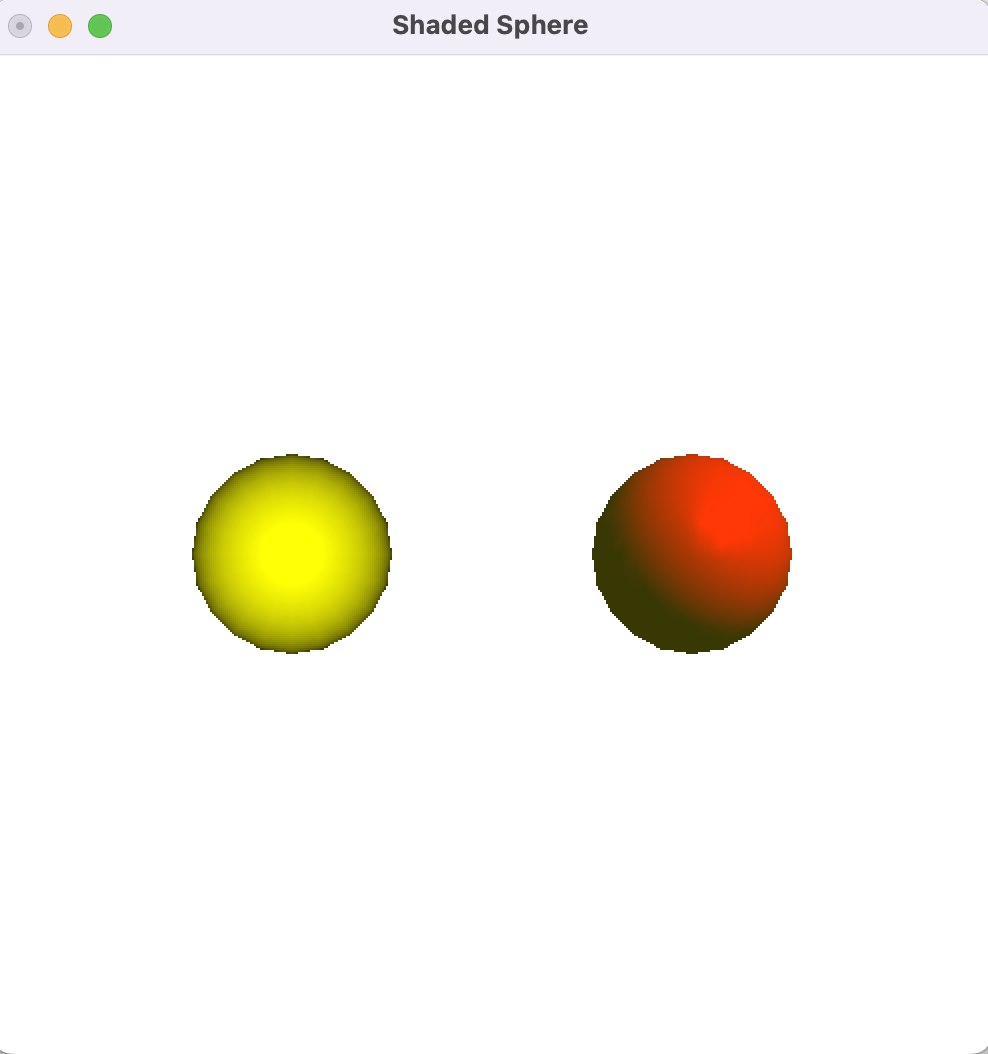
1. //
2. //  main.c
3. //  sphere
4. //
5. //  Created by 張瀛煜 on 2022/9/28.
6. //
8. #include <stdlib.h>
9. #include <GLUT/GLUT.h>

12. **void** shade(**int** mode){
13. **if**(mode==1){
14. GLfloat light0\_pos[]={1.0,1.0,1.0,0.0};
15. GLfloat light0\_diffuse[]={1.0,1.0,0.0,1.0};
16. GLfloat light0\_ambient[]={1.0,1.0,0.0,1.0};
17. GLfloat light0\_specular[]={1.0,1.0,0.0,1.0};
18. GLfloat global\_ambient[]={0.1,0.1,0.1,1.0};
20. GLfloat material\_ambient[]={1.0,1.0,1.0,1.0};
21. GLfloat material\_diffuse[]={1.0,1.0,1.0,1.0};
22. GLfloat material\_specular[]={1.0,1.0,1.0,1.0};
23. GLfloat material\_shinness = 50.0;
25. glLightfv(GL\_LIGHT0,GL\_POSITION,light0\_pos);
26. //glLightfv(GL\_LIGHT0,GL\_AMBIENT,light0\_ambient);
27. glLightfv(GL\_LIGHT0,GL\_DIFFUSE,light0\_diffuse);
28. glLightfv(GL\_LIGHT0,GL\_SPECULAR,light0\_specular);
29. glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT,global\_ambient);
31. //glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_AMBIENT,material\_ambient);
32. //glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_DIFFUSE,material\_diffuse);
33. glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_SPECULAR,material\_specular);
34. glMaterialf(GL\_FRONT,GL\_SHININESS,material\_shinness);
35. }**else** **if**(mode==2){
36. GLfloat light0\_pos[]={1.0,1.0,1.0,0.0};
37. GLfloat light0\_diffuse[]={1.0,0.0,0.0,1.0};
38. GLfloat light0\_ambient[]={1.0,0.0,0.0,1.0};
39. GLfloat light0\_specular[]={1.0,0.0,0.0,1.0};
40. GLfloat global\_ambient[]={0.1,0.1,0.1,1.0};
42. GLfloat material\_ambient[]={1.0,1.0,1.0,1.0};
43. GLfloat material\_diffuse[]={1.0,1.0,1.0,1.0};
44. GLfloat material\_specular[]={1.0,1.0,1.0,1.0};
45. GLfloat material\_shinness = 50.0;
47. glLightfv(GL\_LIGHT0,GL\_POSITION,light0\_pos);
48. //glLightfv(GL\_LIGHT0,GL\_AMBIENT,light0\_ambient);
49. glLightfv(GL\_LIGHT0,GL\_DIFFUSE,light0\_diffuse);
50. glLightfv(GL\_LIGHT0,GL\_SPECULAR,light0\_specular);
51. glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT,global\_ambient);
53. //glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_AMBIENT,material\_ambient);
54. //glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_DIFFUSE,material\_diffuse);
55. glMaterialfv(GL\_FRONT\_AND\_BACK,GL\_SPECULAR,material\_specular);
56. glMaterialf(GL\_FRONT,GL\_SHININESS,material\_shinness);
57. }
58. }
60. **void** inti(**void**){
61. glClearColor (1.0, 1.0, 1.0, 1.0);
62. glColor3f (0.0, 0.0, 0.0);
63. glShadeModel(GL\_SMOOTH);
64. glEnable(GL\_LIGHTING);
65. glEnable(GL\_LIGHT0);
66. glEnable(GL\_DEPTH\_TEST);
67. }
69. **void** myDisplay(**void**){
70. glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);
71. glMatrixMode(GL\_MODELVIEW);
72. glLoadIdentity();
73. glTranslatef(-2.0,0.0,0.0);
74. shade(1);
75. glutSolidSphere(1.0,20,20);
76. glTranslatef(4.0,0.0,0.0);
77. shade(2);
78. glutSolidSphere(1.0,20,20);
79. glFlush();
80. }
82. **void** reshape(**int** w,**int** h){
83. glViewport(0,0,(GLsizei) w,(GLsizei) h);
84. glMatrixMode(GL\_PROJECTION);
85. glLoadIdentity();
86. **if**(w<=h)
87. glOrtho(-5.0,5.0,-5.0\*(GLfloat) h/(GLfloat) w,5.0\*(GLfloat)h/(GLfloat)w,-10.0,10.0);
88. **else**
89. glOrtho(-5.0\*(GLfloat)w/(GLfloat)h,5.0\*(GLfloat) w/(GLfloat) h,-5.0,5.0,-10.0,10.0);
91. }
93. **int** main(**int** argc, **char** \*argv[]){
94. glutInit(&argc, argv);
95. glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE|GLUT\_DEPTH);
96. glutInitWindowSize(500, 500);
97. glutInitWindowPosition(200,400);
98. glutCreateWindow("Shaded Sphere");
99. inti();
100. glutDisplayFunc(myDisplay);
101. glutReshapeFunc(reshape);
102. glutMainLoop();
103. **return** 0;
104. }

3. 运行结果



修改光源位置



修改点光源环境光反射、漫反射和镜面反射的分量值

