**计算机图形学实验报告**

**实 验 二**

**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

内容：

1.按照实验内容完成程序，并调试至成功运行。

2.按实验报告模板完成实验报告。

要求：

1.实现二维Sierpinski镂垫，展示迭代8次后的镂空结果。

2. 实验程序

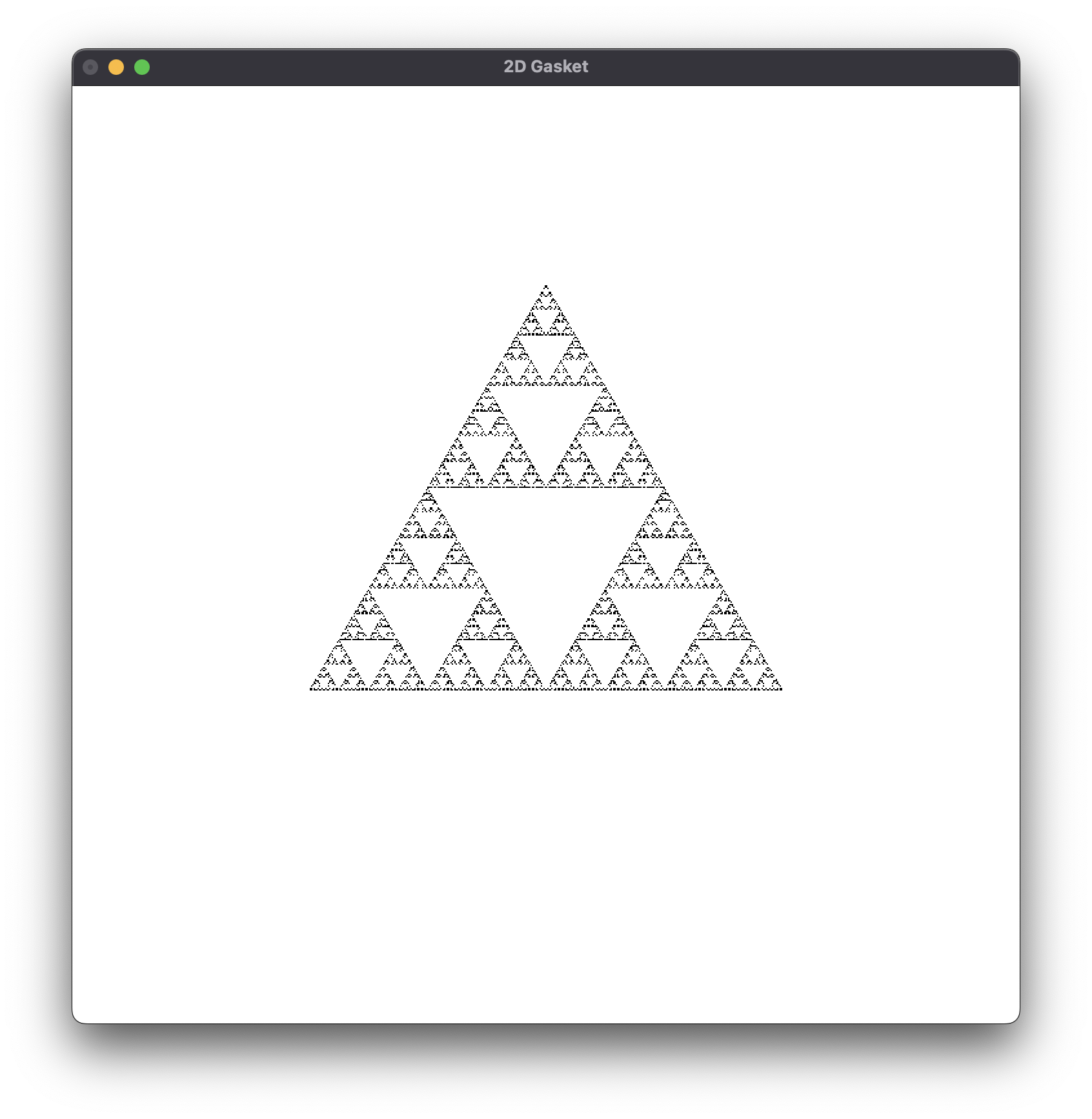
1. //
2. //  main.cpp
3. //  Sierpinshi\_1
4. //
5. //  Created by 張瀛煜 on 2022/9/20.
6. //
8. #include <stdio.h>
9. #include <stdlib.h>
10. #include <GLUT/GLUT.h>
11. #include <OpenGL/OpenGL.h>

14. GLfloat v[3][2] = {{-1.0,-0.58},{1.0,-0.58},{0.0,1.15}};
16. **int** n;

19. //画一个三角形
20. **void** triangle(GLfloat \*a ,GLfloat \*b,GLfloat \*c){
21. glVertex2fv(a);
22. glVertex2fv(b);
23. glVertex2fv(c);
24. }

27. //剖分三角形
28. **void** divide\_triangle (GLfloat \*a,GLfloat \*b,GLfloat \*c,**int** m){
29. GLfloat v0[2],v1[2],v2[2];
30. **int** j;
31. **if** (m>0) {
32. **for** (j=0; j<2; j++) {
33. v0[j]=(a[j]+b[j])/2;
34. }
35. **for** (j=0; j<2; j++) {
36. v1[j]=(a[j]+c[j])/2;
37. }
38. **for** (j=0; j<2; j++) {
39. v2[j]=(b[j]+c[j])/2;
40. }
41. divide\_triangle(a, v0,v1, m-1);
42. divide\_triangle(c, v1,v2, m-1);
43. divide\_triangle(b, v2,v0, m-1);
44. }**else**{
45. triangle(a, b, c);
46. }
47. }
49. **void** display(){
50. glClear(GL\_COLOR\_BUFFER\_BIT);
51. glBegin(GL\_TRIANGLES);
52. divide\_triangle(v[0], v[1], v[2], n);
53. glEnd();
54. glFlush();
55. }
57. **void** myinit(){
58. glMatrixMode(GL\_PROJECTION);
59. glLoadIdentity();
60. gluOrtho2D(-2.0, 2.0, -2.0, 2.0);
61. glMatrixMode(GL\_MODELVIEW);
62. glClearColor(1.0, 1.0, 1.0, 1.0);
63. glColor3f(0.0, 0.0, 0.0);
64. }
66. **int** main(**int** argc, **char** \* argv[]) {
67. n = 8;
68. glutInit(&argc,argv);
69. glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);
70. glutInitWindowSize(500 ,500);
71. glutCreateWindow("2D Gasket");
72. glutDisplayFunc(display);
73. myinit();
74. glutMainLoop();
76. **return** 0;
77. }

3. 运行结果



**二、实验内容 2**

1. 实验内容及要求

实现三维Sierpinski镂垫的面镂空形式，展示迭代8次后的镂空结果。

2. 实验程序

1. //
2. //  main.c
3. //  Sierpinski3D
4. //
5. //  Created by 張瀛煜 on 2022/9/21.
6. //

9. #include <stdio.h>
10. #include <stdlib.h>
11. #include <GLUT/GLUT.h>

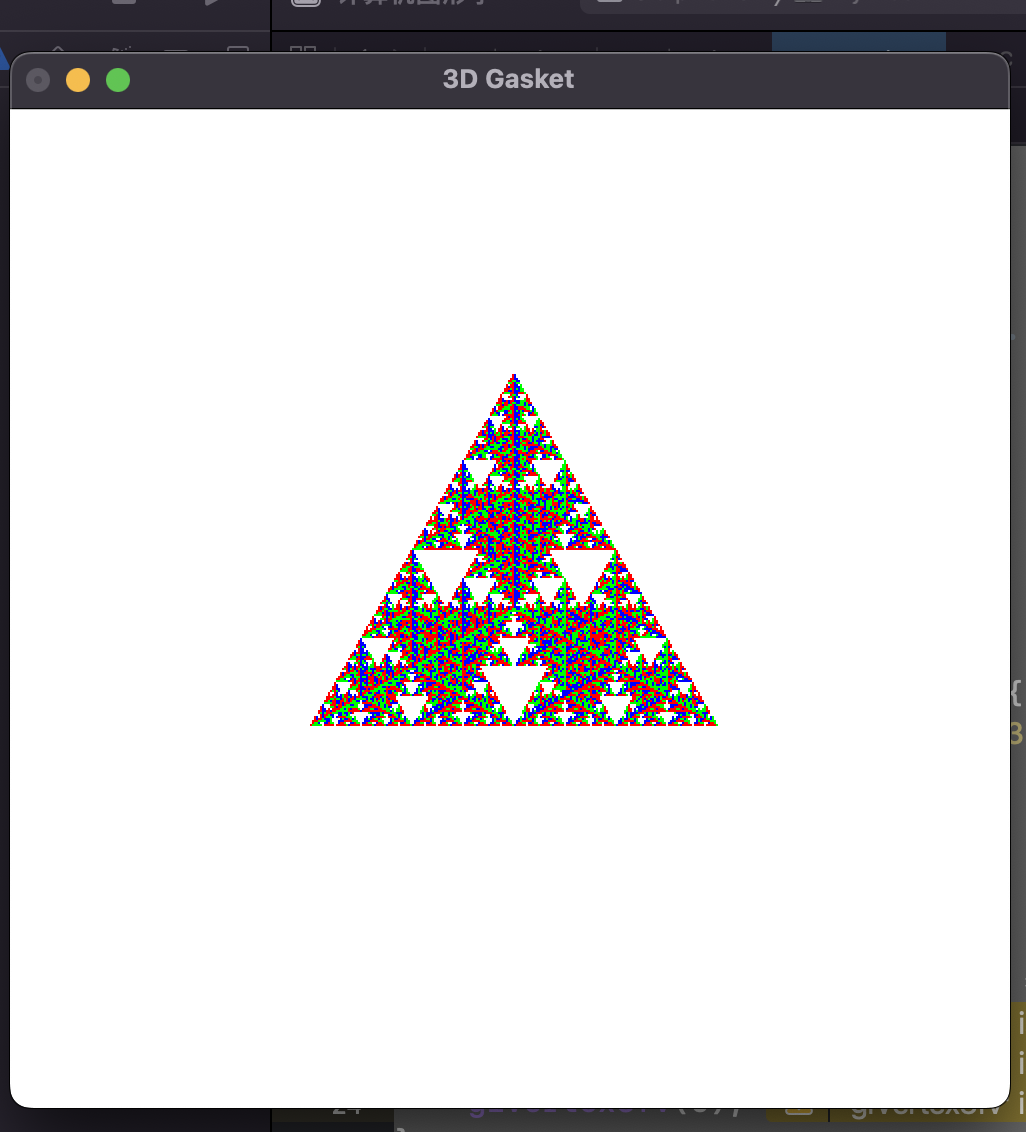
14. GLfloat v[4][3] = {{0.0,0.0,1.0},{0.0,0.942809,-0.33333}
15. ,{-0.816497,-0.471405,-0.33333},{0.816497,-0.471405,-0.33333}};
17. **int** n;

20. //画一个三角形
21. **void** triangle(GLfloat \*a ,GLfloat \*b,GLfloat \*c){
22. glVertex3fv(a);
23. glVertex3fv(b);
24. glVertex3fv(c);
25. }
27. GLfloat colors[4][3] = {{1.0,0.0,0.0},{0.0,1.0,0.0},{0.0,0.0,1.0},{0.0,0.0,0.0}};
29. **void** tetra(GLfloat \*a,GLfloat \*b,GLfloat \*c,GLfloat \*d){
30. glColor3fv(colors[0]);
31. triangle(a, b, c);
32. glColor3fv(colors[1]);
33. triangle(a,c,d);
34. glColor3fv(colors[2]);
35. triangle(a, d,b);
36. glColor3fv(colors[3]);
37. triangle(b, d,c);
38. }

41. **void** divide\_tetra(GLfloat \*a,GLfloat \*b,GLfloat \*c,GLfloat \*d,**int** m){
42. GLfloat mid[6][3];
43. **int** j;
44. **if** (m>0) {
45. **for** (j=0; j<3; j++) {
46. mid[0][j]=(a[j]+b[j])/2;
47. }
48. **for** (j=0; j<3; j++) {
49. mid[1][j]=(a[j]+c[j])/2;
50. }
51. **for** (j=0; j<3; j++) {
52. mid[2][j]=(a[j]+d[j])/2;
53. }
54. **for** (j=0; j<3; j++) {
55. mid[3][j]=(b[j]+c[j])/2;
56. }
57. **for** (j=0; j<3; j++) {
58. mid[4][j]=(c[j]+d[j])/2;
59. }
60. **for** (j=0; j<3; j++) {
61. mid[5][j]=(b[j]+d[j])/2;
62. }
63. divide\_tetra(a,mid[0],mid[1], mid[2],m-1);
64. divide\_tetra(mid[0],b,mid[3], mid[5],m-1);
65. divide\_tetra(mid[1],mid[3], c,mid[4],m-1);
66. divide\_tetra(mid[2],mid[4], d,mid[5],m-1);
68. }**else**{
69. tetra(a, b, c,d);
70. }
71. }
73. **void** display(**void**){

76. glClear(GL\_COLOR\_BUFFER\_BIT|GL\_DEPTH\_BUFFER\_BIT);
77. glBegin(GL\_TRIANGLES);
78. divide\_tetra(v[0], v[1], v[2],v[3], n);
79. glEnd();
80. glFlush();
81. }
83. **void** myReshape(**int** w,**int** h){
84. glViewport(2, 0, w, h );
85. glMatrixMode(GL\_PROJECTION);
86. glLoadIdentity();
87. **if**(w<=h){
88. glOrtho(-2.0, 2.0, -2.0\*(GLfloat) h/(GLfloat)w, 2.0\*(GLfloat)h/(GLfloat)w,-10.0,10.0);
89. }**else**{
90. glOrtho(-2.0\*(GLfloat) h/(GLfloat)w, 2.0\*(GLfloat)h/(GLfloat)w, -2.0, 2.0,-10.0,10.0);
91. }
92. glMatrixMode(GL\_MODELVIEW);
93. glutPostRedisplay();
95. }
97. **int** main(**int** argc, **char** \* argv[]) {
98. n = 5;
99. glutInit(&argc,argv);
100. glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB|GLUT\_DEPTH);
101. glutInitWindowSize(500 ,500);
102. glutCreateWindow("3D Gasket");
103. glutDisplayFunc(display);
104. glutReshapeFunc(myReshape);
106. glEnable(GL\_DEPTH\_TEST);
107. glClearColor(1.0, 1.0, 1.0, 1.0);
108. glutMainLoop();
109. **return** 0;
110. }

3. 运行结果



**三、实验内容 3**

1. 实验内容及要求

实现彩色立方体绘制，在程序中修改顶点坐标展示立方体的不同角度。

2. 实验程序

1. //
2. //  main.c
3. //  彩色立方体
4. //
5. //  Created by 張瀛煜 on 2022/9/21.
6. //
8. #include <stdio.h>
9. #include <GLUT/GLUT.h>
10. GLfloat vertices[8][3] =
11. {{-1.0,-1.0,-1.0},{1.0,-1.0,-1.0},
12. {1.0,1.0,-1.0}, {-1.0,1.0,-1.0},
13. {-1.0,-1.0,1.0},{1.0,-1.0,1.0},
14. {1.0,1.0,1.0}, {-1.0,1.0,1.0}};
15. **typedef** GLfloat point3[3];
17. GLfloat colors[8][3] = {{0.0,0.0,0.0},{1.0,0.0,0.0},{1.0,1.0,0.0},
18. {0.0,1.0,0.0}, {0.0,0.0,1.0}, {1.0,0.0,1.0}, {1.0,1.0,1.0},
19. {0.0,1.0,1.0}};
21. GLubyte cubeIndices[24] =
22. {0,3,2,1,
23. 2,3,7,6,
24. 0,4,7,3,
25. 1,2,6,5,
26. 4,5,6,7,
27. 0,1,5,4};

30. //定义一个面
31. **void** polygon(**int** a,**int** b,**int** c ,**int** d){
32. glBegin(GL\_POLYGON);
33. glColor3fv(colors[a]);
34. glVertex3fv(vertices[a]);
35. glColor3fv(colors[b]);
36. glVertex3fv(vertices[b]);
37. glColor3fv(colors[c]);
38. glVertex3fv(vertices[c]);
39. glColor3fv(colors[d]);
40. glVertex3fv(vertices[d]);
41. glEnd();
42. }
44. **void** colorcube(**void**){
45. polygon(0, 3, 2, 1);
46. polygon(2,3,7,6);
47. polygon(0,4,7,3);
48. polygon(1,2,6,5);
49. polygon(4,5,6,7);
50. polygon(0,1,5,4);
51. }
52. **void** display(){
53. glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);
54. glLoadIdentity();
55. glTranslated(0, 0, -6);
56. glRotatef(40, 1.0, 0.0, 0.0);
57. glRotatef(40, 0.0, 1.0, 0.0);
58. glRotatef(0, 0.0, 0.0, 1.0);
59. colorcube();
60. glutSwapBuffers();
61. }
63. **void** myReshape(**int** w, **int** h)  {    //两个参数：窗口被移动后大小{
64. glViewport(0, 0, w, h);
65. glMatrixMode(GL\_PROJECTION);
66. glLoadIdentity();
67. gluPerspective(45.0f, (GLfloat)w / (GLfloat)h, 0.1f, 100.0f);
68. glMatrixMode(GL\_MODELVIEW);
69. glLoadIdentity();
70. }

73. **int** main(**int** argc, **char** \* argv[]) {
74. glutInit(&argc, argv);
75. glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB |
76. GLUT\_DEPTH);
77. glutInitWindowSize(500, 500);
78. glutCreateWindow("彩色立方体");
79. glutReshapeFunc(myReshape);
80. glutDisplayFunc(display);
82. glEnable(GL\_DEPTH\_TEST);
83. glutMainLoop();
84. **return** 0;
85. }

3. 运行结果

