OpenGL实例：三角形

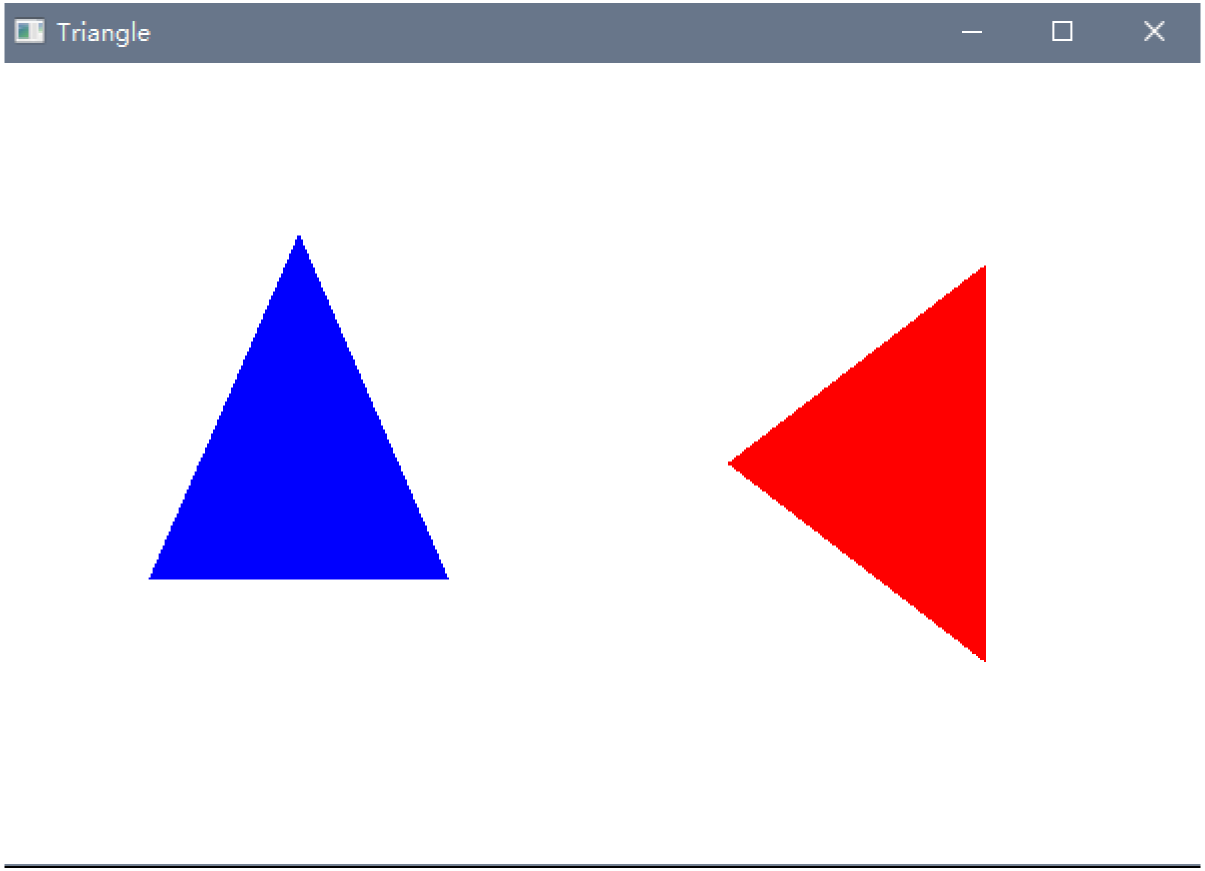
作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

更多请查看：计算机图形学

1. 三角形的旋转

```c
#include <GL/glut.h>
typedef GLfloat point2d[2]; // a point data type
void triangle(point2d a, point2d b, point2d c) // display a triangle
{
    glBegin(GL_TRIANGLES);
    glVertex2fv(a);
    glVertex2fv(b);
    glVertex2fv(c);
    glEnd();
}
void display(void)
{
    point2d v[3] = { {-1.0, -0.58}, {1.0, -0.58}, {0.0, 1.15} }; //initial triangle vertices
    glClear(GL_COLOR_BUFFER_BIT); // Clear display window
    glColor3f(0.0, 0.0, 1.0); // Set fill color to blue
    glLoadIdentity();//应该放在此处，保证每次显示正确
    glViewport(0, 0, 300, 400); // Set left viewport
    triangle(v[0], v[1], v[2]);
    glColor3f(1.0, 0.0, 0.0); // Set fill color to red
    glViewport(300, 0, 300, 400); // Set right viewport
    glRotatef(90.0, 0.0, 0.0, 1.0); // Rotate about z axis
    triangle(v[0], v[1], v[2]); // Display blue triangle
    glFlush();
}
void init()
{
    glMatrixMode(GL_PROJECTION);
    //glLoadIdentity();//如果放在此处，会导致第一次显示正确，再次刷新时图形旋转
    gluOrtho2D(-2.0, 2.0, -2.0, 2.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(1.0, 1.0, 1.0, 1.0);
}
void main(int argc, char **argv)
```

```
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(600, 400);
    glutCreateWindow("Triangle");
    glutDisplayFunc(display);
    init();
    glutMainLoop();
}
```

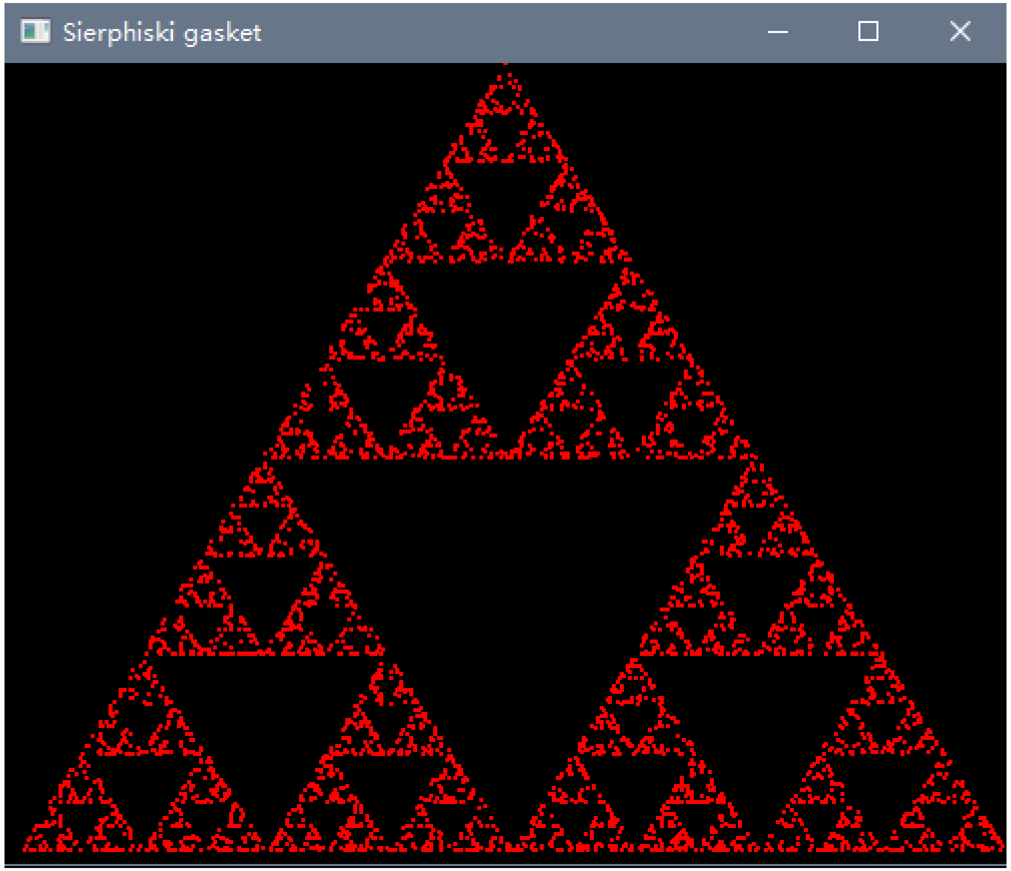


参考网址：[Opengl编程实例 - 红蓝三角形 - 图形学与可视化 - CSDN博客]

2. Sierpinski gasket

方法1：非递归

```
#include <cstdlib>
#include <gl\glut.h>
class GLintPoint
{
public:
    GLint x;
    GLint y;
    GLintPoint(GLint a, GLint b) {
        x = a;
        y = b;
    }
};
void myInit(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glColor3f(1.0, 0.0, 0.0);
    glPointSize(2.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 600.0, 0.0, 600.0);
}
void drawDot(GLint x, GLint y) {
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}
void myDisplay() {
    glClear(GL_COLOR_BUFFER_BIT);
    GLintPoint T[3] = { GLintPoint(10, 10), GLintPoint(600, 10), GLintPoint(300, 600) };
    int index = rand() % 3;
    GLintPoint point = T[index];
    drawDot(point.x, point.y);
    for (int i = 0; i < 5000; i++) {
        index = rand() % 3;
        point.x = (point.x + T[index].x) / 2;
        point.y = (point.y + T[index].y) / 2;
        drawDot(point.x, point.y);
    }
    glFlush();
}
int main(int argc, char *argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_SINGLE);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(500, 400);
    glutCreateWindow("Sierphiski gasket");
    glutDisplayFunc(&myDisplay);
    myInit();
    glutMainLoop();
    return 0;
}
```



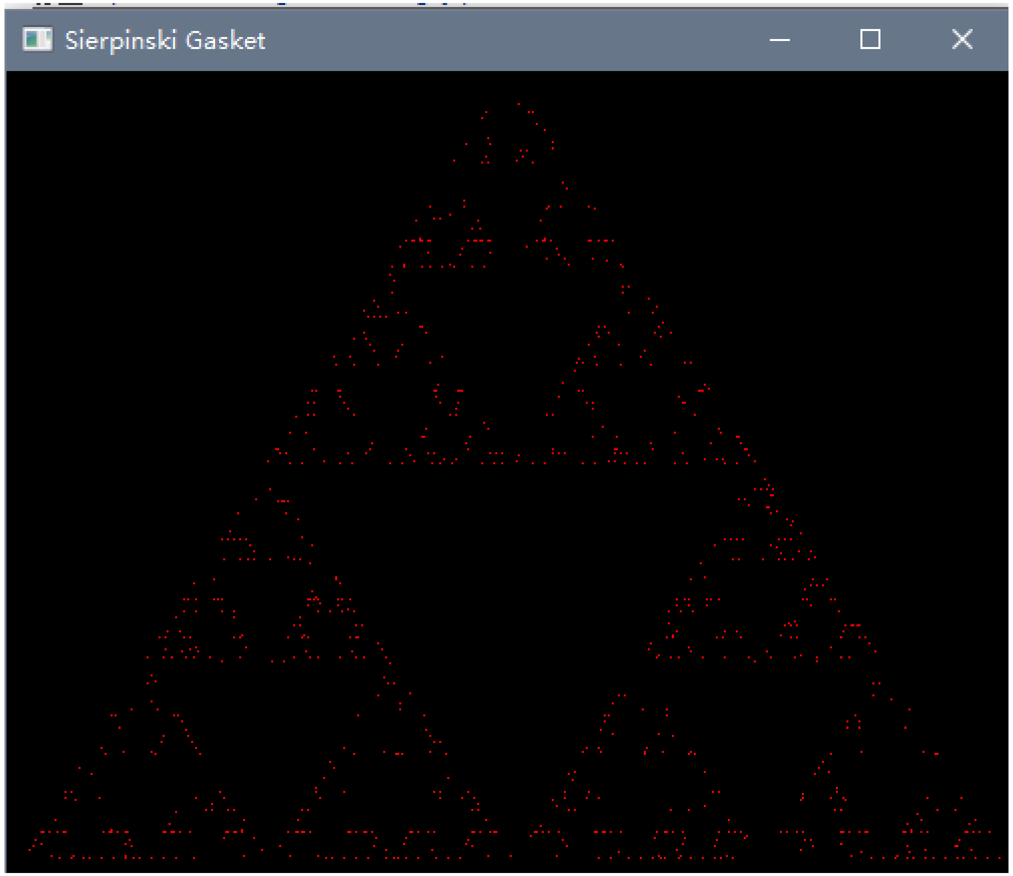
方法2：递归

```
#include <GL/glut.h>
#include<stdlib.h>
/* initial triangle */
GLfloat v[3][2] = { {10.0, 10.0}, {600.0, 10.0}, {300.0, 600.0} };
int n;
void triangle(GLfloat *a, GLfloat *b, GLfloat *c)
/* display one triangle  */
{
```

```c
    glVertex2fv(a);
    glVertex2fv(b);
    glVertex2fv(c);
}
void divide_triangle(GLfloat  *a, GLfloat *b, GLfloat *c, int m)
{
    /* triangle subdivision using vertex numbers */
    GLfloat v0[2], v1[2], v2[2];
    int j;
    if (m > 0)
    {
        for (j = 0; j < 2; j++) v0[j] = (a[j] + b[j]) / 2;
        for (j = 0; j < 2; j++) v1[j] = (a[j] + c[j]) / 2;
        for (j = 0; j < 2; j++) v2[j] = (b[j] + c[j]) / 2;
        divide_triangle(a, v0, v1, m - 1);
        divide_triangle(c, v1, v2, m - 1);
        divide_triangle(b, v2, v0, m - 1);
    }
    else triangle(a, b, c); /* draw triangle at end of recursion */
}
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_TRIANGLES);
    divide_triangle(v[0], v[1], v[2], n);
    glEnd();
    glFlush();
}
void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 600.0, 0.0, 600.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glColor3f(1.0, 0.0, 0.0);
}
void  main(int argc, char **argv)
{
    n = 15;
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 400);
    glutCreateWindow("Sierpinski Gasket");
    glutDisplayFunc(display);
    myinit();
    glutMainLoop();
```

```

}
```


Sierpinski Gasket