1. **问题提出**

通过实现谢尔宾斯基三角形的绘制，掌握opengl的基本使用方法。

1. 算法描述

### Chaos Game

1.用随机的方法（Chaos Game），都可得到谢尔宾斯基三角形：

2.任意取平面上三点A,B,C，组成一三角形

（vertices[3][3]）

3.任意取三角形ABC内的一点P，画出该点

（j = rand()%3;）

4.画出 P和三角形其中一个顶点的中点

(p[0] = (p[0]+vertices[j][0])/2; p[1] = (p[1]+vertices[j][1])/2;)这里应该是p[1]才对，教学在线代码不对，那样只会画出两个固定点。

5.重复1

（算法参考自维基百科，维基百科有:去中心，chaos game，L系统三种方法实现谢尔宾斯基三角形，而chaos game与教学在线代码思路一致，因此在算法后面我补上了教学在线对应的代码）

1. **功能简介**

实现谢尔宾斯基三角形的绘制与展示。

1. **实验结果**

**我修改了一下教学在线的代码，使之可以运行绘制出谢尔宾斯基三角形，修改的地方如下：**

1.增加#include<stdlib.h>语句实现rand()方法，

2.将glutInitAWindowSize(500,500);改为glutInitWindowSize(500,500);（去掉A）

3.将p[0] = (p[0]+vertices[j][0])/2; p[0] = (p[1]+vertices[j][1])/2;改成p[0] = (p[0]+vertices[j][0])/2; p[1] = (p[1]+vertices[j][1])/2;

4. C:\Users\Administrator\AppData\Roaming\Tencent\Users\1418659400\QQ\WinTemp\RichOle\53LB7H[UO8X~%9~UNRDH8E8.png先初始化再调用函数（感觉这样比较合理）

5.在display里 C:\Users\Administrator\AppData\Roaming\Tencent\Users\1418659400\QQ\WinTemp\RichOle\V9BDA4RL_TLX7G]00~Q}LU4.png才能显示背景色白色。

6.感觉红色不好看，在myinit里换成了前景色绿色C:\Users\Administrator\AppData\Roaming\Tencent\Users\1418659400\QQ\WinTemp\RichOle\J3KU1PH5FRAP5`J5KMA_39Q.png

7.为了使图案更清晰，加大了点的size，C:\Users\Administrator\AppData\Roaming\Tencent\Users\1418659400\QQ\WinTemp\RichOle\OJ(9HE876A14A1@Q$ULD@1O.png

**最终代码如下：**

#include<stdlib.h>

#include <GL/glut.h>

void myinit()

{

// attributes

glClearColor(1.0,1.0,1.0,1.0);

glColor3f(0.0,1.0,0.0);

// set up viewing

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 50.0,0.0,50.0);

glMatrixMode(GL\_MODELVIEW);

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

GLfloat vertices[3][3] = {{0.0,0.0,0.0}, {25.0, 50.0,0.0}, {50.0,0.0,0.0}};

// an arbitrary triangle in the plane z = 0;

GLfloat p[3] = {7.5, 5.0, 0.0};

// or set to any desired initial point inside the triangle;

int j,k;

int rand();

glPointSize(2.0);

glBegin(GL\_POINTS);

for (k = 0; k < 5000; k++)

{

/\* pick a random vertex from 0,1,2\*/

j = rand()%3;

// compute new location;

p[0] = (p[0]+vertices[j][0])/2;

p[1] = (p[1]+vertices[j][1])/2;

// display new point

glVertex3fv(p);

}

glEnd();

glFlush();

}

void main(int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(500,500);//drop out A

glutInitWindowPosition(0,0);

glutCreateWindow("Simple OpenGL Example");

myinit();

glutDisplayFunc(display);

glutMainLoop();

}

**最终运行结果如下**

