# 多边形的扫描转换程序设计报告

# 姓名：陶逸群 学号：1704212

## 程序说明

本程序可以通过鼠标在屏幕上绘制多边形的顶点，利用多边形扫描转换算法对多边形内部区域进行填充。程序运行时通过单击鼠标左键确定多边形顶点，然后单击鼠标右键进行填充。

使用活化边表的方式进行多边形的填充避免了许多不必要的判断，具体实现参见代码注释。

## 算法描述

获得多边形的顶点信息后，建立按y值的不同建立分类边表ET，之后建立活化边表：

(1)对于扫描线Y=yc，若对应的ET中非空，则将其所有的边从ET 中取出并且插入到边的活化半表中，并对AET中各边按X递增排序；

(2)若相对于当前扫描线的活化边表AET非空，则将AET中的边两两依次配对，即第1，2边为一对，第3，4 边为一对，依次类推，每一对边与当前扫描线的交点所构成的区段位于多边形内，依次将这些区段上的点进行着色；

(3)将当前的扫描线的纵坐标Y累加1，即Y=Y+1；

(4)将边的活化链表AET中满足Y=ymax的边删去；

(5)将边的活化链表AET中剩下的每一条边的X域累加ΔX，即X=X+ΔX；

(6)重复执行(1)，直到活化边表为空。

## 数据结构及全局变量的说明

//点类

class Point

{

public:

float x, y;

void SetXY(float x, float y) {

this->x = x;

this->y = y;

}

};

//边结构体

typedef struct Edge{

float y\_max,x;

float dx;

Edge \*next;

}Edge;

static int Pointsum = 0;//多边形的点数

static float eps=1e-10;//误差，用于初始化分类边表时float和int的比较

static Point points[100];//多边形顶点

## 函数说明

//初始化绘图函数

void Init(void);

//绘制点

void SetPoint(void);

//绘制线

void SetLine(void);

//显示函数

void display(void);

// 多边形填充函数

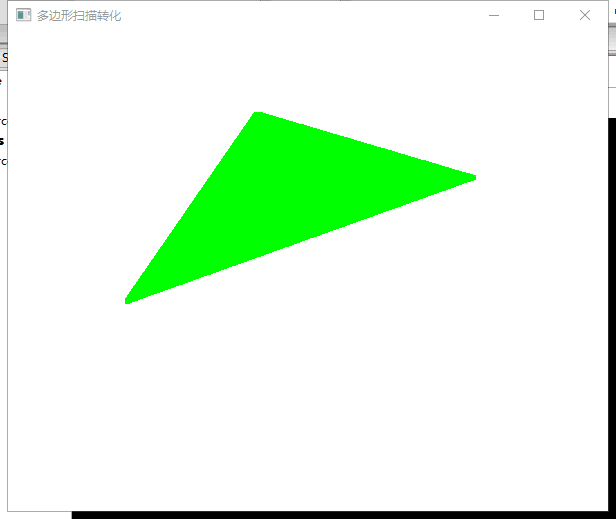
void Filling(void) ;

//添加鼠标事件

void MymouseFunction(int button, int state, int x, int y);

## 运行结果

多边形顶点个数为3时：



多边形顶点个数为12时：

