

# 四边形网格扫描转换

13302010039 童仲毅

## 运行环境

浏览器版本限制：

1. 支持HTML5 Canvas的浏览器。查看[浏览器兼容性](#)。
2. 支持JavaScript (ECMAScript 6)的浏览器。查看[浏览器兼容性](#)。

运行网页前先在 config.js 中选择要运行的测试集和顶点数。

```
const polygon = quadrangle;  
const VERTICES = 4;
```

## 代码结构

代码结构：

- config.js : 测试信息，如顶点位置、画布大小、测试样例等。
- imp.js : 我的实现。

代码块：

1. **Top-level functions:** 根据配置初始化，分别绘制各个多边形和手柄。若多边形被移动，和手柄相连的多边形将重新绘制，以保证移动的多边形在上方。
2. **Drawing functions:** 基本的绘制函数。包括画点、画线、画圆。
3. **Interaction functions:** 交互函数。为canvas添加事件，响应按下、移动、释放操作。
4. **Filling functions:** 扫描转换算法及辅助函数。

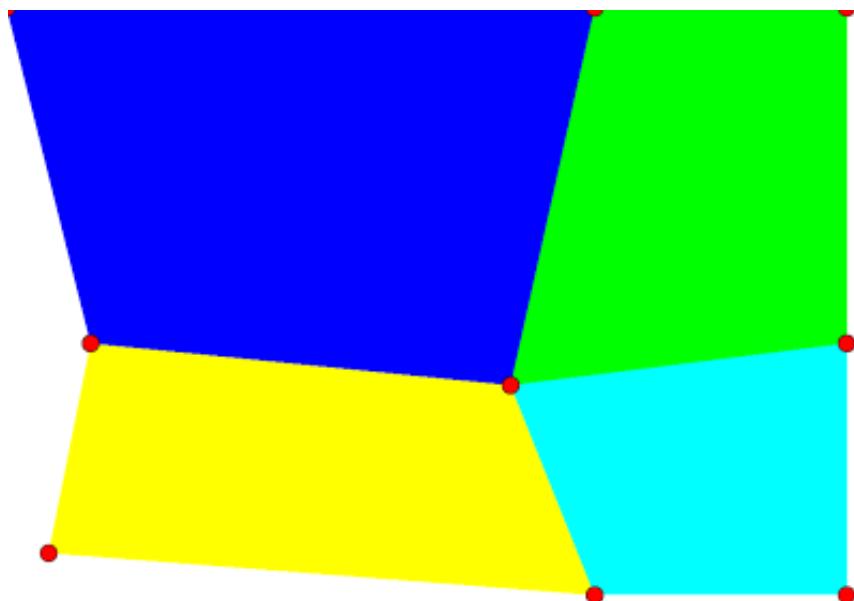
## 算法过程：

1. 这个实现是按行扫描的。首先根据顶点计算最高(y最大)和最低(y最小)的边界。我们在这个边界内扫描。
2. 对于每一行扫描线，计算它与边的交点。优先处理顶点在扫描线上的情况，凸多边形则算一个交点，凹多边形则算两个交点。在后续处理时，跳过这些顶点。接下来计算扫描线和边的其它交点。
3. 对交点集合排序。填充下标满足 $[2n, 2n + 1], n \in N$ 的线段。

## 测试样例

### 人畜无害的四边形

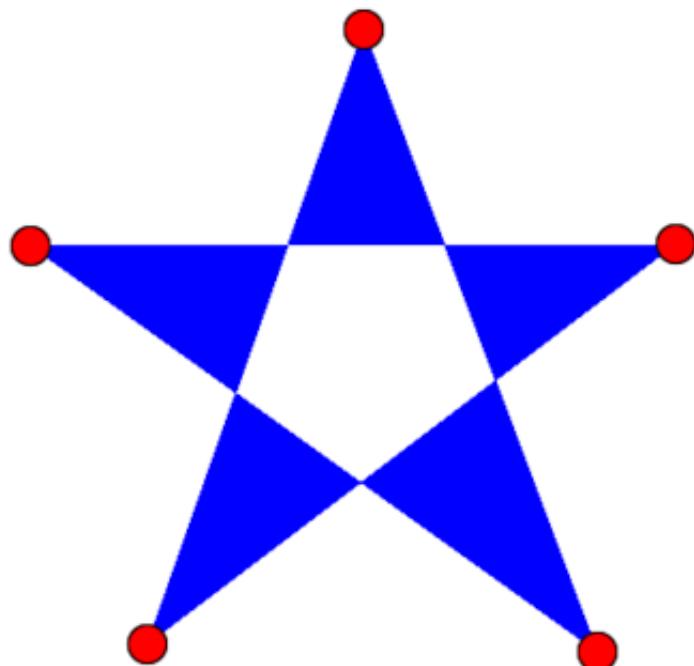
```
// 测试数据
const quadrangle = [
    [0, 1, 4, 3],
    [1, 2, 5, 4],
    [3, 4, 7, 6],
    [4, 5, 8, 7]
];
```



### 花式五边形

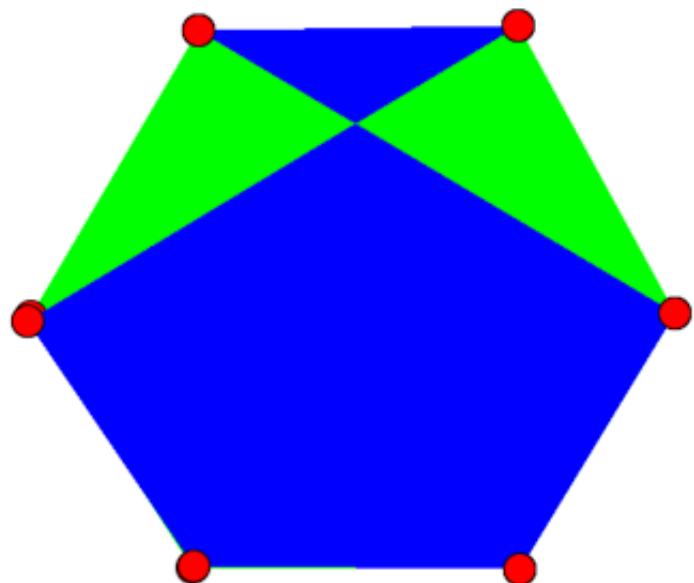
```
// 测试五边形
const pentagon = [
    [0, 1, 4, 3, 5]
```

```
];
```



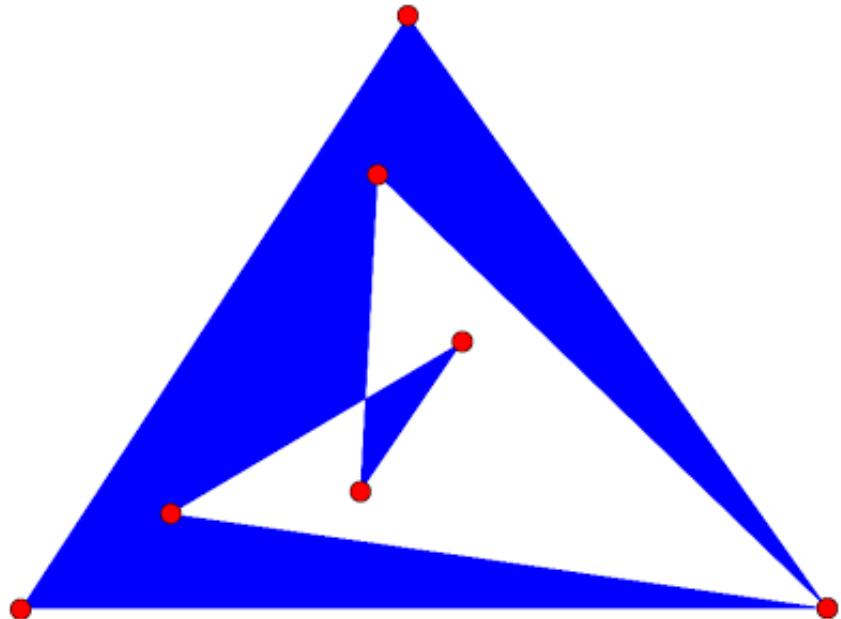
迷之六边形

```
// 测试六边形
const hexagon = [
  [0, 1, 4, 2, 8, 6],
  [1, 0, 4, 2, 7, 5]
];
```



## 什么鬼的八边形

```
// 测试八边形
const octagon = [
    [0, 1, 4, 2, 8, 6, 3, 5]
];
```



另外，凸多边形、凹多边形、含内环的多边形、点在边上这些特殊情形都经过了充分的测试。

## 仍存在的问题

1. 这份代码是凭感觉写下来的，写完之后对照一些教程发现数据结构和算法过程不够清晰。比如扫描线算法中的活性边表(AET)、新边表(NET)等在我的实现中都处理得不够优雅。
2. 主要测试了四边形，没有对所有特殊情形构建测试。

## 课程建议

就是感觉课程进度比较慢？Pj挺好玩的，助教挺萌的：)