

# 《Weiler-Atherton任意多边形裁剪算法》实验报告

## 1、程序的编译方式及可执行文件的使用方式

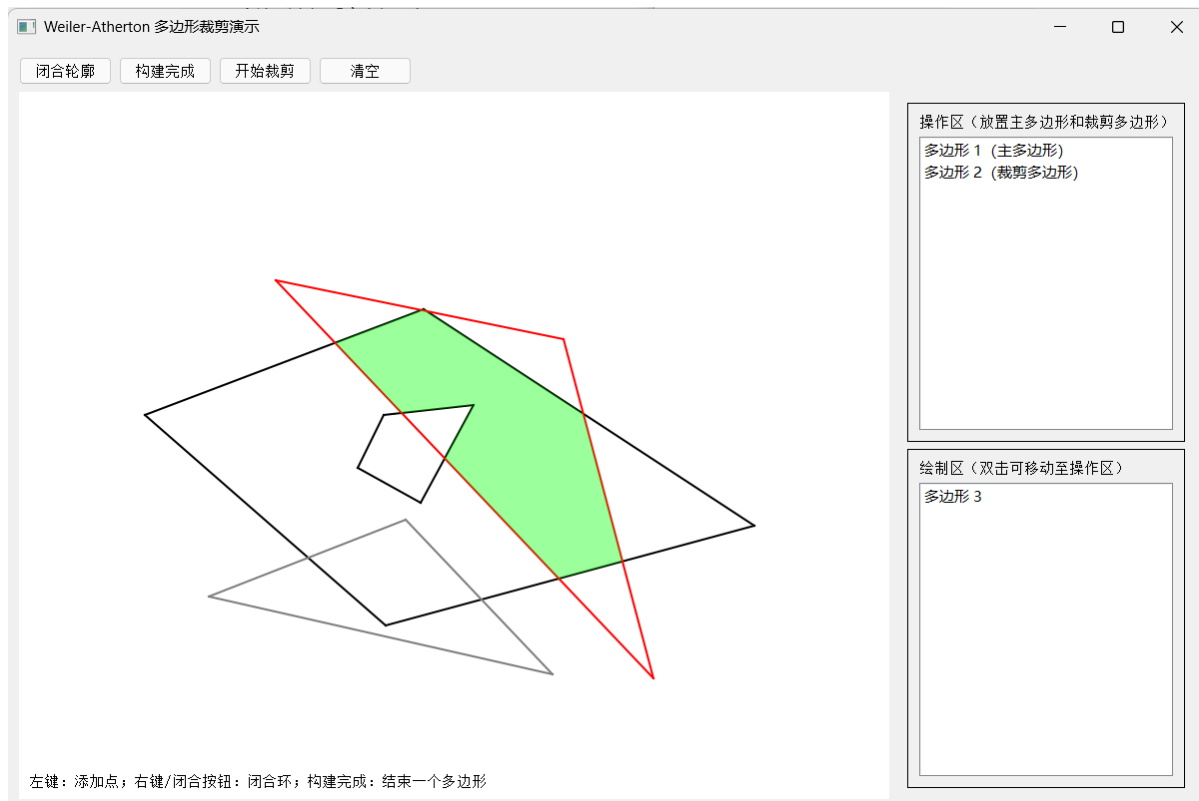
本程序使用python+pyQ5进行界面的编写和交互，需要 PyQt5: `pip install PyQt5`

运行命令：

```
python main.py
```

可执行文件 `main.exe` 在 `dist` 目录中，双击打开即可运行。

## 2、程序的交互方式



### (1) 整体界面

- 上方有四个控制按钮，分别为“闭合”、“构建”、“开始裁剪”、“返回裁剪”；
- 左下方是一个图形窗口，用于用户交互及结果显示；
- 右下方分别有“操作区（放置主多边形和裁剪多边形）”和“绘制区（放置绘制完成的多边形）”，下面放置构建成功的多边形条目，通过双击“绘制区”的多边形条目可将其移动到“操作区”中。

### (2) 交互操作

1. 在画布上**单击鼠标左键**，选点画外环；
2. **单击鼠标右键**，或者点击“闭合轮廓”按钮，闭合外环；
3. 继续在画布上**单击鼠标左键**，选点画内环；
4. **单击鼠标右键**，或者点击“闭合轮廓”按钮，闭合内环，可以转到步骤3绘制多个内环；【默认先画一个外环，后面的均为内环】
5. 点击“构建完成”按钮结束当前多边形的绘制。重复1-5可绘制多个多边形，所有构建成功的多边形都默认放置到“绘制区”中。

6. **双击“绘制区”下任意多边形条目**，可将其移动到“操作区”中，同时更新画布上的多边形边框颜色；默认第一次移动进入的多边形为“主多边形”，第二次移动进入的多边形为“裁剪多边形”，且“操作区”最多容纳两个多边形条目；
7. **点击“开始裁剪”按钮**，图形界面进行裁剪结果的显示；
8. 再次**双击“操作区”下任意多边形条目**，可将其放回“绘制区”，同时更新画布上的多边形边框颜色；
9. **点击“清空”按钮**，图形界面清空所有绘制。

### (3) 显示效果

- 对于每个选点，设置为黑色圆形点。
- 对于未闭合轮廓线，初始都显示为蓝色实线。
- 对于已闭合但未构建完成的轮廓线，初始都显示为蓝色虚线。
- 对于构建完成但尚处在“绘制区”的多边形，设置为灰色边框。
- 对于构建完成且选定为“主多边形”的多边形，设置为黑色边框。
- 对于构建完成且选定为“裁剪多边形”的多边形，设置为红色边框。
- 对于裁剪完成的结果区域，以绿色进行区域填充。

## 3、算法实现方法和细节

### (1) Weiler-Atherton前置知识

- 被裁剪多边形为**主多边形**，裁剪窗口为**裁剪多边形**
- 顶点序列方向：**外环逆时针、内环顺时针**
- **入点**：主多边形边界由此进入裁剪多边形；**出点**：裁剪多边形进入主多边形

### (2) 裁剪流程

- 如果还有未跟踪过的交点，则任取一个作为起点，建空的裁剪结果多边形顶点表，把该交点入结果顶点表。否则算法结束；
- 如果该交点为入点，在主多边形顶点表内跟踪，否则在裁剪多边形顶点表内跟踪；如果跟踪到的是多边形顶点，将其加入结果顶点表，继续跟踪，直到遇到新的交点。重复该步骤，直到回到起点。

### (3) 关键判断

- 判断两线段相交：使用克拉姆法则

对于直线方程  $(y_2 - y_1) * x + (x_1 - x_2) * y = x_1 * y_2 - x_2 * y_1$ ，  
形式如  $Ax + By = C$ ， $Dx + Ey = F$ ，再换为矩阵表示求解即可。

- 判断环的绘制方向：使用鞋带公式，计算有向面积之后，如果面积为正，则是逆时针；为负则是顺时针。**特别需要注意的是**，图形系统（如Canvas、SVG、OpenGL的屏幕空间）使用屏幕坐标系，即Y轴向下，所以判断的时候代码逻辑刚好反过来。
- 判断点是否在环内部：使用射线法，判断该点发射的一条水平射线，与多边形边相交的交点个数，如果为偶数，则点不在环的内部。

## 附：代码结构说明

```
weiler-atherton/  
├─ main.py          # 程序入口：创建窗口与信号连接  
├─ gui.py           # PyQt5 界面与交互逻辑（按钮、列表、画布）  
├─ canvas.py        # 画布绘制 + 鼠标事件（绘点/闭合/点击）  
├─ geometry.py      # 基本几何工具（向量、相交、点内多边形、方向）  
└─ weiler_atherton.py # weiler-Atherton裁剪算法实现
```