Voronoi Diagram

資工碩一 韓定紘 M103040011

軟體規格書

輸出與輸入（資料）規格

輸入

1. 在畫布上用滑鼠任意點擊，畫布大小為630\*630
2. 按下「open」中的「input」，讀入「輸入文字檔」
3. 按下「open」中的「output」，讀入「輸出文字檔」，直接顯示結果到畫布

輸出

1. 按下「save」儲存執行結果(該按鍵只有在成功執行過「run」或「step」後才有效果)

輸入文字檔格式:

先有一列為輸入點數N，接下來N列為輸入點(x y)，即為一組輸入，若符合此格式，一個輸入檔可有多組輸入，以下為範例(N=3)

3

200 200

300 200

400 200

輸出文字檔格式:

以P x y表示圖中的點，E x1 y1 x2 y2表示圖中的邊，裡面的值皆以lexical order排序，以下為輸入範例的輸出

P 200 200

P 300 200

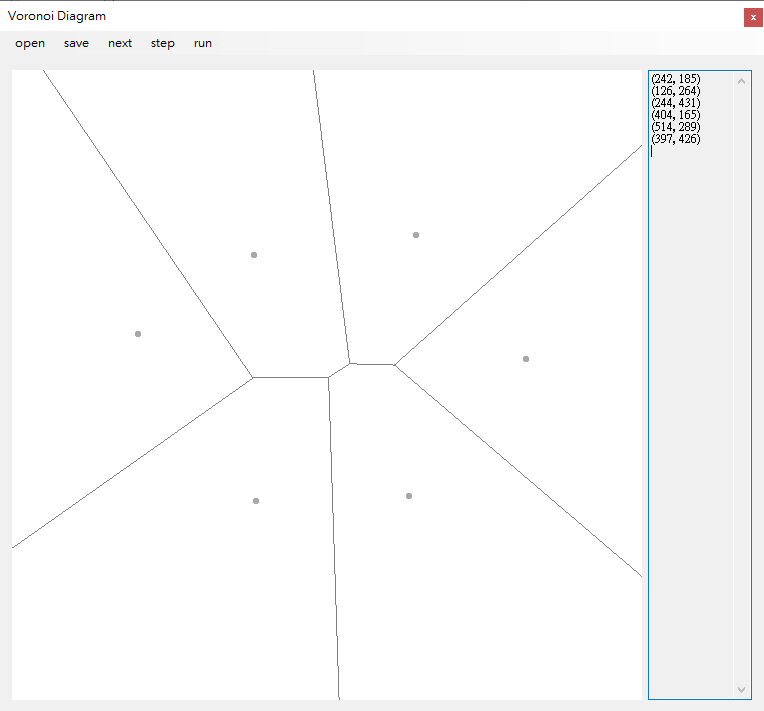
P 400 200

E 250 -62800 250 63200

E 350 -62800 350 63200

功能規格與介面規格

主介面



功能:

1. open:
   * input: 讀取輸入文字檔
   * output: 讀取輸出文字檔
2. save: 儲存執行結果
3. next: 讀取下一筆資料(輸入文檔有多組輸入時)，或是清空畫布
4. step: 針對畫布上的點逐步顯示繪製過程
5. run: 針對畫布上的點顯示最終結果

軟體測試規劃書

主要是以網頁測資為測試依據，或是隨機在畫布上測試，在執行時用若為3點以下會直接解，4到6點會Divide和Merge一次，7點以上會Divide和Merge多次

軟體說明

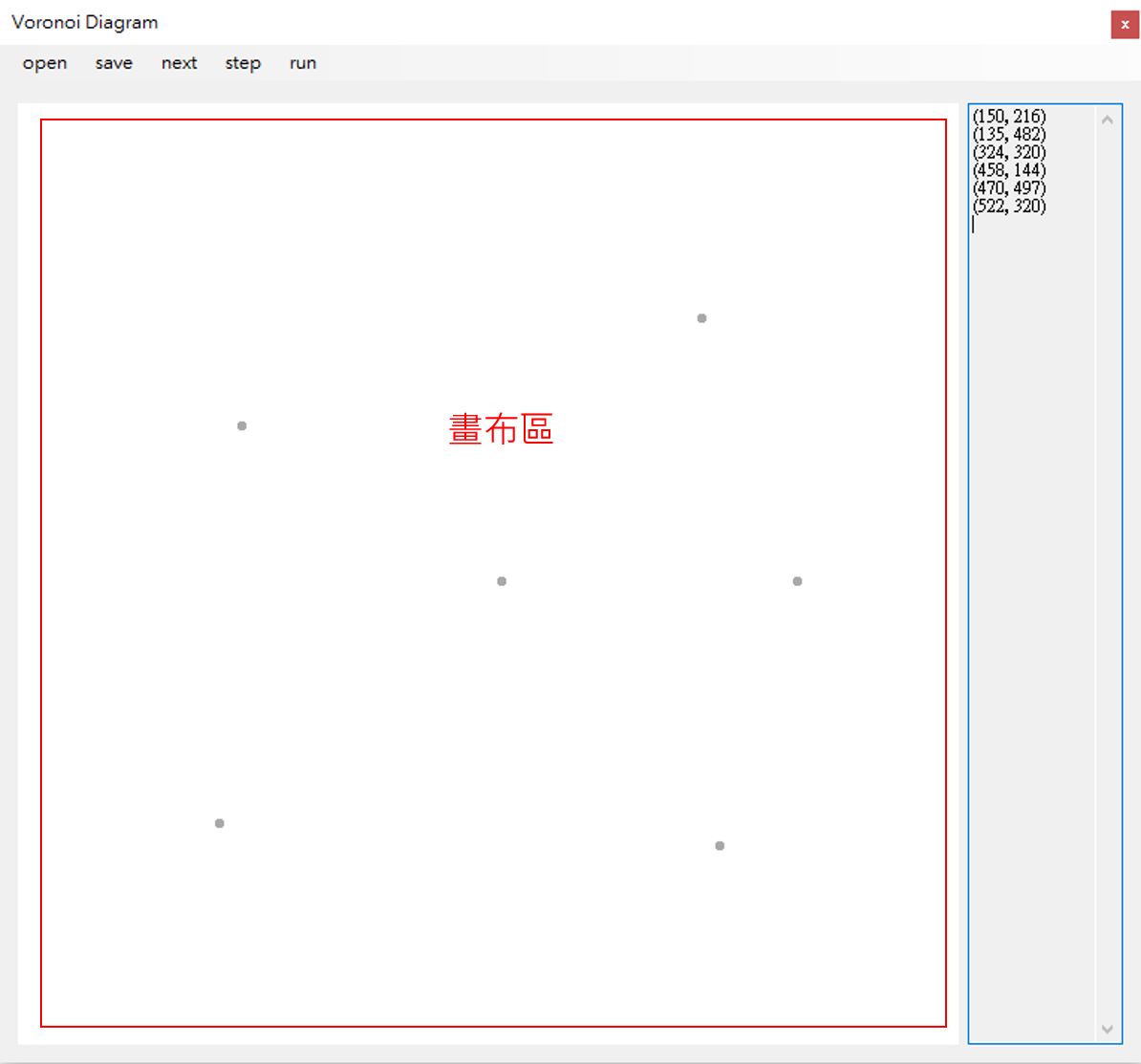
安裝說明

下載主程式後執行「VoronoiDiagram.exe」即可

使用說明

繪製點

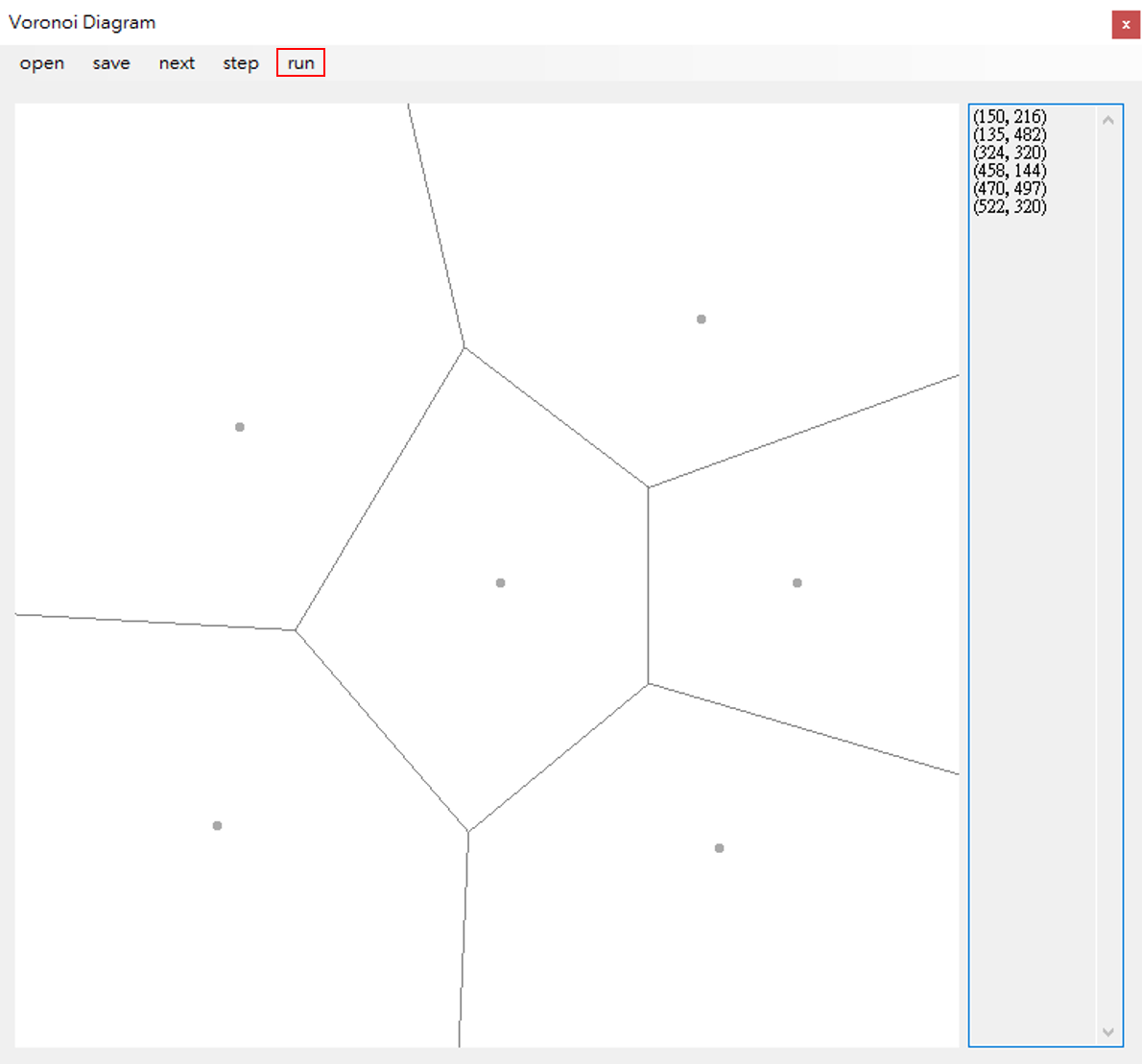
在畫布區繪製點可以直接在畫布上用滑鼠點擊，或是讀取「輸入文字檔」

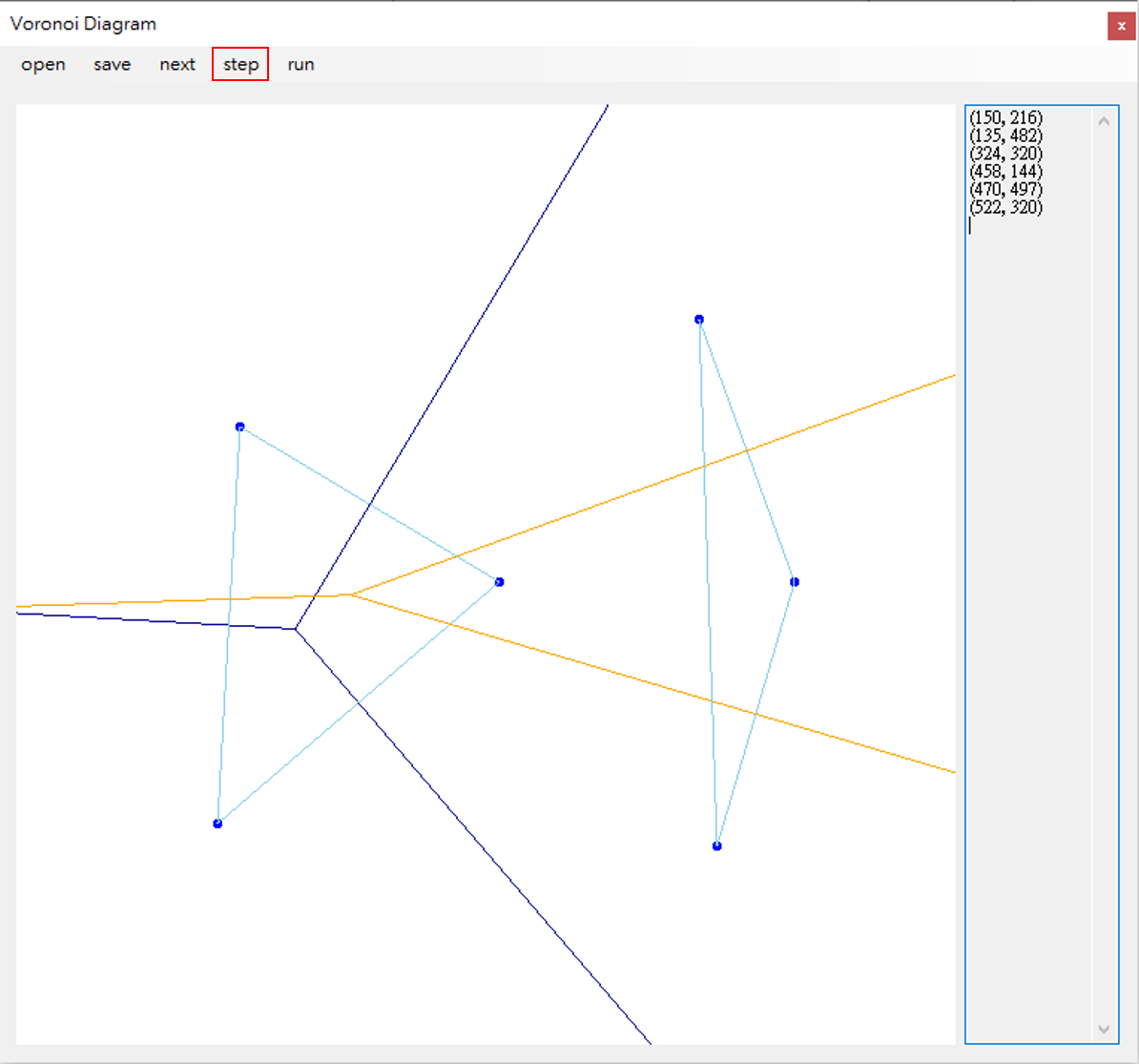


繪製Voronoi Diagram

按下「run」顯示結果

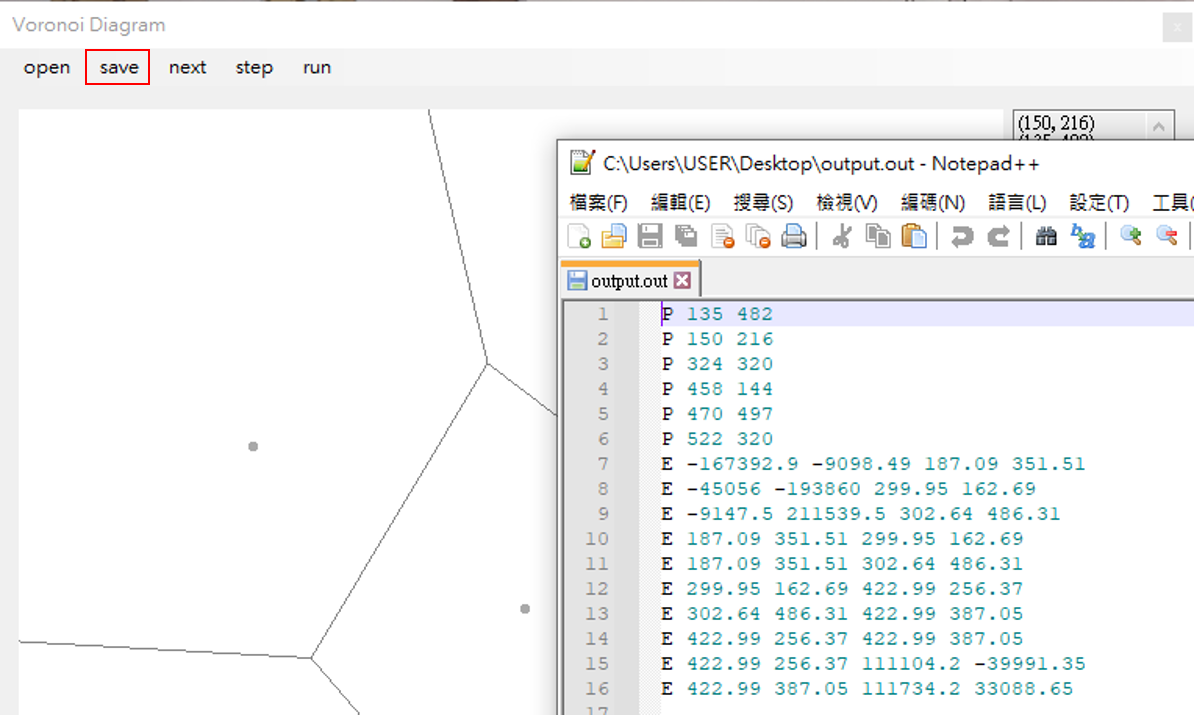
按下「step」逐步顯示結果





輸出結果

按下「save」儲存結果為文字檔



程式設計

資料結構

在C#中有PointF可以用來儲存點座標，因此點的資料結構我就直接使用PointF，另外我又定義了以下兩個資料結構

Edge: 用來儲存線段的資料結構

PointF edgePA; //線段的A點

PointF edgePB; //線段的B點

PointF pointA; //製作此中垂線的A點

PointF pointB; //製作此中垂線的B點

PointEdgeRecoder: 用來儲存Voronoi Diagram的結構，也可以用於紀錄逐步執行

List<PointF> points\_list; //紀錄Voronoi Diagram的點

List<Edge> edges\_list; //紀錄Voronoi Diagram的線段

List<Edge> convex\_list; //紀錄Convex Hull的線段

List<Edge> hyper\_list; //紀錄Hyper Plane的線段

Color point\_color; //紀錄Voronoi Diagram點顯示的顏色

Color edge\_color; //紀錄Voronoi Diagram線段的顏色

Color convex\_color; //紀錄Convex Hull的顏色

Color hyper\_color; //紀錄Hyper Plane的顏色

三點以下

線段

Voronoi Diagram的線段是由兩點的中垂線所組成，中垂線的求法很直覺的會用斜率來算，我一開始也是使用斜率，但是後來發現斜率的方法可能會出現分母為零的情況，想不到怎麼處理。因此我後來採用向量的方法，方法是先算出兩點的向量，接著求垂直該向量的垂直向量，然後將兩點的中點加上足夠倍率的正負垂直向量就可以取得中垂線的兩個端點，即為所求線段

一點

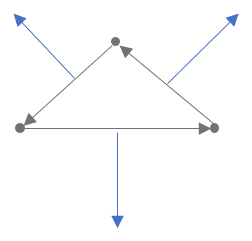
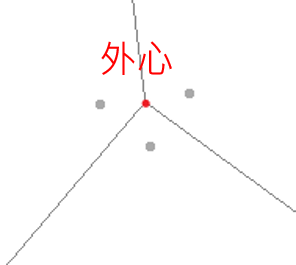
不用做任何處理

兩點

找出兩點的中垂線線段即可

三點

三個點的情況除了基本的中垂線要找外，還要找三點的外心，因為三個中垂線會交於同一點，所以找到其中兩個中垂線的交點即為外心。在這裡線段一端是外心，至於另一端是原本中垂線的其中一端，要決定保留哪一端就成了一個問題。在決定要保留哪一端的方法上，我會先把三個點以逆時鐘排序(可利用差積判斷)，再利用前述的向量法求得中垂線向量，由於三點為逆時鐘排序，此時求得的向量方向會全部都為要去除或是要保留的方向，我把程式調整成方向為要保留的方向(改中垂線向量方向即可)，最後再把外心加上足夠倍率的中垂線向量即為另一端點



三點共線

我程式會去判斷是否三點共線，若為三點共線，直接求得兩中垂線即可

多點

演算法

1. 分割: 將座標排序後，平分點到左右半邊
2. 求Voronoi Diagram: 分別求得左右半邊的Voronoi Diagram
3. 求上下切線: 找到全部點的Convex Hull，橫跨左右兩邊的線段即為上下切線
4. 求Hyper Plane: 從上或下切線進入，碰到Voronoi Diagram轉彎，若沒有再碰到Voronoi Diagram結束
5. 消線: 看Hyper Plane往哪裡轉彎就消除哪邊的線，並把一些完全沒有交線的線段去除