

第二章 自創函數與互動元件

如何有效地提升軟體開發效率是軟體開發者必需面對的課題，本章說明研究者研究 Flash 之特性並結合數學開發而成的自創函數與互動元件，以幫助軟體開發者更有效率的開發互動式教學軟體。

第一節 自創函數

研究者是一位國中數學老師，可以很快的使用數學語言來描述幾何圖形是如何被畫出來的，研究者常想：如果能讓電腦聽懂我的數學語言，如此一來我與電腦之間的隔閡便減少了許多，並且可以提高軟體開發的效率，以畫三角形的外心為例：

(1) 用數學語言描述為：

- ①. 在座標平面上分別標上 A 點(300,200)、B 點(200,400)、C 點(400,400)等三點
- ②. 連接 \overline{AB} 、 \overline{BC} 、 \overline{CA} 線段
- ③. 分別找出 \overline{BC} 、 \overline{CA} 、 \overline{AB} 中點 D、E、F
- ④. 過 D 點作一直線垂直 \overline{BC} 、過 E 點作一直線垂直 \overline{CA} 、過 F 點作一直線垂直 \overline{AB}
- ⑤. 外心 O 即位於 \overline{AB} 、 \overline{BC} 、 \overline{CA} 三邊中垂線的交點上

(2) 套用函數之後的程式語言大略描述如下：

- ①. $pA.x=300, pA.y=200, pB.x=200, pB.y=400, pC.x=400, pC.y=400;$
- ②. **drawLine(pA, pB), drawLine(pB, pC), drawLine(pC, pA);**
- ③. **pD=pointMiddle(pB, pC), pE=pointMiddle(pC, pA), pF=pointMiddle(pA, pB);**
- ④. **drawRec(pD, eqLine(pB, pC)), drawRec(pE, eqLine(pC, pA)), drawRec(pF, eqLine(pA, pB));**
- ⑤. **pO = lineToLine(eqRecline(pD, eqLine(pB, pC)), eqRecline(pE, eqLine(pC, pA)));**

以上這些粗體字即研究者所建立的函數，它的意思大略如下：

- ①. **drawLine(pA, pB)**：連接 AB 線段。
- ②. **pointMiddle(pB, pC)**：找出 B、C 中點座標。
- ③. **eqLine(pB, pC)**：過 BC 兩點的直線方程式。
- ④. **drawRec(pD, eqLine(pB, pC))**：過 D 點畫一直線垂直 BC。
- ⑤. **eqRecline(pD, eqLine(pB, pC))**：過 D 點且垂直 BC 的直線方程式。
- ⑥. **lineToLine**：找出兩直線的交點座標。

套用函數之後的程式碼給了我們一種感覺：寫程式好像在講話一般，差別只在於使用程式語言罷了，而研究者建立這些函數的目的即在於想辦法讓寫程式也能像使用數學語言來描述一般。

雖然 Flash MX 2004 的 Action Script 2.0 功能強大，但它並非是專門用來設計互動式幾何軟體的工具，沒有一個指令可以處理幾何上常用到的旋轉、伸縮、鏡射三種變換，它的繪圖指令更只有畫直線及曲線而已，連個畫圓的指令都沒有，試想我們如何利用它有效率地開發幾何軟體，當每次開發一個軟體需要耗費相當大的心力時，那開發豐富的數位教材是有困難的，所以研究者根據寫幾何軟體常用的描述語言，將好幾個 Action Script 的指令包裝成函數，期望能將撰寫程式的方式改變成像用數學語言在講話一般。

研究者主要利用函數來處理兩件事，其一、依據變數產生新的變數回傳給呼叫函數，如輸入 A、B 兩點座標位置，可產生向量 AB 的變數或直線 AB 的直線方程式變數；其二、依據變數描繪圖形，如輸入 A 點座標位置及長度 100 像素，即可畫出以 A 點為圓心半徑 100 像素的圓。

在建立函數之前，研究者考慮基礎幾何上常用的變數有點、向量、長度、角度、直線方程式及圓方程式，後續的工作則是描述這些變數，並建立一套命名法則，以便讓程式閱讀者可以很容易的了解該變數所代表的內涵。以下分別說明本研究中所使用之命名法則及存取變數資料之方式：

(1) 命名法則：

變數種類	命名法則	舉例
點	以 p 為開頭	pA、pB 分別表示 A、B 兩點的座標
向量	以 vec 為開頭	vecAB、vecBA 分別表示 AB 向量及 BA 向量
長度	以 dis 為開頭	disAB 表示 A、B 兩點間的距離
角度	以 ang 為開頭	angBAC 表示 $\angle BAC$ 的度數
直線方程式	以 line 為開頭	lineAB 表示通過 A、B 兩點的直線方程式
圓方程式	以 circle 為開頭	circleOA 表示以 O 為圓心，A 為圓上一點的圓方程式

(2) 存取變數資料：

- ①. 點：它被看作是座標平面上的一個位置，可以用兩個數字來表示座標平面上任意點的 x 座標及 y 座標，所以用「陣列」來儲放這兩個數字，而 Action Script 又容許使用者用「元素名稱」來儲放資料，所以若要表示座標平面上 A(200, 300)的位置時，可先令 pA 為一個陣列，再用 pA.x 儲放 A 點的 x 座標值 200 及 pA.y 儲放 A 點的 y 座標值 300。
- ②. 向量：使用「陣列」存取向量的值，假設 vecAB 是一個存取 AB 向量的陣列，則 vecAB.x 存放 AB 向量的 x 分量，vecAB.y 存放 AB 向量的 y 分量。
- ③. 長度：使用「數值」變數存取長度的值。
- ④. 角度：使用「數值」變數存取角度的值。
- ⑤. 直線方程式：使用「陣列」存取直線方程式，假設 lineAB 是一個存取 AB 直線方程式的陣列，則 lineAB.a、lineAB.b、lineAB.c 分別表示 AB 直線方程式 $ax+by+c=0$ 的 a、b、c 值。
- ⑥. 圓方程式：使用「陣列」存取圓方程式，假設 circleOA 是一個存取以 O 為圓心， \overline{OA} 為半徑的圓方程式的陣列，則 circleOA.a、circleOA.b、circleOA.r

分別表示圓 O 方程式 $(x-a)^2+(y-b)^2=r^2$ 的 a、b、r 值。

兩千多年前的古希臘數學家歐幾里德，他利用五個公設便能掌握當時所有的幾何知識，並且根據這些已知推出新的幾何知識，這五個公設就像有生命一般，它能一直長出新的東西出來，直到它的極限為止。根據這個想法，研究者首先解決「點」的問題，當有了「點」便可以利用兩個點來建構求兩點「向量」的函數或計算兩點間「距離」的函數，當有了「點」、「向量」及「距離」函數便可建構出求「角度」的函數，有了「點」、「向量」、「距離」及求「角度」的函數，便可建構出「旋轉」變換的函數，就這樣函數一個接著一個被創造出來，並且若遇到新的需要也能很快的利用舊有函數組合成新函數來解決。以下為本研究經常使用的函數，說明如下：

(1) 點、向量、距離、角度、內(外)積

1	語法	point(moviename:MovieClip)
	說明	回傳影片 moviename 的座標
2	語法	getMouse ()
	說明	回傳滑鼠的座標位置
3	語法	putMovie(moviename:MovieClip, x:Number, y:Number)
	說明	將影片 moviename 放至(x, y)的座標位置
4	語法	vector (pA:Array, pB:Array)
	說明	回傳 AB 向量

5	語法	vecNormal (pA:Array, pB:Array)
	說明	回傳 AB 向量的法向量
6	語法	angle (pA:Array, pB:Array, pC:Array)
	說明	回傳 $\angle ABC$ 的度數
7	語法	inner_product (vecAB:Array, vecAC:Array)
	說明	回傳向量 vecAB 及 vecAC 的內積
8	語法	outer_product (vecAB:Array, vecAC:Array)
	說明	回傳向量 vecAB 及 vecAC 的外積
9	語法	disPointPoint (pA:Array, pB:Array)
	說明	回傳 pA、pB 兩點間的距離
10	語法	disPointLine (pC:Array, lineAB:Array)
	說明	回傳點 pC 至直線 lineAB 的距離

(2) 中點、垂點、分點、交點

1	語法	pointMiddle (pA:Array, pB:Array)
	說明	回傳 pA、pB 兩點的中點
2	語法	pointRec (pC:Array, lineAB:Array)
	說明	回傳點 pC 在直線 lineAB 的垂點座標
3	語法	pointSeparate(pA:Array, pB:Array, coem:Number, coen:Number)
	說明	回傳與 pA、pB 的距離比分別為 coem 及 coen 的一點座標
4	語法	lineToLine (line1:Array, line2:Array)
	說明	回傳直線 line1 及 line2 的交點座標
5	語法	lineToCircle (line:Array, circle:Array)
	說明	回傳直線 line 與圓 circle 的交點座標

6	語法	circleToCircle (circle1:Array, circle2:Array)
	說明	回傳兩圓 circle1 與 circle2 的交點座標
7	語法	pointTangent (circleO:Array, pA:Array)
	說明	回傳圓外一點 pA 在圓 circleO 的切點座標

(3) 直線與圓的方程式

1	語法	eqLine (pA:Array, pB:Array)
	說明	回傳通過 pA、pB 兩點的直線方程式
2	語法	eqParaline (pC:Array, lineAB:Array)
	說明	回傳通過點 pC 且平行直線 lineAB 的直線方程式
3	語法	eqRecline(pC:Array, lineAB:Array)
	說明	回傳通過點 pC 且垂直直線 lineAB 的直線方程式
4	語法	eqBisectline (pA:Array, pB:Array, pC:Array)
	說明	回傳平分 $\angle ABC$ 的直線方程式
5	語法	eqCircle (pO:Array, disR:Number)
	說明	回傳以點 pO 為圓心且半徑等於 disR 的圓方程式

(4) 旋轉、平移、對稱

1	語法	tranForm(pB:Array,pA:Array,angle:Number,scale:Number)
	說明	回傳以 pA 為中心點將 pB 旋轉 angle 角並伸縮 scale 百分比比例所得到的新座標
2	語法	tranSlip(pA:Array, vecAB:Array)
	說明	回傳將點 pA 平移向量 vecAB 所得到的新座標
3	語法	tranFlipPoint (pA:Array, pO:Array)
	說明	回傳以 pO 為中心點將 pA 作鏡射所得到的新座標

4	語法	tranFlip (pA:Array, lineCD:Array)
	說明	回傳以 lineCD 為對稱軸將 pA 作鏡射所得到的新座標
5	語法	movieRotation(moviename:MovieClip,pA:Array, angle:Number)
	說明	將影片 moviename 以點 pA 為中心點旋轉 angle 的角度。
6	語法	movieSlip (moviename:MovieClip, vecAB:Array)
	說明	將影片 moviename 平移向量 vecAB

(5) MovieClip 的繪圖方法

1	語法	drawLine (pA:Array, pB:Array, style, wide)
	說明	在 MovieClip 上畫出通過 pA、pB 兩點的直線，當 style=0 時畫線，style=1 時畫直線，style=2 時畫射線
2	語法	drawBisect (pA:Array, pB:Array, pC:Array, style, wide)
	說明	在 MovieClip 上畫出平分 $\angle ABC$ 的直線
3	語法	drawRec (pC:Array, lineAB:Array, style, wide)
	說明	在 MovieClip 上畫出通過 pC 且垂直直線 lineAB 的直線
4	語法	drawPara (pC:Array, lineAB:Array, wide)
	說明	在 MovieClip 上畫出通過 pC 且平行直線 lineAB 的直線
5	語法	drawTriangle(pA:Array, pB:Array, pC:Array)
	說明	在 MovieClip 上畫出以 pA、pB、pC 為頂點的三角形
6	語法	drawQuadrangle(pA:Array, pB:Array, pC:Array, pD:Array)
	說明	在 MovieClip 上畫出以 pA、pB、pC、pD 為頂點的四邊形
7	語法	drawPolygon (pO:Array, pStart:Array, sides:Number)
	說明	在 MovieClip 上畫出以 pO 為中心點，pStart 為一頂點的正多邊形
8	語法	drawCircle (pO:Array, disR:Number)
	說明	在 MovieClip 上畫出以 pO 為圓心 disR 為半徑的圓形

9	語法	drawSector(pO:Array,pA:Array, anglevalue:Number, style)
	說明	在 MovieClip 上畫出以 pO 為圓心 pA 為圓上一點的扇形(style=null 或 0)、弓形(style=1)、圓弧(style=其它不等於 0 或 1 的正整數)

(6) 極座標與平面座標的變換

1	語法	polarToPlane (pO:Array, disR:Number, angle:Number)
	說明	將輸入極座標中心點 pO，半徑 disR，角度 angle，傳回平面座標
2	語法	planeToPolar (pA:Array, pO:Array)
	說明	將 pA 以 pO 為中心點轉換成極座標

(7) 將點固定在特定路徑上

1	語法	getLineXY (lineAB:Array, pMove:Array)
	說明	傳回任意點 pMove 在直線 lineAB 相對應的座標
2	語法	getCircleXY (circle:Array, pA:Array)
	說明	傳回任意點 pMove 在圓 circle 相對應的座標

第二節 互動元件

在製作一個 Flash 影片時，我們常常會將一些元件從元件庫中拉出放入場景，並藉由圖層、影格的運作讓整個影片活絡起來，這樣的方法將會使得整個影片的檔案大小獲得有效控制，並且能套用他人製作的元件或將自己創作的元件供他人使用，以達到資源分享、事半功倍的效果。

Flash 的元件分成三類：圖像(Graphic)、按鈕(Button)、影片片段(MovieClip)，其中 MovieClip 是最重要也最常用，但一般軟體開發者只是將它拿來當作「小動畫影片」使用，研究者為了更進一步提昇 MovieClip 的功用，將 MovieClip 製作成可以和軟體使用者互動的元件。以三角形為例，它不再只是一個單純的靜態圖片或炫麗的動畫影片片段，而是一個可以和使用者互動的元件，使用者可以藉由滑鼠改變三角形的形狀或顏色，也可以讓它隨滑鼠而旋轉、平移、縮放、鏡射，當然也能利用滑鼠將它刪除或複製一個和它完全一樣的三角形，當這樣的互動元件一旦製作完成後，日後只要將它從元件庫拉入場景便能使用，而其他的軟體開發者也能利用這樣功能強大的元件，因此互動式軟體的開發將更有效率。

本研究已製作許多的互動元件，所以當開發一項新的軟體時，只需將這些互動元件排列組合便能完成，以下列出本研究所完成的重要互動元件。

元件名稱	元件參數	功能介紹
三角形	<ol style="list-style-type: none"> ①. pV: 三角形三頂點座標集合的陣列，其中 pV[0]、pV[1]、pV[2] 分別表示三個頂點的座標陣列 ②. fillColor: 三角形的內部顏色 ③. lineColor: 三角形的三邊顏色 ④. tA: 表示三個點的名稱集合的陣列，其中 tA[0]、tA[1]、tA[2] 分別表示三角形三頂點名稱的字串變數 	<ol style="list-style-type: none"> ①. 刪除、複製 ②. 改變三角形內部及三邊的顏色 ③. 改變三角形內角、外角的顏色 ④. 顯示內角、外角的度數，顯示三角形邊長的長度 ⑤. 旋轉、伸縮、鏡射、平移三角形 ⑥. 拖曳頂點改變三角形的形狀，但下列其中一種性質保持不變：面積不變、周長不變、拖曳的頂角不變。 ⑦. 使頂點在下列固定路徑上移動：直線、圓形、多邊形 ⑧. 拖曳任意一邊改變三角形的形狀，但下列其中一種性質保持不變：角度不變、樞紐性質
四邊形	<ol style="list-style-type: none"> ①. pV: 四邊形四頂點座標集合的陣列，其中 pV[0]、pV[1]、pV[2]、pV[3] 分別表示四個頂點的座標陣列 	<ol style="list-style-type: none"> ①. 刪除、複製 ②. 改變四邊形內部及四邊的顏色。 ③. 改變四邊形內角、外角的顏色

	<ul style="list-style-type: none"> ②. fillColor：四邊形的內部顏色 ③. lineColor：四邊形的四邊顏色 ④. tA：表示四個點名稱集合的陣列，其中 tA[0]、tA[1]、tA[2]、tA[3] 分別表示四邊形四個頂點名稱的字串變數 	<ul style="list-style-type: none"> ④. 顯示內角、外角的度數及邊長的長度 ⑤. 旋轉、伸縮、鏡射、平移四邊形 ⑥. 拖曳頂點改變四邊形的形狀，但下列其中一種性質保持不變：面積不變、周長不變、拖曳的頂角不變。 ⑦. 使頂點在下列固定路徑上移動：直線、圓形、多邊形 ⑧. 拖曳任意一邊改變四邊形的形狀，但下列其中一種性質保持不變：角度不變、周長不變
平行四邊形	<ul style="list-style-type: none"> ①. pV：四邊形四頂點座標集合的陣列，其中 pV[0]、pV[1]、pV[2]、pV[3] 分別表示四個頂點的座標陣列 ②. fillColor：四邊形的內部顏色 ③. lineColor：四邊形的四邊顏色 ④. tA：表示四個點名稱集合的陣列，其中 tA[0]、tA[1]、tA[2]、tA[3] 分別表示四邊形四個頂點名稱的字串變數 	<ul style="list-style-type: none"> ①. 刪除、複製 ②. 改變四邊形內部及四邊的顏色。 ③. 改變四邊形內角、外角的顏色 ④. 顯示內角、外角的度數及邊長的長度 ⑤. 旋轉、伸縮、鏡射、平移四邊形 ⑥. 拖曳頂點改變四邊形的形狀，但下列其中一種性質保持不變：對角線長不變、對角線夾角不變 ⑦. 使頂點在下列固定路徑上移動：直線、圓形、多邊形 ⑧. 拖曳任意一邊改變四邊形的形狀，但下列其中一種性質保持不變：面積不變、角度不變、周長不變
圓形	<ul style="list-style-type: none"> ①. pV：圓形圓心及圓上一點座標集合的陣列，其中 pV[0]、pV[1] 分別表示圓心及圓上一點座標陣列 ②. fillColor：圓形的內部顏色 ③. lineColor：圓周的顏色 ④. tA：表示圓心及圓上一點名稱集合的陣列，tA[0]、tA[1] 分別表示圓心及圓上一點名稱的字串變數 	<ul style="list-style-type: none"> ①. 刪除、複製 ②. 改變圓形內部及圓周的顏色。 ③. 顯示半徑

直線	<ul style="list-style-type: none"> ①. pV:表示直線上兩點座標集合的陣列，其中 pV[0]、pV[1]分別表示這兩點的座標陣列 ②. lineColor：直線的顏色 ③. tA：表示直線上兩點名稱集合的陣列，其中 tA[0]、tA[1]分別表示這兩點名稱的字串變數 	<ul style="list-style-type: none"> ①. 刪除、複製 ②. 改變直線的顏色。 ③. 顯示線段的長度 ④. 使頂點在下列固定路徑上移動：直線、圓形、多邊形
點	<p>pV：表示該點的座標陣列</p> <p>fillColor：點的顏色</p> <p>tA：表示該點名稱的字串變數</p>	<ul style="list-style-type: none"> ①. 刪除、複製 ②. 使頂點在下列固定路徑上移動：直線、圓形、多邊形
角度	<ul style="list-style-type: none"> ①. pV:控制角度的三點座標集合的陣列，其中 pV[0]、pV[1]、pV[2]分別依序表示為一邊上的點、該角度的頂點及另一邊上點的座標陣列 ②. fillColor：角度的顏色 ③. lineColor：兩邊的顏色 ④. tA：表示三個點的名稱集合的陣列，其中 tA[0]、tA[1]、tA[2]分別依序表示為一邊上的點、該角度的頂點及另一邊上點名稱的字串變數 	<ul style="list-style-type: none"> ①. 刪除、複製 ②. 改變角度及兩邊的顏色。 ③. 顯示該角的度數 ④. 拖曳一角的两邊時，依照設定可有下列其中一種性質：任意改變角度大小、旋轉角度、轉換成它的對頂角。
角度測量工具	<ul style="list-style-type: none"> ①. 不用設定任何參數 	<ul style="list-style-type: none"> ①. 輸入角度名稱即可得知角度的大小 ②. 顯示或隱藏角度的度數
長度測量工具	<ul style="list-style-type: none"> ①. 不用設定任何參數 	<ul style="list-style-type: none"> ①. 輸入長度名稱即可得知長度的大小 ②. 顯示或隱藏長度的大小
方格紙	<ul style="list-style-type: none"> ①. unit：方格紙的單位大小 ②. lineColor：方格紙格線的顏色 ③. lineAlpha：方格紙線段的 alpha 值 ④. Bd：表示方格紙的大小的陣列。 	<ul style="list-style-type: none"> ①. 任意拖曳 ②. 提供單位調整鈕以便改變方格紙的單位大小 ③. 改變格線的顏色
量角器	<ul style="list-style-type: none"> ①. disR：量角器的大小 ②. degrees：量角器的刻度大小 	<ul style="list-style-type: none"> ①. 顯示度數 ②. 提供調整鈕以便改變量角器的大小 ③. 在多邊形上移動

函數(function)其實只是具有某種功能的小程式，也可以將它看成是一個神奇的小盒子，輸入一些資料之後，可以得到處理過的結果。要解決一個大問題並不容易，但若把它分開成許多小問題，然後將它各個擊破就比較容易。分析開發平面幾何教材經常處理的問題，研究者建構一個有系統的函數，透過函數的使用，教材開發者可以用更自然熟悉的方式撰寫 Flash 程式，而不必讓自己的心思耗費在鎖碎的 ActionScript 指令中，由於所設計的函數具有結構性，因此能不斷的由舊有函數中組合出新的函數。再者，Flash 之所以適合拿來開發數位教材，除了它集美工繪圖及編程開發於一身外，更因它有一個其它程式語言所沒有的類別：MovieClip。研究者利用 MovieClip 強大的功能，善用所創造的函數，撰寫一系列協助開發平面幾何教材的互動元件，藉由不同互動元件的組合，軟體開發者可以更效率的開發數位教材。

