

# 用 VB 來教繪圖應用的經驗

國立台灣師大附中 李啓龍 老師

## 余憶童稚時.....

還記得，在我小學四年級的時候，參加學校的電腦營隊，學著在方格紙上畫圖（如果記得沒錯的話，當時畫的是坦克車，呵呵），然後用 Basic 語言的畫線指令，在解析度極低的綠色螢幕上，一筆一畫地繪製我心目中的坦克車，雖然畫得不好，但是給我的成就感卻不小。記得當時在我幼小的心靈中，出現了一個聲音，『哇！原來我也可以自己用電腦指令來繪圖啊！』

當了老師以後，我也開始利用 Visual Basic 來教學生學習電腦繪圖。學習電腦繪圖，可以訓練學生的空間觀念、邏輯觀念並且可以幫助學生瞭解數學的角度觀念 等效果。在實際教學時，發現學生的學習狀況相當不錯，始終維持著高度的學習興趣，當學生發現自己下的繪圖指令，成功地在螢幕上繪製出圖形時，其學習成就感相當強大；而且，在教學過程中，發現多數同學，都想自己去創造出更具有創意的圖形，這樣的教學經驗與內容，可以提供大家參考。

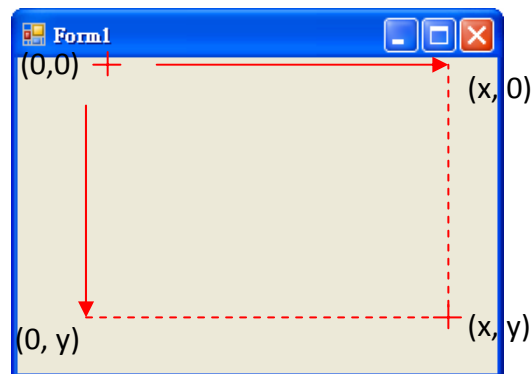
在我的教學經驗中，我發現教授 Visual Basic 的繪圖功能，可以成功開發引導出，學生對於電腦程式設計的熱情，以下介紹我在教導 VB 繪圖功能時的課程內容。

## VB 繪圖功能的課程內容.....

### ● VB 繪圖基礎：

#### ➤ 座標格式：

要瞭解 VB 的繪圖基礎與原理，必須先瞭解其座標格式。在 VB 的座標格式中，表單上所採用的是類似數學上的直角座標系(x,y)，原點(0,0)位於表單的左上角，x 座標向右增加，y 座標向下增加，x,y 皆為整數，如下圖所示。



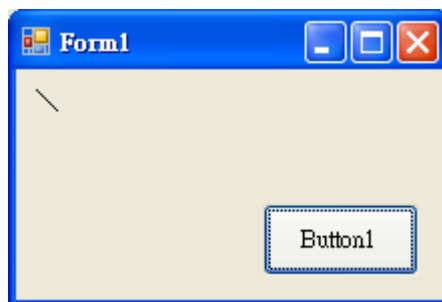
在 VB 的視窗程式裡，您可以在特定的控制項上，以函式來繪圖，如在

表單或圖片方塊上繪圖。但是在繪圖之前，您必須要透過一個繪圖物件 Graphics，才能使用 VB 所提供的繪圖方法。

下面是用繪圖物件 Graphics 在表單上畫一條線的程式碼。

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim g As Graphics '產生一個名為 g 的 Graphics 變數，但 g 並未指派任何實體。
    g = Me.CreateGraphics '請 Me(也就是指 Form1)產生一個 Graphics 物件，並指定到
        'g 的位置，所以現在的 g 指的就是 Form1 所派來的物件。
    g.DrawLine(Pens.Black, 10, 10, 20, 20) '請 g 畫一條線，顏色為黑色、起始點(10,10)、
        '終止點(20,20)。
    g.Dispose() 'g 使用完後，將他釋放。
End Sub
```

執行結果畫面。

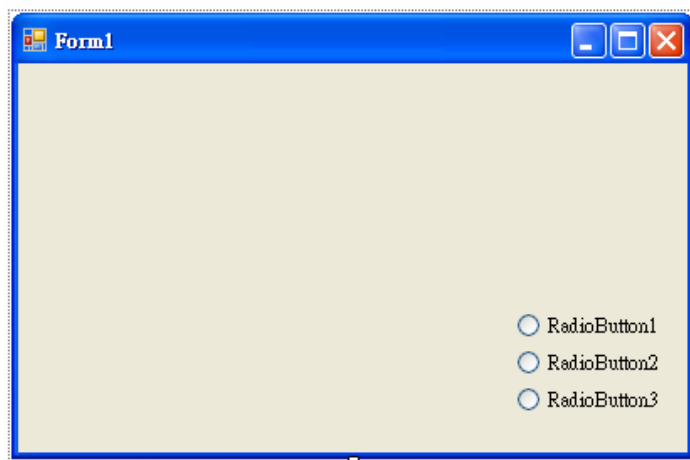


## 程式練習題：小畫家

學習重點：Graphics 物件的使用

### 一、表單配置圖例

新增三個 RadioButton，如下圖。

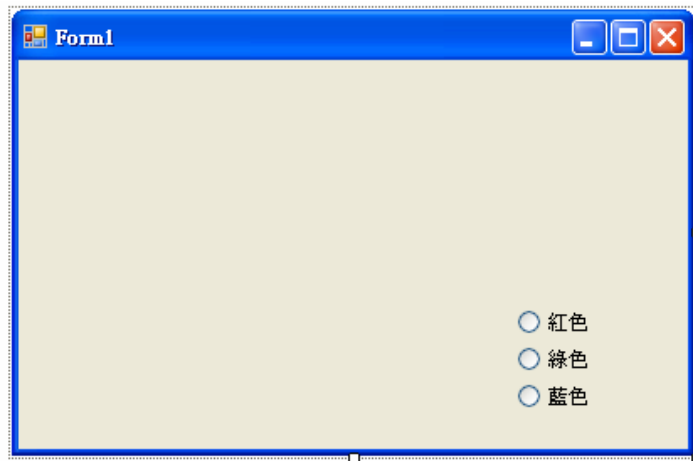


## 二、屬性值設定

屬性值設定表格：

物件	屬性	屬性值
RadioButton1	(Name)	rbtRed
	Text	紅色
RadioButton2	(Name)	rbtGreen
	Text	綠色
RadioButton3	(Name)	rbtBlue
	Text	藍色

設定屬性後的表單。



## 三、編輯程式碼

列數	程式碼
1	Dim Down As Boolean
2	Private Sub Form1_MouseDown(...) Handles Me.MouseDown
3	Down = True
4	End Sub
5	Private Sub Form1_MouseUp(...) Handles Me.MouseUp
6	Down = False
7	End Sub
8	Private Sub Form1_MouseMove(...) Handles Me.MouseMove
9	If Down Then
10	Dim Color
11	If rbtRed.Checked Then
12	Color = Pens.Red
13	Elseif rbtGreen.Checked Then

14	Color = Pens.Green
15	Else
16	Color = Pens.Blue
17	End If
18	
19	Dim g As Graphics
20	g = Me.CreateGraphics
21	g.DrawLine(Color, e.X, e.Y, e.X + 1, e.Y + 1)
22	g.Dispose( )
23	End If
24	End Sub

#### 四、程式碼解說

第 1 列宣告一個全域的 Boolean 變數 Down，記錄目前滑鼠是否按下，其預設初始值為 False。

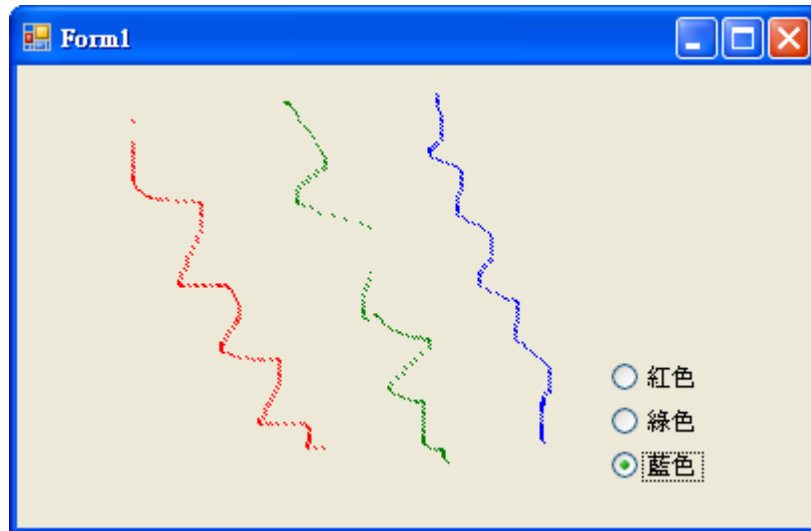
第 2~4 列當滑鼠按下時驅動此事件處理，將 Down 設為 True。

第 5~7 列當滑鼠放開時驅動此事件處理，將 Down 設為 False。

第 8~24 列當滑鼠在表單上移動時驅動此事件處理。若滑鼠在表單上移動並且同時按下了左鍵，Down 會是 True，就會進入 If 的程式區塊。區塊內先用 If 判斷來得知使用者所選的是哪個選項按鈕，是紅色、綠色或是藍色。針對不同的選項狀況，指定顏色至 Color 變數。

在 19~22 列會依據滑鼠的座標，畫出一條條的小線段。第 19 列宣告名為 g 的 Graphics，但目前尚未有實體。第 20 列由 Me，也就是 Form1 來產生 Graphics 實體，指定給 g，此時 g 便代表該 Graphics 實體的名稱。第 21 列透過 g 的 DrawLine 方法來畫線段。其中 e.X 得知目前滑鼠的 X 位置，e.Y 得知目前滑鼠所在的 Y 位置。傳遞 5 個參數項目，包括顏色、線段的 X 起始位置、Y 起始位置、X 終止位置以及 Y 的終止位置。第 22 列將呼叫 Dispose 方法，將 g 所佔用的資源釋放掉。

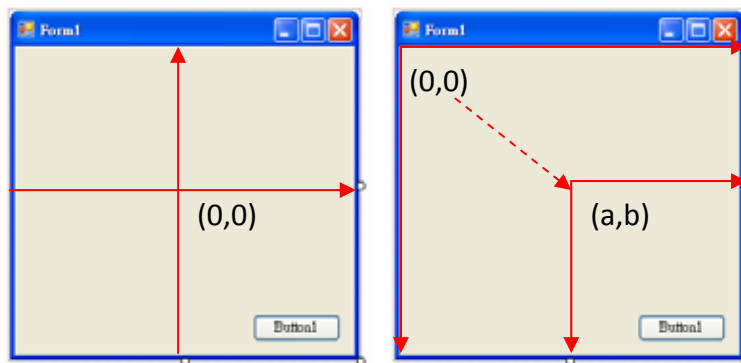
#### 五、執行結果



### 程式練習題：二元一次方程式圖形

學習重點：Graphics 物件的使用、座標轉換

【說明】在表單上以描點的方式，畫出二元一次方程式  $y = x^2$  之圖形。由於數學上所使用的座標系(左圖)，與 VB 所使用的(右圖)，有所差異，所以必須經過座標轉換，才能繪出正確的圖形。



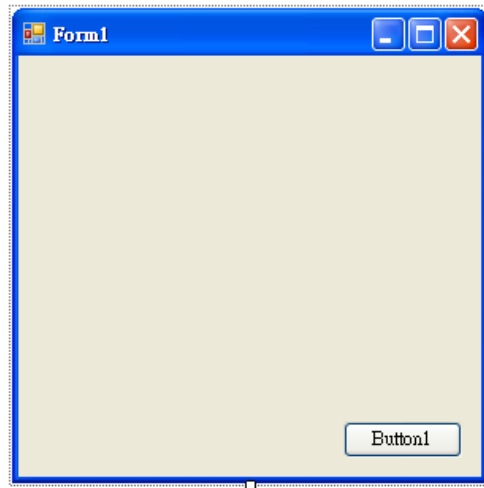
Step1：原點須向右下角移動到中央的位置，其中央位置的座標假設為  $(a, b)$ ；另假設其原始座標為  $(x, y)$ ，因此在 VB 的座標系統其位置為  $(a+x, b+y)$ 。

Step2：由於 VB 的  $y$  軸方向與數學座標的  $y$  軸方向相反，所以  $y$  軸的部分需加上負號修正，所以轉換方式變成  $(a+x, b-y)$ 。

註： $a, b$  通常為長寬之半，如此會使原點移至表單的正中央。

#### 一、表單配置圖例

新增一個 Button，如下圖。



## 二、屬性值設定

屬性值設定表格：

物件	屬性	屬性值
Button1	Text	繪製圖形

設定屬性後的表單。



## 三、編輯程式碼

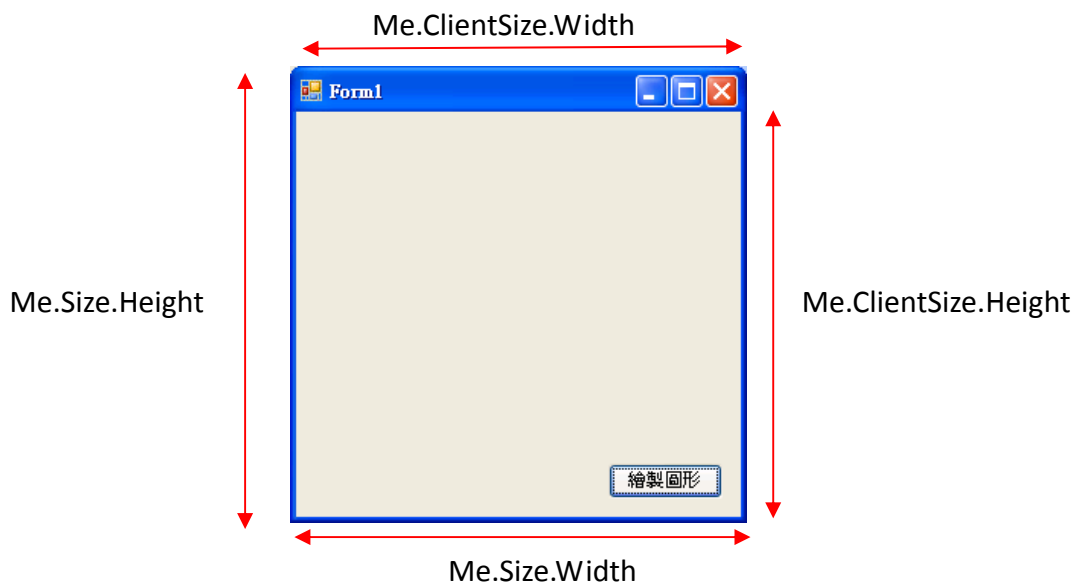
列數	程式碼
1	Private Sub Button1_Click(...) Handles Button1.Click
2	Dim g As Graphics = Me.CreateGraphics
3	Dim a As Integer = Me.ClientSize.Width \ 2
4	Dim b As Integer = Me.ClientSize.Height \ 2
5	Dim x, y As Integer
6	Dim lastx, lasty As Integer
7	

8	<code>g.DrawLine(Pens.Red, 0, b, 2 * a, b)</code> '以紅線畫出 X 軸
9	<code>g.DrawLine(Pens.Red, a, 0, a, 2 * b)</code> '以紅線畫出 Y 軸
10	<code>For x = -a To a</code> '由表單的左至右，畫出 $Y=X^2$ 的圖形，以直線段逼近
11	<code>y = x ^ 2</code>
12	<code>If x &gt; -a Then g.DrawLine(Pens.Black, lastx + a, b - lasty, x + a, b - y)</code>
13	<code>lastx = x</code>
14	<code>lasty = y</code>
15	<code>Next</code>
16	<code>g.Dispose( )</code>
17	<code>End Sub</code>

#### 四、程式碼解說

第 2 列宣告 `g` 做為 `Graphics` 繪圖物件，並呼叫 `Me(Form1 本身)的 CreateGraphics` 方法來產生物件的實體。

第 3、4 列宣告 `a`、`b` 分別代表表單寬度與高度的一半。注意到在這裡我們需要的是 `Me.ClientSize` 的寬度(`Width`)與高度(`Height`)，`Me.ClientSize` 所代表的範圍是標題列以下，使用者真正使用到的部份，有別於表示整個表單大小的 `Me.Size`。



第 5、6 列宣告 `x`、`y` 整數變數來儲存要描繪的點，而 `lastx`、`lasty` 整數變數，用來保留上一次所計算出的 `x`、`y` 值。

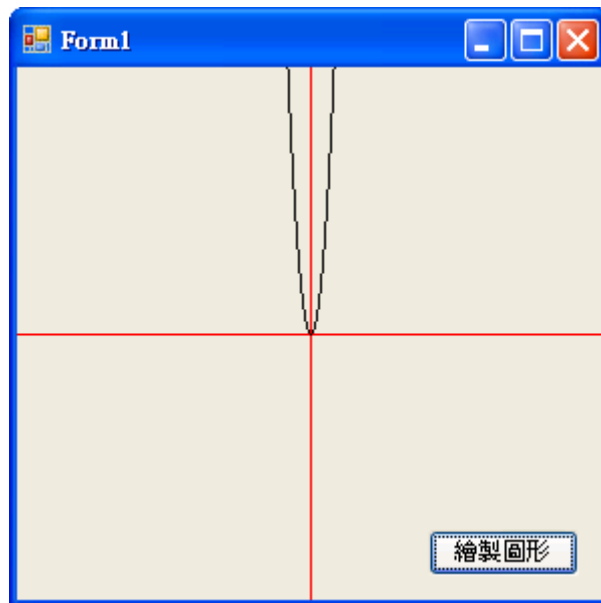
第 8、9 列以紅線標示 X、Y 軸。

第 10~15 列使用迴圈將 `x` 依序代入 `-a` 至 `a` 之間的整數值，在第 11 列算出該 `x` 所對應到的 `y`。第 12 列判斷 `x` 是否大於 `-a`，假如不是，代表現在是第 1 次進入迴圈，還沒有前一次所計算的座標；假如 `x` 大於 `-a`，代表 `lastx`、`lasty`

含有上一回合所計算出來的座標，此時便使用線段，將上回的座標(lastx, lasty)與本次的座標(x,y)連接，以直線段逼近  $y=x^2$  曲線。

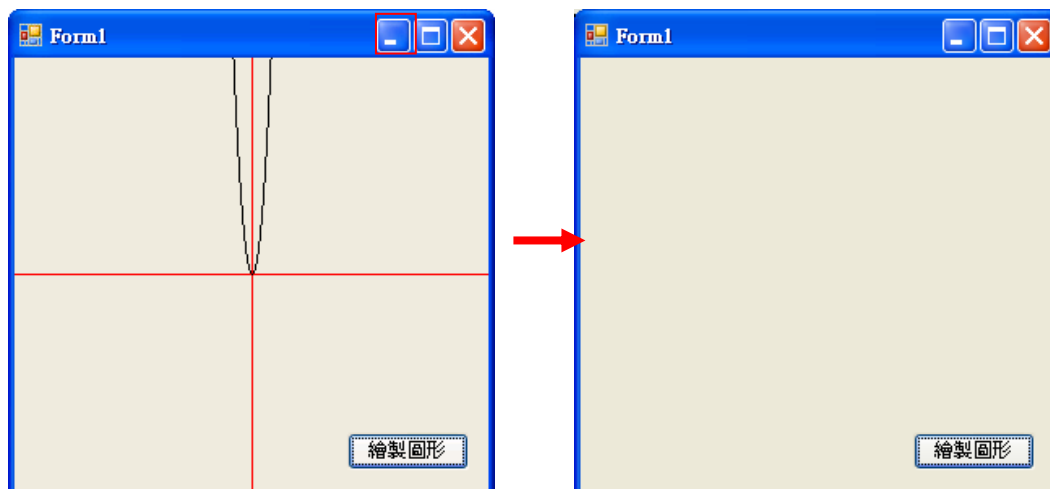
第 16 列將 Graphics 物件 g 釋放。

## 五、執行結果



### 小提示：圖形的常駐

以上面的範例而言，繪製圖形完畢後，若按下表單右上角的「最小化」按鈕，然後復原表單，結果原本的圖形便消失了，如圖：



要解決此問題，可以改成使用 Paint 事件，在圖形被抹去後，將會自動執行重繪，因此將程式修改如下：

1. 將原先程式內容移至 Me.Paint 之事件處理，也就是移到 Form1 物件的(Form1 事件)的 Paint 方法中。





2. 修改第一列的 Me.CreateGraphics 為 e.Graphics，因為在 Paint 事件處理中，已預先產生一個 Graphics 物件，我們不需再要求 Form1 產生另一個 Graphics 物件，以減少系統資源的消耗。

```

1 Private Sub Form1_Paint(...) Handles Me.Paint '修改 1
2     Dim g As Graphics = e.Graphics '修改 2
3     Dim a As Integer = Me.ClientSize.Width \ 2
4     Dim b As Integer = Me.ClientSize.Height \ 2
5     Dim x, y As Integer
6     Dim lastx, lasty As Integer
7
8     g.DrawLine(Pens.Red, 0, b, 2 * a, b)
9     g.DrawLine(Pens.Red, a, 0, a, 2 * b)
10    For x = -a To a
11        y = x ^ 2
12        If x > -a Then g.DrawLine(Pens.Black, lastx + a, b - lasty, x + a, b - y)
13        lastx = x
14        lasty = y
15    Next
16    e.Graphics.Dispose()
17 End Sub

```

- 框線繪圖方法：

使用 Graphics 繪圖物件所提供的 DrawLine 函數，來繪製線段。以下將一一說明 VB 提供的常用繪圖函數。

- 基本類：

## 1. DrawLine 畫線方法：

使用格式：

```
g.DrawLine(Pen 物件, x1, y1, x2, y2)
```



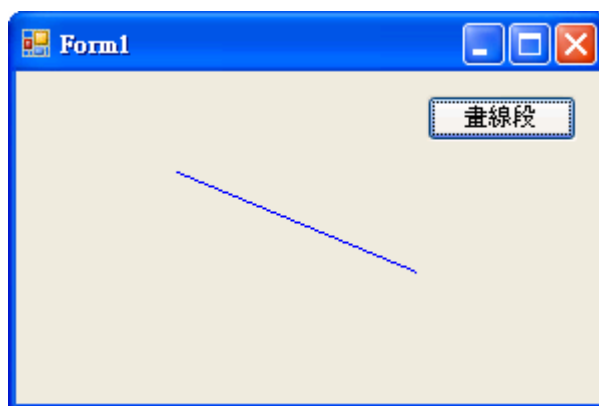
說明：DrawLine 函數會根據傳入的端點座標來繪製線段，起點為(x1,y1)，終點為(x2,y2)。其中 Pen 物件，可以從 Pens 這個集合中取得，例如：Pens.Blue 代表藍色畫筆，Pens.Black 代表黑色畫筆。

範例：

```
Private Sub Button1_Click(...) Handles Button1.Click  
    Dim g As Graphics = Me.CreateGraphics  
    g.DrawLine(Pens.Blue, 80, 50, 200, 100)  
    g.Dispose()  
End Sub
```

繪製起點為(80,50)、終點為(200,100)的線段。

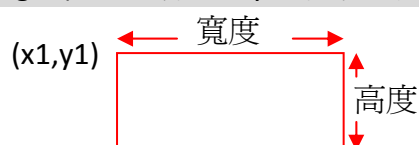
執行結果：



## 2. DrawRectangle 畫矩形框：

使用格式：

```
g.DrawRectangle (Pen 物件, x1, y1, 寬度, 高度)
```

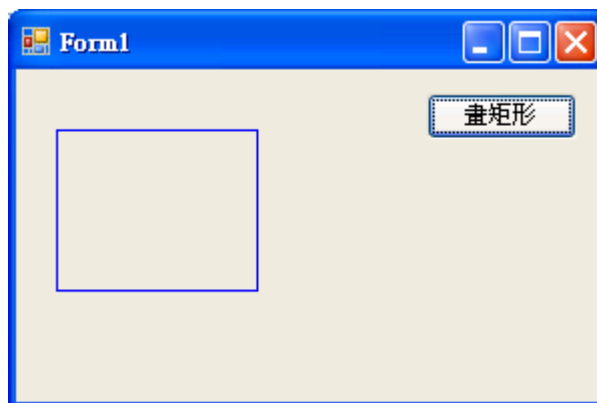


範例：

```
Private Sub Button1_Click(...) Handles Button1.Click  
    Dim g As Graphics = Me.CreateGraphics  
    g.DrawRectangle(Pens.Blue, 20, 30, 100, 80)  
    g.Dispose( )  
End Sub
```

繪製左上角起點為(20, 30)、寬為 100、高為 80 的矩形。

執行結果：

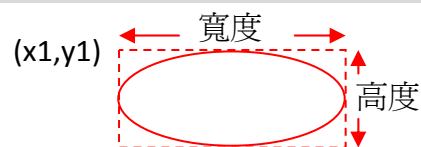


➤ 圓弧類：

1. DrawEllipse 畫橢圓形框：

使用格式：

```
g.DrawEllipse (Pen 物件, x1, y1, 寬度, 高度)
```



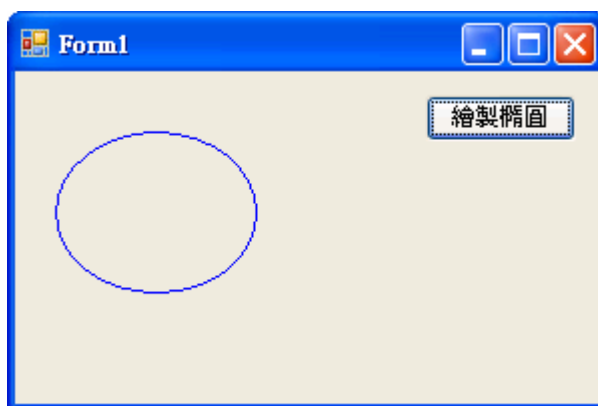
範例：

```
Private Sub Button1_Click(...) Handles Button1.Click  
    Dim g As Graphics = Me.CreateGraphics  
    g.DrawEllipse(Pens.Blue, 20, 30, 100, 80)  
    g.Dispose( )  
End Sub
```

End Sub

繪製一橢圓形，該橢圓可恰好內切於一左上角座標為(20, 30)、寬為 100 且高為 80 的矩形框。

執行結果：

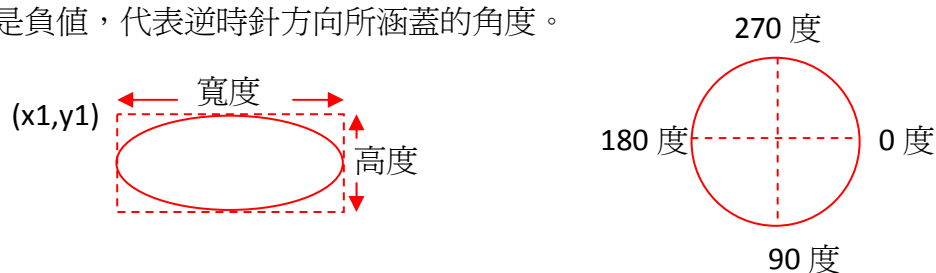


## 2. DrawArc 畫弧形：

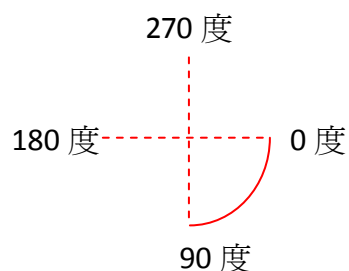
使用格式：

`g.DrawArc (Pen 物件, x1, y1, 寬度, 高度, 起始角度, 角度變化)`

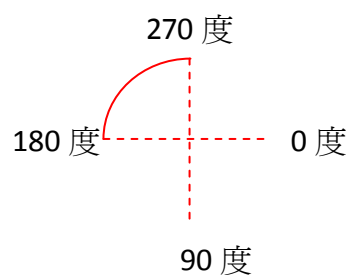
畫弧形的寬度、高度部分與畫橢圓時相同，不同處在於多加上了起始角度與角度變化兩項。起始角度可以是 0~359，若超出，則代表的是同位角。其中 0 度從數學的 X 軸方向開始，以順時針方向增加。角度變化為弧線自起始角開始，順時針所涵蓋的角度範圍，90 度即代表弧線佔了 1/4 圓，180 度則代表半個圓。角度變化也可以是負值，代表逆時針方向所涵蓋的角度。



例如：起始角度為 0，角度變化為 90，所代表的弧形。



例如：起始角度為 270，角度變化為-90，所代表的弧形。



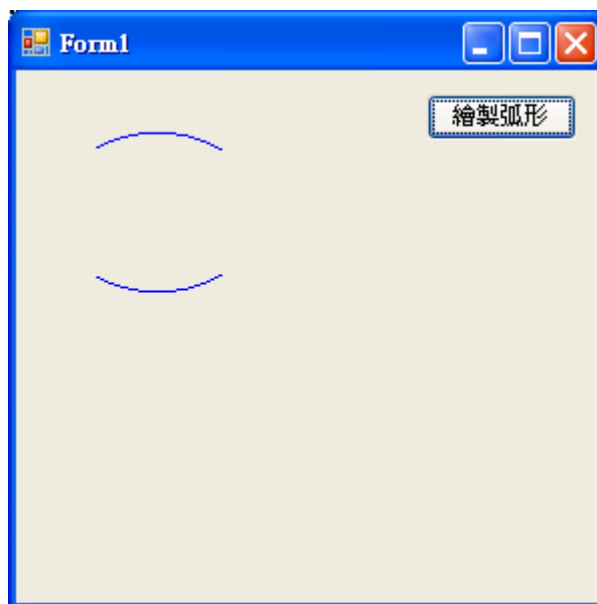
範例：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim g As Graphics = Me.CreateGraphics
    g.DrawArc(Pens.Blue, 20, 30, 100, 80, 45, 90)
    g.DrawArc(Pens.Blue, 20, 30, 100, 80, -45, -90)
    g.Dispose( )
End Sub
```

End Sub

繪製兩弧形，弧形分別屬於橢圓形的 45~135 度，以及 225~315 度(-135~-45 的同位角)，該橢圓可恰好內切於一左上角為(20, 30)、寬為 100 且高為 80 的矩形框。

執行結果：



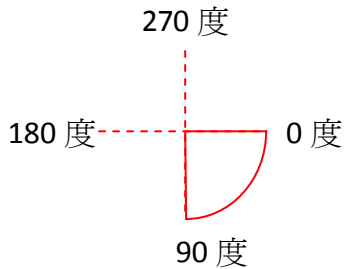
### 3. DrawPie 畫扇形框：

使用格式：

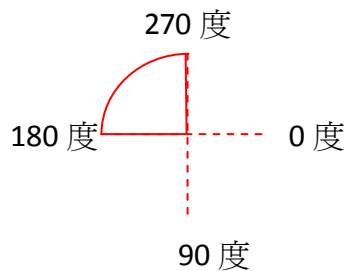
### g.DrawPie (Pen 物件, x1, y1, 寬度, 高度, 起始角度, 角度變化)

畫扇形的原則與方法和先前畫弧形的方式幾乎一模一樣，只有指令的差別。畫出的圖形之差別在於畫扇形時，除了畫出圓弧還會畫上半徑，而畫弧形則只會畫外圓弧的部份而已。

例如：起始角度為 0，角度變化為 90，所代表的扇形。



例如：起始角度為 270，角度變化為-90，所代表的扇形。



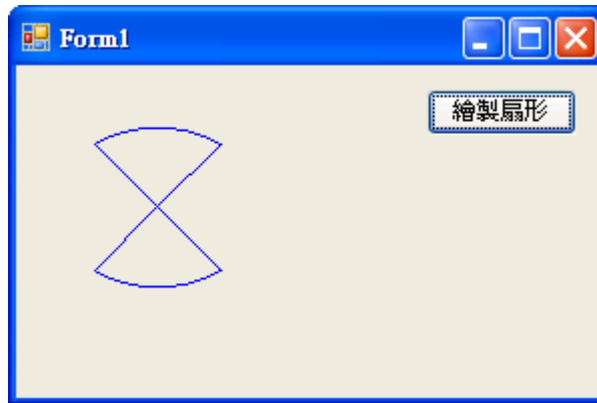
#### 範例：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim g As Graphics = Me.CreateGraphics
    g.DrawPie(Pens.Blue, 20, 30, 100, 80, 45, 90)
    g.DrawPie(Pens.Blue, 20, 30, 100, 80, -45, -90)
    g.Dispose()
End Sub
```

End Sub

繪製兩扇形，扇形分別屬於橢圓形的 45~135 度，以及 225~315 度(-135~-45 的同位角)，該橢圓可恰好內切於一左上角為(20, 30)、寬為 100 且高為 80 的矩形框。注意扇形除了弧以外，還加上了半徑的線段。

#### 執行結果：



## 程式練習題：井字遊戲

學習重點：DrawEllipse 與 DrawLine 函數的應用

### 一、表單配置圖例

本範例不需任何物件配置。

### 二、屬性值設定

本範例不需任何屬性設定。

### 三、編輯程式碼

列數	程式碼
1	Dim Turn As Integer = 1
2	Dim w, h As Integer
3	
4	Private Sub Form1_Paint(...) Handles Me.Paint
5	w = Me.ClientSize.Width \ 3
6	h = Me.ClientSize.Height \ 3
7	Dim g As Graphics = Me.CreateGraphics
8	g.DrawLine(Pens.Gray, 0, h, w * 3, h)
9	g.DrawLine(Pens.Gray, 0, h * 2, w * 3, h * 2)
10	g.DrawLine(Pens.Gray, w, 0, w, h * 3)
11	g.DrawLine(Pens.Gray, w * 2, 0, w * 2, h * 3)
12	g.Dispose( )
13	End Sub
14	Private Sub Form1_MouseClick(...) Handles Me.MouseClick
15	Dim g As Graphics = Me.CreateGraphics
16	Dim x1, y1 As Integer

17	If e.X < w Then 'e.X 取得滑鼠的 X 座標
18	x1 = 0
19	Elseif e.X < w * 2 Then
20	x1 = w
21	Else
22	x1 = w * 2
23	End If
24	If e.Y < h Then 'e.Y 取得滑鼠的 Y 座標
25	y1 = 0
26	Elseif e.Y < h * 2 Then
27	y1 = h
28	Else
29	y1 = h * 2
30	End If
31	If Turn = 1 Then
32	g.DrawEllipse(Pens.Blue, x1, y1, w, h) '畫圓
33	Turn = 2
34	Else
35	g.DrawLine(Pens.Red, x1, y1, x1 + w, y1 + h) '畫左上到右下的線
36	g.DrawLine(Pens.Red, x1 + w, y1, x1, y1 + h) '畫右上到左下的線
37	Turn = 1
38	End If
39	End Sub

#### 四、程式碼解說

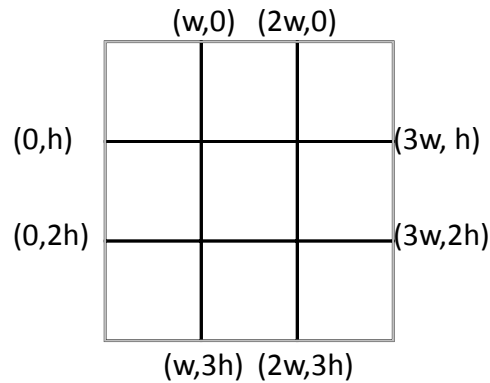
第 1 列宣告 Turn 變數來記錄目前換到的是「O」或者是「X」，1 代表前者，2 代表後者。

第 2 列宣告 w 及 h 分別用於儲存表單的  $\frac{1}{3}$  長與  $\frac{1}{3}$  高。

第 4~13 列為 Form1 的 Paint 事件之程序，會在表單上用線段分割成九宮格的模樣。第 5、6 列設置 w 與 h 的變數值，Me.ClientSize.Width \ 3 會計算出表單寬的  $\frac{1}{3}$ ，而「\」的計算結果，將會捨去商的小數部分，因為表單上只有整數的座標。同理 Me.ClientSize.Height \ 3，即代表表單高的  $\frac{1}{3}$ 。



第 8~11 列在表單畫上兩條直線、兩條橫線，起點與終點如下圖所示。



第 12 列將呼叫 `Dispose` 方法，將 `g` 所佔用的資源釋放掉。

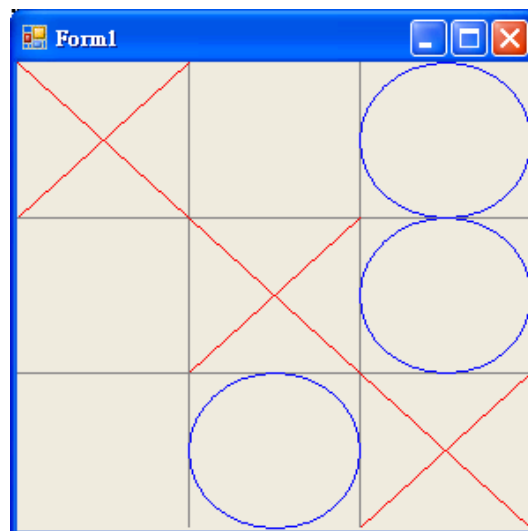
第 14~39 列的程式負責在玩家按下特定位置後，於指定格子畫上圈圈或叉叉。`x1` 與 `y1` 代表指定格子的左上角位置。要得知滑鼠按下的位置，可以透過 `e.X` 與 `e.Y` 取得座標。

第 17~23 列判斷滑鼠所在的 X 軸方向範圍，`if` 判斷式所表達的意思為：「`0~w`」代表最左欄，令 `x1=0`；「`w~2w`」代表中間欄，令 `x1=w`；「`2w~3w`」代表最右欄，令 `x1=2*w`。

第 24~30 列判斷滑鼠所在的 Y 軸方向範圍，`if` 判斷式所表達的意思為：「`0~h`」代表最上列，令 `y1=0`；「`h~2h`」代表中間列，令 `y1=h`；「`2w~3w`」代表最右欄，令 `y1=2*h`。

第 31~38 列判斷目前輪到「O」或是「X」，當 `Turn` 記錄的是 1，代表輪到「O」，所以使用 `DrawEllipse` 在指定的(`x1,y1`)位置畫圓，並將 `Turn` 改為 2。假如 `Turn` 為 2，代表輪到「X」，此時呼叫兩次 `DrawLine` 方法，分別畫左上到右下的線以及畫右上到左下的線，在指定位置繪出「X」的模樣。

## 五、執行結果



## ● 區塊繪圖方法：

在上一節中，所使用的繪圖方法僅能描繪邊框，如：`DrawRectangle`(矩形框)、`DrawEllipse`(橢圓框)、`DrawPie`(扇形框)。假如想要繪製一個填滿的圖形，則需要使用相對應的 `Fill` 方法，如：`FillRectangle`、`FillEllipse`、`FillPie`，並將原先傳遞 `Pen` 物件修改為傳遞 `Brush` 物件即可。`Brush` 與 `Pen` 物件，同樣皆為可設定顏色之工具，唯 `Brush` 是用來表示區塊的顏色，而 `Pen` 用於表示框線的顏色。

### 1. `FillRectangle` 填滿矩形區塊：

使用格式：

`g.FillRectangle (Brush 物件, x1, y1, 寬度, 高度)`



範例：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim g As Graphics = Me.CreateGraphics
    g.FillRectangle(Brushes.Blue, 20, 30, 100, 80)
    g.Dispose()
End Sub
```

繪製左上角起點為(20, 30)、寬為 100 且高為 80 的矩形區塊。

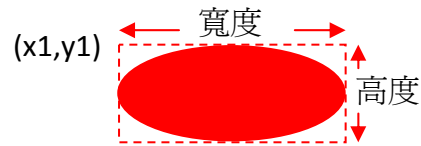
執行結果：



## 2. FillEllipse 填滿橢圓區塊：

使用格式：

`g.FillEllipse (Brush 物件, x1, y1, 寬度, 高度)`



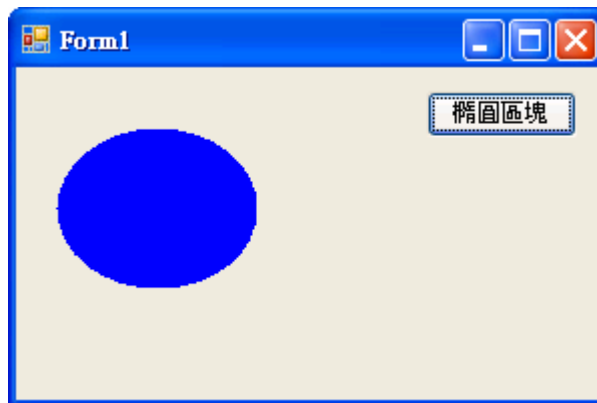
範例：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim g As Graphics = Me.CreateGraphics
    g.FillEllipse(Brushes.Blue, 20, 30, 100, 80)
    g.Dispose()
End Sub
```

End Sub

填滿一橢圓區塊，該橢圓可恰好內切於一左上角為(20, 30)、寬為 100 且高為 80 的矩形框。

執行結果：

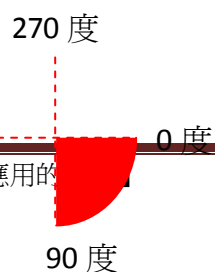


## 3. FillPie 填滿扇形區塊：

使用格式：

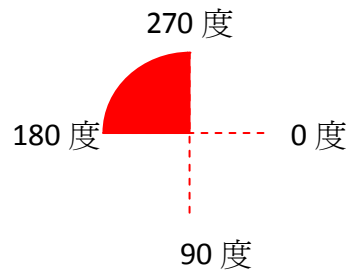
`g.FillPie (Brush 物件, x1, y1, 寬度, 高度, 起始角度, 角度變化)`

例如：起始角度為 0，角度變化為 90，所代表的扇形區塊。





例如：起始角度為 270，角度變化為-90，所代表的扇形區塊。

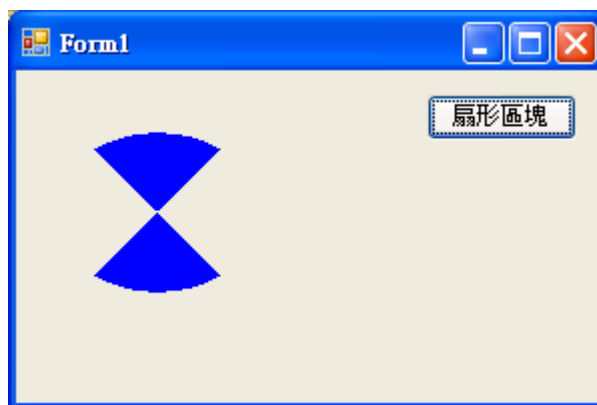


範例：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim g As Graphics = Me.CreateGraphics
    g.FillPie(Brushes.Blue, 20, 30, 100, 80, 45, 90)
    g.FillPie(Brushes.Blue, 20, 30, 100, 80, -45, -90)
    g.Dispose( )
End Sub
```

繪製兩扇形，扇形分別屬於橢圓形的 45~135 度，以及 225~315 度(-135~-45 的同位角)，該橢圓可恰好內切於一左上角為(20, 30)、寬為 100 且高為 80 的矩形框。

執行結果：

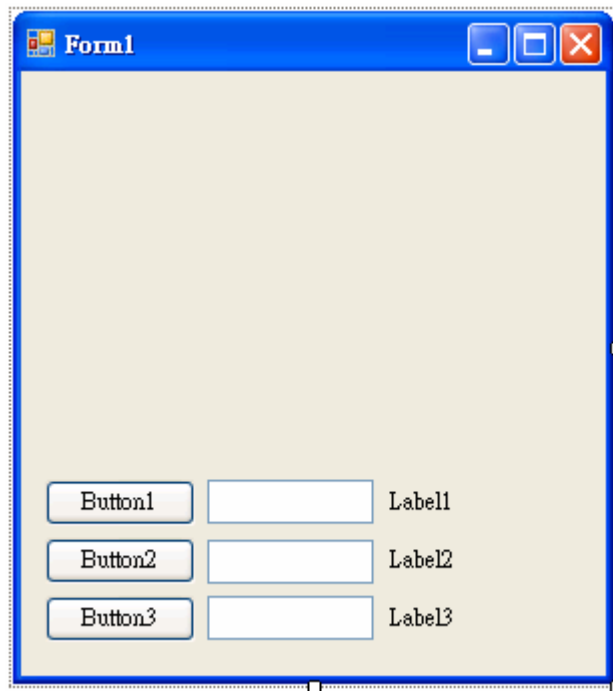


程式練習題：繪製民意調查圓餅圖

學習重點：FillPie 函數的應用

## 一、表單配置圖例

新增三個 Button、三個 TextBox 和三個 Label，如下圖。



## 二、屬性值設定

屬性值設定表格：

物件	屬性	屬性值
Form1	Text	吉祥物票選
Button1	Text	小猴子
Button2	Text	小狗兒
Button3	Text	小企鵝
Label1	Text	0.00%
	BackColor	Blue
Label2	Text	0.00%
	BackColor	Red
Label3	Text	0.00%
	BackColor	Yellow

設定屬性後的表單。



### 三、編輯程式碼

列數	程式碼
1	Dim vote(2) As Integer
2	Dim rate(2) As Single
3	Private Sub Button_Click(ByVal sender As System.Object, ...) Handles _
4	<b>Button1.Click, Button2.Click, Button3.Click</b>
5	If (sender.Equals(Button1)) Then
6	vote(0) += 1
7	Elseif (sender.Equals(Button2)) Then
8	vote(1) += 1
9	Else
10	vote(2) += 1
11	End If
12	
13	Dim sum As Integer = 0
14	For i As Integer = 0 To 2
15	sum += vote(i)
16	Next
17	For i As Integer = 0 To 2
18	rate(i) = vote(i) / sum
19	Next
20	Dim g As Graphics = Me.CreateGraphics

21	<code>g.FillPie(Brushes.Blue, 50, 10, 150, 150, 0, 360 * rate(0))</code>
22	<code>g.FillPie(Brushes.Red, 50, 10, 150, 150, 360 * rate(0), 360 * rate(1))</code>
23	<code>g.FillPie(Brushes.Yellow, 50, 10, 150, 150, 0, -360 * rate(2))</code>
24	<code>g.DrawPie(Pens.Black, 50, 10, 150, 150, 0, 360 * rate(0))</code>
25	<code>g.DrawPie(Pens.Black, 50, 10, 150, 150, 360 * rate(0), 360 * rate(1))</code>
26	<code>g.DrawPie(Pens.Black, 50, 10, 150, 150, 0, -360 * rate(2))</code>
27	<code>g.Dispose( )</code>
28	<code>TextBox1.Text = vote(0)</code>
29	<code>TextBox2.Text = vote(1)</code>
30	<code>TextBox3.Text = vote(2)</code>
31	<code>Label1.Text = Format(rate(0), "0.00%")</code>
32	<code>Label2.Text = Format(rate(1), "0.00%")</code>
33	<code>Label3.Text = Format(rate(2), "0.00%")</code>
34	<code>End Sub</code>

#### 四、程式碼解說

第 1 列宣告 `vote` 整數陣列存放三位候選者之得票數。

第 2 列宣告 `rate` 浮點數陣列存放三位候選者之得票率。

第 3~34 列為 `Button1`、`Button2` 及 `Button3` 按下後所驅動的事件處理，注意到第 3、4 列 `Handles` 後面的寫法，代表此程序在 `Button1.Click`、`Button2.Click` 及 `Button3.Click` 等事件中皆會被驅動。

第 5~11 列判斷究竟是哪一個 `Button` 驅動此事件。這時需要一個小技巧，用到之前常被「...」忽略掉的「`sender`」傳入值，它在第 3 列事件處理的開頭被宣告。`sender` 傳入值所指的是事件所屬的物件。在本例中，「`Click`」事件可能發生在三個 `Button` 上，`sender` 為其中的一個 `Button`。利用 `sender.Equals( ButtonX )`，來判斷 `sender` 與 `ButtonX` 是否代表了同個物件，假如相同即回傳「`True`」，不同則回傳「`False`」。

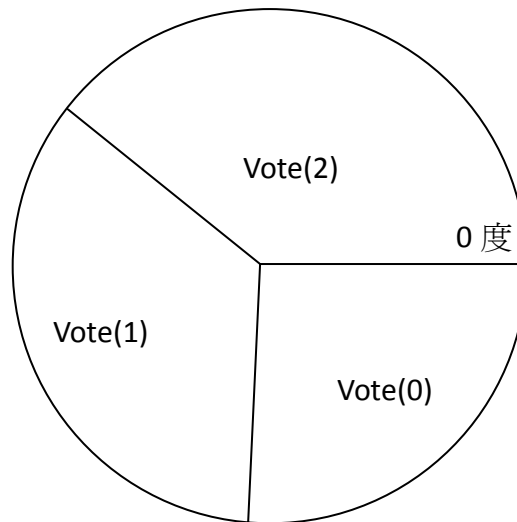
例如：使用者按下了「小猴子」，也就是 `Button1`。

接著 `vote(0)` 的值增加 1。

第 14~16 列計算總共的投票數。

第 17~19 列以[票數]/[總票數]，可得到 3 位候選人之得票率。

第 21~23 列進行圓餅圖之填色，此圓之大小為 `150*150`，而左上角位於 `50, 10`。



第 21 列以藍色刷從 0 度位置開始,畫上 vote(0)所跨過的度數,可以用公式(360 度\*得票率)來計算。第 22 列以紅色刷從 vote(0)結束的角度位置開始,所跨過的度數也是套用上述公式。特別的是第 23 列扇形改由 0 度往逆時針方向描繪,所以 360 前加上一個負號,表示以逆時針方向跨越該範圍。

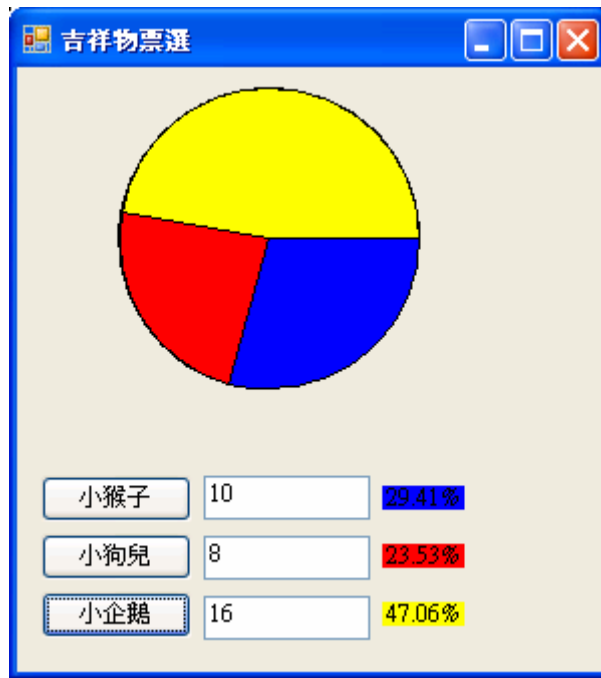
第 24~26 列以同樣的參數傳遞至 DrawPie 函數,為此 3 區域加上黑邊框。

第 28~33 列顯示三者之得票數與得票率。第 31~33 列,為透過 Format 函式,將數字資料轉換成以百分比的方式來顯示。例如,Format(0.3516, "0.00%"),會將運算式 0.3516 乘以 100,轉換成 35.16%的字串型態。

## 五、執行結果

投給小猴子 10 票、小狗兒 8 票、小企鵝 16 票,所繪得的圓餅圖。





● 練習題：

➤ 選擇題

- ( ) 1. 請問在 VB 表單上繪製圖形需透過下列何物件？
- Graphics
  - Images
  - Pictures
  - Drawer
- ( ) 2. g 代表一個 Graphics 物件，試問下列對於「g.Dispose」的描述何者錯誤？
- 釋放物件所佔用的資源
  - Dispose 為 Graphics 的一個方法
  - 執行後，接著仍可使用 g.DrawRectangle 方法畫矩形
  - 變數 g 在執行「g.Dispose」之後仍存在
- ( ) 3. g 代表一個 Graphics 物件，試問 g.DrawPie(Pens.Black, 0, 0, 20, 20, 150, -270)所繪出的圖形為何？
- - 
  - 
  -
- ( ) 4. 對於 VB 表單上的座標格式之敘述何者正確？
- 座標原點位於表單左上角，Y 軸方向向下
  - 座標原點位於表單正中央，Y 軸方向向上

- (c) 座標原點位於表單左上角，Y 軸方向向上  
 (d) 座標原點位於表單正中央，Y 軸方向向下
- ( ) 5. Dim x as Double = 0.875，試問 Format(x, "0.00%") 之回傳值為何？  
 (a) 0.88%  
 (b) 0.87%  
 (c) 88.00%  
 (d) 87.50%
- ( ) 6. 假設表單之寬、高為 W, H，直角座標系上之一點(x,y)，應當透過下列哪一組公式之轉換，才能在表單上呈現出直角座標系的模式：原點在正中央、X 軸向右增加、Y 軸向上增加？  
 (a)  $(x,y) = (x + W/2, y + H/2)$   
 (b)  $(x,y) = (x + W/2, H/2 - y)$   
 (c)  $(x,y) = (x - W/2, H/2 - y)$   
 (d)  $(x,y) = (W/2 - x, y + H/2)$

### ➤ 問答題

請寫下用 Graphics 物件，繪製以下圖案的程式碼。



### 總結.....

以上將我教導學生學習 VB 繪圖功能的教學內容，提供給大家參考！我們可以發現，學生學習繪圖功能，有很大的自行設計空間，只要學會基本的繪圖指令，就可以組合變化出各種圖形。

觀察學生對於 VB 繪圖功能的學習過程，發現學生表現出相當高的學習興趣，也始終維持著學習熱情，甚至想要嘗試各種繪圖功能的排列組合，以設計創造出新的有趣圖形，甚至同儕之間還會比賽誰設計的圖形較有創意。

當我們在教授資訊課時，讓學生快樂且有成就感地學電腦，是一件很重要的事情；而我覺得教授 VB 的繪圖功能，正是可以有效地達到這個目的。教授 VB 的繪圖功能，可以讓學生實際完成一些繪圖成果，並且可以學到空間觀念、邏輯觀念和數學的角度觀念，最重要的是可以讓學生發揮創意，在學會了基本繪圖功能之後，自行去設計具有新意的圖形。總之，透過教授 Visual Basic 的繪圖功

能，可以讓學生喜歡上電腦課喔！

野人獻曝，謹供大家參考，讓我們在提升資訊教育品質的路上，一同打拚，一起加油，謝謝大家 ^\_^