

## 第 7 章 螢幕保護程式

### 簡介：

螢幕保護程式就是製作全螢幕程式的技巧，它原本的功能是避免螢幕畫面固定太久會加速螢幕的老化，所以製作的基本原則就是畫面要覆蓋整個螢幕，並儘量讓畫面各個部分都不要停滯太久。當然有時候也可以加入密碼，變成一個保護電腦不被外人擅用的安全機制，或者讓畫面與滑鼠有些互動，以增加一些趣味性，這些都是本章的教學內容。

### 7-1 建立關閉機制與全螢幕視窗

因為視窗變成全螢幕之後，會失去邊框上的關閉視窗按鈕，本範例也不會建立功能表，程式因此無法用一般方式結束，所以建立專案之後的第一件事是設計關閉程式的機制。請先到程式碼的 Form1 事件中選擇 Click 事件寫入程式碼：

```
Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Click
    Application.Exit()
End Sub
```

意思就是此程式開啟後，用滑鼠點一下表單就會自動結束！接著將表單的 WindowState 屬性設為 Maximized(放到最大)，同時將 FormBorderStyle(邊框樣式)設為 None(沒有邊框)，執行程式就會看到一個全螢幕的畫面了！這時要關閉程式就必須用滑鼠點擊畫面，想想如果沒寫好前面的程式，現在會有多尷尬？如果你不喜歡灰灰的預設背景色，可以到表單的 BackColor 屬性修改。

### 7-2 作個會移動的小時鐘

現在的電腦已經兼具時鐘的功能，許多螢幕保護程式都會顯示時間，我們就先作一個簡單的小時鐘吧！當然因為是螢幕保護程式，時鐘不應該固定在一處。方法是叫用一個 Label 物件將字型放大一些，同時叫用一個 Timer1 物件，雙擊它寫入以下程式：

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick
    Label1.Text = Now
    Label1.Left -= 5
    If Label1.Right < 0 Then
        Label1.Left = Me.Width
    End If
End Sub
```

Now 就是現在的時間，將它顯示於 Label1 並定時更新就是一個時鐘了！我們在前面的電子琴單元裡也作過，但是此地還要讓它移動。Timer 的預設時間間隔是 100 毫秒，也就 0.1 秒，上面程式就讓它每 0.1 秒往左移動 5 個像素點。當然最後一定會跑出畫面左邊，就是 Label1 的右邊(Right)座標小於 0 的時候，我們要用程式將它重新拉回到螢幕最右邊，也就是座標為視窗寬度(Me.Width)的地方。當然程式應該也要啟動 Timer1，你可以直接將它的 Enabled 屬性改為 True，或者在程式的 Form\_Load 事件中寫程式如下：

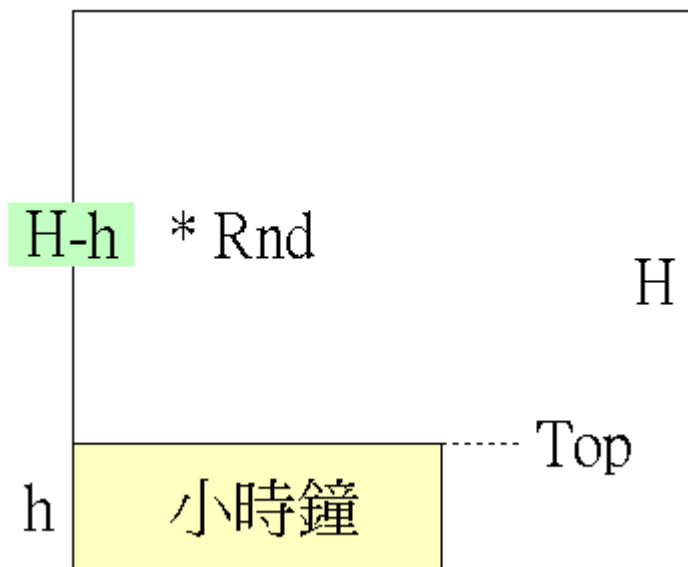
```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Label1.Text = Now
    Timer1.Start()
End Sub
```

Start 是啟動的意思，表示 Timer1 事件內的程式會開始反覆執行了！但是 Timer 的運作方式是等到第一個間隔時間(1/10 秒)到之後才會執行第一次動作，所以我們在此加了一個顯示時間的程式，這樣表示表單載入的第一時間就會有時間顯示，否則開啟時會有個 Label 文字一閃而過。試試看，一個會顯示時間且往左飄移的小時鐘作好了！測試時請確定它移出畫面之後會正確的重新在右邊出現。

這樣的小時鐘會固定在同一個高度上，如果希望每次回頭時都可以隨機的換個高度可以將 Timer1 事件程式拉回標籤的部分改寫如下：

```
If Label1.Right < 0 Then
    Label1.Left = Me.Width
    Label1.Top = Rnd() * (Me.Height - Label1.Height)
End If
```

Rnd 是一個大於等於 0，小於 1 的隨機數字，上面程式的示意圖如下：H 對應於螢幕可活動區的高度，h 是小時鐘高度，我們將兩者差(H-h)乘以介於 0~1 之間的任意數 Rnd 作為小時鐘的 Top 值，小時鐘的位置就會界於最頂與最底部了！當 Rnd 趨近於 0 的時候時鐘會在視窗最上方，趨近於 1 時則位於視窗最下方。現在時鐘應該每次折返時都會換個高度了！



請注意到前面程式的 Start 與 Rnd 後面都會有一對小括號，事實上即使你不寫，VB 也會在換行時自動加上。它的意義是副程式(或函數)置放參數的地方，如果剛好不需要參數，就空著了！在物件導向程式設計的概念中，物件的靜態性質如 Top 或 Left 是「屬性」(Attributes)，通常不需要夾帶參數(但可能有下一層的屬性，如 Size->Width)；物件可以執行的「功能」或動作如 Start 及 Rnd(產生亂數)等等則稱為「方法」(Method)，預設就會有參

數的欄位。所以即使是空的小括號也可以讓我們知道它是個程式的動作，對閱讀程式有點幫助的！

### 7-3 背景隨機繪圖

接下來我們想在視窗的背景，每秒鐘隨機的畫出大小與位置不等的圓形，首先要叫用一個 Timer2，在其事件中寫程式如下，當然也必須在 Form\_Load 事件中啟動它(Timer2.Start)：

```
Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer2.Tick
    Dim D As Integer = Rnd() * Me.Height
    Dim X As Integer = Rnd() * (Me.Width - D)
    Dim Y As Integer = Rnd() * (Me.Height - D)
    Dim bg As New Bitmap(Me.Width, Me.Height)
    Dim g As Graphics = Graphics.FromImage(bg)
    g.FillEllipse(Brushes.Black, X, Y, D, D)
    Me.BackgroundImage = bg
End Sub
```

上面程式中我們先使用亂數決定圓的直徑 D，大小最大不能超出螢幕，而一般螢幕都是寬大於高，所以 D 就是螢幕(在此也是視窗)高度(Height)乘上一個介於 0 與 1 之間的隨機數字(比例)，這樣 D 最大只能與螢幕一樣高。直徑決定之後再決定圓要畫的位置(X, Y)，X 是圓的左緣，必須介於 0 到螢幕寬度(Me.Width)減去直徑(D)的範圍，超過就會有一部分或全部的圓圈會畫出視窗；同理可證，Y 的範圍就是高度(Me.Height)減去直徑(D)了！一樣的，我們用亂數隨機選出 X 與 Y 的繪圖座標。這裡如果不太懂，一樣可以參考前一節隨機調整小時鐘高度的示意圖，狀況非常相似。

接下來是繪圖的程式碼，先宣告(建立)一個點陣圖(Bitmap)物件 bg，接著建立對應的繪圖物件(Graphics) g，執行 FillEllipse 方法將圓畫在 bg 上面，Fill 是填滿的意思，Ellipse 是橢圓的意思，就是繪製實心的橢圓之意！如果要畫空心圓應該使用 DrawEllipse 方法。FillEllipse 的五個參數分別是筆刷(填充色)、X(左緣)與 Y(上緣)座標，以及寬和高度，讓寬高都等於 D 就是正圓形了！最後不要忘了我們的繪圖動作是畫在 bg 上的，必須將此點陣圖貼到表單的背景影像(BackgroundImage)屬性之中。試試看，你應該會看到隨機出現大小與位置不等的圓形了，如果靜悄悄的不動，就看看你的 Timer2 是不是忘了啟動(Start)。

會不會覺得老是同一個顏色很單調？接下來可以試試修改筆刷部分的程式如下：

```
Dim C As Color = Color.FromArgb(255 * Rnd(), 255 * Rnd(), 255 * Rnd())
Dim B As New SolidBrush(C)
g.FillEllipse(B, X, Y, D, D)
```

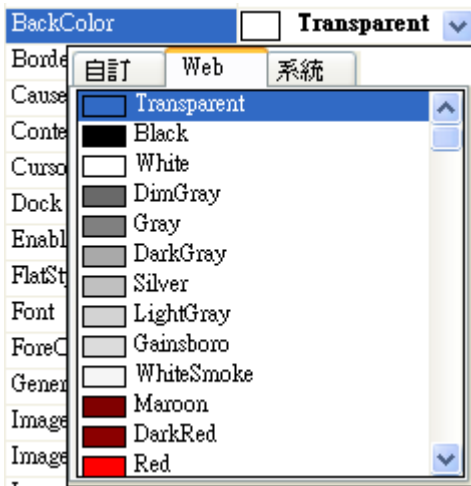
先是宣告一個顏色 C，讓它的三原色是 0~255 之間的任意數字，其中 Color.FromArgb 意義是使用 ARGB 四個參數建構一個顏色，RGB 分別指紅綠藍色，A 指透明度，0 是全透明，255 是全不透明(預設值)，如果只寫三個參數，程式會自動認為是 RGB，且全不透明(A=255)。接著建立一個筆刷物件 B 將它的顏色定義為 C，再套用到畫圓的程式即可。這裡比較詭異的是必須使用 SolidBrush 作筆刷物件，而非 Brush 或 Brushes，很多初學者常常會因此誤會筆刷顏色是不能變更的！當然你也可以試試看繪製矩形(FillRectangle)或者扇形

(FillPie)，不會用的話可以注意看它們顯示的工具提示，或寫出來之後選取這個字按下 F1 鍵，會出現詳細的說明文件。

如果細心一點，你會發現目前的程式每次執行時圓圈出現的順序是完全一樣的！如果想打亂它們，請在 Form\_Load 事件中加上一行 Randomize()，這個指令有類似「洗牌」的功能，所以程式就不會每次都一樣了！你可以想像，當我們要設計遊戲時，這個指令就非常重要了！如果每次發牌順序都一樣，遊戲還怎麼玩啊？

#### 7-4 透明背景的標籤

還有一個小地方美中不足，就是時鐘標籤如果是「透明」的就好了！目前狀態是一個灰灰的方塊，真的不太專業。在 VB6 的時代大家都很容易可以將背景樣式設為透明(BackColor=透明)，現在好像找不到一個定義「透明」或「實心」之類的屬性了，只有一個 BackColor 可以修改。其實在新的 VB 架構概念中「透明」也是一個顏色！請到 BackColor 屬性的「Web」頁籤選項，找到位於第一個的 Transparent(透明)，選下去，標籤背景就透明了！



這時候你就知道英文程度對於寫程式來說還是有差別的，我就是因為認識這個字才學到這一招的！哈哈！執行時的部分畫面如下，已經呈現出透明的效果了！



## 7-5 滑鼠互動

在此我們希望程式可以呼應滑鼠移動而有一些繪圖動作反應，本來應該將程式寫在 MouseMove 事件之中，但是前面的 Timer2 已經有繪圖相關程式，為了簡化程式碼，我們在 MouseMove 事件中就只是記下滑鼠的座標而已，等 Timer2 時間到再一起畫圖，會有一點延遲，但也是另一種所謂的「跟隨」效果啦！我們就試試看作個以滑鼠位置為中心的十字型準心吧！包含 Timer2 的修改完整程式碼如下：

```
Dim mmX As Integer, mmY As Integer

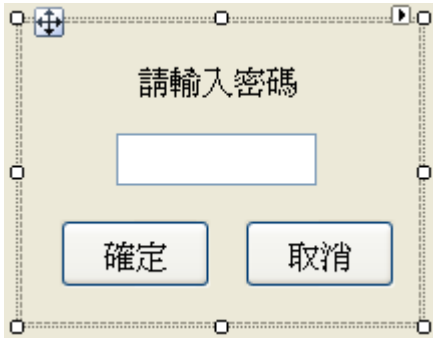
Private Sub Form1_MouseMove(ByVal sender As Object, ByVal e As System.Windows.Forms.MouseEventArgs) _
Handles Me.MouseMove
    mmX = e.X
    mmY = e.Y
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer2.Tick
    Dim D As Integer = Rnd() * Me.Height
    Dim X As Integer = Rnd() * (Me.Width - D)
    Dim Y As Integer = Rnd() * (Me.Height - D)
    Dim bg As New Bitmap(Me.Width, Me.Height)
    Dim g As Graphics = Graphics.FromImage(bg)
    Dim C As Color = Color.FromArgb(255 * Rnd(), 255 * Rnd(), 255 * Rnd())
    Dim B As New SolidBrush(C)
    g.FillEllipse(B, X, Y, D, D)
    g.DrawLine(Pens.Black, 0, mmY, Me.Width, mmY)
    g.DrawLine(Pens.Black, mmX, 0, mmX, Me.Height)
    Me.BackgroundImage = bg
End Sub
```

首先是宣告公用的全域變數 mmX 與 mmY 在 MouseMove 事件中記錄滑鼠座標，接著在 Timer2 的繪圖程式再加兩行畫直線(DrawLine)的程式，畫線程式的參數分別為筆，以及 X1，Y1，X2 與 Y2，就是起終點的座標啦。試試看！當你移動滑鼠之後，一個十字型的準心會在一秒鐘之後跟隨過去。

## 7-6 加入密碼

請先在表單上建立如下介面，底層是工具箱中「容器」分類的 Panel(面板)物件，上面是一個標籤(Label2)寫「請輸入密碼」，中間是一個文字方塊(TextBox1)，下方是兩個按鍵(「確定」為 Button1，「取消」為 Button2)。請在 TextBox1 的 PasswordChar 屬性欄中打一個「\*」號，這樣就會有密碼輸入時的隱藏字元效果了！



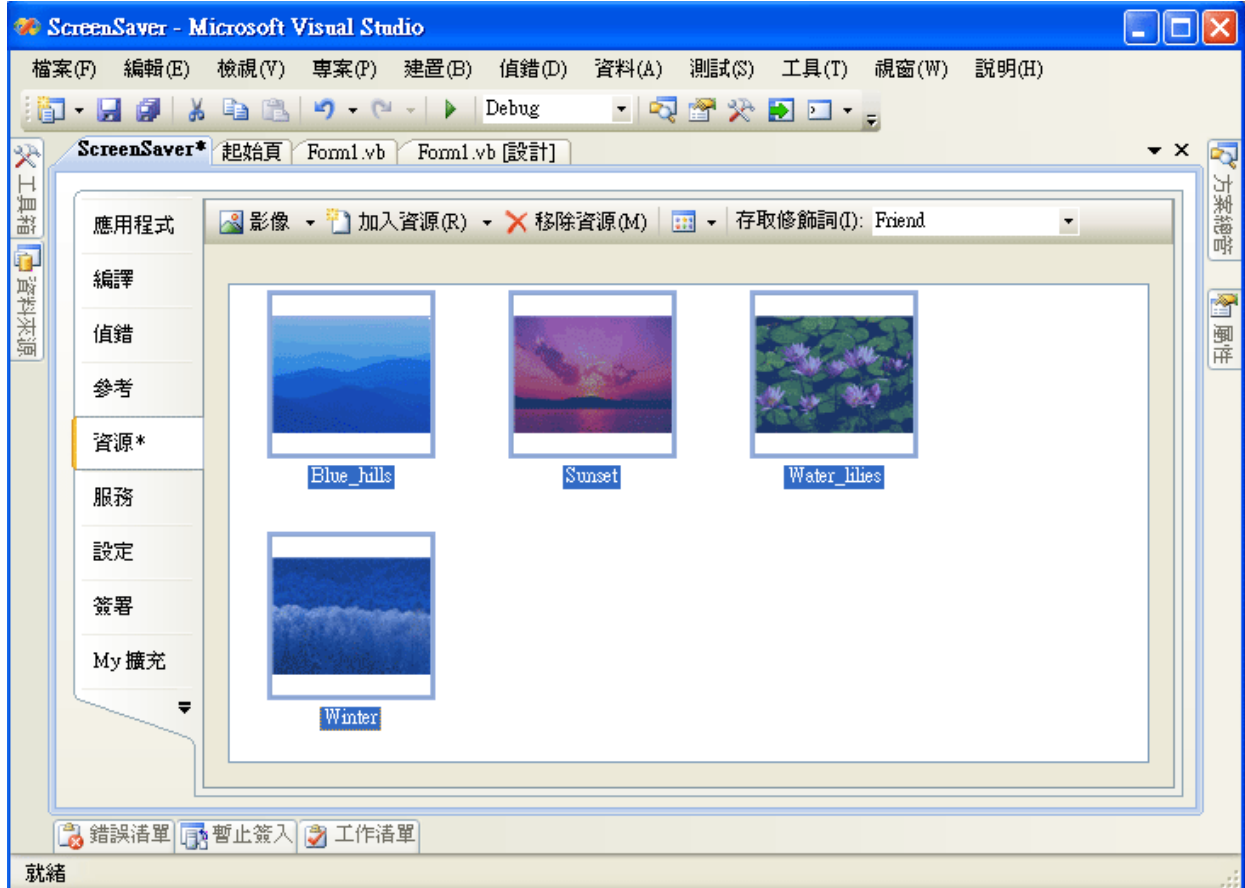
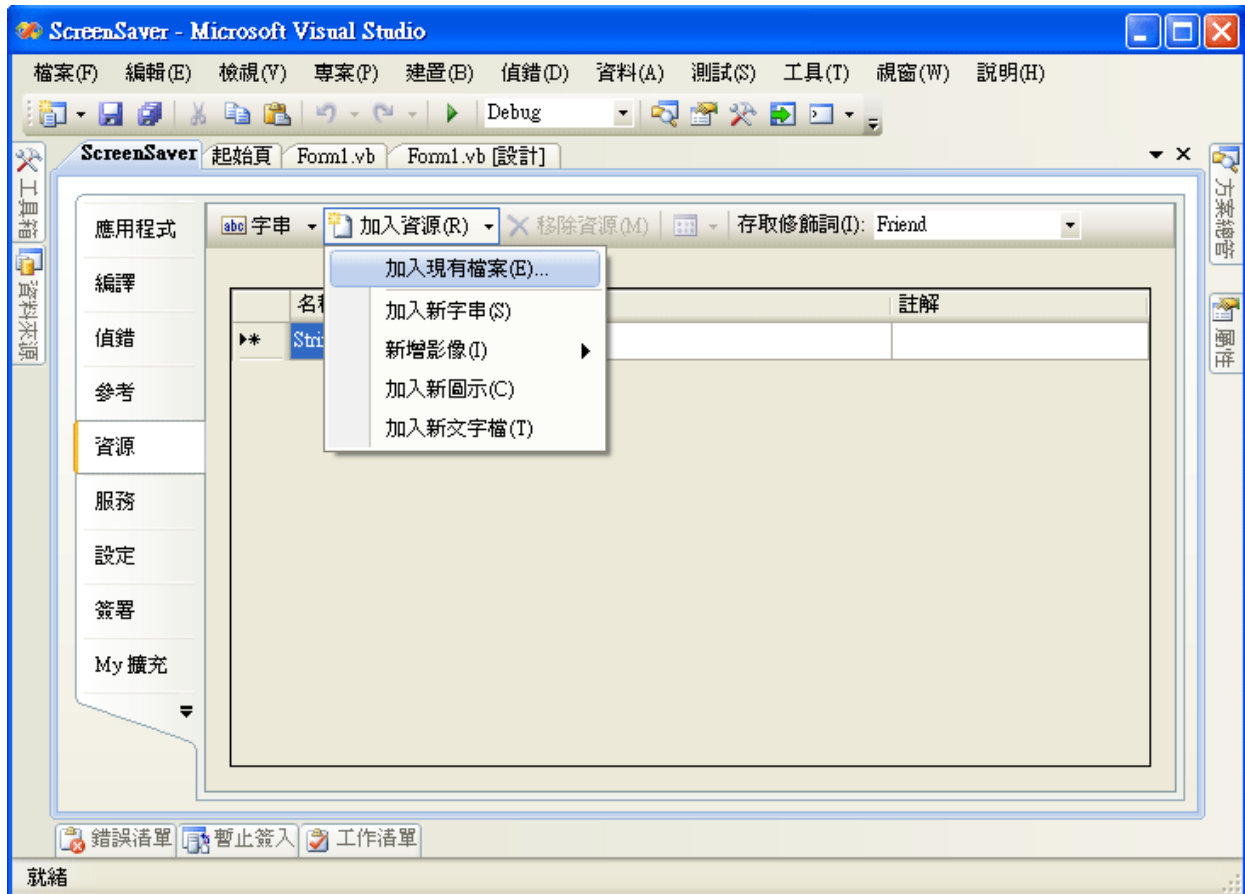
這個東西當然是使用者準備關閉程式時才應該出現，所以請將 Panel1 的 Visible 屬性設為 False 隱藏起來，當使用者點擊表單觸發前面寫過的 Form\_Click 事件時再讓它出現，還應該用程式將它放到螢幕中央。程式碼如下：

```
Private Sub Form1_Click(ByVal sender As Object, ByVal e As System.EventArgs) _  
    Handles Me.Click  
    Panel1.Left = (Me.Width - Panel1.Width) / 2  
    Panel1.Top = (Me.Height - Panel1.Height) / 2  
    Panel1.Visible = True  
End Sub  
  
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles Button1.Click  
    If TextBox1.Text = "xyz" Then  
        Application.Exit()  
    Else  
        MsgBox("密碼錯誤!")  
    End If  
End Sub  
  
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles Button2.Click  
    Panel1.Visible = False  
End Sub
```

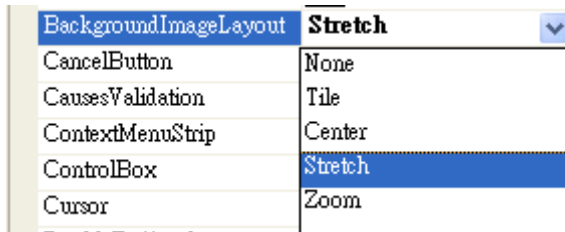
首先我們修改 Form1\_Click 事件副程式，前兩行是令面板物件(Panel1)置中，接著讓它顯示。Button1 被按下代表使用者已經輸入密碼並要求核對身分了！如果密碼與程式設定的"xyz"相同，就讓程式結束，使用者就可以結束螢幕保護程式繼續其他的工作了！否則(Else)就跳出訊息警告密碼錯誤！至於 Button2 取消鍵只是讓使用者可以放棄輸入密碼的動作，所以只要將 Panel 再度隱藏(Visible=False)即可！拜託不要忘記自己設定的密碼哦！如果不幸發生，請按 Ctrl+Pause(Break)按鍵可以暫停程式回到程式設計的畫面。

## 7-7 播放圖片

銀幕保護程式也可做成如 PowerPoint 一樣的播放形式，這對很多商家應該很有用，只要將產品圖片放上去就可以當輪播的廣告看板了！在此我們將一般 Windows 作業系統預設的幾張照片匯入到我們的程式專案來。方法是開啟方案總管中的 My Project，就是專案屬性檔，選擇「資源」頁籤→「加入資源」→「加入現有檔案」，之後就是一一將圖片載入了！操作畫面如下：



接著請至表單再加入一個計時器 Timer3，將 Interval 設定為 2000(兩秒鐘一次)；同時修改表單的 BackgroundImageLayout(背景影像展示方式)屬性為 Stretch，預設值為 Tile，就是磁磚的意思，如果圖片不夠大時會如磁磚一樣拼貼。選擇 Stretch 的效果就是將單一影像延展填滿整個表單(螢幕)了！



接著請寫程式如下：

```
Dim curP As Integer = 0

Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer3.Tick
    curP += 1
    If curP > 4 Then curP = 1
    Select Case curP
        Case 1
            Me.BackgroundImage = My.Resources.Blue_hills
        Case 2
            Me.BackgroundImage = My.Resources.Sunset
        Case 3
            Me.BackgroundImage = My.Resources.Water_lilies
        Case 4
            Me.BackgroundImage = My.Resources.Winter
    End Select
End Sub
```

以上程式碼的意義是：宣告一個公用的全域變數 curP 用來記住目前播到哪一張圖片，接著在 Timer3 裡面每次時間到就將 curP 加 1！如果 curP 大於 4 表示播完了，就要從第一張再播一次。Select Case 的語法表示多選項的單選狀況，上面程式依據 curP 的值決定要播哪一張？播放時就是將已經載入的資源(Resource)檔案貼到表單的背景影像(BackgroundImage)而已。這和秀圖軟體單元的投影片播放有點像，但差別是資源檔是程式一開始就全部載入的，執行中就不再需要到磁碟去讀檔案了！這個程式執行速度快，但是想變更影像內容時就不如秀圖軟體程式有彈性了。

因為這個操作與前面的 Timer1 小時鐘及 Timer2 的繪圖可能有些衝突(畫面會交替出現)，測試時請先將小時鐘(Label1)隱藏(Visible=False)，並將 Timer1.Start 及 Timer2.Start「註解掉」，就是在程式碼前面打一撇，VB 會將它當作「註解」而不執行。修改過的 Form\_Load 應該是這樣的：

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles MyBase.Load
    'Label1.Text = Now
    'Randomize()
    'Timer1.Start()
```



```
'Timer2.Start()  
Timer3.Start()  
End Sub
```

## 7-8 進階挑戰

一、如何讓表單的背景也可以隨機變色？

提示：使用 `Color.FromArgb` 功能隨機設定表單的 `BackColor`。

二、如何讓密碼輸入時按下 `Enter` 鍵也能完成輸入動作？

提示：使用 `TextBox1` 的 `KeyDown` 事件，檢查按鍵是否為 `Keys.Enter`。

三、如何讓 `Timer2` 有時畫方，有時畫圓？

提示：使用隨機亂數 `Rnd` 大於 `0.5` 時畫方，小於 `0.5` 畫圓即可。

## 課後閱讀

### 談談 MsgBox 訊息小視窗

很多時候程式中我們會使用到 `MsgBox` 這個東西，最簡單的情境譬如本單元中出現的 `MsgBox("密碼錯誤！")`，它會跳出來的視窗外觀像這樣：

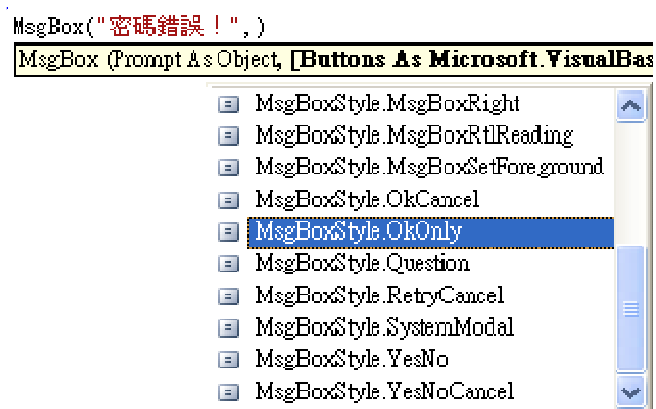


事實上，它包括了三個可以改變的屬性，最重要的當然是訊息的內容，如上例是「密碼錯誤！」除此之外還有視窗的「標題」，以及下方的按鈕(目前是「確定」)。預設的視窗標題是專案的名稱，在此就是「ScreenSaver」；預設按鈕則是最簡單的一個「確定」，如果在英文版作業系統之下，就會變成一個「OK」按鈕！如何改變標題與按鈕事實上在你寫程式過程中就有提示，請看看下面的畫面：

```
MsgBox("密碼錯誤！")  
MsgBox(  
MsgBox (Prompt As Object, [Buttons As Microsoft.VisualBasic.MsgBoxStyle = MsgBoxStyle.DefaultButton1], [Title As Object
```

設計環境提供的自動提示先告訴你第一個參數叫作「Prompt」就是訊息內容了！後面還有逗點分開的第二個以及第三個參數，就是引導你必要時可以加入按鍵(Buttons)種類及標題(Title)內容！請注意，後兩個參數用「方型中括號」括住表示可以寫，也可以不寫，都不寫時就是用「確定」鈕加上專案名稱了！

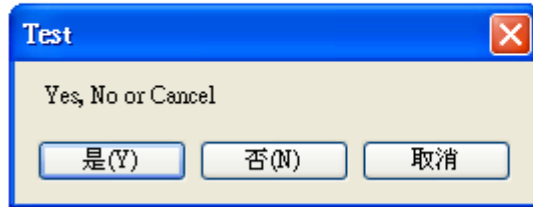
通常如果要讓此訊息更加清楚，如上的標題可以改成「密碼輸入訊息」，程式碼變成這樣：`MsgBox("密碼錯誤！","密碼輸入訊息")`，可以用兩個逗點跳過第二個按鍵選擇的參數，電腦就知道我們後面的字串是要當作標題了！另一方面我們也可以看看有那些按鈕的選擇：



「OkOnly」就是預設選項(只有確定鈕)，常用的還有畫面下方的 `MsgBoxStyle.YesNo`，選它會出現「是」與「否」兩個選項；`MsgBoxStyle.YesNoCancel` 則會再多一個「取消」的

選項。重要的是：如果使用者真的選了不同的按鈕(是、否或取消)，程式該如何採取不同的反應呢？一個比較嚴謹古板的寫法是這樣的：

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles MyBase.Load
    Dim B As MsgBoxResult = MsgBox("Yes, No or Cancel", MsgBoxStyle.YesNoCancel, "Test")
    Select Case B
        Case MsgBoxResult.Yes
            MsgBox("Yes")
        Case MsgBoxResult.No
            MsgBox("No")
        Case MsgBoxResult.Cancel
            MsgBox("Cancel")
    End Select
End Sub
```



先宣告一個 `MsgBoxResult`(訊息的結果或答覆)型態的變數 `B` 來承接 `MsgBox` 的結果(使用者選擇)，這個訊息框可能有三種答案(Yes, No or Cancel)，因此就以 `Select Case` 的語法分別在 Yes, No 與 Cancel 答案出現時作不同的回應。當然多數時候簡單的是與否選擇可能是這樣寫的：

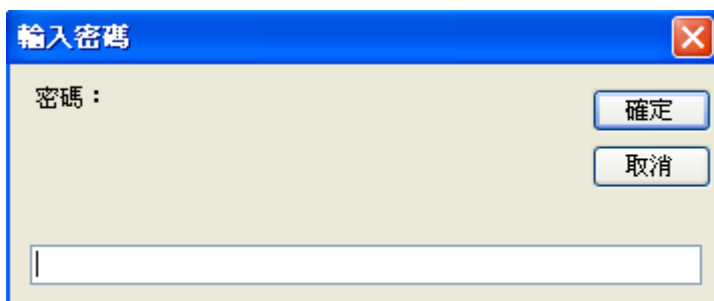
```
If MsgBox("Yes or No", MsgBoxStyle.YesNo, "Test") = MsgBoxResult.Yes Then
    MsgBox("Yes")
Else
    MsgBox("No")
End If
```

重點是 `MsgBox` 除了提示訊息的功能之外，也是可以接受使用者的「答案」而作不同回應的！希望同學可以學會這個技巧，很多時候這可以少寫很多程式碼，非常好用的！

還有一個類似的，現成的小視窗函數是 `InputBox`，如果你不介意畫面是不是很漂亮，輸入密碼的程式也可以寫成這樣：

```
If InputBox("密碼：", "輸入密碼") = "xyz" Then
    End
End If
```

跳出的畫面如下：



簡單說，`InputBox` 也是有回傳值(答案)的，它的回傳值就是使用者輸入的文字，如果密碼對了，程式就會結束(End)！「End」是 VB 程式專屬的結束程式指令，與 `Application.Exit` 意義相同。