# 第 8 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舊式的 CRT 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必須覆蓋整個螢幕，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盜用的安全機制，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。

## 8-1 建立關閉機制與全螢幕視窗

[建立 Form1_Click 事件]

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。請先選擇 Form1 物件至其屬性視窗點選閃電標誌，切換至事件選擇頁籤，雙擊 Click 建立事件副程式框架，寫入程式碼：

```
private void Form1_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

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。

## 8-2 作個會移動的小時鐘

[定時移動物件的程式]

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(36 或更大)，同時叫用一個 timer1 物件，雙擊它寫入以下程式：

```
private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToString();
    label1.Left -= 5;
    if (label1.Right < 0)
    {
        label1.Left = this.ClientSize.Width;
    }
}
```

其中 DateTime.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，上面程式就是讓它每 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(this.Width)的地方。當然程式應該也要啟動 timer1，你可以在設計階段直接將它的 Enabled 屬性改為 True，或者在 Form_Load 事件中寫程式如下：

```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToString();
    timer1.Start();
}
```

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行顯示時間的程式，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"label1"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畫面右邊出現。

[Random 物件的使用]

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 事件程式改寫如下：

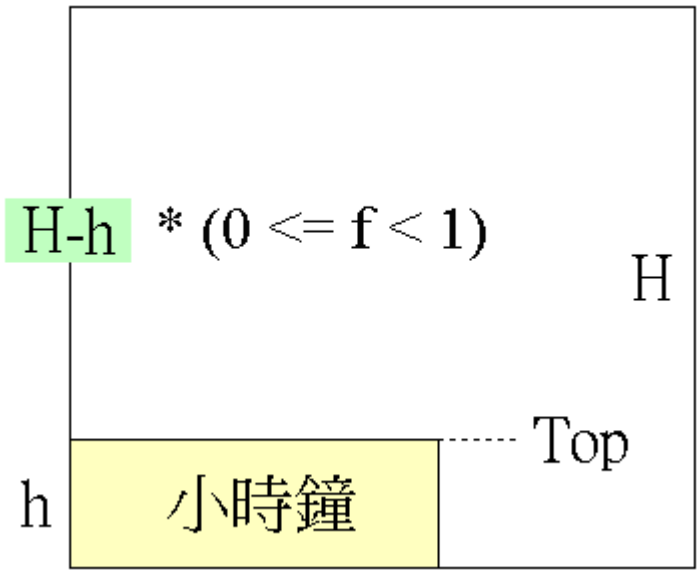
```
Random R = new Random();
private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = DateTime.Now.ToString();
    label1.Left -= 5;
    if (label1.Right < 0)
    {
        label1.Left = this.ClientSize.Width;
        label1.Top = R.Next(this.Height - label1.Height);
    }
}
```

首先在副程式外面先定義了一個亂數產生器 R，Random 原意是隨機的意思，當我們在程式中需要用到隨機亂數時就必須先建置這個物件。建置(new)時 Random 後面的小括號內可以給一個數字稱為 Seed(種子)，如果你給的數字一樣就會產生一樣次序的一組亂數，不給的話系統會依時間給不同的數字，當然這樣反而更有隨機的效果，所以如果你不想作弊的話就用空括號吧！

[隨機改變物件高度的程式]

接著在時間標籤拉回右邊的程式碼下面我們加了一行程式，其中 R.Next()就是取得下一個隨機數的意思，小括號內是此亂數的最大範圍(0 到此數字之間)。在此，我們程式的示意

圖如下：H 對應於螢幕可活動區的高度，h 是小時鐘高度，兩者的差(H-h)就是小時鐘可以上下移動又不會跑出畫面的範圍了！我們以此為亂數的最大值，當亂數趨近於 0 的時候時鐘會在視窗最上方，趨近於此最大值時則位於視窗最下方(螢幕座標以下為正)。現在時鐘應該每次折返時都會換個高度了！




**8-3 背景隨機繪圖**

[隨機改變物件位置與大小的程式]

```
private void timer2_Tick(object sender, EventArgs e)
{
    int D = R.Next(this.Height);
    int X = R.Next(this.Width  - D);
    int Y = R.Next(this.Height  - D);
    Bitmap bg = new Bitmap(this.Width,  this.Height);
    Graphics g = Graphics.FromImage(bg);
    g.FillEllipse(Brushes.Blue,  X, Y, D, D);
    this.BackgroundImage  = bg;
}
```

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(此例中等於視窗)高度，正常螢幕是寬大於高，不大於螢幕高應該就不會跑出螢幕了。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(this.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(this.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。這一部分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(Form 物件沒有 Image 屬性)。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)。

[隨機改變筆刷顏色]

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：

```
Color C = Color.FromArgb(R.Next(256), R.Next(256), R.Next(256));
SolidBrush B = new SolidBrush(C);
g.FillEllipse(B, X, Y, D, D);
```

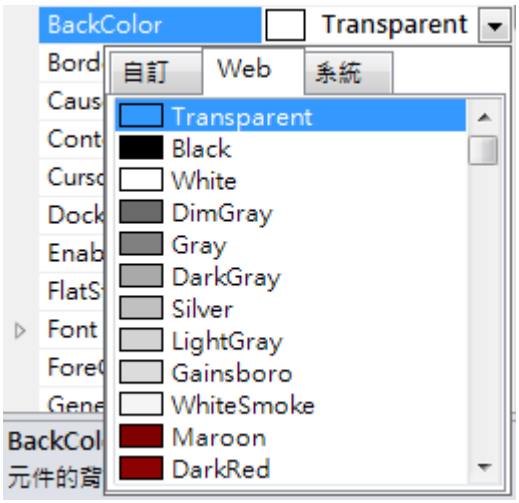
先是宣告一個顏色變數 C，讓它的三原色是 0~255 之間的任意數字，Color.FromArgb 意義是使用 ARGB 四個參數建構一個顏色，RGB 分別指紅綠藍色，A 指透明度，0 是全透明，255 是全不透明(預設值)，如果只寫三個參數，程式會自動認為是 RGB，且全不透明(A=255)。接下來 R.Next(X)會回傳小於 X 且大於 0 的任意整數，所以需要 0~255 的隨機數就應該寫成：R.Next(256)。

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)，不會用這些指令的話可以注意看它們顯示的工具提示，或在程式編輯器中寫出該字後選取它按下 F1 鍵，就會出現詳細的說明文件。

### 8-4 透明背景的標籤

[透明也是一種顏色]

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！目前狀態是如果底圖圓形與標籤相疊時標籤呈現一個灰灰的方塊，真的不太專業。在.NET 繪圖架構概念中其實「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！執行時的部分畫面如下，已經呈現出透明的效果了！





**8-5 滑鼠互動**

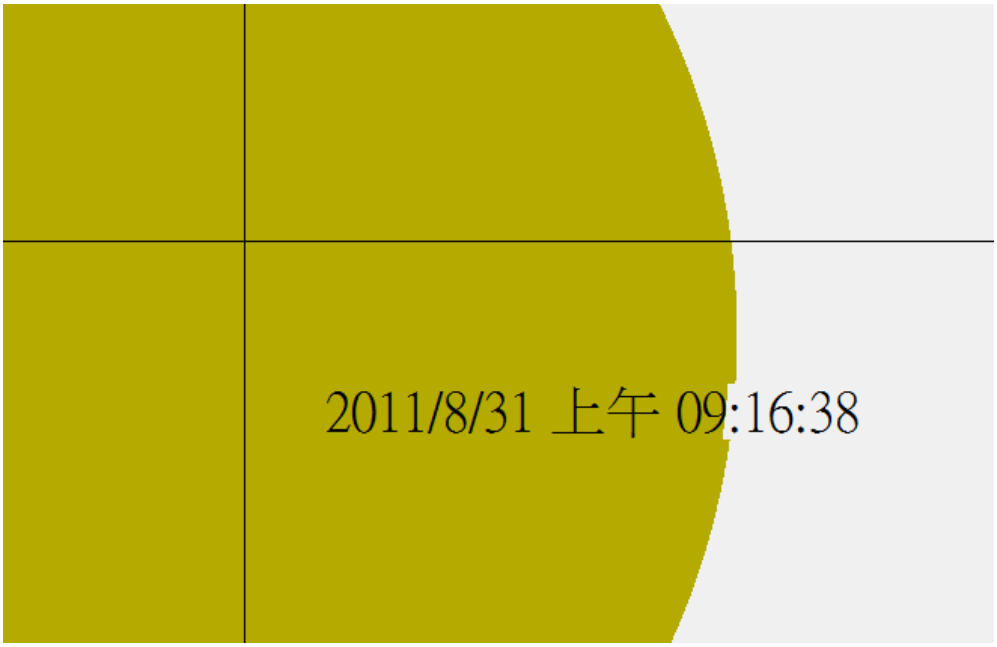
[繪製垂直與水平線]

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，本來應該將此繪圖程式寫在 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「跟隨」的效果吧？我們就試試看作個以滑鼠位置為中心的十字型

```
    int mmx, mmy;
    private void Form1_MouseMove(object sender, MouseEventArgs e)
    {
         mmx = e.X;
         mmy = e.Y;
    }

private void timer2_Tick(object sender, EventArgs e)
{
    int D = R.Next(this.Height);
    int X = R.Next(this.Width  - D);
    int Y = R.Next(this.Height  - D);
    Bitmap bg = new Bitmap(this.Width,  this.Height);
    Graphics g = Graphics.FromImage(bg);
    Color C = Color.FromArgb(R.Next(256),  R.Next(256), R.Next(256));
    SolidBrush B = new SolidBrush(C);
    g.FillEllipse(B, X, Y, D, D);
     g.DrawLine(Pens.Black, 0, mmy, this.Width, mmy);
     g.DrawLine(Pens.Black, mmx, 0, mmx, this.Height);
    this.BackgroundImage  = bg;
}
```

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物件(Pens)，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內跟隨過去。

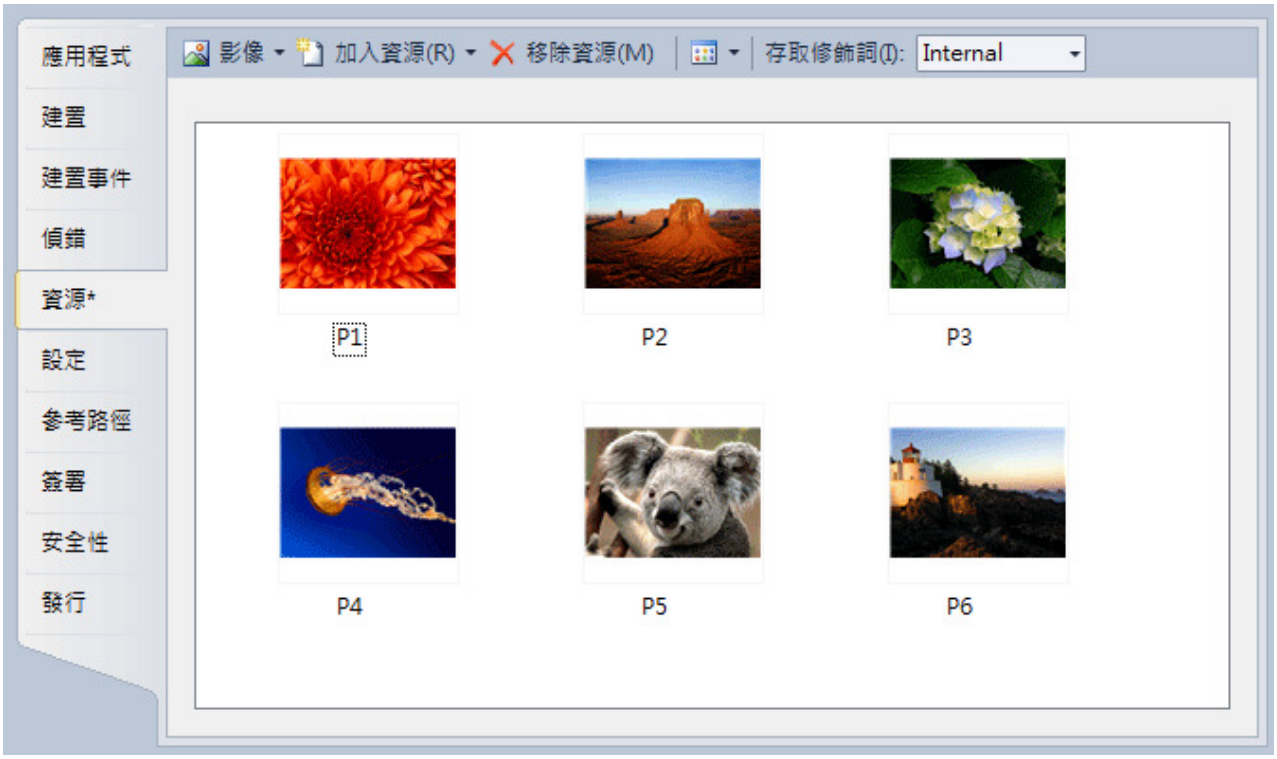



**8-6 播放圖片**

[加入專案資源檔]

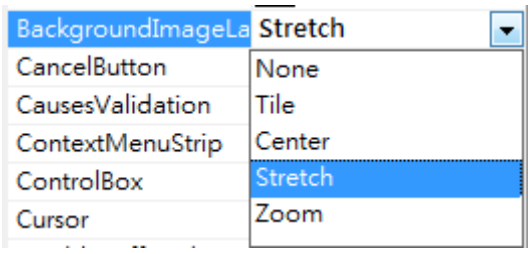
銀幕保護程式也可做成如 PowerPoint 一樣的播放形式，這對很多商家應該很有用，只要將產品圖片放上去就可以把螢幕保護程式當輪播的廣告看板了！在此我們將一般 Windows

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 Properties，就是專案屬性檔，選擇「資源」頁籤→「加入資源」→「加入現有檔案」，之後就是將圖片載入了！操作畫面如下，這是來自 Windows 7 系統的範例圖片，你可以任意選擇喜歡的圖載入，但是為了後續寫程式方便請將資源圖的名稱改成如下的 P1、P2、P3…。




[設定為全螢幕播放]

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，此值預設為 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！




[用註解停止程式碼的執行]

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)，所以測試時須先將小時鐘(label1)隱藏(Visible=False)，並將 timer1.Start 及 timer2.Start 等程式碼前面加兩斜線「註解掉」，C#會將它當作「註解」而不執行。修改過的 Form_Load 應該是這樣的：

```
private void Form1_Load(object sender, EventArgs e)
{
    //label1.Text = DateTime.Now.ToString();
    //timer1.Start();
    //timer2.Start();
    timer3.Start();
}
```

[定時輪播影像的程式]

接著請寫 timer3 的事件副程式如下：

```
int curP = 0;
private void timer3_Tick(object sender, EventArgs e)
{
    curP += 1;
    if (curP > 4) { curP = 1; }
    switch (curP)
    {
        case 1:
            this.BackgroundImage = Properties.Resources.P1;
            break;
        case 2:
            this.BackgroundImage = Properties.Resources.P2;
            break;
        case 31:
            this.BackgroundImage = Properties.Resources.P3;
            break;
        case 4:
            this.BackgroundImage = Properties.Resources.P4;
            break;
        case 5:
            this.BackgroundImage = Properties.Resources.P5;
            break;
        case 6:
            this.BackgroundImage = Properties.Resources.P6;
            break;          }
}
```

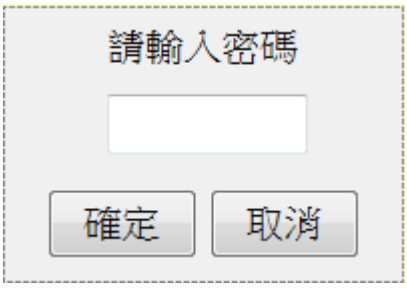
以上程式碼的意義是：宣告一個公用的全域變數 curP 用來記住目前播到哪一張圖片，接著在 timer3 裡面每次時間到就將 curP 加 1！如果 curP 大於 6 表示播完了，就要從第一張再播一次。switch 的語法表示多選項的單選狀況，上面程式依據 curP 的值決定要播哪一張？播放時就是將已經載入的資源(Resource)檔案貼到表單的背景影像(BackgroundImage)而已。這和秀圖軟體單元的投影片播放有點像，但差別是資源檔是程式設計階段就全部載入的，執行中就不再需要到磁碟去讀取檔案了！

**8-7 加入密碼**

[密碼輸入介面]

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！




[密碼輸入與檢核]

這個東西是使用者準備關閉程式(結束全螢幕)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(置中)。程式碼如下：

```
//點擊表單
private void Form1_Click(object sender, EventArgs e)
{
    panel1.Left = (this.Width - panel1.Width) / 2;
    panel1.Top = (this.Height - panel1.Height) / 2;
    panel1.Visible = true;
}
//確定密碼鍵
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "xyz")
    {
        Application.Exit();
    }
    else
    {
        MessageBox.Show("密碼錯誤！");
    }
}
//取消密碼鍵
private void button2_Click(object sender, EventArgs e)
{
    panel1.Visible = false;
}
```

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 panel1 再度隱藏(Visible=False)即可！

[強迫中斷程式的方法]

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)按鍵可以中斷程的執行，但是這一招只在還在設計程式進行偵錯的狀態下有用，一旦安裝為正式程式時就

無效了，否則密碼不就沒意義了嗎？真有那種狀況時可以試試看按 Ctrl+Alt+Delete 按鍵，呼叫作業系統的工作管理員程式，直接關掉這個螢幕保護程式。

**8-8 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 R.Next(2)，等於 1 時畫方，等於 0 畫圓即可。

**課後閱讀**

**談談資源檔**

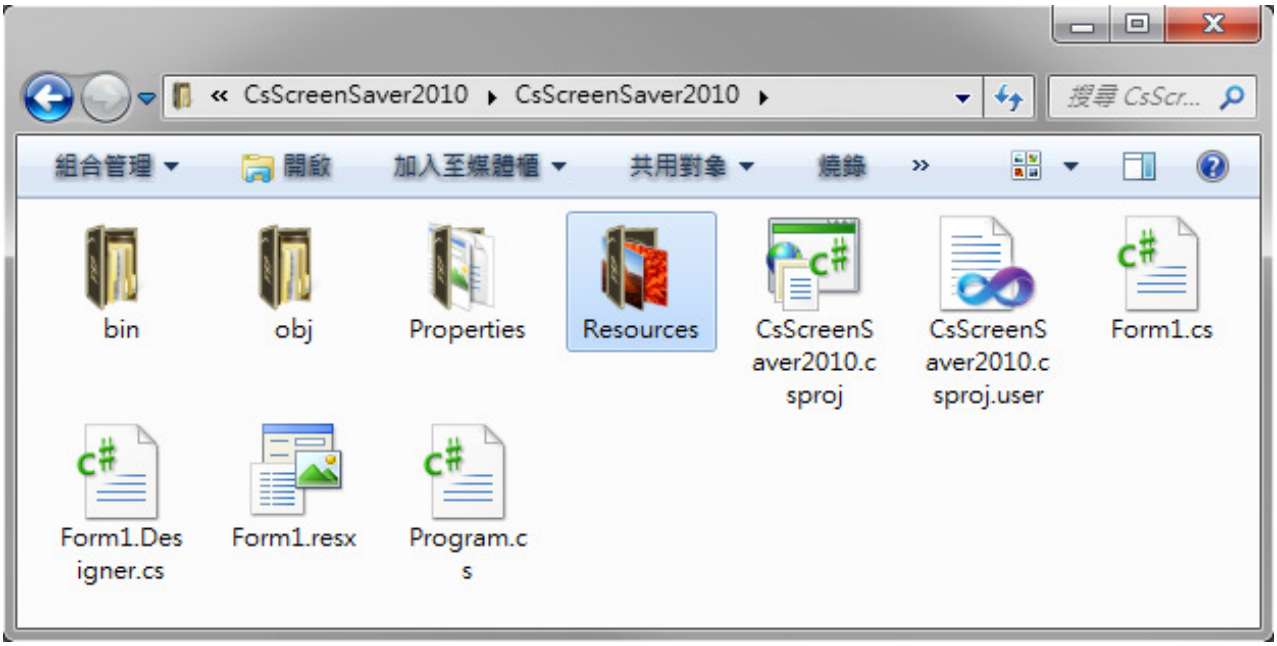
[為何需要資源檔？]

在一般人的概念中『程式』與『資料』是兩種不同的東西，但是很多時候我們希望程式本身也包含一些固定會用到的資料，譬如物件的圖案、說明文字檔、音效甚至多媒體影片等等。筆者最近的一項研究還可以將互動式網頁直接放到程式裡面咧！要做到這種效果，我們當然可以將所需要的資料檔案隨著程式檔一起拷貝到要安裝使用程式的地方。但是如果可以將這些資料直接『封裝』到專案裡面，變成如 Label 或 TextBox 之類程式內的物件，操作起來就會更為方便，安裝程式時也不必擔心遺漏這些檔案了，這就是『資源檔』的用處！

[資源檔的設定位置]

在 C#中，資源檔案屬於專案(Project)的屬性(Properties)之一！所以要處理資源檔案必須開啟方案總管的 Properties 項目進行。基本上任何電腦檔案都可以作為資源檔，專案會先試著將檔案分類，如果可以辨識為影像、文字或音效等等程式預設可以處理的格式，就會自動將它們分類，之後用到它們時就會當作圖片、文字或音效來使用；如果是不認識的格式就歸為『檔案』類，之後如果用到時就會當作位元組陣列(一串數字)來使用。這對很多程式設計者實在是一大福音！譬如像設計遊戲程式一定會用到非常多的圖片以及音效，想一想！如果每次安裝程式都必須分別處理這些大量的檔案會有多麻煩？

當使用者將檔案加入專案之後這些資源檔會被拷貝到一個 Resources 目錄內(如下圖)，並給它取一個專案內使用的名字，如果你不刻意修改它，應該就是檔案原來的名稱，告訴你這件事情是因為如果你設計中途想修改資源檔案的內容，可以直接到此地打開檔案修改，不必重複的刪除資源再加入資源，譬如程式的說明檔如果加入資源之後一定會常常修改的。



另一方面，如果你改了資源名稱，如本範例我們將複雜的圖檔名改成 P1、P2 等等，在專案目錄中的圖檔名並不會改變，如下是本專案的 Resources 目錄內容，專案內會記錄 P1 對應到哪一張圖，所以不要任意在檔案總管內更改圖名，這會讓專案產生混亂，這種資料不同步的錯誤也較難處理。