第 10 章打地鼠遊戲

簡介：

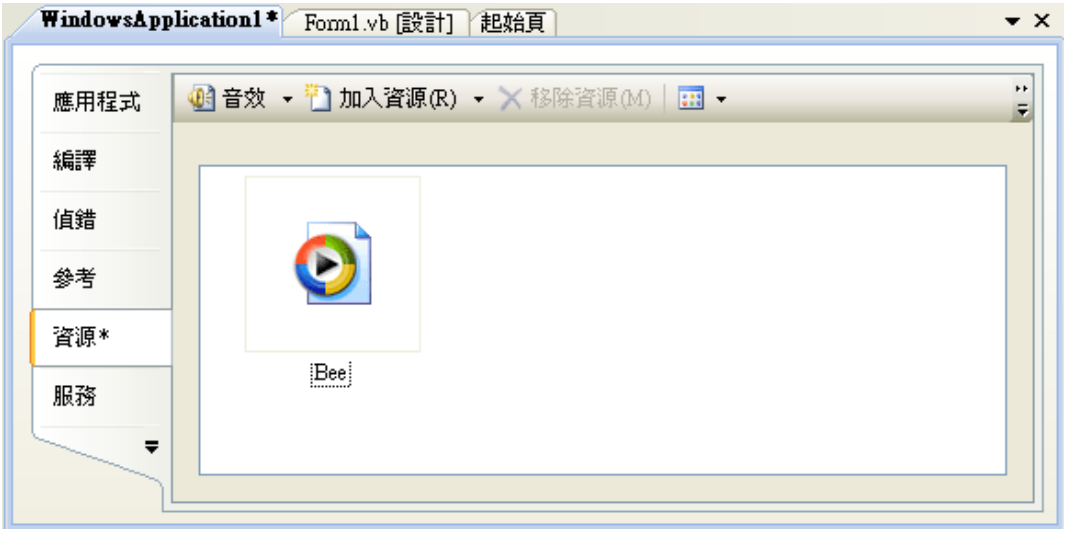
本單元要介紹一個互動性較高的打地鼠遊戲，目標圖案會在預設的幾個位置隨機出現，短時間內再度消失，玩家必須用滑鼠及時點擊目標得分，遊戲還會限時結束且能計分。包含的程式技巧有資源檔案的使用，隨機亂數的操作，計時器的使用，以及用全域變數計分的概念，這些都是遊戲程式常常用到的基本技術。

10-1 加入資源圖檔

當我們需要在程式專案內使用影像時，基本上有兩種方式：一是在需要使用影像物件的 Image 或 BackgroundImage 屬性欄中選取檔案載入；一是將圖檔載入為專案的「資源」檔。前者的影像隨後就專屬於該物件，後者則變成專案內任何物件都能以程式碼操作隨時引用的「資源」。在本單元中地鼠物件的圖案在執行中需要切換，所以我們選擇採用資源檔的方式載入。

同樣的，音效檔案也可以在設計階段事先載入為資源檔。不論使用哪一種方式，都告訴我們一個概念！我們可以將很多專案需要的圖檔、音效，甚至文字或文件檔案先放在專案內，變成專案內部的東西，就像工具箱物件一樣，使用起來就很方便了！換言之，你的專案需要的很多東西都可以與專案直接整合，不必製作成程式後還得帶著很多檔案同行！

載入資源檔的動作是：開啟專案後開啟「方案總管」中的 My Project 項目，進入「專案屬性」頁，選擇「資源」頁籤，選擇「加入資源」→「加入現有檔案」。依序加入兩個圖檔(face.gif & boom.gif)，以及一個音效檔(Bee.wav)，兩圖的長寬都是 100 像素點，這是稍後當作目標以及目標被點中後的圖案，音效檔則作為目標被點中時的聲音。



請注意到載入的資源檔案專案會自動將它們分類，如上面的兩個畫面，不管你是一一載入或一起載入，音效與影像都會分在兩個頁面，不要誤會載入失敗而重複載入。當然這些資源檔不用時也應該將它們刪除，以免你的最終程式負載太多無用的東西，為了寫程式方便，在此也可以將它們的名字變更。

10-2 表單物件佈置

將表單標題修改為『打地鼠遊戲』，置入一個容器 Panel 物件，Dock(停駐)屬性設定為在表單的下方(Bottom)。在 Panel 中置入一個計分用的標籤，寫『分數=0』(Label1)；置入一個計時標籤，寫『時間=30』(Label2)；加入一個啟動按鍵(Button1)文字寫 Play。

在視窗內容區(ClientSize)加入六個有臉形圖案的標籤(Label)，做法是先將 AutoSize 屬性設為 False；刪除 Text 屬性中的預設文字；選擇 Image 屬性的影像為資源檔中的臉形圖案，長寬調整為影像的 100X100。並將六個標籤分別改名為 M1~M6(Name 屬性)以簡化後續程式碼的寫作，完成畫面如下：

或許有同學會問：顯示影像的物件不是應該用 PictureBox 嗎？為何在此要用 Label？圖片還不太好設定咧！主要原因是 PictureBox 相對於 Label 是個比較複雜功能較多的物件，這會佔去較多的記憶體，多出的功能如影像縮放旋轉等等，我們在此又用不到！如果可以節省下這些電腦資源，你的程式會運轉更快，也有更大的設計空間，譬如放置更多地鼠。這其實是一個概念的學習，雖然現在的電腦硬體都很好，記憶體也很大，但是同樣的環境下，如果你可以讓自己的程式更小更快始終是會有優勢的！

10-2 啟動遊戲→使圖案隨機出現

要讓地鼠定時的隨機出現一隻，最簡單的方法是利用一個計時器，每次時間到時先將所有地鼠隱藏，再用亂數隨機選一隻出來顯示。同時間「Play」的按鍵就用來啟動這個計時器。請自工具箱的「元件」類加入一個 Timer1，將其 Interval(時間間隔)屬性設為 1000(等於一秒)。接著雙擊 Play 按鍵及 Timer1 物件寫程式如下：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Button1.Click
    Randomize()
    Timer1.Start()
End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer1.Tick
    For i As Integer = 1 To 6
        Controls("M" + i.ToString).Visible = False
    Next
    Dim j As Integer = Int(Rnd() * 6) + 1
    Controls("M" + j.ToString).Visible = True
End Sub
```

首先在 Play(Button1)按鍵程式中我們使用 Randomize()指令將亂數序列打亂，少了它每次程式開始時地鼠出現的順序都會一樣，就少了許多懸疑性！接著就是啟動計時器(Timer1)了。計時器程式是先用一個迴圈將 M1~M6 物件都隱藏起來，一個有趣的語法是：Controls("M"+i.ToString)，Controls 代表控制項，括號內的參數是控制項的名稱(Name 屬性)，"M"加上"i"變數就分別代表 M1~M6 了！譬如 Controls("M1")就等同於物件 M1，這種取得物件的方式在遊戲設計中常常會用到。

程式碼 Dim j As Integer = Int(Rnd() * 6) + 1 是宣告一個 1 到 6 之間隨機選擇的整數，Rnd 是一個 0 到 1 之間(小於 1)的隨機實數，乘上 6 之後就變成 0~5 之間的實數，取整數後變成 0~5 的隨機「整數」，再加 1 就是 1 到 6 之間的隨機整數了！接著再用 Controls 的技巧取得 M"j"這個物件讓它顯示出來(Visible=True)就 OK 了！

10-3 打地鼠囉

這個遊戲需要計時計分，所以程式一定需要全域的公用變數來紀錄這兩個參數，請先在程式開始處這樣宣告。

```
Dim S As Integer = 0 '分數
Dim T As Integer = 30 '時間
```

接下來我們要加入打地鼠的互動程式。地鼠(目標)被點中的事件可以寫在 Click 事件中，但是我們有六隻地鼠，所以必須再度使用到共用事件副程式的技巧，否則就必須寫六個事件副程式了。寫法是先點選產生其中一個物件的事件副程式框架，然後在副程式標題行後面的 Handles ...後面依序加入要共用此副程式的事件，分別是：M2.Click, M3.Click..., M6.Click。

在此我們加上計分機制，宣告一全域公用的計分變數 S，每次目標被點中時分數會加 1 並顯示於 Label1 的看板；同時發出一個音效；還不夠炫嗎？就讓它連圖案都一起變化吧！

```
Private Sub M1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles M1.Click, M2.Click, M3.Click, M4.Click, M5.Click, M6.Click
    S += 1
    Label1.Text = "得分" + S.ToString
    My.Computer.Audio.Play(My.Resources.Bee, AudioPlayMode.Background)
    sender.Image = My.Resources.boom
End Sub
```

上述程式 S+=1 就是加分，Label1 顯示得分，播放音效則由 My.Audio.Play 捷徑功能執行，其中的 AudioPlayMode.Background 選項是指「背景播放」，就是播音效時其它的視窗動作都不要暫停的意思。最後是將事件觸發者(sender)的影像(Image)改成 boom 就是一個鞭炮爆炸的影像。不過既然在此將影像改掉表示被打中，那下次重新出現之前就必須重設影像，Timer1 程式可以修改如下：

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer1.Tick
    For i As Integer = 1 To 6
        Controls("M" + i.ToString).Visible = False
    Next
    Dim j As Integer = Int(Rnd() * 6) + 1
    Dim Q As Label = Controls("M" + j.ToString)
    Q.Visible = True
    Q.Image = My.Resources.face
End Sub
```

上述程式迴圈部分與之前一樣，接著宣告一個 Q 代表被選定的地鼠物件，將其影像屬性(Image)用資源檔(My.Resource.face)取代，就是復原為原來的臉形圖案啦！

10-4 防止快手搶分

至此地鼠圖案會正常的切換了！只是有個小漏洞，如果手腳快的玩家可以趁同一隻地鼠未消失之前連續點擊得分，這好像有點不合理？我們必須將地鼠是否已經被點擊的狀態記住，被點過就不能再回應點擊動作(加分)了，下次出現之前又要恢復為未點擊狀態，傳統技術上好像必須建立一個陣列來記住這個資訊。但事實上多數物件都有個 Tag(標籤)屬性可以擔任這個暫時註記的工作，我們就用它來記錄吧！被點過的 Tag = 1，未被點就是 0 了！加入此一機制之後程式碼如下：

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer1.Tick
        ...
        ...
    Q.Image = My.Resources.face
    Q.Tag = 0
End Sub

Private Sub M1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles M1.Click, M2.Click, M3.Click, M4.Click, M5.Click, M6.Click
    If sender.Tag = 0 Then
        S += 1
        Label1.Text = "得分" + S.ToString
        sender.Image = My.Resources.boom
        My.Computer.Audio.Play(My.Resources.Bee, AudioPlayMode.Background)
        sender.Tag = 1
    End If
End Sub
```

簡單說就是在地鼠出現時定義牠的 Tag 是 0，被點擊時先檢查牠的 Tag，如果是 0 才可以執行加分等動作，最後讓 Tag=1。

10-5 計時機制

接下來要加入一個倒數計時的機制讓遊戲可以按時結束。方法是使用一個新的計時物件 Timer2，Interval 屬性也是設定為 1000(一秒)，每次減一秒並顯示於

看板(Label2)，時間到了就會關掉 Timer1 和它自己(Timer2)，遊戲也就結束了，應該出現一個 Game Over 的訊息！當然，Timer2 本身也必須在遊戲開始時開啟，而且分數與時間都必須在此重設！

在此為了增加視覺效果，可以增加一個工具箱中通用控制項的 ProgressBar 物件，將它停駐在控制面板的底部，Maximium 屬性設為最大時間的 30，Value 也設為 30，表示從此開始倒數，之後程式碼只要將時間變數 T 與 ProgressBar1 的 Value 屬性同步，我們就會看到如沙漏一般的倒數計時圖形化介面了。最後 Play 按鍵與 Timer2 計時器的程式如下：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Button1.Click
    Randomize()
    S = 0
    Label1.Text = "得分=" + S.ToString
    T = 30
    Label2.Text = "時間=" + T.ToString
    Timer1.Start()
    Timer2.Start()
End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Timer2.Tick
    T -= 1
    ProgressBar1.Value = T
    Label2.Text = "時間=" + T.ToString
    If T = 0 Then
        Timer1.Stop()
        Timer2.Stop()
        MsgBox("遊戲結束！" + Label1.Text)
    End If
End Sub
```

執行中畫面如下：

10-4 多個地鼠同時出現

多數這一類遊戲到了後半段都會變快且同時出現不只一隻地鼠。要讓我們的程式也可以同時出現好幾隻地鼠相當簡單，只要加上一個迴圈就可以了！還記得挑選「一」隻地鼠的程式段落嗎？用迴圈將它包起來就 OK 了！

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
 Handles Timer1.Tick
    For i As Integer = 1 To 6
        Controls("M" + i.ToString).Visible = False
    Next
    For k As Integer = 1 To 3
        Dim j As Integer = Int(Rnd() * 6) + 1
        Dim Q As Label = Controls("M" + j.ToString)
        Q.Visible = True
        Q.Image = My.Resources.face
        Q.Tag = 0
    Next
End Sub
```

請注意到，迴圈重複執行亂數選取時每次都是獨立事件，可能會有「相同」或「不同」的結果，上面的迴圈 For k As Integer = 1 To 3 會無條件執行三次選擇地鼠的動作，它其實有可能會重複，所以有時候只出現兩隻，甚至一隻地鼠都算是正常的！

10-5 忽快忽慢的地鼠

地鼠出現的快慢其實是由 Timer1 物件的 Interval 屬性決定的，發揮一下想像

力，Interval 也可以隨時改變的！我們可以每次時間到除了做預設動作之外，也將 Interval 做一個幅度內的隨機改變，如下程式可以將 Interval 在 0.5~1.5 秒之間隨機變化，將這行程式加入 Timer1 事件中，每次時間的間隔就都會變化了！

```
Timer1.Interval = 500 + 1000 * Rnd()
```

10-6 進階挑戰

一、可以讓地鼠不要限制在固定位置嗎？

提示：可以只用一個地鼠物件，每次出現時隨機改變其位置。

二、如何設計不同難度的關卡？

提示：可用 Interval 調整速度，用 10-4 的迴圈控制地鼠出現的個數。

課後閱讀

沒有物件陣列怎麼辦？

寫程式時常常會使用陣列來處理大量同質性的資料，譬如我們可以宣告全班的分數為一個數字陣列，然後用迴圈給全班一樣的分數 80 分，程式碼如下：

```
Dim S(50) As Integer
For i As Integer=0 to 50
S(i)=80
Next
```

但是在本單元碰到我們想將六隻地鼠一起隱藏起來的狀況，卻發現我們沒辦法將六個「物件」變成陣列的關係，因此也好像不能用迴圈簡單處理了！這是目前「.NET」程式預設的限制，事實上在以往的 VB6 時代是可以有「物件陣列」的！時代好像倒退了，以前真的可以寫如下的迴圈(現在這樣寫一定是錯的)：

```
For i As Integer=0 to 5
Label1(i).Visible=False
Next
```

說實話，我也不知道新版 VB 如此限制的原因，但是不用迴圈真的是一大災難，本單元中只有六支地鼠，寫六行程式讓地鼠消失也就罷了，如果是打磚塊遊戲裡的幾百個磚塊要一起消失呢？複製幾百行程式看起來就會非常愚蠢了！

各位已經看到我們的程式範例提出了解決之道，就是使用物件的名稱(Name 屬性)去取得物件的參考，譬如在程式之中 Me.Controls("M1")事實上與直接寫 M1 的效果相同！Me.Controls("M1").Visible=False 與 M1.Visible=False 都可以使 M1 消失，也因此我們可以用有系統的命名，如 M1，M2，M3…M6，再搭配名稱變數，就可以用迴圈處理大量物件了，前提是「有系統的命名」，程式如下：

```
For i As Integer = 1 To 6
   Controls("M" + i.ToString).Visible = False
Next
```

i.ToString 讓 i 的數字屬性變成文字，前面加上"M"就是 M1，M2…了！程式中的 Controls 表示控制項的「集合」，所以 Me.Controls 的意義是表單內所有控制項的集合之意，如果省略 Me 預設也是表單的意思。在此有個小陷阱，就是控制項的歸屬是有階層性的！如果你將一群物件(控制項)放到某一個「容器」如 Panel1 之內，那麼要取得這些物件的程式就必須是 Me.Panel1.Controls(物件名稱)了，直接寫 Me.Controls(物件名稱)會找不到物件的。如果你夠細心可能還會注意到本單元還有下面這種程式的語法：

Dim Q As Label = Controls("M" + j.ToString)

這麼寫的意思是宣告一個物件變數 Q 來承接我們用 Controls 抓到的物件，好處是接下來如果要對此物件作好幾個動作時，只要用 Q 變數就可以了！但是這裡也有一個小陷阱，可能稍後你就會學到，物件的標準宣告常常是要加上 New 關鍵字的！像這樣：

Dim K As New Label

不寫 New 表示 K 只是一個名稱，如同未出生先命名的孩子，實體是不存在的，如果你接著寫 K.Visible=False 可能就會收到錯誤訊息了(也可能是程式沒反應)！但是在本單元你千萬不能寫 New(寫了反而會錯誤)，因為這個物件的實體不是來自你的宣告，而是來自 Controls 函數抓到的現有物件！簡單說，Q 只是這個已經存在物件暫時使用的別名而已！

這種技巧一般書籍很少提到，但是在遊戲設計程式中非常重要，因為我們常常必須處理大量物件的狀態變化，如果沒有這一招，又不能使用物件陣列實在非常困擾！當然我們不能太低估「.NET」的能力，嚴格講他還是可以使用物件陣列的，只是不在預設功能之內，光是設定使用就煩死人了！他們自己也宣稱這些只是為了與舊版的 VB6 相容而作的「不得已」的設計！

總之，記住這個技巧就不必擔心大量物件處理的程式該怎麼寫了！