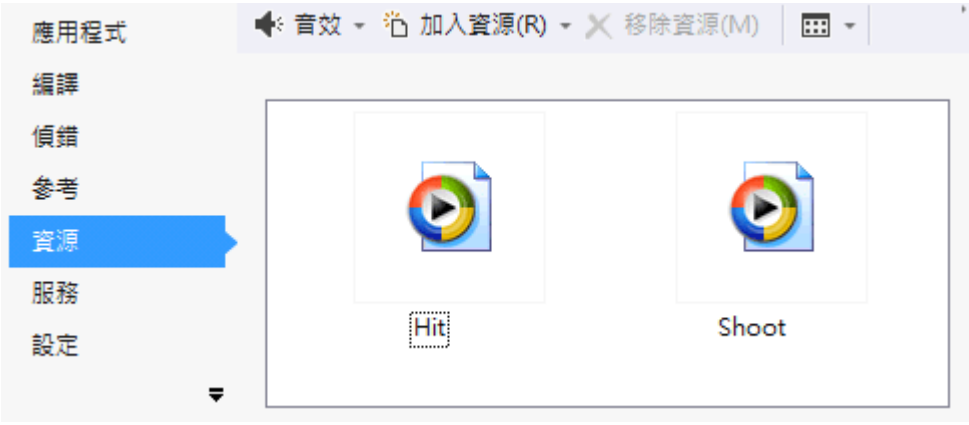
第 15 章射擊遊戲

簡介：

射擊遊戲是非常盛行的遊戲種類，它的基本元素包括：可以移動及射擊的槍枝、會移動的目標、會自動飛行並擊破目標的砲彈、開槍以與擊中目標的音效，當然還要有計分與結束遊戲的機制等等。本章將以最精簡的程式，具有代表性的設計出以上的各項功能，讓讀者們體驗一下此類程式運作的核心機制。

15-1 加入資源檔案

請仿照之前打地鼠遊戲等單元的方式加入書附光碟中的兩個圖檔以及兩個音效，結果顯示於專案屬性資源頁如下圖：

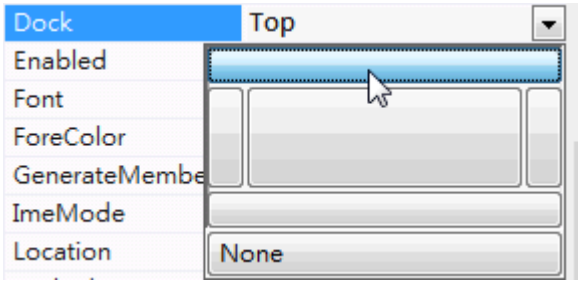


其中的 Shooter 圖檔將作為發射砲彈的「槍枝」，Target 就是要被射擊的目標了！音效部分 Shoot 是開槍的聲音，Hit 則是擊中目標時的聲音。如果你只是練習使用，事實上搜尋自己電腦裡面「*.wav」的檔案就會找到很多可用的素材。圖案部分也是如此，可用的免費圖庫資料網路上很多，自行繪製亦可。

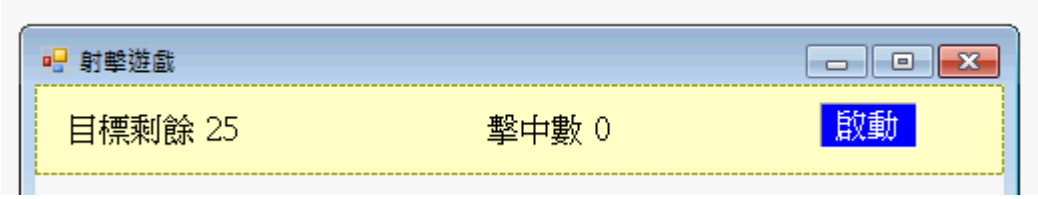
15-2 表單物件佈置→資訊面板

首先將設計表單拉大到約 500 像素點長寬，標題改為「射擊遊戲」，到工具箱的「容

器」項目載入一個面板(Panel)物件，選擇 Dock 屬性置於表單上方(Top)。如下圖：



在 Panel 物件中佈置五個標籤(Label)，分別寫入「目標剩餘」、「25」、「擊中數」、「0」以及「啟動」，如下圖：



將顯示「25」的標籤改名為「Tn」，用於顯示剩餘目標數，本遊戲會依序跑出定額的目標，目標跑完遊戲就結束了！所以這個 25 的數字在遊戲進行中是會遞減的。接著將顯示「0」的標籤改名為「Hn」表示擊中的目標數目，最後將「啟動」標籤更名為「StB」當作啟動程式的按鈕。

在此有點特別的是好像用 Label 物件(StB)取代了應該是 Button 的工作！原因之一是我們稍後會使用到向左(←)向右(→)的按鍵驅動我們的槍枝，但是如 Button 之類的物件常常預設會攔截這些特殊鍵，作為物件焦點移轉(譬如 Button1 到 Button2)，或物件內的選項移動，一個簡單規避的方法就是使用 Label 這種無法獲得焦點的物件，但是它依舊可以產生 Click 事件，也就可以當作 Button 來使用了！

15-3 表單物件佈置→槍枝與目標

接著在 Panel 物件以下的區域，加入三個 PictureBox 物件在比較靠上的位置作為射擊目標，三者水平等間距分布，影像設為資源檔的 Target，要讓 PictureBox 的大小適合內部的影像，只需將 SizeMode 屬性設為 AutoSize 即可。Tag 屬性則設定為「T」，這是讓後續程式辨識它們是「敵機」的記號。上述三個目標在遊戲一開始時應該是看不見的，請將 Visible 屬性都設為 False。接著再加一個 PictureBox 置於下方，設定影像為資源檔的 Shooter，物件更名為 P，這是作為槍枝的物件，完成的設計畫面如下圖：

15-4 槍支的移動

本範例中槍枝的移動是用鍵盤的向左(←)與向右(→)按鍵，首先必須將表單的 KeyPreview 屬性改為 True，接著在表單 Form1 的 KeyDown 事件中寫程式如下：

```
'移動槍枝
Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown
    Select Case e.KeyCode
        Case Keys.Left '槍枝左移
            P.Left -= 5
        Case Keys.Right '槍枝右移
            P.Left += 5
    End Select
End Sub
```

如程式所示，按向左鍵槍枝 P 會左移五點，向右鍵 P 右移五點。試試看，執行程式後你的槍枝應該可以用鍵盤左右移動了！

在打乒乓球的單元中我們有替球拍寫限制不能移出畫面的程式，此處似乎也要寫才對？但事實是用滑鼠拖曳的球拍一旦移出畫面，因為無法再度用滑鼠抓取移回(點不到它了)，所以限制範圍的程式非寫不可！相對的，使用鍵盤移動的物件即使已經跑出畫面，仍然可以用按鍵繼續操作將它移回來，限制範圍的程式可以選擇不寫。

15-5 製作砲彈的程式

類似打磚塊遊戲中製作的磚塊，在此砲彈也是程式動態產生的物件，但是它有一個特性是永遠應該出現在槍口正前方！所以不必經由參數定義它的起始位置，而是在作砲彈的副程序內直接將新砲彈放在槍口就好，我們將此副程序命名為 Blt，程式碼如下：

```
'新增砲彈物件
Sub Blt()
    Dim B As New Label '產生新砲彈
    With B
        .Tag = "B" '標記為砲彈
        .Width = 3
        .Height = 10
        .BackColor = Color.Red ‘紅色
        .Left = P.Left + P.Width / 2 - B.Width / 2 'X座標調至槍枝中央
        .Top = P.Top - B.Height 'Y座標調至槍枝正上方
    End With
    Me.Controls.Add(B) '加入表單
End Sub
```

砲彈是一個 Label 物件，Tag 屬性設定為「B」，稍後的程式就用此屬性知道它是一顆砲彈，可以繼續追蹤並定義它應有的行為，譬如持續飛行、擊中目標的反應以及何時該銷毀等等。它的形狀是細長的紅色，模擬夜空的炮火。位置在槍枝正前方的中央。

15-6 發射砲彈的程式

接著我們用鍵盤的空白鍵(Space)當作射擊紐，KeyDown 程式加一個 Case 如下：

```
'移動槍枝與發射砲彈
Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown
    Select Case e.KeyCode
        Case Keys.Left '槍枝左移
            P.Left -= 5
        Case Keys.Right '槍枝右移
            P.Left += 5
        Case Keys.Space '發射砲彈(空白鍵)
            Blt()
            My.Computer.Audio.Play(My.Resources.Shoot, AudioPlayMode.Background) '槍聲
    End Select
End Sub
```

每次發射砲彈時就呼叫 Blt 副程序產生一顆新的砲彈，同時發出設計的音效。此時執行程式，試試看按空白紐發射砲彈，並用左右按鍵移動槍枝，結果如下：

15-7 移動砲彈的程式

如上圖，砲彈是會依據移動的槍枝位置，總是出現在槍口了！但是並不會如預期地往前跑。此時請到設計表單從工具箱加入一個 Timer1 元件，先將 Enabled 屬性設為 True，雙按該物件寫程式如下：

```
'砲彈移動
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    For Each i In Me.Controls
        If i.Tag = "B" Then '每一顆砲彈
            i.Top -= 5 '上移五點
            If i.Bottom < Panel1.Height Then '砲彈超出範圍
                i.Dispose() '刪除
            End If
        End If
    Next
End Sub
```

上述程式使用 For Each 迴圈一一取得表單上所有的控制項(Me.Controls)，如果控制項的 Tag 屬性為「B」代表它是炮彈，應該持續往前(上)飛行，所以 Top 值減 5。但是如果它已經飛出遊戲可見的畫面之外(砲彈 Bottom 小於 Panel1 的高度)，則此砲彈應該自動銷毀(Dispose)。這樣砲彈就會動了，而且跑出畫面後會自動銷毀，這很重要，因為一直新增砲彈物件卻永遠不會銷毀，最後就是塞爆記憶體了！移動槍枝並連續發射砲彈的畫面大概如下：

當然你可以調整 Timer1 的 Interval 屬性(預設是 100 毫秒，相當 0.1 秒)，讓砲彈的速度合乎你的需求。

15-8 移動目標的程式

我們的射擊目標就是畫面上方的三架飛機，但是必須讓它們動起來，機頭向右就該向右飛行，而且超出畫面之後還要拉回左邊重新出場，像螢幕保護程式單元中移動的小時鐘一樣。這必須再加入一個 Timer2 物件，也是先將 Enabled 屬性設為 True，並寫程式如下：

```
'控制目標移動(多個目標循環出現)
Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles Timer2.Tick
    For Each i In Me.Controls
        If i.Tag = "T" Then '每個目標
            i.Left += 5 '右移五點
            If i.left > Me.ClientSize.Width Then '超出表單
                i.Left = -i.Width - Rnd() * 20 '左邊出場，加入20點的不確定性
                i.Visible = True '目標可見
            End If
        End If
    Next
End Sub
```

程式一開始是抓取 Tag 為「T」的物件為處理對象，記得嗎？我們設計時將三個目標的 Tag 都設為「T」了！往右移就是 Left+=5，超出畫面寬度(Me.ClientSize.Width)時將 Left 值拉回左邊「-Width」的位置，就是讓飛機整隻藏在畫面外等待上場。但是目標間距永遠一樣會讓遊戲區少變化，在此加上一個 Rnd() * 20 的數值就是 0~19 的隨機數字，目標間距就會隨機的略有不同了！當然你可以調整 20 這個數字，讓它變化更大或更小。因為目標飛行中有可能被擊中，稍後程式會讓它因此消失(Visible=False)，所以重新出場時也必須重新設定飛機的屬性 Visible = True。

15-9 剩餘目標計數

此時執行程式，目標飛機會循環的從左邊出現，飛到右邊消失。本範例想要設計的遊戲結束狀況是目標 25 個全部飛完後就不再出現新目標，程式也等於結束了。這個機制必須先宣告一個公用變數(NT)記憶已經飛出了幾個目標，Timer2 相關程式應修改成：

```
Dim NT As Integer = 25 '目標總數
Dim H As Integer = 0 '擊中目標數
'控制目標移動(四個目標循環出現)
Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles Timer2.Tick
    For Each i In Me.Controls
        If i.Tag = "T" Then '每個目標
            i.Left += 5 '右移五點
            If i.left > Me.ClientSize.Width Then '超出表單
                i.Left = -i.Width - Rnd() * 20 '加入20點的不確定性
                i.Visible = True '目標可見
                NT -= 1 '目標總數減一
                If NT < 0 Then '目標跑完了
                    i.Visible = False '隱藏
```

```
                    NT = 0 '剩餘目標0
                End If
                Tn.Text = NT '顯示剩餘目標數
            End If
        End If
    Next
End Sub
```

其中的 NT 就是應有的目標總數，每飛出一架飛機 NT 值就減 1，NT 小於或等於 0 時就算遊戲結束了！此處結束程式時沒有直接關閉 Timer1，意義是讓已經出現的目標可以繼續跑完全程，但是 NT <= 0 時新目標一直都是看不見的(Visible=False)的。此處順便也宣告了擊中目標數的變數 H，初始值當然是 0！

15-10 啟動按鍵

之前為了測試方便我們將兩個 Timer 的 Enabled 屬性都設為 True，就是程式啟動直接開啟計時器，現在請將啟動權還給程式設計的啟動按鍵(StB)，先將兩個 Timer 的 Enabled 屬性都回歸預設的 False，再雙按啟動鍵寫程式如下：

```
'啟動
Private Sub StB_Click(sender As Object, e As EventArgs) Handles StB.Click
    NT = 25 : Tn.Text = NT '重設目標數
    H = 0 : Hn.Text = H '重設擊中數
    Timer1.Start() '啟動砲彈控制
    Timer2.Start() '啟動目標控制
End Sub
```

就是將目標數與擊中數回歸初始值，並啟動兩個 Timer。如前所述，NT > 0 時新目標就會繼續出現，所以這個按鍵其實也是重玩鍵！

15-11 碰撞偵測的程式

接下來要處理砲彈擊中目標時的遊戲反應，首先必須寫一個檢查砲彈與目標是否有碰撞的自訂副程序 chkHit。基本邏輯與打磚塊遊戲的碰撞相似，但是此處碰撞的物件不需要反彈，所以程式碼簡單很多：

```
'碰撞偵測程式
Function chkHit(B As Label, T As PictureBox) As Boolean
    If B.Right < T.Left Then Return False '球在物件之左
    If B.Left > T.Right Then Return False '球在物件之右
    If B.Bottom < T.Top Then Return False '球在物件之上
    If B.Top > T.Bottom Then Return False '球在物件之下
    Return True '碰撞
End Function
```

參數 B 與 T 就是需要檢查碰撞與否的兩個物件，在此例中 B 是砲彈，T 就是敵機了！比較兩者的邊緣座標：Left, Right, Top 與 Bottom 等，排除四種確定無碰撞的情況(Return False)之後其他的狀況就是有碰撞(Return True)了！相關邏輯解釋請參閱打磚塊

遊戲單元。

接著是在砲彈移動時檢查該砲彈有無與任何一個目標碰撞？程式碼是加在 Timer1 裡面的，擴充的 Timer1 程式如下：

```
'砲彈移動
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    For Each i In Me.Controls
        If i.Tag = "B" Then '每一顆砲彈
            i.Top -= 5 '上移五點
            For Each j In Me.Controls
                If j.Tag = "T" And j.Visible = True Then '每一個可見目標
                    If chkHit(i, j) Then '砲彈擊中目標(碰撞)時
                        i.Dispose() '刪除砲彈
                        j.Visible = False '目標消失
                        H += 1 '擊中數加一
                        Hn.Text = H.ToString
                        '擊中音效
                        My.Computer.Audio.Play(My.Resources.Hit, AudioPlayMode.Background)
                    End If
                End If
            Next
            If i.Bottom < Panel1.Height Then '砲彈超出範圍
                i.Dispose() '刪除
            End If
        End If
    Next
End Sub
```

就是在每一顆砲彈移動之後逐一取出 Tag 值為「T」，且尚未被擊中(Visible=True)的可見目標，檢查該目標與砲彈是否有碰撞？如果有，就將砲彈物件刪除(Dispose)，也將被擊中的目標隱藏(Visible=False)。擊中之後當然還必須計分(H+=1)，以及加入音效(Hit)。

碰撞的砲彈與目標一個刪除一個隱藏，處理方式不同，是因為砲彈此後不會繼續使用，目標則還要繼續循環使用。但是已經被打到一次的目標(Visible=False)繼續被其他砲彈打到應該不計分，所以限制擊中與否的判斷僅限於可見的目標。

本範例程式至此全部完成，遊戲進行中畫面可能如下：

15-12 進階挑戰

一、如何製作打中會扣分的目標(友軍)？

提示：可製作具有不同 Tag 值的目標為敵或友軍，打到敵軍加分，打到友軍扣分。

二、如何製作可以選擇武器圖案的介面？

提示：製作選擇圖檔介面切換武器圖案。

專案設計頁面與程式碼

Form1：


```
Public Class Form1

    '移動槍枝與發射砲彈
    Private Sub Form1_KeyDown(sender As Object, e As KeyEventArgs) Handles MyBase.KeyDown
        Select Case e.KeyCode
            Case Keys.Left '槍枝左移
                P.Left -= 5
            Case Keys.Right '槍枝右移
                P.Left += 5
            Case Keys.Space '發射砲彈(空白)
                Blt()
                My.Computer.Audio.Play(My.Resources.Shoot, AudioPlayMode.Background) '開槍音效
        End Select
    End Sub
```

```
'新增砲彈物件
Sub Blt()
    Dim B As New Label '產生新砲彈
    With B
        .Tag = "B" '標記為砲彈
        .Width = 3
        .Height = 10
        .BackColor = Color.Red '紅色
        .Left = P.Left + P.Width / 2 - B.Width / 2 'X座標調至槍枝中央
        .Top = P.Top - B.Height 'Y座標調至槍枝正上方
    End With
    Me.Controls.Add(B) '加入表單
End Sub

'砲彈移動
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    For Each i In Me.Controls
        If i.Tag = "B" Then '每一顆砲彈
            i.Top -= 5 '上移五點
            For Each j In Me.Controls
                If j.Tag = "T" And j.Visible = True Then '每一個可見目標
                    If chkHit(i, j) Then '砲彈擊中目標(碰撞)時
                        i.Dispose() '刪除砲彈
                        j.Visible = False '目標消失
                        H += 1 '擊中數加一
                        Hn.Text = H.ToString
                        '擊中音效
                        My.Computer.Audio.Play(My.Resources.Hit, AudioPlayMode.Background)
                    End If
                End If
            Next
            If i.Bottom < Panel1.Height Then '砲彈超出範圍
                i.Dispose() '刪除
            End If
        End If
    Next
End Sub

'碰撞偵測程式
Function chkHit(B As Label, T As PictureBox) As Boolean
    If B.Right < T.Left Then Return False '球在物件之左
    If B.Left > T.Right Then Return False '球在物件之右
    If B.Bottom < T.Top Then Return False '球在物件之上
    If B.Top > T.Bottom Then Return False '球在物件之下
    Return True '碰撞
End Function

Dim NT As Integer = 25 '目標總數
Dim H As Integer = 0 '擊中目標數
```

```
    '控制目標移動(四個目標循環出現)
    Private Sub Timer2_Tick(sender As Object, e As EventArgs) Handles Timer2.Tick
        For Each i In Me.Controls
            If i.Tag = "T" Then '每個目標
                i.Left += 5 '右移五點
                If i.left > Me.ClientSize.Width Then '超出表單
                    i.Left = -i.Width - Rnd() * 20 '加入20點的不確定性
                    i.Visible = True '目標可見
                    NT -= 1 '目標總數減一
                    If NT < 0 Then '目標跑完了
                        i.Visible = False '隱藏
                        NT = 0 '剩餘目標0
                    End If
                    Tn.Text = NT '顯示剩餘目標數
                End If
            End If
        Next
    End Sub

    '啟動
    Private Sub StB_Click(sender As Object, e As EventArgs) Handles StB.Click
        NT = 25 : Tn.Text = NT '重設目標數
        H = 0 : Hn.Text = H '重設擊中數
        Timer1.Start() '啟動砲彈控制
        Timer2.Start() '啟動目標控制
    End Sub

End Class
```

課後閱讀

一、談談變數與程式碼架構

我們先看看程式碼的階層架構，最外層是 Public Class Form1 與 End Class，接著在此框架內會隨程式設計過程，建立很多的 Sub 或 Function 副程序。每一個副程序的內部依據程式需要還可能有一些程式區塊，譬如 If…EndIf、Select Case…End Select、For…Next 等等。

作者常用的比喻是：Class 有如學校，有前門(Public Class Form1)，也有後門(End Class)；Sub 或 Function 等則有如學校建築，如教學大樓、行政大樓等等，也是各有前後門(Sub 與 End Sub 等)；進到建築物之後還有教室與辦公室，那就是 If…EndIf 等等再次級的程式區塊了！它們依然必須有前後門，譬如有 For 開始就必須有 Next 結束。了解與熟悉這些程式架構之後，寫程式出錯的機會就大為減少了！

程式碼有階層架構，變數也是！一般來說我們宣告的變數只能在同階層，及以下的階層使用。在此階層(或者說程式單元)以上(或說以外)就不具意義。譬如在某個 Sub A 副程序內宣告的變數「C」與另一個 Sub B 副程序內的變數 C 其實是不同的變數，猶如不同班級同座號的人並不相同一樣！甚至如果你在一個 If…EndIf 程式區塊(或者 For…Next)內宣告一個變數，這個變數也只能在該區塊內有效！出了這個程式區塊，即使還在同一個 Sub 之內，這個變數都是沒有意義的。

那麼如果我希望一個變數可以通用於多個副程序怎麼辦？就是必須將它宣告在所有副程序之外，也就是它的地位須與副程序平行，是同一個「階層」的東西！我們就稱它為全域變數了。譬如本單元的剩餘目標數(NT)與擊中目標數(H)就是如此，我們在多個副程序中都必須用到它們，此時就不能將變數宣告在個別的副程序內。

但是宣告全域變數的位置不限定要在程式的哪個地方(譬如最前面)，如上面的比喻所說 Sub 是學校建築物，Sub 之間的空間就是建築物之間的走道了，任何出現在這些走道邊的布告都是效果相同的全域變數。也因此你可以將變數宣告在主要使用它們的 Sub 前面，這樣比較容易閱讀。就像某大科學家將來校演講，布告可以貼在校園的任何地方，但是不宜只貼在某教室內，而且貼在理工學院門口會比貼在中文系前面好！

還有一個很容易忽略的概念是，即使在同一個副程序內的變數，如果你執行此 Sub 結束，**離開之後再度回到此副程序，原來的變數也都是會全部重設的**！如果你堅持某個變數應該在此副程序內一直延續被記憶著，宣告時就不能用 Dim 而必須改用 **Static** Q As Integer 這樣的語法。Static 原意是靜態的，也就是不變動的意思。

二、屬性 Enabled 的意義與用法

幾乎每一個工具箱物件都有 Enabled 這個屬性，我們已經用過幾次的是 Timer 物件的 Enabled 屬性，True 就是開，False 就是關。Enabled 字面上的意義是「可用」與否，那怎麼樣是可用？怎樣是不可用呢？簡單說就是可不可以回應「**事件**」！一旦物件被設

定為 Enabled=False 之後，所有該物件的**事件副程序**都會失效！譬如 Timer1.Tick 事件。又譬如你先為一個 Button1 寫了 Click 副程序，稍後卻將它設為 Enabled=False，那麼畫面上會看到 Button1 顏色變得灰暗，滑鼠點下時 Click 事件程式也不會動作了。

這種設計在程式設計中非常重要，闢如一個文書或影像軟體一定要在已經開啟檔案之後才有可能「儲存」或「關閉」檔案。「儲存」與「關閉」檔案的程式碼當然必須要在設計時就先寫好，但是不能一開始就啟用，否則在沒有開啟檔案的情況下要關閉或儲存根本不存在的東西，程式就會當掉了！這時設計者就可以使用 Enabled 屬性來控制何時該副程序可用或不可用。

當然我們可以辯說：使用者自己明明知道沒有開檔，卻硬要去關檔是他自己太笨！不是我的錯啊？但是如果真的這樣設計出來的軟體，連你在內任何人都會很討厭的！就算我不笨也有可能按錯嘛！你就不能在設計時幫忙防範一下哦？讓按鍵根本不能按，就不會按錯了嘛？好的程式這種小地方還是必須妥善設計的！所以你打開功能表時常常會看到很多灰灰的選項(如下圖)，這就是設計者費心讓你不會按錯的苦心啦！

但是其實多數的程式設計師都不喜歡寫這一部份的程式，認為只要使用者聰明一點這些工作根本就不需要嘛！因此產生了一個很刻薄的形容詞，稱這些工作為「**防呆**」機制！你想他們是罵誰呆呢？哈哈！