

```
Form1 - 1
'***注意單引號後面是程式註解說明用，在真正考試寫程式是不必寫的
'本表單名稱是Form1都採用簡易寫法，所以要先在<已打次數>和<打中次數>的Caption填上0，在<剩餘時間>的Caption填上5
```

```
Private Sub Form_Load() '程式一開始執行時，會先執行的事件，用來做變數初值設定
Randomize '因為有使用到<Rnd>這函數，所以要加這行，使其亂數完全沒有規律性
End Sub
```

```
Private Sub 地鼠計時器_Timer() '記得在Interval這個屬性上填300，代表每0.3秒跳動一次，固定時間週期內會來執行一遍的事件
For i = 0 To 8 '有9隻地鼠
    地鼠(i).Visible = False '先把9隻地鼠全部消失
Next '因為亂數函數只能產生0.0000000-0.9999999之間的數
抽到的號碼 = Int(Rnd * 9) '有9隻地鼠，所以亂數函數乘9，不要有小數所以取Int
地鼠(抽到的號碼).Visible = True '抽中的那隻老鼠，就讓他出現
End Sub
```

```
Private Sub Form_Click() '玩者用滑鼠點表單
已打次數 = 已打次數 + 1 '點中表單，代表沒打中地鼠，所以只要加<已打次數>就可以了
打中率 = Round((打中次數 / 已打次數) * 100, 2) & "%" '只要<已打次數>有變化就得重算打中率, 乘100是代表打中率是百分之多少，Round(??, 2), 其中2代表取小數2位的精確度。
End Sub
```

```
Private Sub 地鼠_Click(Index As Integer) '當圖片被打到，就代表打中地鼠
已打次數 = 已打次數 + 1 '打中，沒打中，<已打次數>當然都得加一
打中次數 = 打中次數 + 1 '即然打中了，當然<打中次數>也要加一
打中率 = Round((打中次數 / 已打次數) * 100, 2) & "%" '只要<已打次數>有變化就得重算打中率, 乘100是代表打中率是百分之多少，Round(??, 2), 其中2代表取小數2位的精確度。
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer) '當按了表單右上角X時會發生的事件'考試時不必寫此段程式
Form4.Show '因為是多重表單，所以在本表單結束前要啟動選項表單。考試時不必寫
End Sub '考試時此段不必寫
```

```
Private Sub 倒數計時器_Timer() '記得在Interval這個屬性上填1000，代表每1秒執行本程式一次'固定時間週期內會來執行一遍的事件
剩餘時間 = 剩餘時間 - 1 '每秒會執行一次，把剩下時間減一
If 剩餘時間 <= 0 Then '每次減一後馬上檢查是否到達零了
    回答 = MsgBox("時間結束，是否再玩一次", vbYesNo + vbQuestion) '使用msgbox函數呈現說明，並詢問玩者是否再玩一次
    If 回答 = vbYes Then '比對玩者選的鍵，注意是<vbYes>不可只寫Yes
        剩餘時間 = 5 '如果玩者選<是>的鍵，就把所有狀態恢復原始
        已打次數 = 0 '所以改<剩餘時間><已打次數><打中次數>這幾項
        打中次數 = 0
        打中率 = ""
    Else
        MsgBox "Bye Bye..." '如果不是選擇<是>，那就是不玩了，顯示再見的訊息
        End '把程式結束。
    End If
End If
End Sub
```

′***注意單引號後面是程式註解說明用，在真正考試寫程式是不必寫的

```
Dim 鴨子移動距離 As Integer
Dim 烏鴉移動距離 As Integer
Dim 飛彈已發射 As Boolean
```

′宣告鴨子移動用的變數
′宣告烏鴉移動用的變數
′宣告布林來記住飛彈<是, 否>正往上飛

```
Private Sub Form_Load()
    鴨子移動距離 = 300
    烏鴉移動距離 = 300
    飛彈已發射 = False
End Sub
```

′程式一開始執行時，會先執行的事件，用來做變數初值設定
′先決定鴨子每次移動的距離，此值越大，移動越快
′先決定烏鴉每次移動的距離，此值越大，移動越快

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If 飛彈已發射 = False Then
    飛彈.Left = X - 飛彈.Width / 2
End If
End Sub
```

′當滑鼠在表單移動，就檢查飛彈
′如果飛彈沒發射，就要跟著滑鼠左右移動
′沒射出去的飛彈的左界保持跟滑鼠的X值相關，而飛彈上界不要跟，這樣飛彈就會左右受控，上下不變。
′這樣看起來，就像拿飛彈在瞄準鴨子

```
Private Sub Form_Click()
If 飛彈已發射 = False Then
    飛彈已發射 = True
    發射數量 = 發射數量 + 1
    狀況 = "已發射...."
End If
End Sub
```

′表單被點一下，代表玩的人要發射一個飛彈
′如果飛彈還未射出
′就讓那個飛彈的狀態設為真，這樣飛彈就發射出去了
′每發射了一個飛彈，當然發射數量得加一

```
Private Sub 鴨子計時器_Timer()
    鴨子.Left = 鴨子.Left + 鴨子移動距離
    If 鴨子.Left + 鴨子.Width > Me.Width Then
        鴨子移動距離 = -300
    End If
    If 鴨子.Left < 0 Then
        鴨子移動距離 = 300
    End If
End Sub
```

′固定時間週期內會來執行一遍的事件
′鴨子每次往一個方向移動固定距離
′檢查是否到達表單右界的外面
′如果是，代表已加太多，把移動距離符號相反，改成負的
′檢查是否到達表單左界的左邊
′如果是，代表已減太多(加太多負?)，把移動距離符號相反，改成正的

```
Private Sub 烏鴉計時器_Timer()
    烏鴉.Left = 烏鴉.Left + 烏鴉移動距離
    If 烏鴉.Left + 烏鴉.Width > Me.Width Then
        烏鴉移動距離 = -300
    End If
    If 烏鴉.Left < 0 Then
        烏鴉移動距離 = 300
    End If
End Sub
```

′固定時間週期內會來執行一遍的事件
′烏鴉每次往一個方向移動固定距離
′檢查是否到達表單右界的外面, Me是代表本表單
′如果是，代表已加太多，把移動距離符號相反，改成負
′檢查是否到達表單左界的左邊
′如果是，代表已減太多(加太多負?)，把移動距離符號相反，改成正的

```
Private Sub 飛彈計時器_Timer()
If 飛彈已發射 = True Then
    飛彈.Top = 飛彈.Top - 150
    X = 飛彈.Left + 飛彈.Width / 2
    Y = 飛彈.Top + 200
    If Y < 頂線.Y1 Then
        飛彈.Top = 底線.Y1 - 飛彈.Height - 60
        飛彈已發射 = False
        狀況 = "損失飛彈一枚"
    End If
End If
```

′固定時間週期內會來執行一遍的事件
′如果飛彈已發射，才須往上移動及比較，如果不是在這裡不必管它
′每次往上移150點，可放此值，改變飛彈上昇的整度
′算出目前這個飛彈的抵火位置
′把飛彈化成一個X, Y作標，使圖形重疊的比對變得容易
′如果飛彈上界小於頂線，代表飛到表單上面去了(注意電腦用第四象限)
′把該個飛彈收回到底線上方一點，準備下次發射。
′就把該枚飛彈的發射狀態設為假，使其不再飛行

Form2 - 2

```
If X > 鴨子.Left And X < 鴨子.Left + 鴨子.Width And Y > 鴨子.Top And Y < 鴨子.Top + 鴨子.Height Then    '如果X在鴨子的左右界之間
    飛彈.Top = 底線.Y1 - 飛彈.Height - 60    ''且Y在鴨子的上下界之間, 收回飛彈。
    飛彈已發射 = False    '打中數量加一，並如上面方式
    狀況 = "打中目標"
    命中數量 = 命中數量 + 1    '代表圖形重疊，飛彈打中鴨子
End If
If X > 烏鴉.Left And X < 烏鴉.Left + 烏鴉.Width And Y > 烏鴉.Top And Y < 烏鴉.Top + 烏鴉.Height Then    '如果X在鴨子的左右界之間
    飛彈.Top = 底線.Y1 - 飛彈.Height - 60    ''且Y在烏鴉的上下界之間, 並如上面方式收回飛彈。
    飛彈已發射 = False    '代表圖形重疊，飛彈打中烏鴉
    狀況 = "打錯目標"
End If
End If
End Sub
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer) '當按了表單右上角X時會發生的事件
Form4.Show    '因為是多重表單，所以在本表單結束前要啟動選項表單。
End Sub    '考試時此段不必寫
```

```
Form3 = 1  
`***注意單引號後面是程式註解說明用，在真正考試寫程式是不必寫的  
`記得把表單的<KeyPreview>設為<True>
```

```
Dim 格點大小 As Integer    `宣告這個變數，使格點遊戲所有大小及移動距離相同，都用這個值
```

```
Private Sub Form_KeyPress(KeyAscii As Integer)    `有人按下按鍵，要使這事件發生，記得把表單的KeyPreview設為True  
Select Case Chr(KeyAscii)    `按鍵按下時會傳入所按的鍵的內碼，就以傳來的內碼當判斷條件  
Case "2":    `因為傳來是內碼，經CHR()函數換成原入的字  
    蛇的方向 = "下"    `如果相同，代表剛才玩者按了2這個鍵，就把方向設為<下>  
Case "4":    `同理判斷其他想用的鍵內碼，有相同時  
    蛇的方向 = "左"    `就設成相對的字，以便等下蛇移動時的方向依據。  
Case "6":  
    蛇的方向 = "右"  
Case "8":  
    蛇的方向 = "上"  
Case "5":  
    蛇的方向 = "停"  
End Select  
End Sub
```

```
Private Sub Form_Load()  
Randomize    `程式一開始執行時，會先執行的事件，用來做變數初值設定  
格點大小 = 蛇(0).Width    `因為有使用到<Rnd>這函數，所以要加這行，使其亂數完全沒有規律性  
蛇(0).Height = 格點大小    `為使所有蛇及青蛙等大，所以先把<蛇(0).Width>存在宣告過的<格點大小>這變數裡  
    `把蛇頭的高度設成跟寬度一樣，這樣蛇頭就會是正方形
```

```
青蛙.Width = 格點大小    `把青蛙也設成等大  
青蛙.Height = 格點大小
```

```
蛇(0).Left = 取X值    `利用已寫好的格點化函數，來使一開始<蛇(0)>隨意在表單中找一個新位置  
蛇(0).Top = 取Y值
```

```
青蛙.Left = 取X值    `同樣的，把青蛙也隨機放在表單的某個格點上  
青蛙.Top = 取Y值  
End Sub
```

```
Private Function 取X值()  
最大倍數 = Int(Me.Width / 格點大小)    `自寫的副程式，用來在表單中隨機找到一個格點大小整倍數的X位置  
取X值 = Int(Rnd * 最大倍數) * 格點大小    `拿表單寬度除格點大小，算出目前橫的可放幾隻蛇  
    `依所能放的倍數取Rnd化成整數再乘<格點大小>，這樣保證取到的位置是蛇寬的倍數  
End Function
```

```
Private Function 取Y值()  
表單實際高度 = Me.Height - 600    `自寫的副程式，用來在表單中隨機找到一個格點大小整倍數的Y位置  
    `注意是Me.Height不是Width，因為表單有個藍色標題區，會佔用約600的高度，所需減掉，  
最大倍數 = Int(表單實際高度 / 格點大小)    `拿表單實際高度，除格點大小，算出目前縱的可放幾隻蛇  
取Y值 = Int(Rnd * 最大倍數) * 格點大小    `不過由於表單的Y軸是由藍色標題下方算起，所以要先減一個值，以免剛好放在看不到的地方  
    `依所能放的倍數取Rnd化成整數再乘<格點大小>，這樣保證取到的位置是蛇寬的倍數  
End Function
```

```
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)    `當按了表單右上角X時會發生的事件  
Form4.Show    `因為是多重表單，所以在本表單結束前要啟動選項表單。考試時此處不必寫。  
End Sub    `考試時此段不必寫
```

```
Private Sub 計時器_Timer()  
If Me.WindowState = vbMinimized Then    `固定時間週期內會來執行一遍的事件  
Exit Sub  
End If  
`下面是先把蛇頭位置記住，以便移蛇身程式使用  
蛇頭X = 蛇(0).Left    `移動蛇頭前先把目前位標記在X, Y變數裡，以便後面的蛇身能跟在蛇頭後方  
蛇頭Y = 蛇(0).Top
```

´下面是處理<蛇頭移動>的程序

```
Select Case 蛇的方向
```

```
Case "上":
```

```
蛇(0).Top = 蛇(0).Top - 格點大小
```

```
Case "下":
```

```
蛇(0).Top = 蛇(0).Top + 格點大小
```

```
Case "左":
```

```
蛇(0).Left = 蛇(0).Left - 格點大小
```

```
Case "右":
```

```
蛇(0).Left = 蛇(0).Left + 格點大小
```

```
End Select
```

´下面是處理<蛇吃到青蛙>的程序

```
If 蛇(0).Top = 青蛙.Top And 蛇(0).Left = 青蛙.Left Then
```

```
    青蛙.Left = 取X值
```

```
    青蛙.Top = 取Y值
```

```
    Load 蛇(蛇的長度)
```

```
    蛇(蛇的長度).Caption = "X"
```

```
    蛇(蛇的長度).BackColor = vbYellow
```

```
    蛇(蛇的長度).Visible = True
```

```
    計時器.Interval = 計時器.Interval * 0.8
```

```
    蛇的長度 = Val(蛇的長度) + 1
```

```
    If Val(最高紀錄) < Val(蛇的長度) Then
```

```
        最高紀錄 = 蛇的長度
```

```
    End If
```

```
End If
```

´下面是處理<蛇身移動>的程序

```
For i = 1 To 蛇的長度 - 1
```

```
    蛇身X = 蛇(i).Left
```

```
    蛇身Y = 蛇(i).Top
```

```
    蛇(i).Left = 蛇頭X
```

```
    蛇(i).Top = 蛇頭Y
```

```
    蛇頭X = 蛇身X
```

```
    蛇頭Y = 蛇身Y
```

```
Next
```

´下面是處理<蛇撞牆>程序

```
If 蛇(0).Left < 0 Or 蛇(0).Left > Me.Width Or 蛇(0).Top < 0 Or 蛇(0).Top > Me.Height - 600 Then
```

```
    MsgBox "撞到了"
```

```
    蛇(0).Left = 取X值
```

```
    蛇(0).Top = 取Y值
```

```
    蛇的方向 = "停"
```

```
    For i = 1 To 蛇的長度 - 1
```

```
        Unload 蛇(i)
```

```
    Next
```

```
    蛇的長度 = 1
```

```
    計時器.Interval = 300
```

```
End If
```

´下面是處理<咬蛇自盡>程序

```
If 蛇的方向 <> "停" Then
```

```
    有咬到自己 = False
```

```
    For i = 1 To 蛇的長度 - 1
```

```
        If 蛇(0).Left = 蛇(i).Left And 蛇(0).Top = 蛇(i).Top Then
```

```
            有咬到自己 = True
```

```
        End If
```

´依據<蛇的方向>目前呈現的字來決定蛇要往那個方向走一個<格點大小>距離

´如果是上下，則改變的是蛇頭的上界<蛇(0).Top>，每次改變一個蛇身距離

´如果是左右，則改變的是蛇頭的上界<蛇(0).Left>，每次改變一個蛇身距離

´停不必處理，因為如果是停就不必移動。

´蛇頭移動完，蛇身還沒動時先判斷有沒有吃到青蛙，以便先把蛇身加長

´如果青蛙與蛇(0)的上界跟左界相同代表蛇吃到青蛙了

´青蛙被咬到，就替青蛙換個新的位置

´複製一個蛇身

´使蛇身上沒有字

´使蛇身跟蛇頭不同色

´使新產生的這個蛇身出現

´為了增加效果，吃一隻青蛙就讓蛇加速百分之20

´並把<蛇的長度>加一，以便下次吃到青蛙時會呈現下一節

´<最高紀錄>與<>都是字串，所以一定要先用val化成數值來比較

´否則會發生<"9">比<"10">大的現象。

´可是要改變某個變數時，等號左邊的變數卻不可加上val, 只直接用變數名稱

´移動蛇身程式，讓每一節都跟著前一節走, 因為只要處蛇身所以 For 是從 1 算起的。

´要往前一位置移之前先把目前座標留下在X2, Y2

´真的把目前這蛇移到蛇頭位置

´硬把X, Y位置設為剛移走這節蛇的位置，使下一節蛇身誤以為

´他前面的那節就是蛇頭，這樣就會一節跟著一節走

´判斷是否撞牆

´如果蛇(0)的左界跑到表單的左右外面或蛇(0)的上界跑到表單上下的外面

´代表撞到了，這時有六件事要做，以恢復蛇的起始狀態。

´一. 秀個訊息 用msgbox指令，

´二. 替蛇(0)找一個新的位

´三. 因為有些蛇身已呈現，再把所有蛇身全部拿掉

´四. 把蛇的長度改成基本長度，就是1節

´五. 因為重定位，怕玩者還沒看清楚，就撞牆了，所以<蛇的方向>設為"停"。

´六. 恢復原始移動速度，以免重玩，一開始就很快

´判斷是否蛇有咬到自己，不過首先必須不在<停>的狀態中，否則按<停>就咬到了

´實際有蛇身長度的蛇頭不算，所以只判斷到<蛇的長度>減一節

´如果<蛇(0)>跟蛇的某節<蛇(J)>的左界跟上界相同就代表咬到自已了

```
Next
If 有咬到自己 = True Then
    MsgBox "好痛"
    蛇(0).Left = 取X值
    蛇(0).Top = 取Y值
    蛇的方向 = "停"
    For i = 1 To 蛇的長度 - 1
        Unload 蛇(i)
    Next
    蛇的長度 = 1
    計時器.Interval = 300
End If
End If
End Sub
```

´ 咬到自己跟前面撞到牆一樣要做六件事來重新啟動貪吃蛇

´ 因為是一節節檢查蛇身是否有跟蛇頭在同位置，有咬到自己時會把蛇身消除