

Two Timers

Using one timer to fire another timer can add a great deal of functionality to a program.

Lets create a program which displays a duck moving back and forth on the screen the way you might see in a shooting gallery.

First lets start with our duck image.

```
Public Class Form1
```

```
    Dim duck1Path As String = _  
        "C:\Documents and Settings\James\My Documents\Visual Studio 2005\Projects\DuckTimers\duck1.jpg"  
    Dim duck1 As Image = Image.FromFile(duck1Path)
```

```
End Class
```



Beginning Code

```
Public Class Form1
    Dim duck1Path As String = _
        "C:\Documents and Settings\James\My Documents\Visual

    Dim duck1 As Image = Image.FromFile(duck1Path)

    Dim x As Integer = 0
    Dim y As Integer = 0
```

Path(cut off to save space)

Image Declaration

```
Sub MoveRight ()
    picDuck.Location = New Point(x, y)
    x = x + 1
End Sub
```

MoveRight() cause picDuck to move

```
Private Sub Timer1_Tick(ByVal sender As System.Object)
    MoveRight ()
End Sub
```

Timer calls Sub MoveRight()

```
Private Sub btnStart_Click(ByVal sender As System.Object)
    Timer1.Start ()
End Sub
End Class
```

Button starts Timer

Duck Gallery

```
Sub MoveRight()  
    picDuck.Location = New Point(x, y)  
    x = x + 5  
End Sub  


---

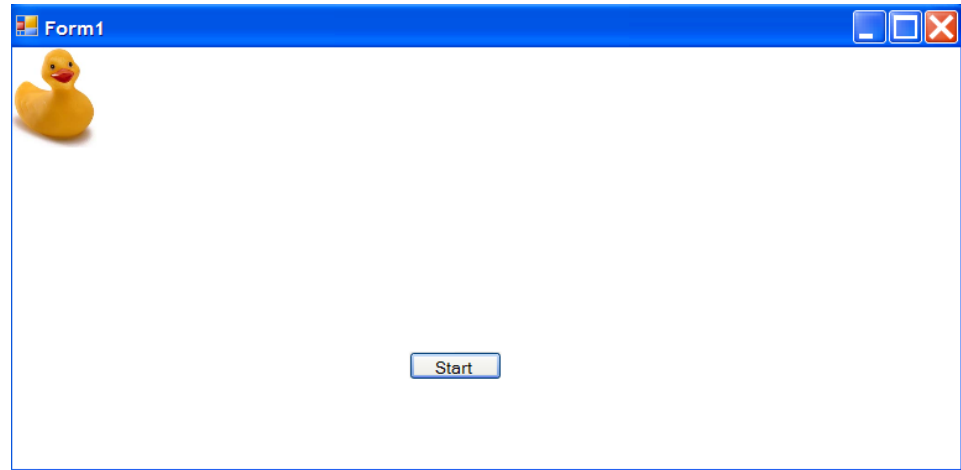
  
Sub MoveLeft()  
    picDuck.Location = New Point(x, y)  
    x = x - 5  
End Sub  


---

  
Private Sub Timer1_Tick(ByVal sender As S  
    If (x <= 700) Then  
        MoveRight()  
    Else  
        Timer1.Stop()  
        Timer2.Start()  
    End If  
  
End Sub  


---

  
Private Sub Timer2_Tick(ByVal sender As S  
    If (x >= 0) Then  
        MoveLeft()  
    Else  
        Timer2.Stop()  
        Timer1.Start()  
    End If  
End Sub
```



**Timer1 starts duck across screen by calling MoveRight().
When x reaches 700 Timer1 is shut off and Timer2 is started
which starts calling MoveLeft(). When x reaches 0, Timer2 is
shut off and Timer1 is started again.**

**The end result is that the duck keeps moving back and forth
across the screen.**

Changing Direction

```
Dim duck1Path As String = _  
"C:\Documents and Settings\James\My Documents\Visual Studio 2005\Projects\DuckTimers\duck1.jpg"  
Dim duck2Path As String = _  
"C:\Documents and Settings\James\My Documents\Visual Studio 2005\Projects\DuckTimers\duck2.jpg"  
  
Dim duck1 As Image = Image.FromFile(duck1Path)  
Dim duck2 As Image = Image.FromFile(duck2Path)
```

To make our duck movements a little more realistic, lets make the duck appear to turn around when it heads left so that it doesn't look like the duck is going backward.

To do this you can flip the image of duck1 in an image editing program.

In Fireworks the commands are Modify >>> Transform >>> Flip Horizontally

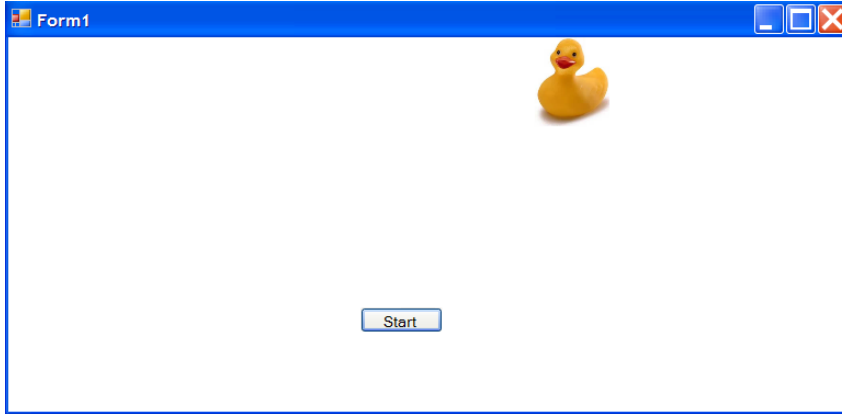


Duck1



Duck2

Switch Duck Images



```
Private Sub Timer1_Tick(ByVal sender As  
    If (x <= 700) Then  
        MoveRight ()  
    Else  
        picDuck.Image = duck2  
        Timer1.Stop ()  
        Timer2.Start ()  
    End If
```

When x reaches 700, image of duck2 is loaded into picDuck picture box.

```
End Sub  
Private Sub Timer2_Tick(ByVal sender As  
    If (x >= 0) Then  
        MoveLeft ()  
    Else  
        picDuck.Image = duck1  
        Timer2.Stop ()  
        Timer1.Start ()  
    End If  
End Sub
```

When x reaches 0, image of duck1 is re-loaded into picDuck picture box.

```
Private Sub btnStart_Click(ByVal sender  
    picDuck.Image = duck1  
    Timer1.Start ()  
End Sub
```

When Start is clicked, image of duck1 is loaded into picDuck picture box.

What Fun is a Shooting Gallery Without Shooting? Adding Weaponry

```
Sub Shoot ()  
    picBullet.Location = New Point(a, b)  
    b = b - 8  
End Sub
```

```
Private Sub Timer3_Tick(ByVal sender As Sy  
    Shoot ()  
End Sub
```

```
Private Sub btnFire_Click(ByVal sender As  
    Timer3.Start ()  
    b = 340  
    picBullet.Location = New Point(a, b)  
End Sub
```

Soon-to-be-dead
duck!

