Chapter 23

Creating Animation

- ❖ Learning how to create Animation in VB2010

Although VB2010 is a serious programming language designed for creating business and other industrial applications and not for creating animation, you can use it to create animation and other fun programs. In this Chapter, we will show you just that.

23.1 Moving an object

In this section, we will show you how to move an object by pressing a command button. You need to make use of the **Top** and **Left** properties of an object to create animation. The Top property defines the distance of the object from the top most border of the screen while the Left property defines the distance of the object from leftmost border of the screen. By adding or subtracting the distance of the object, we can create the animated effect of moving an object.

Start a new project and name it as Movable Object, or any name you wish. Now insert a Picture Box and in its Image property import a picture from your hard drive or other sources. Next, insert four command buttons and set up their properties as follows:

Control	**Property**
Button1	Name: MoveUpBtn Text : Move Up
Button2	Name: MoveDownBtn Text : Move Down
Button3	Name: MoveLeftBtn Text : Move Left
Button4	Name: MoveRightBtn Text : Move Right

Table 23.1

Now, click on the buttons and key in the following code:

```
Private Sub MoveDownBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MoveDownBtn.Click

PictureBox1.Top = PictureBox1.Top + 10

End Sub

Private Sub MoveLeftBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MoveLeftBtn.Click

PictureBox1.Left = PictureBox1.Left - 10

End Sub

Private Sub MoveRightBtn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MoveRightBtn.Click

PictureBox1.Left = PictureBox1.Left + 10

End Sub
```

Explanation:

Each time the user clicks on the Move Down button, the distance of the PictureBox increases by 10 pixels from the top border, creating a downward motion. On the other hand, each time the user clicks on the Move Up button, the distance of the PictureBox decreases by 10 pixels from the top borders, thus creating an upward motion. In addition, each time the user click on the Move Left button, the distance of the PictureBox decreases by 10 pixels from the left border, thus creating a leftward motion. Lastly, each time the user click on the Move Right button, the distance of the PictureBox increases by 10 pixels from the left border, thus creating a rightward motion.

The interface is shown in Figure 23.1

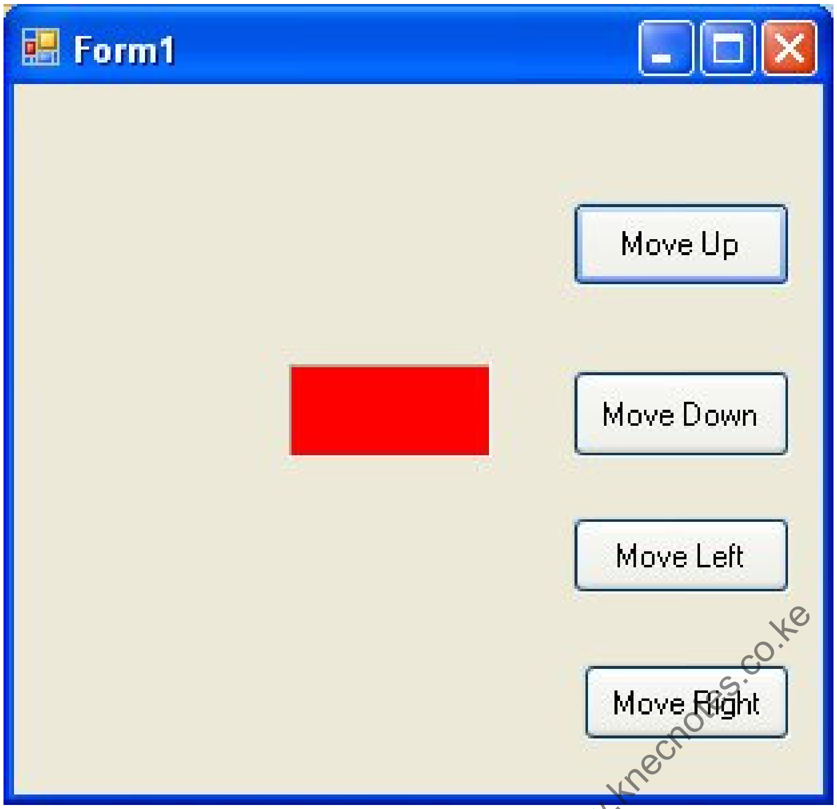


Figure 23.1

23.2 Using Timer to Create Animation

We can create continuous animation using timer without the need to manually clicking a command button. We can create left to right or top to bottom motion by writing the necessary code.

First, insert a PictureBox into the form. In the PictureBox properties window, select the image property and click to import an image file from your external sources such as your hard drive, your Pendrive or DVD. We have inserted an image of a bunch of grapes. Next, insert a Timer control into the form set its interval property to 100, which is equivalent to 0.1 second. We also set its Enabled property to be false so that the animation won't start when the form is loaded. Finally, add two command buttons to the form, name one of them as AnimateBtn and the other one as StopBtn, and change to caption to Animate and Stop respectively.

We make use of the Left property of the PictureBox to create the motion. PictureBox.Left means the distance of the PictureBox from the left border of the Form. Now click on the Timer control and type in the following code:

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer1.Tick

If PictureBox1.Left < Me.Width Then

PictureBox1.Left = PictureBox1.Left + 10

Else

PictureBox1.Left = 0

End If

End Sub
```

In the code above, Me.Width represents the width of the Form. If the distance of the PictureBox from the left is less than the width of the Form, a value of 10 is added to the distance of the PictureBox from the left border each time the Timer tick, or every 0.1 second in this example. When the distance of the PictureBox from the left border is equal to the width of the form, the distance from the left border is set to 0, which move the PictureBox object to the left border and then move right again, thus creates an oscillating motion from left to right. We need to insert a button to stop the motion. The code for the Stop button is:

```
Timer1.Enabled = False
```

To animate the PictureBox object, we click the Animate button and key in the following code:

```
Timer1.Enabled = True
```

The Interface of the Animation program is show in Figure 23.2

When the user click on the Animate button, the image will move from left to right continuously until the user press the Stop button.

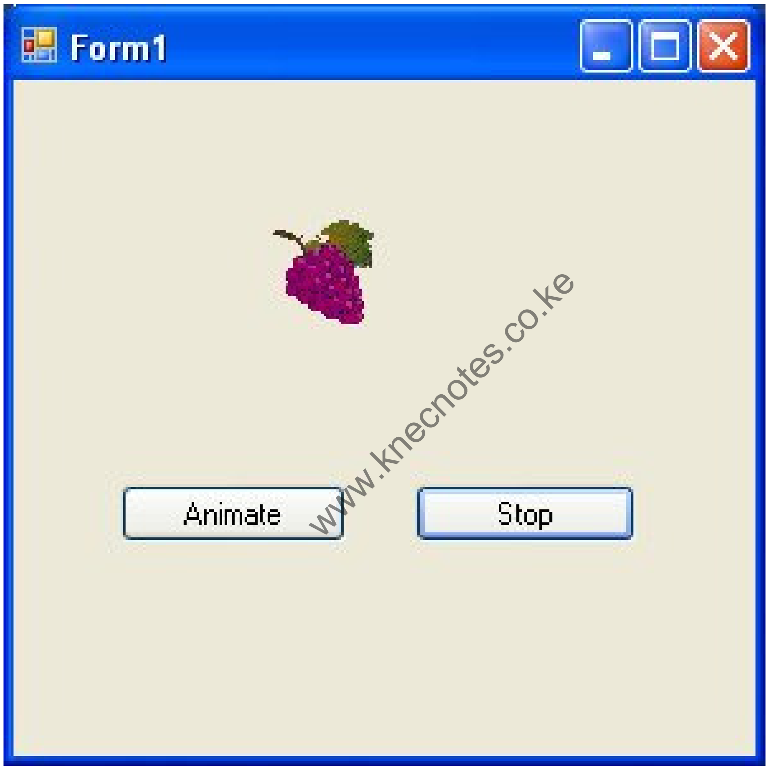


Figure 23.2

23.3 Creating a Simple Lucky Seven Slot Machine

A lucky seven slot machine allows a player to win the jackpot if three rows of 7 appear when the spin stopped. In this example, we can create a simple version of the lucky seven slot machine with only one row of numbers.

In this program, we insert three labels, a button and a timer into the form. Set the properties of the controls as follows:

Control	**Property**
Label1, Label2 and Label3	Backcolor: Navy Blue Forecolor: Yellow Font: Microsoft Sans Serif, Size 14, Bold
Button1	Name : BtnSpin Text : SPIN Font: Microsoft Sans Serif, Size 14, Bold
Button1	Name : BtnSpin Text : EXIT Font: Microsoft Sans Serif, Size 14, Bold

We need to write code that generates nine numbers that changes randomly according to the Timer's interval and displays them on the three labels. The process stops after a certain period. The program will then examine the final three numbers. If the three numbers are found to be seven, a message will pop up to tell the player that he has won the jackpot, otherwise the message will display the text " No luck, try again".

To generate random numbers, we use the Rnd() function. The syntax to generate nine integers from 1 to 9 is shown below:

```
Int(1 + Rnd( ) * 9)
```

You need to declare three variables n1, n2, and n3 as integers so that the randomly generated numbers will be assigned to them and to be displayed on three labels. We also declare a variable m to control the duration of the spin. The value of m increases by 10 after each timer's interval and the process stops after the value of m exceeds 1000.

Enter the code as shown below:

```
Public Class Form1
    Dim m, n1, n2, n3 As Integer

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Timer1.Tick
        m = m + 10
        If m < 1000 Then
            n1 = Int(1 + Rnd() * 9)
            Label1.Text = n1
            n2 = Int(1 + Rnd() * 9)
            Label2.Text = n2
            n3 = Int(1 + Rnd() * 9)
            Label3.Text = n3
        Else
            Timer1.Enabled = False
            m = 0
            If n1 = 7 And n2 = 7 And n3 = 7 Then
                MsgBox("You strike Jackpot and won $1,000,000")
            Else
                MsgBox("No luck, try again")

            End If
        End If

    End Sub


    Private Sub BtnSpin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnSpin.Click
        Timer1.Enabled = True
    End Sub

    Private Sub BtnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnExit.Click
        End
    End Sub
End Class
```

The runtime interface is shown in Figure 23.3

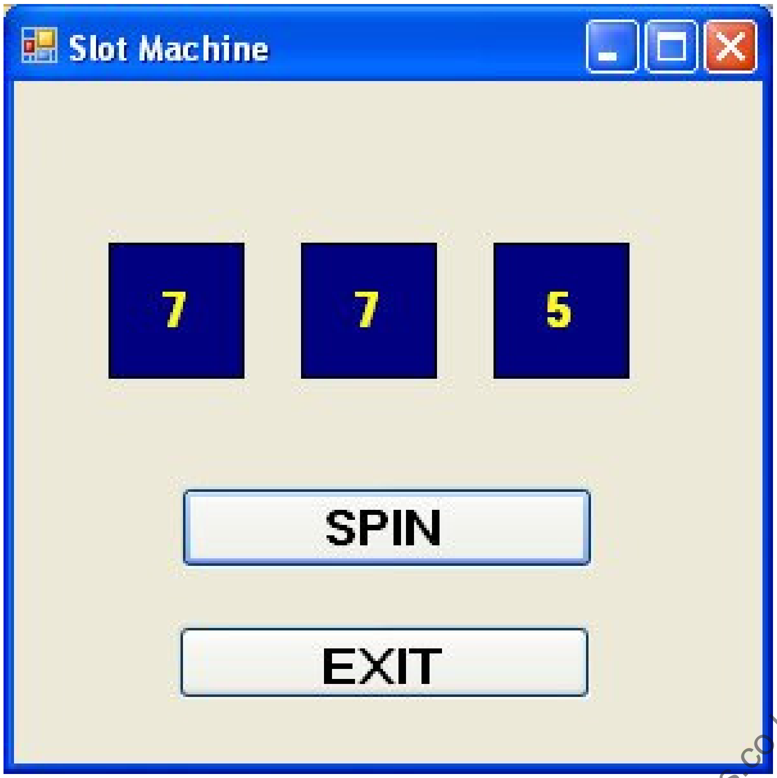


Figure 23.3

23.4 Creating a Graphical Slot Machine

We will modify the previous slot machine make it into a graphical one. Instead of displaying random numbers, this slot machine shows random images. To show the images randomly, we use the same logic as the previous example, but instead of generating random numbers, the program loads images randomly. The syntax to load image is:

```
Image.FromFile("C:\ My Documents\Image\image. jpg")
```

In this example, we insert three picture boxes for loading three different images. This program is for demonstration only so we use only three images so that the code does not look too complicated. We use the Select Case...End Select control structure to load images according to the values of the randomly generated numbers. The procedure to load the images randomly is:

```
n1 = Int(1 + Rnd( )* 3)

n2 = Int(1 + Rnd( )* 3)
```

```
n3 = Int(1 + Rnd( )* 3)

    Select Case n1
    Case 1
    PictureBox1.Image = Image.FromFile("C:\ MyDuments\Image\image1. jpg")
     Case 2
     PictureBox2.Image = Image.FromFile("C:\ MyDuments\Image\image2. jpg")
     Case 3
     PictureBox3.Image = Image.FromFile("C:\ MyDuments\Image\image3 jpg")

    End Select
```

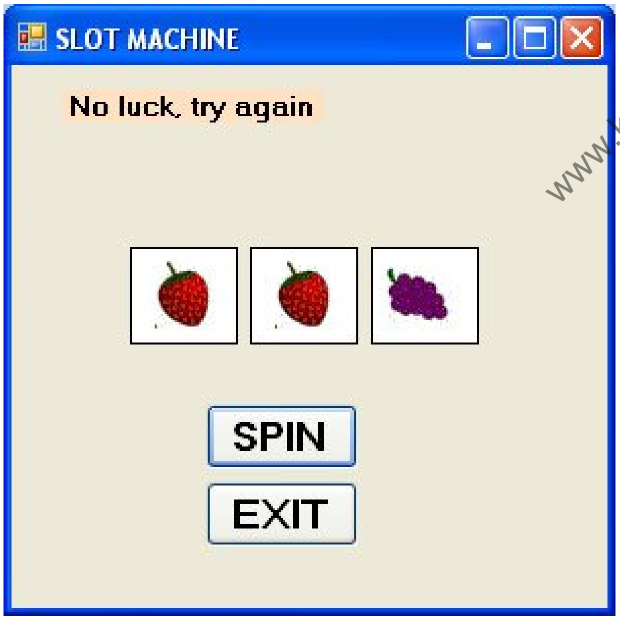
We need to insert a timer to create the animation effect. In the properties window, set the property enabled to be false

The full code looks like this:

```
Public Class Form1

 Dim m, n1, n2, n3 As Integer


Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick

 m = m + 10
 If m < 1000 Then

     n1 = Int(1 + Rnd() * 3)

     n2 = Int(1 + Rnd() * 3)

     n3 = Int(1 + Rnd() * 3)

     Select Case n1

       Case 1
       PictureBox1.Image = Image.FromFile("C:\ MyDuments\Image\image1. jpg")
        Case 2
       PictureBox1.Image = Image.FromFile("C:\ MyDuments\Image\image2. jpg")
 Case 3
  PictureBox1.Image = Image.FromFile("C:\ MyDuments\Image\image3. jpg")

End Select

  Select Case n2

 Case 1
 PictureBox2.Image = Image.FromFile("C:\ MyDuments\Image\image1. jpg")
 Case 2
 PictureBox2.Image = Image.FromFile("C:\ MyDuments\Image\image2. jpg")
 Case 3
  PictureBox3.Image = Image.FromFile("C:\ MyDuments\Image\image3. jpg")

End Select

Select Case n3

  Case 1
  PictureBox3.Image = Image.FromFile("C:\ MyDuments\Image\image1. jpg")
   Case 2
   PictureBox3.Image = Image.FromFile("C:\ MyDuments\Image\image2. jpg")
   Case 3
   PictureBox3.Image = Image.FromFile("C:\ MyDuments\Image\image3. jpg")

 End Select


  Else
     Timer1.Enabled = False
     m = 0
     If n1 = n2 And n1 = n3 Then

        Label1.Text = "Jackpot! You won $1,000,000"
     Else
        Label1.Text = "No luck, try again"

     End If

  End If

End Sub
```

```
   Private Sub BtnSpin_Click(ByVal sender As System.Object, ByVal e As
   System.EventArgs) Handles BtnSpin.Click
      Label1.Text = ""
      Timer1.Enabled = True
   End Sub
   Private Sub BtnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnExit.Click
      End
   End Sub
End Class
```

The interface is shown in Figure 23.4



No luck, try again

Figure 23.4

Summary

In this chapter, you learned how to create animation.

- In section 23.1, you learned how to write code that move an object.
- In section 23.2, you learned how to use Timer to create animation.
- In section 23.3, you learned how to create a lucky seven slot machine.
- In section 23.4, you learned how to create a graphical slot machine.