

HANDLING KEYBOARD EVENTS

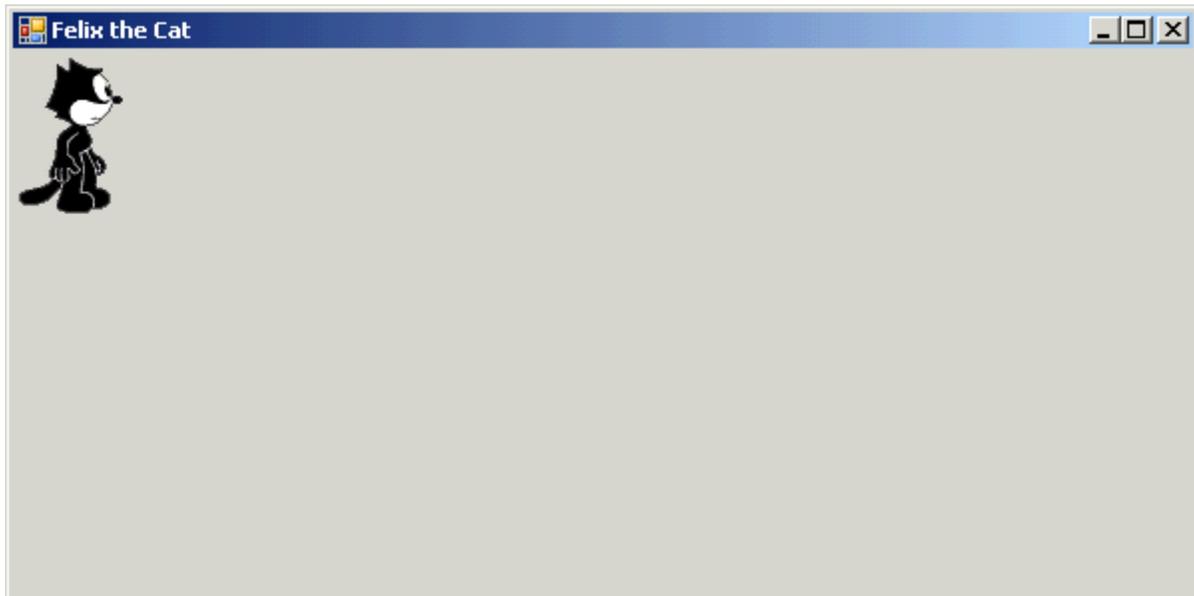
Key events occur when keyboard keys are pressed and released. There are three key events: **KeyPress**, **KeyUp** and **KeyDown**.

The **KeyPress** event occurs when the user presses a key that represents an ASCII character. The specific key can be determined with the property **KeyChar** of the event handler's **KeyPressEventArgs** argument. ASCII is a 128-character set of alphanumeric symbols.

The **KeyDown** event occurs when a key is initially pressed and, unlike the **KeyPress** event, it also indicates whether modifier keys (e.g. SHIFT, ALT, and CTRL) were pressed when a key event occurred.

The **KeyUp** event occurs when a key is released.

In the following example, we are going to create a program that allows the user to move a character up, down and across the screen using the arrow keys.



The first thing we will need to do is declare some global variables:

```
Imports System.IO

Public Class Felix

    'Declare global variables
    Public cat(1) As Image
    Public east As Boolean
    Public xPos, yPos As Integer

End Class
```

We will use an array called **cat** that will store two images: one image of Felix facing east and another of Felix facing west. We will also use a Boolean variable called **east** that is going to keep track of whether Felix is facing east or west so that we know which image needs to be displayed. Finally, we set the x- and y-position of the cat by setting its location to a new point on the form.

Next, we will need to fill the array with the two images and set the default values for our variables:

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

```
    'Initialize array of images
```

```
    cat(0) = Image.FromFile(Directory.GetCurrentDirectory &
        "\images\catEast.png")
```

```
    cat(1) = Image.FromFile(Directory.GetCurrentDirectory &
        "\images\catWest.png")
```

```
    'Set Felix facing east
```

```
    east = True
```

```
    'Set the x- and y-position of the cat
```

```
    picFelix.Location = New Point(0, 0)
```

```
    'Output the image
```

```
    picFelix.Image = cat(0)
```

```
End Sub
```

Now it's time to program the arrow keys. Since we are using the arrow keys and arrow keys are not represented as an ASCII character, we will need to program the arrow keys in the **KeyDown** procedure. The first key we will program is the right key so that when the user clicks the right arrow key, Felix is going to move east 10 pixels if he's facing east. If Felix is not facing east, then we are simply going to change the image to Felix facing west.

```
Private Sub Felix_KeyDown(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
```

```
    'User clicks the right key
```

```
    If e.KeyCode = Keys.Right Then
```

```
        If east = True Then
```

```
            picFelix.Location = New Point(picFelix.Location.X + 10,
                picFelix.Location.Y)
```

```
        Else
```

```
            east = False
```

```
            picFelix.Image = cat(0)
```

```
        End If
```

```
    End If
```

When the user clicks the left arrow key, we need Felix to move west 10 pixels if he's facing west. If Felix is not facing west, then we need to change the image to Felix facing west.

```
'User clicks the left key
If e.KeyCode = Keys.Left Then
    If east = True Then
        east = False
        picFelix.Image = cat(1)
    Else
        picFelix.Location = New Point(picFelix.Location.X - 10,
            picFelix.Location.Y)
    End If
End If
```

When the user clicks the up arrow key, we need Felix to move up 10 pixels.

```
'User clicks the up key
If e.KeyCode = Keys.Up Then
    picFelix.Location = New Point(picFelix.Location.X,
        picFelix.Location.Y - 10)
End If
```

When the user clicks the down arrow key, Felix needs to move down 10 pixels.

```
'User clicks the down key
If e.KeyCode = Keys.Down Then
    picFelix.Location = New Point(picFelix.Location.X,
        picFelix.Location.Y + 10)
End If
```

Run the program and hit the arrow keys to make sure that Felix is responding the way he should. Do you notice a problem? What happens when you keep hitting any one of the arrow buttons? Felix inevitably disappears off the screen. We need to fix this problem. Let's start with the right arrow button. Although we want Felix to move across the screen when the user hits the right arrow key, we only want Felix to go as far as the Form allows. To fix this problem we are going to need to modify our code:

```
'User clicks the right key
If e.KeyCode = Keys.Right Then
    'If Felix is facing east
    If east = True Then

        'Increase Felix's x-position by 10 pixels
        picFelix.Location = New Point(picFelix.Location.X + 10,
            picFelix.Location.Y)

        'If, after moving Felix 10 pixels, the right side of Felix
        hits the right side of the form
        If xPos + picFelix.Width >= Me.Width Then
```

```
        'Set Felix's position so that he stays against the right
        side of the form
        picFelix.Location = New Point(Me.Width - picFelix.Width,
        picFelix.Location.Y)
    End If
Else
    east = True
    picFelix.Image = cat(0)
End If
End If
```

By adding an additional condition that only adds 10 pixels to Felix's x-position only if his current position + the width of the picture box is not greater than the width of the form (minus an extra 15 pixels which accounts for the borders around the Form), Felix will never travel east past the Form.

See if you can do the same for when the user clicks the up, down and left arrow keys.