



Eventos

José Antônio da Cunha
IFRN

Eventos

- Eventos

Os eventos do Silverlight estão ligados a um teclado e alguns eventos de entrada do mouse (como MouseMove e KeyDown).

Eventos

- Tabela de Eventos (UIElement)

Evento	Descrição
KeyDown	Ocorre quando uma tecla é pressionada.
KeyUp	Ocorre quando uma tecla é liberado.
GotFocus	Ocorre quando o foco muda de elemento.
LostFocus	Ocorre quando o elemento recebe o foco.
MouseLeftButtomDown	Ocorre quando o botão esquerdo do mouse é pressionado enquanto o ponteiro do mouse é posicionado sobre o elemento.
MouseLeftButtomUp	Ocorre quando o botão do mouse é liberado.
MouseLeave	Ocorre quando o ponteiro do mouse move para fora do elemento.
MouseMove	Ocorre quando o mouse move sobre um elemento.
MouseWheel	Ocorre quando o usuário faz girar a roda do mouse sobre um elemento

Eventos

- Tabela de Eventos (FrameWorkElement)

Evento	Descrição
Loaded	Ocorre após um elemento ter sido criado e adicionado para a árvore do objeto.
SizeChanged	Ocorre após o tamanho do elemento muda.
LayoutUpdated	Ocorre após o layout do elemento muda.
BindingValidationError	Ocorre quando é remetido uma exceção.

Eventos

Exemplo do evento MouseMove

```
<Grid x:Name="LayoutRoot" Background="White"  
MouseMove="LayoutRoot_MouseMove">  
    <TextBlock x:Name="lblInfo"></TextBlock>  
</Grid>
```

```
private void LayoutRoot_MouseMove(object sender, MouseEventArgs e)  
{  
    Point pt = e.GetPosition(this);  
    lblInfo.Text = String.Format("Você está na coordenada ({0},{1}):", pt.X, pt.Y);  
}
```

Eventos

Exemplo do evento MouseWheel

```
d:DesignHeight="300" d:DesignWidth="400"  
MouseWheel="UserControl_MouseWheel">
```

```
<ScrollViewer VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Auto" >  
  <Viewbox x:Name="viewbox" Height="250" Width="350">  
    <Grid x:Name="LayoutRoot" Background="White" Height="250" Width="350">  
  </Grid>  
  </Viewbox>  
</ScrollViewer>
```

Eventos

Implementando o evento MouseWheel

```
private void UserControl_MouseWheel(object sender, MouseEventArgs e)
{
    //Delta está em unidades de 120, então dividir por 120 dá um fator de escala
    de 1,09 (120/110).
    //Em outras palavras, um ponto da roda do mouse expande ou encolhe a
    Viewbox por cerca de 9%.
    double scalingFactor = (double)e.Delta / 110;

    if (scalingFactor > 0)
    {
        //Expande
        viewbox.Width *= scalingFactor;
        viewbox.Height *= scalingFactor;
    }
    else
    {
        //Encolhe
        viewbox.Width /= -scalingFactor;
        viewbox.Height /= -scalingFactor;
    }
}
```

Eventos

Capturando o Mouse

```
<Canvas x:Name="parentCanvas" MouseLeftButtonDown="canvas_Click"  
Background="White">
```

```
</Canvas>
```

O código C# está nos slides a seguir:

Eventos

```
private bool isDragging = false;
private Point mouseOffset;

private void canvas_Click(object sender, MouseButtonEventArgs e)
{
    if (!isDragging)
    {
        Ellipse ellipse = new Ellipse();
        ellipse.Fill = new SolidColorBrush(Colors.Red);
        ellipse.Width = 50;
        ellipse.Height = 50;

        //Use a posição corrente do mouse para centralizar a elipse.
        Point point = e.GetPosition(this);
        ellipse.SetValue(Canvas.TopProperty, point.Y - ellipse.Height / 2);
        ellipse.SetValue(Canvas.LeftProperty, point.X - ellipse.Width / 2);

        //Assista os cliques no botão esquerdo
        ellipse.MouseLeftButtonDown += ellipse_MouseDown;

        //Adicionando Elipse para canvas
        parentCanvas.Children.Add(ellipse);
    }
}
```

Eventos

```
private void ellipse_MouseDown(object sender, MouseButtonEventArgs e)
{
    //Inicia modo arrastar
    isDragging = true;
    Ellipse ellipse = (Ellipse)sender;

    //Posição relativa da elipse
    mouseOffset = e.GetPosition(ellipse);

    //Muda cor da elipse
    ellipse.Fill = new SolidColorBrush(Colors.Green);

    ellipse.MouseMove += ellipse_MouseMove;
    ellipse.MouseLeftButtonUp += ellipse_MouseUp;

    //Captura o mouse
    ellipse.CaptureMouse();
}
```

Eventos

```
private void ellipse_MouseMove(object sender, MouseEventArgs e)
{
    if (isDragging)
    {
        Ellipse ellipse = (Ellipse)sender;
        //Posição relativa da elipse no canvas.
        Point point = e.GetPosition(parentCanvas);
        //Move elipse
        ellipse.SetValue(Canvas.TopProperty, point.Y - mouseOffset.Y);
        ellipse.SetValue(Canvas.LeftProperty, point.X - mouseOffset.X);
    }
}
```

Eventos

```
private void ellipse_MouseUp(object sender, MouseButtonEventArgs e)
{
    if (isDragging)
    {
        Ellipse ellipse = (Ellipse)sender;

        //Muda cor da elipse
        ellipse.Fill = new SolidColorBrush(Colors.Orange);

        ellipse.MouseMove -= ellipse_MouseMove;
        ellipse.MouseLeftButtonUp -= ellipse_MouseUp;
        ellipse.ReleaseMouseCapture();

        isDragging = false;
    }
}
```