Routed Events

Introduction

- WPF enhances the .NET event model. WPF supports the concept of events routing, that allows an event to take place in one element and be raised by another one.
- While WPF supports more than a few different types of routed events, Silverlight supports only one. Silverlight supports bubbled routed events only. Events that rise up the containment hierarchy from the deepest nested element to the containers that hold it.

Introduction

- The Silverlight events bubbling is linked to a few keyboard and mouse input events only (e.g. MouseMove and KeyDown) and the support for events bubbling exists for few low level elements only.
- Silverlight doesn't allow us to use events routing in our own custom elements.

Core Events

- The Silverlight elements inherit their events from `UIElement` and `FrameworkElement`. All Silverlight elements inherit from these two classes.

Higher Level Events

- In some cases an higher level event will replace a UIElement event. This is the case – for instance – when a user presses a button

Events Bubbling

- These are events that travel up the containment hierarchy till they are handled. The `MouseLeftButtonDown` event is one of them.

Handled Events

- The `MouseLeftButtonDown` and the `MouseLeftButtonUp` events are the only ones that Silverlight allows controls to handle, and by doing so suppressing their bubbling.
- Assigning the `Handled` property (the event object has) with the value `true` will stop its bubbling.

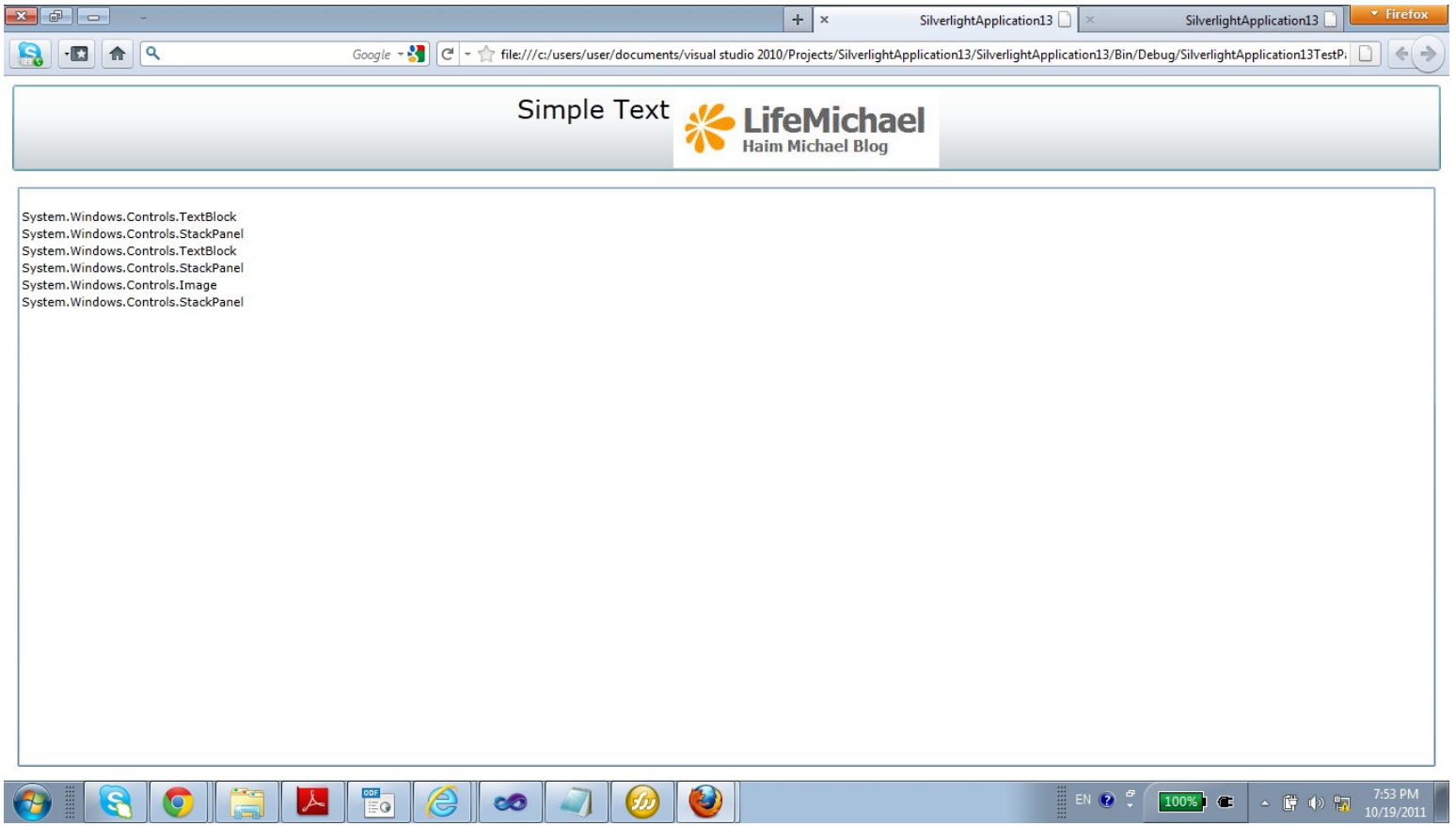
Handled Events

```
namespace SilverlightApplication13
{
    public partial class MainPage : UserControl
    {
        public MainPage()
        {
                InitializeComponent();
        }
        private void MouseClicked(
                object sender, MouseButtonEventArgs e)
        {
                msg.Text += "\n"+sender.ToString();
        }
    }
}
```

Handled Events

```
<UserControl x:Class="SilverlightApplication13.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Grid Margin="3" MouseLeftButtonDown="SomethingClicked">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"></RowDefinition>
        <RowDefinition Height="*"></RowDefinition>
    </Grid.RowDefinitions>
    <Button Margin="5" Grid.Row="0">
            <StackPanel MouseLeftButtonDown="MouseClicked"
                        Orientation="Horizontal">
                <TextBlock  Margin="3" FontSize="24"
                            MouseLeftButtonDown="MouseClicked"
                            HorizontalAlignment="Center" Text="Simple Text">

                </TextBlock>
                <Image  Source="lifemichael_logo.jpg" Stretch="None"
                        MouseLeftButtonDown="MouseClicked">
                </Image>
            </StackPanel>
        </Button>
        <TextBox x:Name="msg" Grid.Row="1" Margin="10"></TextBox>
    </Grid>
</UserControl>
```

Handled Events



Mouse Clicks

- When the user right click his mouse the Silverlight execution environment shows a pop-up menu through which the user can change the Silverlight execution environment setting.
- In order to show our own context menu and prevent the silverlight setting pop-up menu display we should assign the Handled property with `true`. Doing so we will actually suppress the silverlight setting pop-up menu.

Mouse Clicks

- In order to specify the functions we want to invoke when the user right click or left click his mouse we should refer the `MouseLeftButtonDown` and the `MouseRightButtonDown` properties.
- Other properties include the `MouseLeftButtonUp` and the `MouseRightButtonUp`.

Mouse Clicks

```
<UserControl x:Class="SilverlightApplication14.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">

    <Canvas x:Name="DrawingCanvas"
            Background="Yellow"
            MouseLeftButtonDown="CanvasLeftClick"
            MouseRightButtonDown="CanvasRightClick">
    </Canvas>

</UserControl>
```

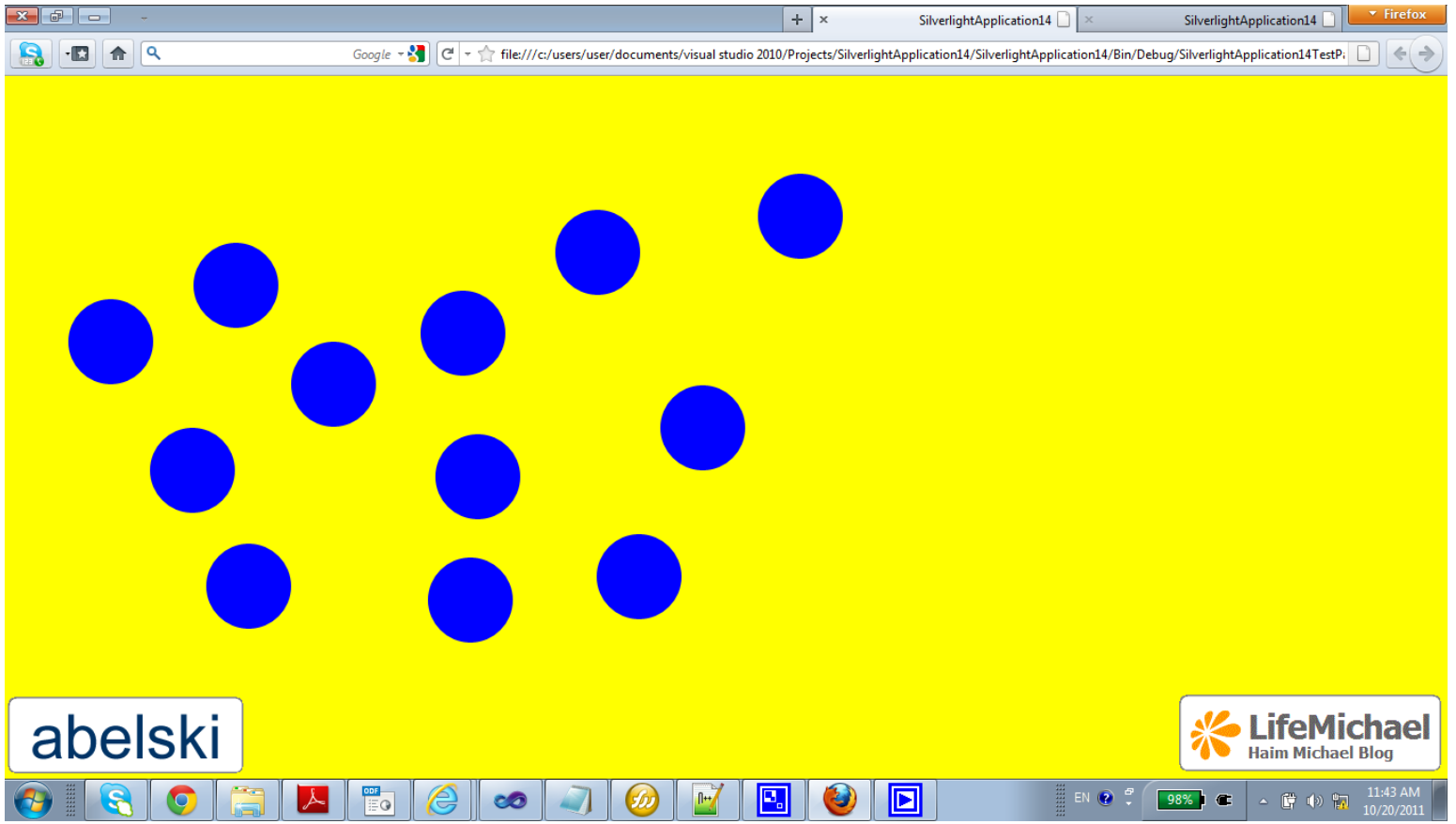
Mouse Clicks

```
namespace SilverlightApplication14
{
    public partial class MainPage : UserControl
    {
        public MainPage()
        {
            InitializeComponent();
        }
        private void CanvasLeftClick(object sender,
            MouseButtonEventArgs eventArgs)
        {
            var ellipse = new Ellipse {Fill =
                new SolidColorBrush(Colors.Blue),
                Width = 80, Height = 80};
            var point = eventArgs.GetPosition(this);
            ellipse.SetValue(Canvas.TopProperty, point.Y - ellipse.Height/2);
            ellipse.SetValue(Canvas.LeftProperty, point.X - ellipse.Width/2);
            DrawingCanvas.Children.Add(ellipse);
        }
```

Mouse Clicks

```
        private void CanvasRightClick(object sender,
            MouseButtonEventArgs eventArgs)
        {
            eventArgs.Handled = true;
        }
    }
}
```

Mouse Clicks



Mouse Movements

- When the mouse pointer moves the Silverlight platform fires the `MouseEnter`, `MouseLeave` and the `MouseMove` events.
- Each one of these three events provide us with the same information using a `MouseEventArgs` object.
- Calling `GetPosition()` method on the `MouseEventArgs` object we shall get the coordinates in relation to the element we specify.

Mouse Movements

```
<UserControl x:Class="SilverlightApplication13.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition></ColumnDefinition>
            <ColumnDefinition></ColumnDefinition>
        </Grid.ColumnDefinitions>
    <Canvas x:Name="Surface" Grid.Column="0">
        <Ellipse Fill="Blue" Width="80" Height="80"
                 MouseMove="EllipseMouseMove"
                 MouseEnter="EllipseMouseEnter"
                 MouseLeave="EllipseMouseLeave"
                 MouseWheel="EllipseMouseWheel"
                 >
        </Ellipse>
    </Canvas>

        <TextBox FontSize="20" x:Name="Msg" Grid.Column="1"></TextBox>

    </Grid>
</UserControl>
```

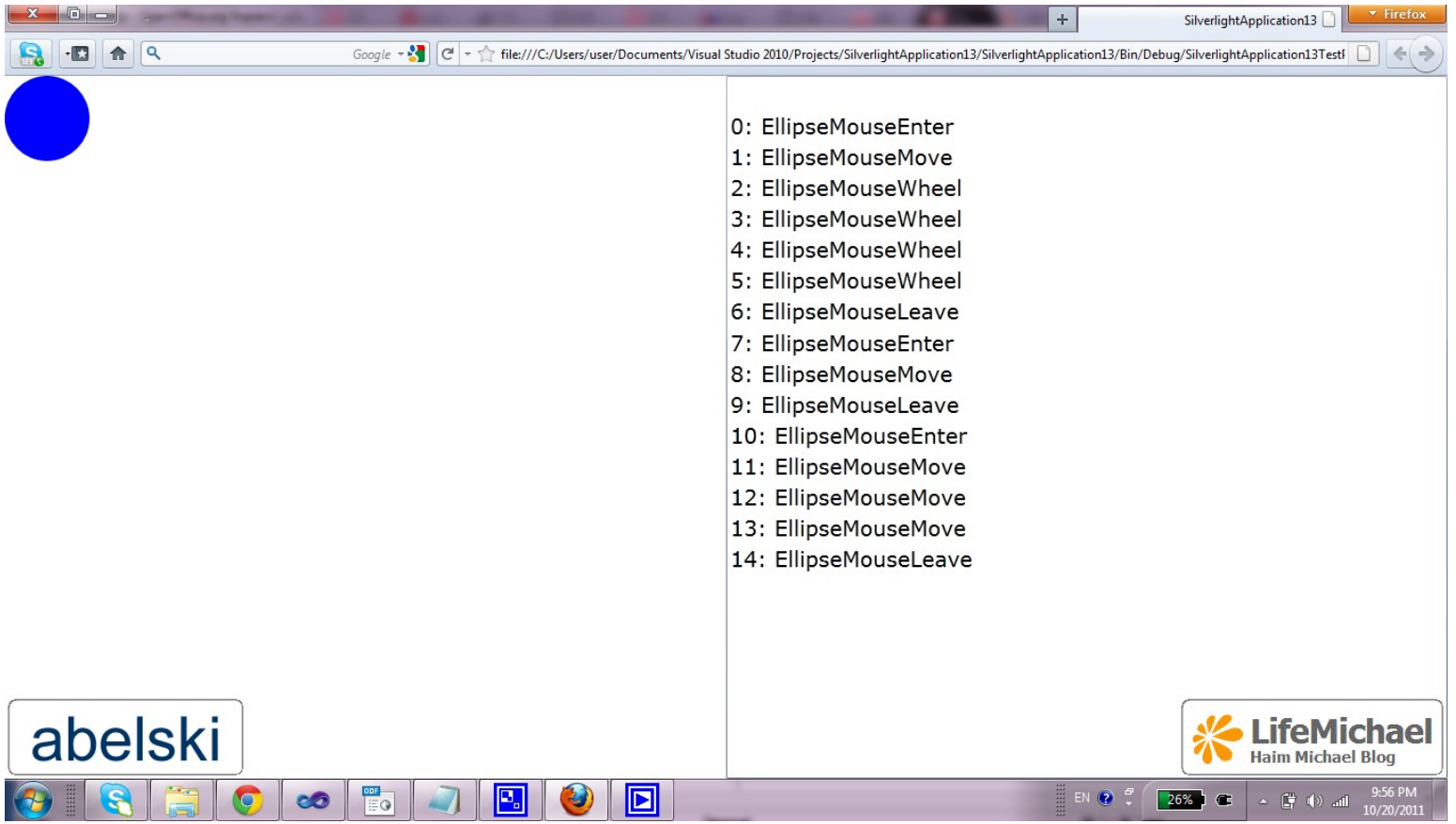
Mouse Movements

```
namespace SilverlightApplication13
{
    public partial class MainPage : UserControl
    {
        private int counter = 0;
        public MainPage()
        {
            InitializeComponent();
        }

        private void EllipseMouseEnter(object sender, MouseEventArgs e)
        {
            Msg.Text += "\n"+ counter + ": EllipseMouseEnter";
            counter++;
        }

        private void EllipseMouseMove(object sender, MouseEventArgs e)
        {
            Msg.Text += "\n"+ counter + ": EllipseMouseMove";
            counter++;
        }
```

Mouse Movements

```
        private void EllipseMouseLeave(object sender, MouseEventArgs e)
        {
            Msg.Text += "\n"+ counter + ": EllipseMouseLeave";
             counter++;
        }

        private void EllipseMouseWheel(object sender, MouseEventArgs e)
        {
            Msg.Text += "\n"+ counter + ": EllipseMouseWheel";
             counter++;
        }
    }
}
```

Mouse Movements



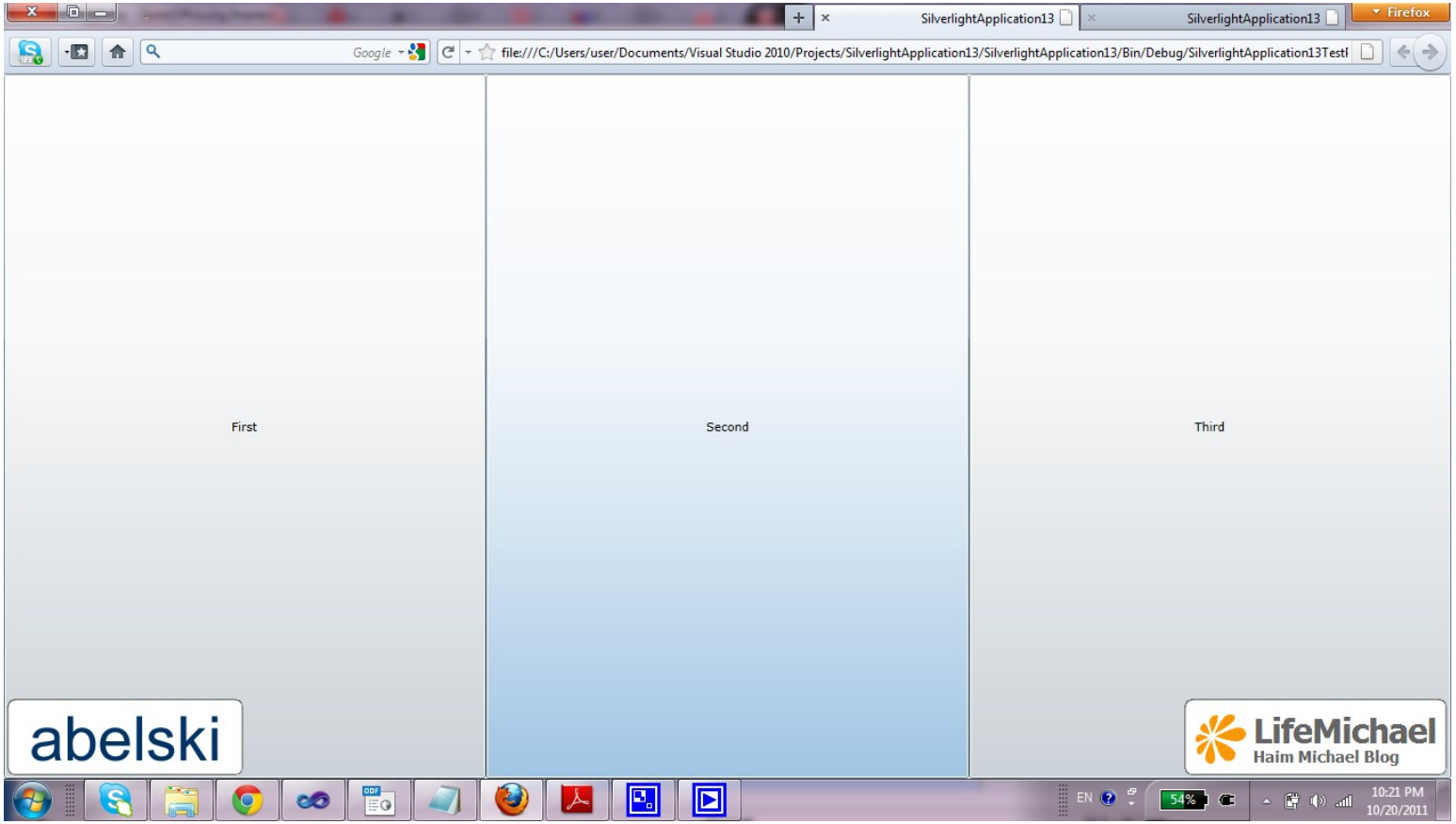
Mouse Cursors

- We can change the cursor by referring the Cursor property.
- The possible cursors are represented by objects instantiated from the `System.Windows.Input.Cursor` class.

Mouse Cursors

```
<UserControl x:Class="SilverlightApplication13.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition></ColumnDefinition>
            <ColumnDefinition></ColumnDefinition>
            <ColumnDefinition></ColumnDefinition>
        </Grid.ColumnDefinitions>
        <Button Grid.Column="0" Content="First" Cursor="Hand" />
        <Button Grid.Column="1" Content="Second" Cursor="IBeam" />
        <Button Grid.Column="2" Content="Third" Cursor="Eraser" />
    </Grid>
</UserControl>
```

Mouse Cursors



Key Events

- Silverlight elements use the `KeyDown` and `KeyUp` events in order to notify when a key was pressed.
- The `KeyEventArgs` object provide additional information such as the `Key` and the `PlatformKeyCode`.