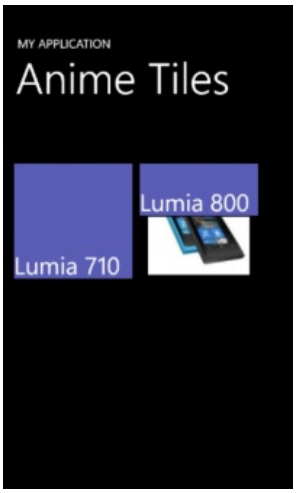
Creating custom user control in Expression Blend for Windows Phone

This article describes creating a custom user control for Windows Phone using Expression Blend.

Introduction

The article shows how to develop custom user controls and explains the communication between items on main page and the user control. The custom user control used in this application is similar to ME hub shortcut in the homescreen of windows phone. It consists of an image in the background and a rectangle(with textual description) that animates vertically on top of the image

The application is explained in Expression Blend because that is the best environment for design. The user control created complies with the Windows Phone design guidelines. The popular application SGTransport uses a similar control.



Creating the bare bones

1. Create a Blank Windowsphone application page in Expression Blend.
2. Change title of the application page title to "Anime Tiles"
3. To create a new control, goto File->New Item->User control. Give it a name, I have named it AnimBox

The User control

The ME hub tile on homescreen animates in the homescreen. The user control we will create in this application is similar to this ME Hub tile. It consists of an image in the background and a rectangle(with textual description) that animates vertically on top of the image. This kind of widgets or controls are also common in windows phone application and they also give a rich appearance. To flash designers, a user control is the equivalent of a movieclip. Step 3 gave as two new files, AnimBox.xaml and AnimBox.xaml.cs

4. In the XAML create three items inside. Image, rectangle and TextBlock. Give them names - img, rectangle and txt respectively.

5. Create a Clipping rectangle with the same dimensions of the image, so that only things visible in that region is shown. To those familiar with Flash, Clipping is equivalent of masking.

6. Next is the task of animating the rectangle and txt. Animations are done via Storyboards in Silverlight. These storyboards are textual descriptions of animation that are visible in the XAML. The storyboard name is "animate". Since textually describing creation of storyboard is not appropriate, I am pasting the XAML below.

```
<UserControl
 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"
 x:Class="AnimTile.AnimBox"
 d:DesignWidth="480" d:DesignHeight="800" Width="100" Height="100">
 <UserControl.Resources>
  <Storyboard x:Name="animate">
   <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.RenderTransform).(CompositeTransform.TranslateY)" Storyboard
    <EasingDoubleKeyFrame KeyTime="0" Value="-7"/>
    <EasingDoubleKeyFrame KeyTime="0:0:1.4" Value="14"/>
    <EasingDoubleKeyFrame KeyTime="0:0:2.9" Value="45"/>
    <EasingDoubleKeyFrame KeyTime="0:0:4.5" Value="45"/>
    <EasingDoubleKeyFrame KeyTime="0:0:5" Value="100"/>
    <EasingDoubleKeyFrame KeyTime="0:0:6" Value="-7"/>
   </DoubleAnimationUsingKeyFrames>
   <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.RenderTransform).(CompositeTransform.TranslateX)" Storyboard
    <EasingDoubleKeyFrame KeyTime="0" Value="0"/>
    <EasingDoubleKeyFrame KeyTime="0:0:1.4" Value="0"/>
```


(C) Copyright Nokia 2013. All rights reserved.

```
    <EasingDoubleKeyFrame KeyTime="0:0:2.9" Value="0"/>
    <EasingDoubleKeyFrame KeyTime="0:0:4.5" Value="0"/>
    <EasingDoubleKeyFrame KeyTime="0:0:5" Value="0"/>
    <EasingDoubleKeyFrame KeyTime="0:0:6" Value="0"/>
   </DoubleAnimationUsingKeyFrames>

   <DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(UIElement.RenderTransform).(CompositeTransform.TranslateY)" Storyboard
    <EasingDoubleKeyFrame KeyTime="0" Value="-7"/>
    <EasingDoubleKeyFrame KeyTime="0:0:1.4" Value="14"/>
    <EasingDoubleKeyFrame KeyTime="0:0:2.9" Value="45"/>
    <EasingDoubleKeyFrame KeyTime="0:0:4.5" Value="45"/>
    <EasingDoubleKeyFrame KeyTime="0:0:5" Value="100"/>
    <EasingDoubleKeyFrame KeyTime="0:0:6" Value="-7"/>
   </DoubleAnimationUsingKeyFrames>

  </Storyboard>
 </UserControl.Resources>

 <Grid x:Name="LayoutRoot" Background="Transparent" Width="100" Height="100">
  <Image x:Name="img" Margin="8"/>
  <Rectangle x:Name="rectangle" Fill="#FF5C5CB6" Margin="0,-101,0,0" Stroke="Black" RenderTransformOrigin="0.5,0.5" Height="100" Ver
   <Rectangle.RenderTransform>
    <CompositeTransform/>
   </Rectangle.RenderTransform>
  </Rectangle>
  <TextBlock x:Name="txt" Height="23" Margin="1,-27,-5,0" TextWrapping="Wrap" Text="TextBlock" VerticalAlignment="Top" RenderTransfo
   <TextBlock.RenderTransform>
    <CompositeTransform/>
   </TextBlock.RenderTransform>
  </TextBlock>
  <Grid.Clip>
          <RectangleGeometry Rect="0,0,100,100" />
      </Grid.Clip>
 </Grid>
</UserControl>
```

7. Now to start this animation at a random time, we should start a thread that is local to this AnimBox.

8. In AnimBox.xaml.cs file, in the constructor, create a DispatcherTimer that fires every random x seconds.

9. In the tick handler of the DispatcherTimer, start the animation.

```
public partial class AnimBox : UserControl
{
 private int timeperiod = 0;
 private Random rnd;
 private DispatcherTimer dt;
 public AnimBox()
 {
  // Required to initialize variables
  InitializeComponent();
  rnd = new Random();
  dt = new DispatcherTimer();
  timeperiod = rnd.Next(7,10);
  dt.Interval = new TimeSpan(0,0,timeperiod);
  dt.Start();
  dt.Tick +=new System.EventHandler(dt_Tick);
 }

 private void dt_Tick(object sender, System.EventArgs e)
 {
  // TODO: Add event handler implementation here.
  System.Diagnostics.Debug.WriteLine("tick" + this.Name);
  animate.Begin();
 }
}
```

With this, creation of the user control is complete. This control can be used in other projects as well, if imported.

10. Now, to instantiate the user control on the main page, place instances of this user control(AnimBox). You can find the user control in the Assets toolbar in Expression Blend. If you don't find it, Build your project and you'll see it later.

11. Drag instances of the AnimBox to the mainPage.

12. Assign the instance of AnimBox with a suitable name, aBox1 and aBox2 are my names.

13. Now in the constructor of MainPage.xaml.cs, assign image source and text label values. Note that elements inside a user control are available to the main page.

```
    public partial class MainPage : PhoneApplicationPage
    {
        public MainPage()
        {
            InitializeComponent();
  aBox1.img.Source = (ImageSource)( new BitmapImage(new Uri("lumia710.jpg",UriKind.RelativeOrAbsolute)) );
            aBox2.img.Source = (ImageSource)(new BitmapImage(new Uri("lumia800.jpg", UriKind.RelativeOrAbsolute)));

  aBox1.txt.Text = "Lumia 710";
  aBox2.txt.Text = "Lumia 800";
        }
    }
```

Download

You can download the entire source code File:AnimTile.zip

Summary

(C) Copyright Nokia 2013. All rights reserved.

This article concentrates on designing and integrating of custom user controls.

 (C) Copyright Nokia 2013. All rights reserved.

 (C) Copyright Nokia 2013. All rights reserved.