Module

Controls

Table of contents

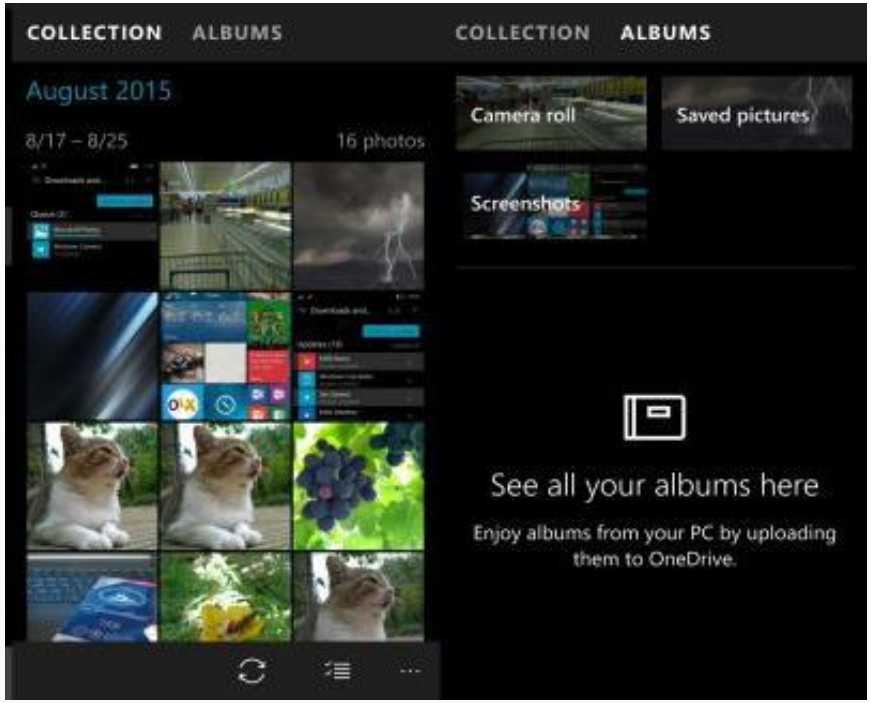
Introduction

- Hub Control

http://forums.wpcentral.com/windows-phone-8-1/264671-windows-phone-8-1-9-modern-ui-concept.html

Introduction

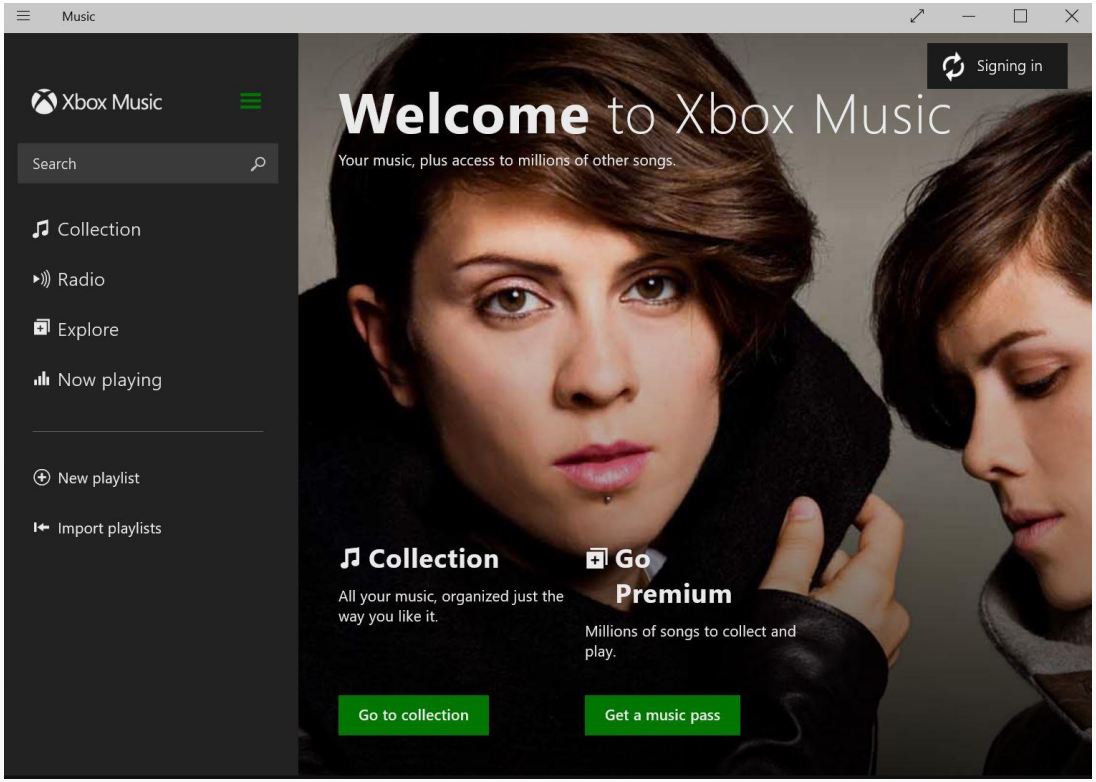
- Pivot Control



http://www.windowsphonearea.com/microsoft-photos-updated-with-pivot-controls/

Introduction

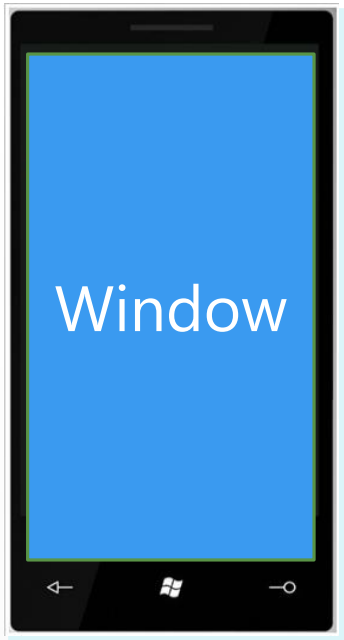
- SplitView Control

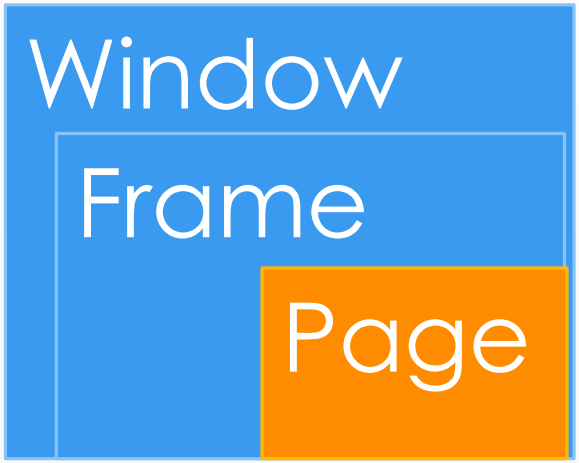


Introduction

- On phone, apps have only one window
- The window contains a single frame, sized at 100% of area
- The frame contains pages, also typically sized at 100% of the area available to the window

  = the container of the pages





Introduction

- A page is built like a tree with a root parent and some children
  - By default : a grid is the parent
- How the controls behave (sizing , margins, padding) is influenced in part by the parent for each control

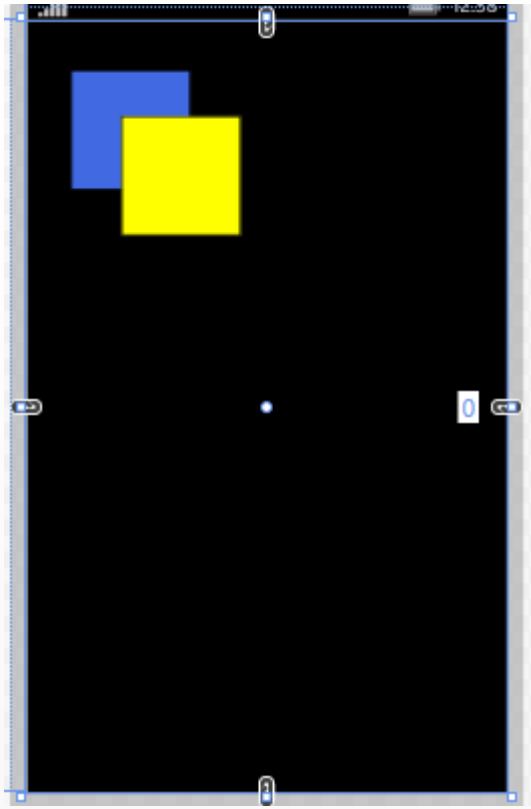
```
<Page
  x:Class="FlickClient.MainPage" ….>

  <Grid>
    <ListView ItemsSource="{Binding Forecast}" ItemTemplate="{StaticResource WeatherTemplate1}"/>
  </Grid>
</Page>
```

Layout

- Process of sizing and positioning visual objects
  - In a container control derived from *Panel* or another container object
  - Such as *Canvas*, *StackPanel* and *Grid*
- Canvas
  - Absolute layout
  - Controls positions
    - Set x/y coordinates
- StackPanel or Grid
  - Dynamic layout
  - Automatically sizing of the UI to various screen resolutions

Absolute Layout

- Canvas



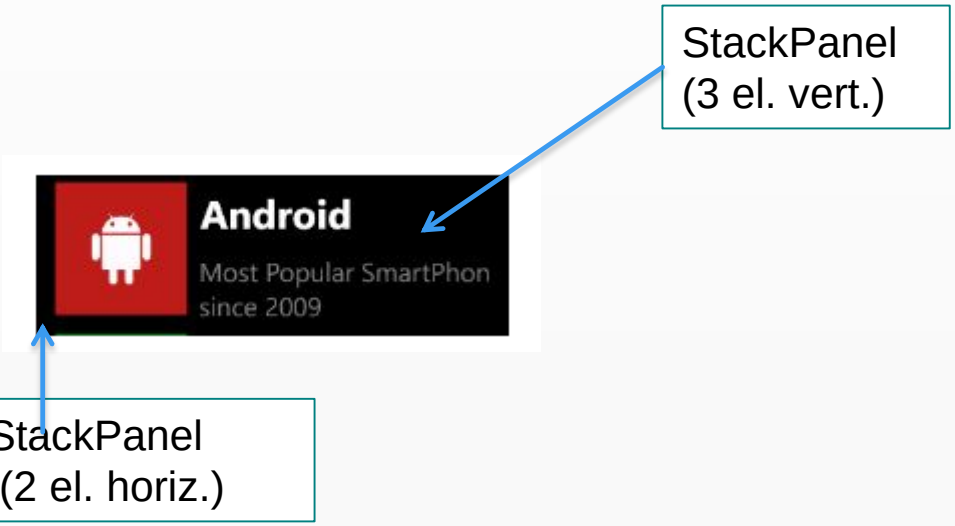
```
<Canvas Background="Transparent">
    <Rectangle Fill="RoyalBlue" Height="100" Canvas.Left="36" Stroke="Black" Canvas.Top="40" Width="100"/>
    <Rectangle Fill="Yellow" Height="100" Canvas.Left="78" Stroke="Black" Canvas.Top="78" Width="100"/>
</Canvas>
```

Absolute Layout

- Canvas
  - Specify the chidren locations relative to their parent element
  - Absolute positioning doesn't consider the size of the screen
    - If the app requires absolute positioning of UI elements, the developer can design different pages for different screen resolutions or uses scaling as an alternative

Dynamic Layout

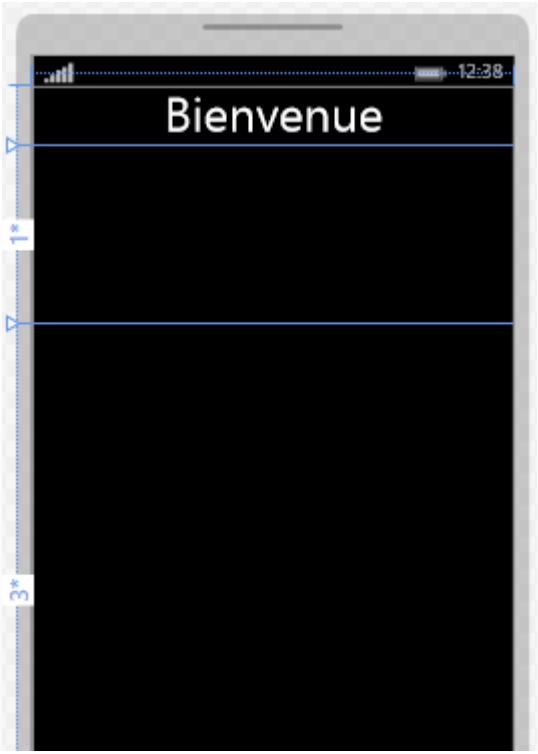
- StackPanel
  - Child elements into a single line that can be oriented horizontally or vertically



http://www.c-sharpcorner.com/UploadFile/167ad2/windows-phone-layout%E2%80%99s-demonstration-in-silverlight/

Dynamic Layout

- Grid




```
<Grid>
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition  Height="*"/>
        <RowDefinition Height="3*"/>
    </Grid.RowDefinitions>
    <TextBlock TextAlignment="Center" FontSize="40" HorizontalAlignment="Center" TextWrapping="Wrap" Text="Bienvenue" VerticalAlignment="Center"/>
</Grid>
```

Dynamic Layout

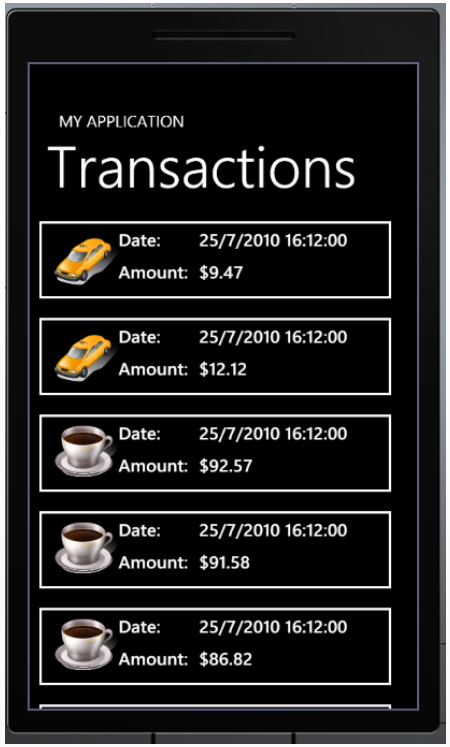
- The user interface appears correctly on various screen resolutions
- Specify how the children should be arranged and how they should wrap relative to their parent
  - Example : horizontally or vertically in the StackPanel
- To use automatic or proportional sizing
  - Assign special values to the Height and Width properties

Dynamic Layout

- Recommended settings
  - Set the Height and Width of the control to Auto
    - When these values are used for controls in the Grid layout, the control fills the cell that contains it
  - Controls that contain text
    - Remove the Height and Width properties
    - Set the MinWidth or MinHeigh properties
      - This prevents the text from scaling down to a size that's unreadable
  - Proportional values for the RowDefinition and ColumnDefinition elements in a Grid layout ("*")

Grouping Controls

- ListBox
  - Properties
    - ItemSource
    - DisplayMemberPath
- PathListBox

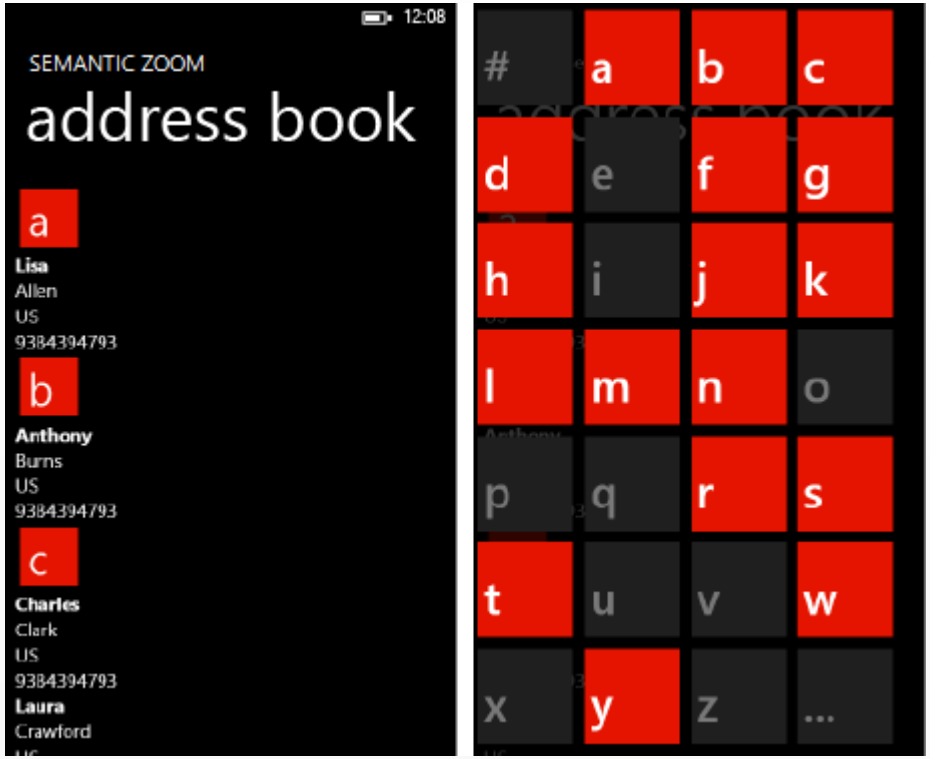


Grouping Controls

- ListView

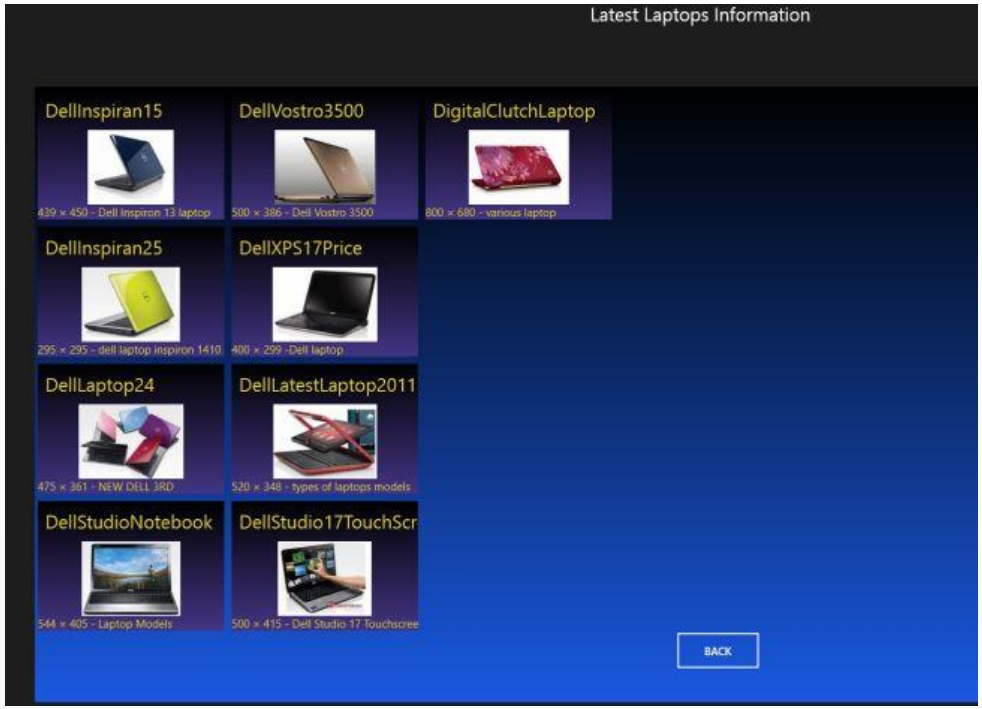
Grouping Controls

- SemanticZoom



Grouping Controls

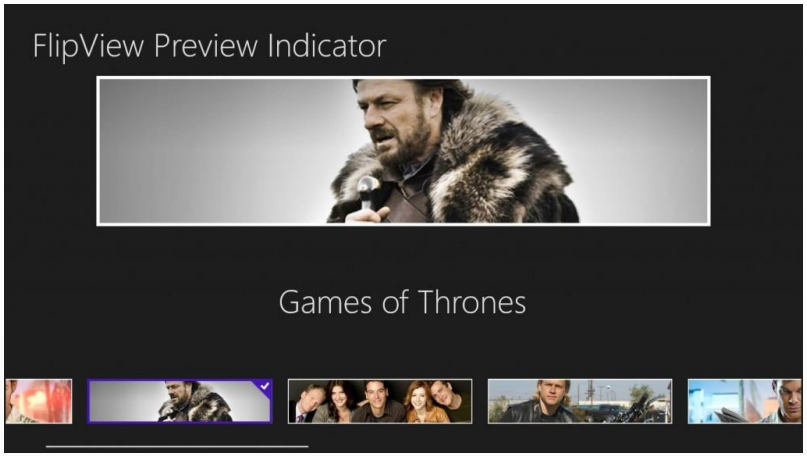
- GridView



http://www.c-sharpcorner.com/UploadFile/c25b6d/image-binding-in-gridview-and-listview-in-windows-8-apps-usi/

Grouping Controls

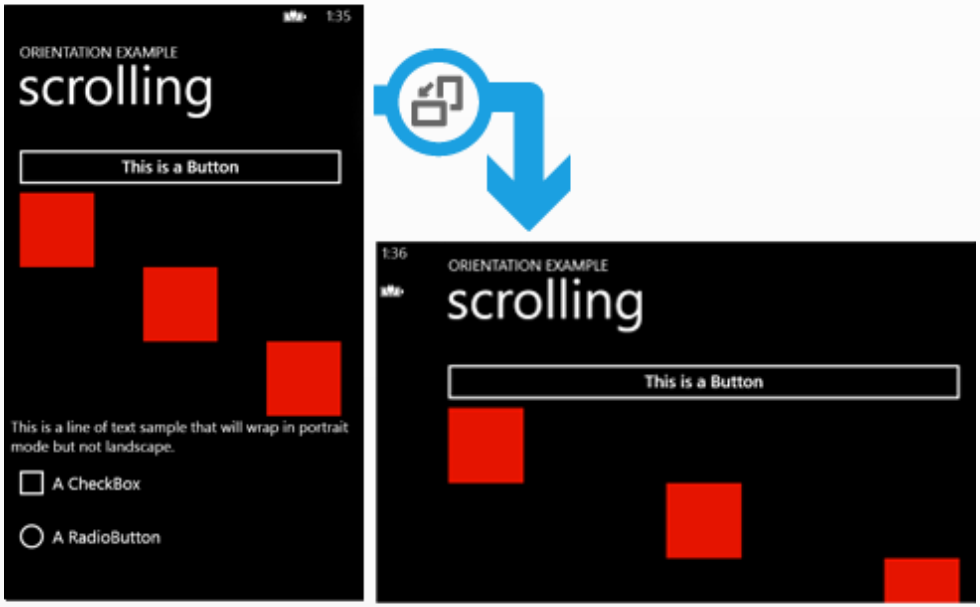
- FlipView



http://archive.renauddumont.be/post/2012/12/04/Windows-8-C-XAML-FlipView-Context-Indicator-et-Previsualisation

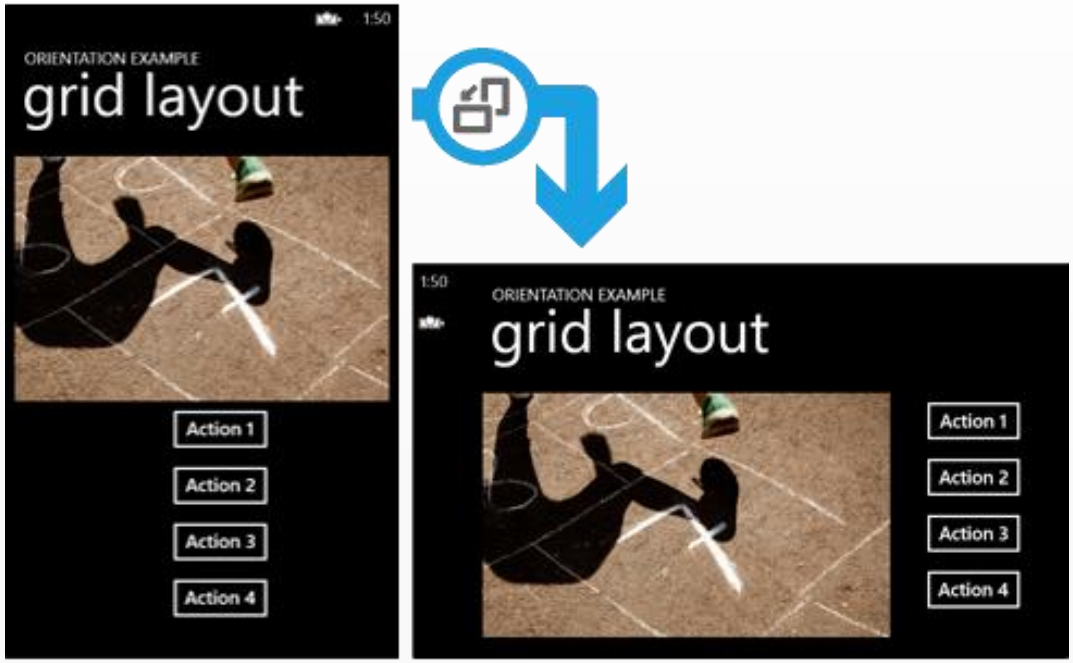
Orientations

- Orientations
  - StackPanel



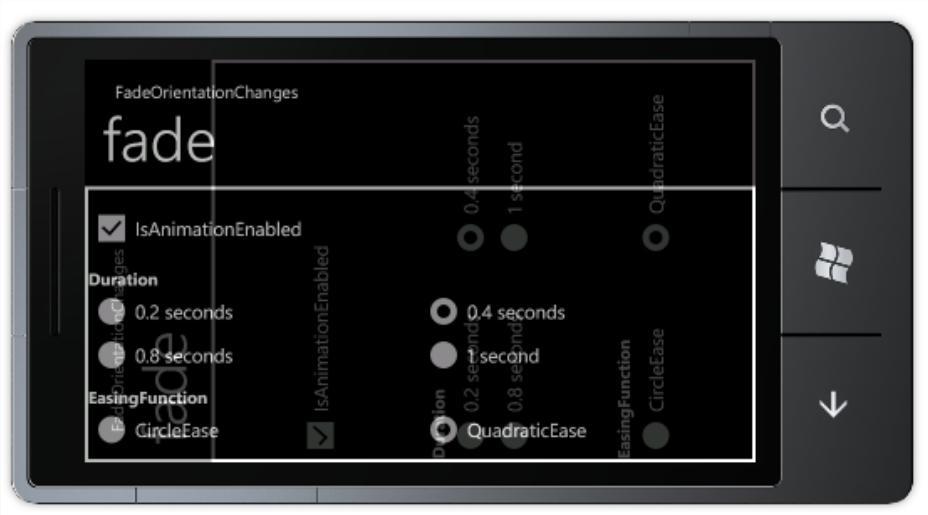
Orientations

- Grid



Orientations

- During the orientation changes,
  - We can create an image overlay which animates the Opacity property of the image



http://blogs.msdn.com/b/delay/archive/2010/09/28/this-one-s-for-you-gregor-mendel-code-to-animate-and-fade-windows-phone-orientation-changes-now-supports-a-new-mode-hybrid.aspx

Orientations

- The content may also rotate

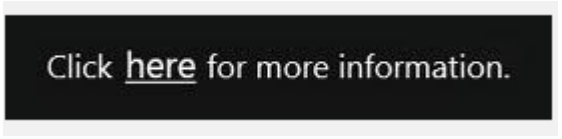
http://blogs.msdn.com/b/delay/archive/2010/09/28/this-one-s-for-you-gregor-mendel-code-to-animate-and-fade-windows-phone-orientation-changes-now-supports-a-new-mode-hybrid.aspx

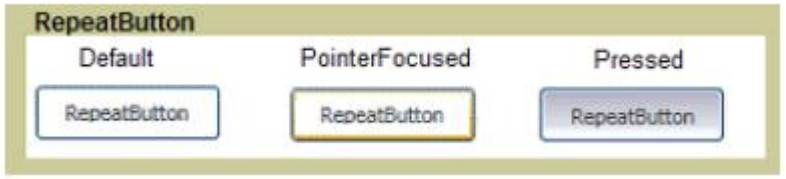
Controls

- Button, Selection, Progress Controls
- Text Controls
- List Controls
- HTML Control
- Image, Map and Media Controls
- Date and Time
- Navigation Controls

Button, Selection and Progress Controls

- Button
- HyperlinkButton
- RepeatButton
- ToggleButton
- CheckBox
- RadioButton

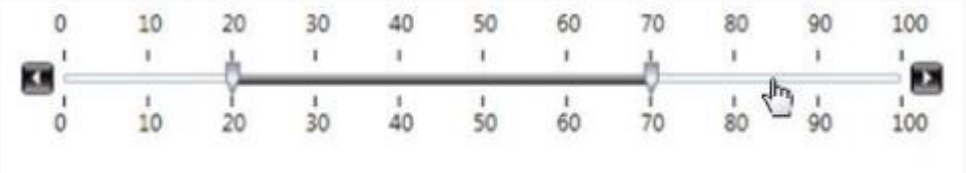


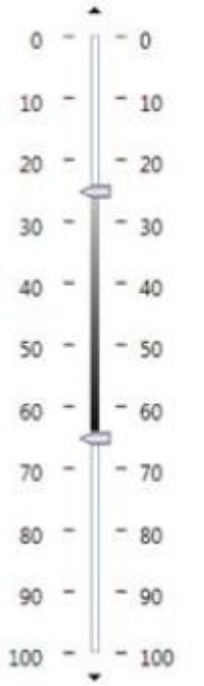


http://msdn.microsoft.com/en-us/library/windowsphone/design/

Button, Selection and Progress Controls

- Slider
  - Useful for selecting values in a range





Button, Selection and Progress Controls

- ProgressBar
  - During time-consuming operations, provides an indication to the user that the app is busy

  - If the activity is kown and progress can be periodically determined, a determinate progressbar is used

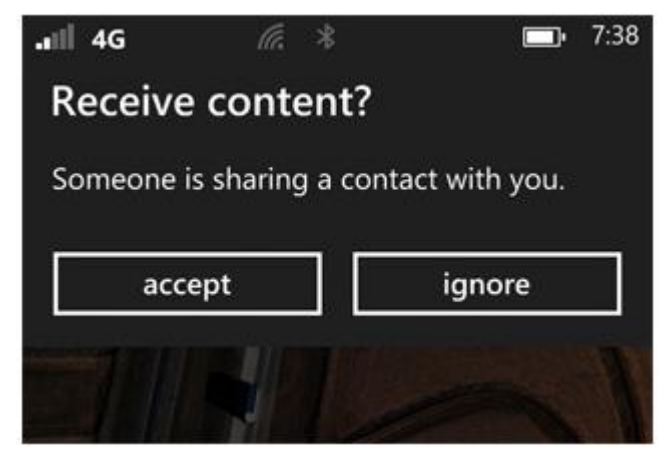
Button, Selection and Progress Controls

- ProgressIndicator
  - To control the progress bar in the status bar

```
<phone:PhoneApplicationPage …shell: SystemTray.IsVisible =
"True"   …>
    <shell:SystemTray.ProgressIndicator>
            <shell:ProgressIndicator IsVisible ="True"
                                     IsIndeterminate = "True"
                                    Text  = " Connecting… " />
   <shell:SystemTray.ProgressIndicator>
</phone:PhoneApplicationPage>
```

Button, Selection and Progress Controls

- Popup



http://winsupersite.com/windows-phone/windows-phone-8-tip-share-nfc

Text Controls

This textbox uses text wrapping. This is a useful feature to avoid truncated words. I've added a scroll bar to the WPF TextBox to enable the user to read all the text in those cases when the TextBox is too

- TextBox
  - Displays and edits data
  - Data in Text Property (string)
  - Support for copy and paste
  - TextWrapping Property
  - Number, Text, Email, Chat, TelephoneNumber, …

SIP : Soft Input Panel


```
<TextBlock Text="E-Mail Address" Margin="12,0,0,0" Style="{StaticResource PhoneTextNormalStyle}" />
<TextBox InputScope="EmailSmtpAddress" Style="{StaticResource PhoneTextBoxStyle}" />
```

Text Controls

- PasswordBox

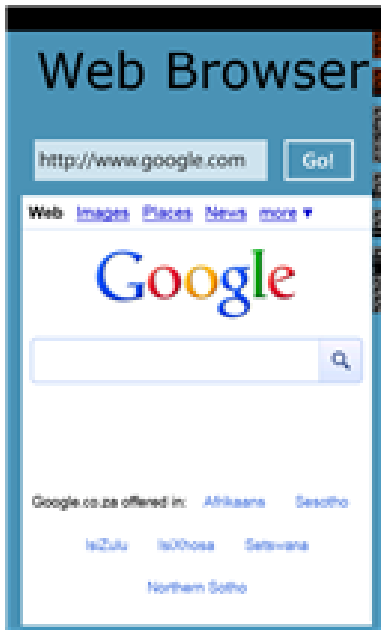
http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202898(v=vs.105).aspx

- TextBlock

- RichTextBlock
  - Rich text container that supports formatted text, hyperlinks, inline images, …

HTML Control

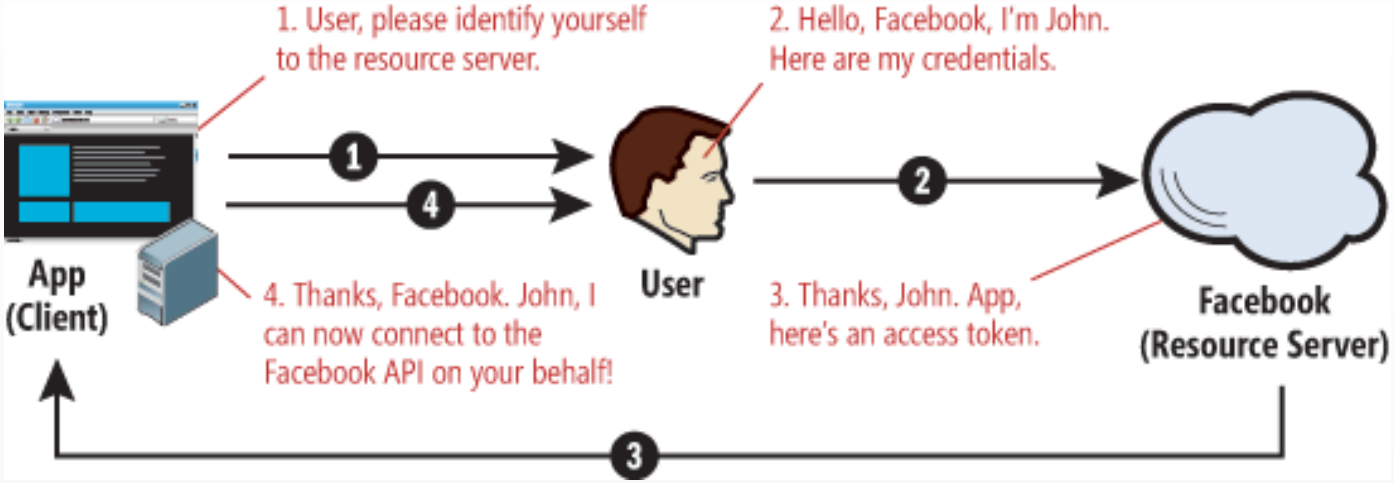
- WebView



http://www.windowsphone.com/en-us/store/app/webbrowser/3dcf1906-302b-4dd9-b320-5a55b97f9a31

HTML Control

- OAuth



http://msdn.microsoft.com/fr-fr/magazine/jj883954.aspx

HTML Controls

- WebView (…)
  - Example : identification OAuth
    - In a component of the app, the developer can call a page identifying an identity provider
    - Then the OAuth flow follows his path to retrieve a token Access
    - The WebView component is closed
    - The token will be used in the rest of the app during the communications with the remote service

Image, Map and Media Controls

- Image
  - Provide bitmap images to work with the scaling system
  - Build responsive UI that adapts to the usable screen size
  - Make use of *DisplayInformation* for granular display information

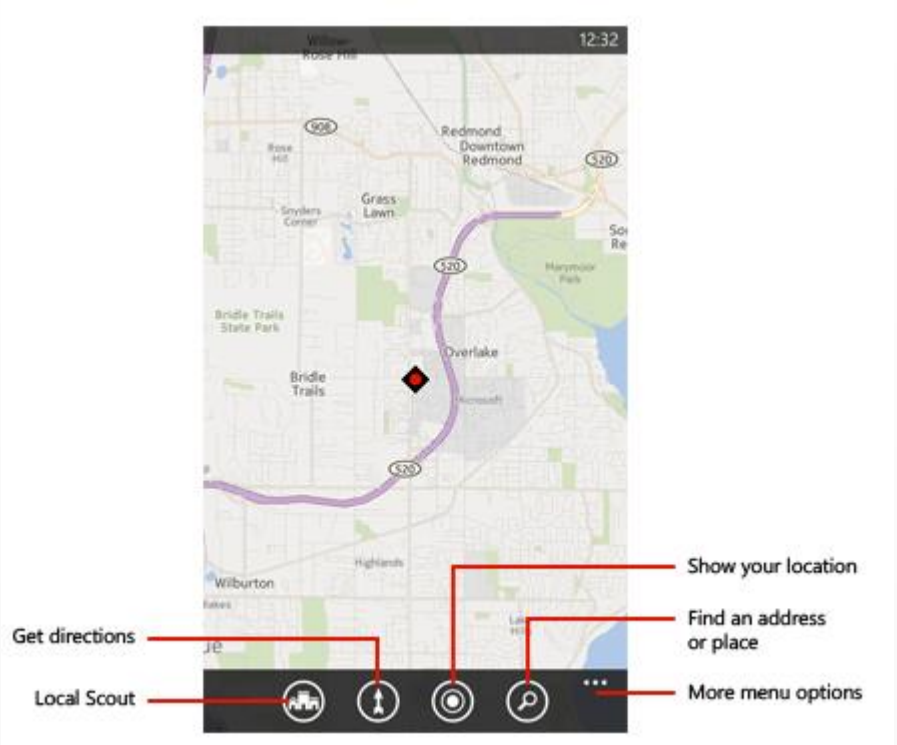
Image, Map and Media Controls

- MediaElement

http://thewire.vertigo.com/tag/timgreenfield/

Image, Map and Media Controls

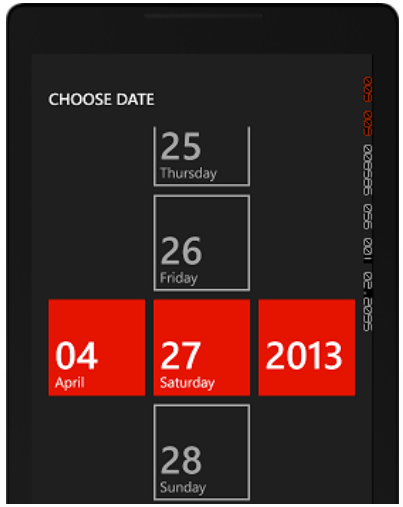
- Map

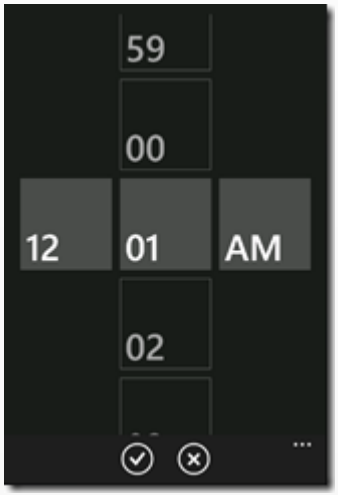


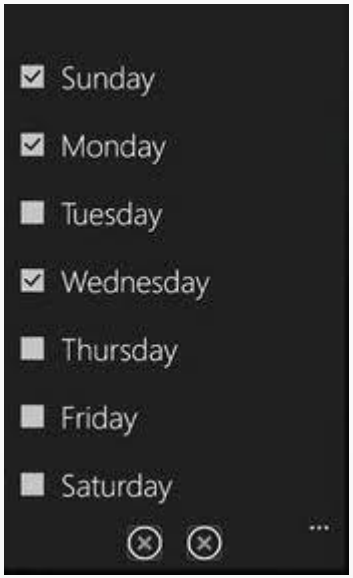
http://www.windowsphone.com/en-us/how-to/wp8/web/use-maps

Date and Time

- DatePicker
  - The user can select a date
- TimePicker
  - The user can select a time

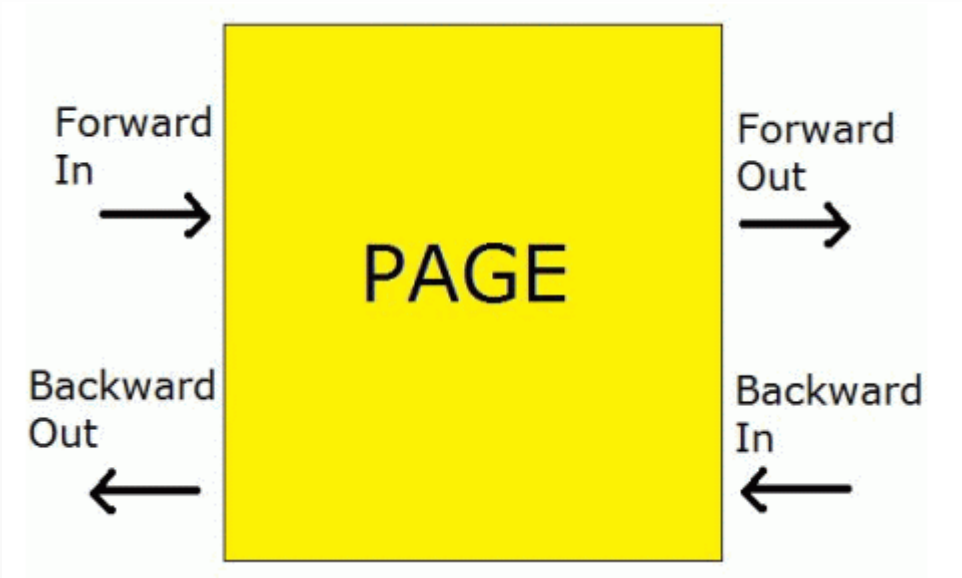






Navigation Controls

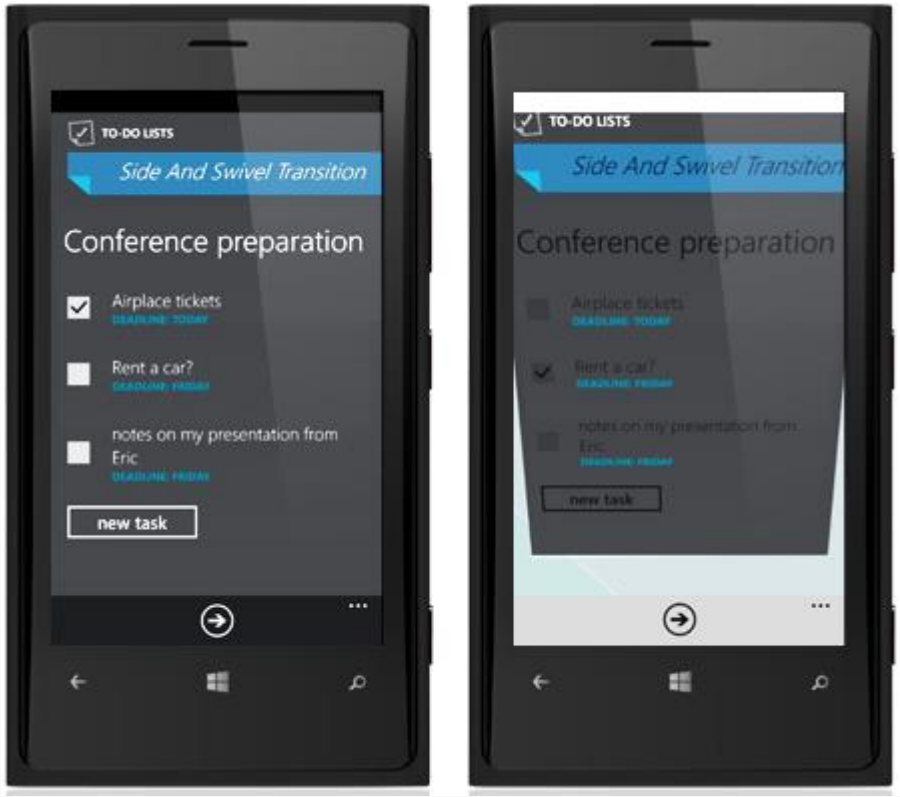
- Animated Page Transitions
  - RollTransition
  - RotateTransition
  - SlideTransition
  - SwiveTransition
  - TurnstileTransition



http://www.c-sharpcorner.com/UploadFile/ae8050/page-transition-animation-in-windows-phone-7/

Navigation Controls

- Example : SwiveTransition



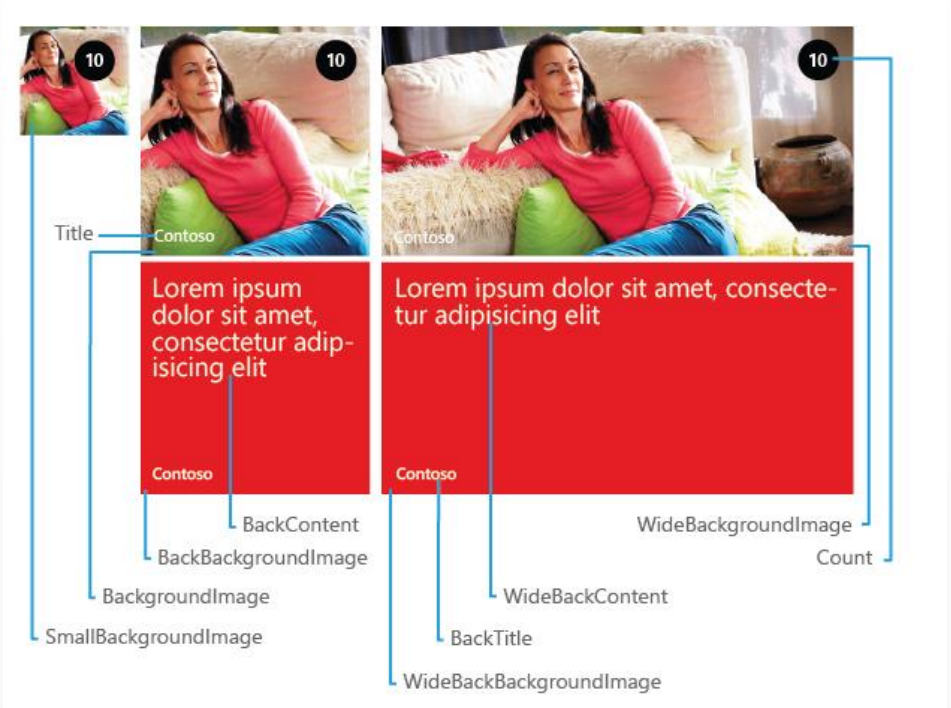
http://www.telerik.com/products/windows-phone/overview/all-controls/applicationframe.aspx

Tiles

- Image that represents the app on the Start screen (default Tile, at least one by app)
- Support multiple sizes, customizable by the user
- One or more secondary tiles
  - Example : the app displays the current weather of a city on its Tile. By supporting secondary Tiles, the customer could choose to pin a Tile for each city that he or she chooses
- Choice of how the Tile updates
  - Never : by using a static Tile
  - Programmatically by using
    - Immediate or scheduled local notifications
    - A cloud service with push notifications

Tiles

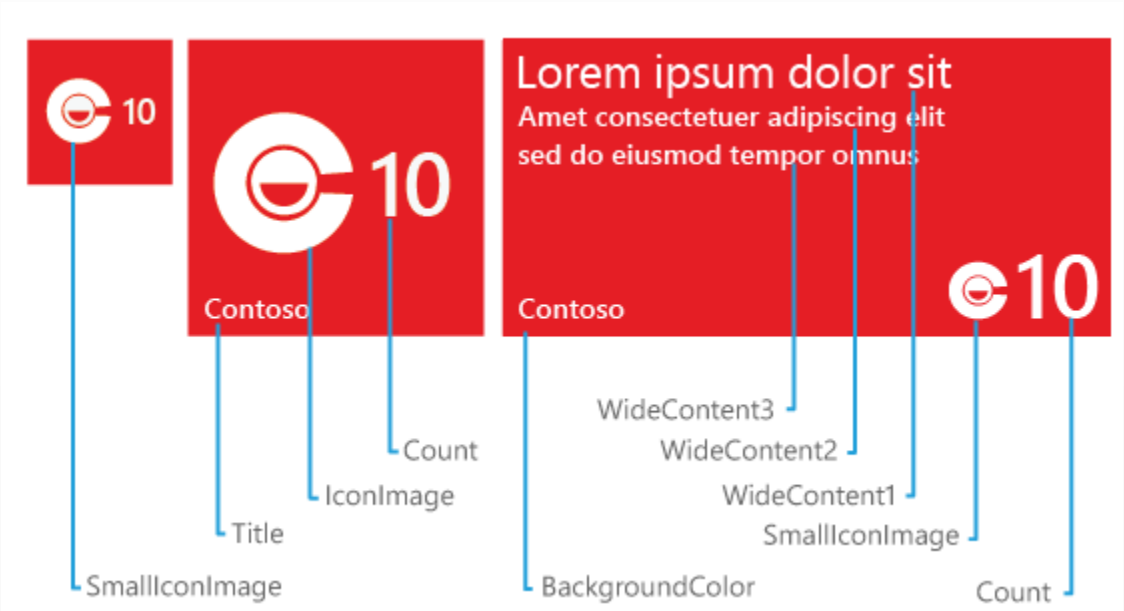
- Three visual templates to help the app stand out
  - Cycle
  - Flip



http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206971(v=vs.105).aspx

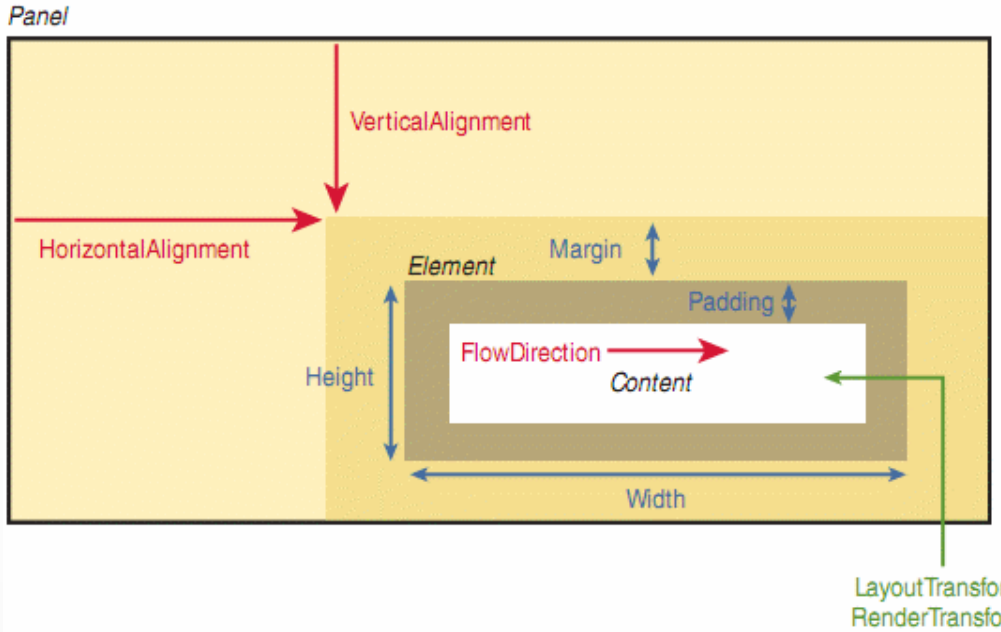
Tiles

- Three visual templates (continue)
  - Iconic



Basics

- Size
  - Height – Width
  - Parent can always override
  - Related propreties
    - MinWidth
    - MinHeight
    - MaxWidth
    - MaxHeight
  - Autosize
    - Value : Auto
    - Works better for larger fonts and translation

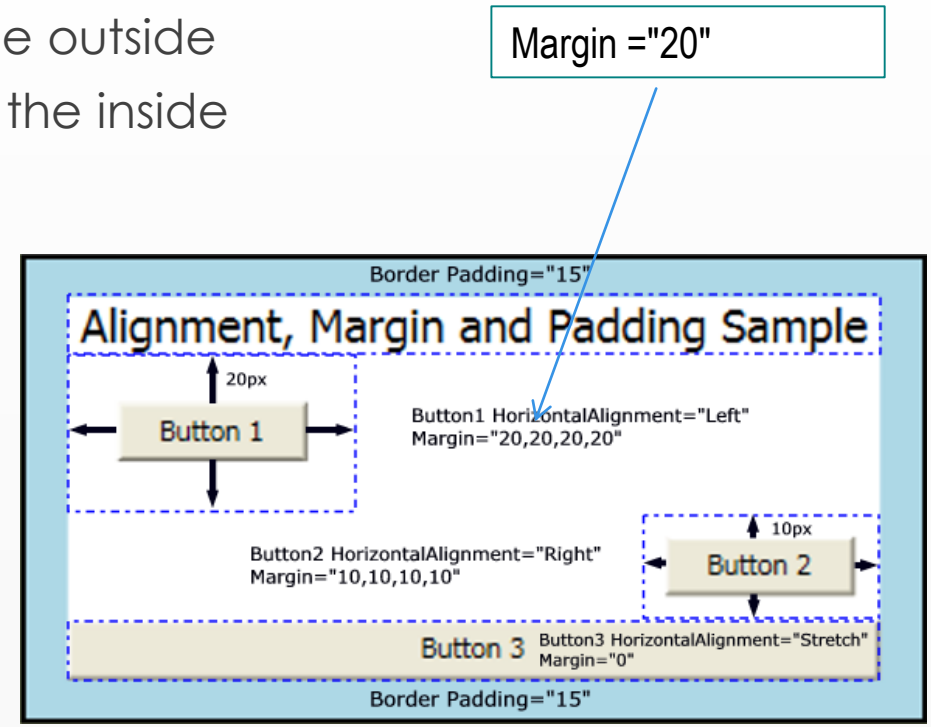


http://msdn.microsoft.com/en-us/library/ms751709(v=vs.110).aspx

Basics

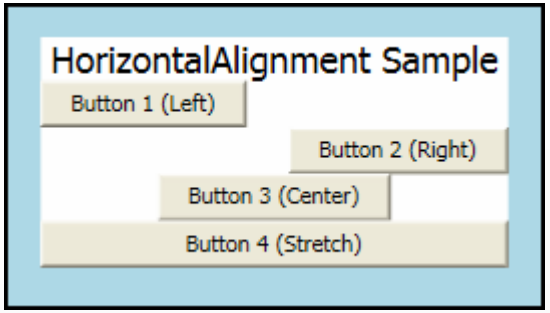
- Margin and Padding
  - Margin : spaces to the outside
  - Padding : spaces to the inside

- Specifying
  - One number
  - Two numbers : Left, Top
  - Four numbers : Left, Top, Right, Bottom



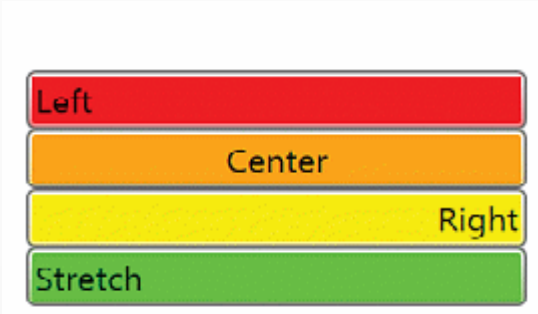
Basics

- Alignment
  - HorizontalAlignment
  - VerticalAlignment



- ContentAlignment

- Visibility
  - Visible
  - Collapsed



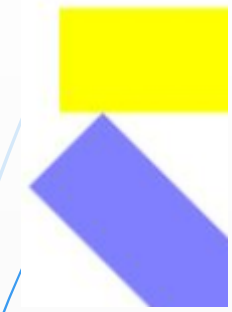
Basics

- RenderTransform / RenderTransformOrigin
  - Gets or sets the center point of any possible render transform declared by RenderTransform, relative to the bounds of the element

```
<Button Content="Flip me!" Padding="5" RenderTransformOrigin="0.5,0.5">
        <Button.RenderTransform>
                <ScaleTransform ScaleY="-1" />
        </Button.RenderTransform>
</Button>
```

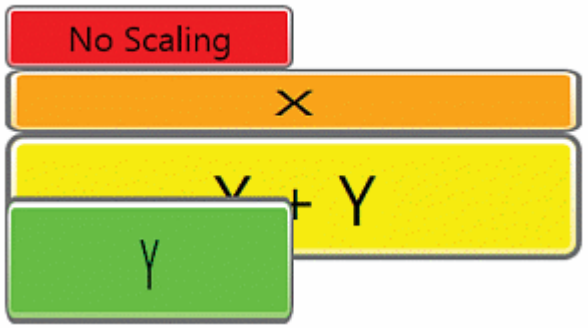
Basics

- Transformation of an element in 2D x-y coordinate system
  - RotateTransform
    - Angle
    - CenterX ,CenterY



```
<Rectangle Width="200" Height="50" Fill="Blue"  Opacity="0.5" Margin="61,27,117,184">
            <Rectangle.RenderTransform>
                <RotateTransform CenterX="-50" CenterY="50" Angle="45" />
            </Rectangle.RenderTransform>
</Rectangle>
```

  - ScaleTransform
    - ScaleX , ScaleY
    - CenterX, CenterY
  - TranslateTransform
    - X et Y



Basics

- SkewTransform
  - AngleX, AngleY
  - CenterX, CenterY


```
<!-- Skew the text by using a SkewTransform. -->
<TextBlock
  FontSize="32"
  FontWeight="Bold"
  Foreground="Maroon"
  Text="Skewed Text">
  <TextBlock.RenderTransform>
    <SkewTransform AngleX="-30" AngleY="0" />
  </TextBlock.RenderTransform>
</TextBlock>

<TextBlock
  Canvas.Top="60"
  FontSize="32"
  FontWeight="Bold"
  Foreground="Maroon"
  Text="Skewed Text">
  <TextBlock.RenderTransform>
    <SkewTransform AngleX="30" AngleY="0" />
  </TextBlock.RenderTransform>
</TextBlock>
```

Basics

- MatrixTransform
  - Affine matrix transformation that is used to manipulate objects or coordinate systems to create custom transformations that are not provided by the RotateTransform, SkewTransform, ScaleTransform or TranslateTransform classes
  - Matrix 3*3

m11	m12	0
m21	m22	0
offSetX	offSetY	1

translation values

```
<Rectangle Width="60" Height="60" Fill="Blue">
  <Rectangle.RenderTransform>
    <MatrixTransform>
      <MatrixTransform.Matrix >
        <Matrix OffsetX="30" OffsetY="100" M12="0.5" />
      </MatrixTransform.Matrix>
    </MatrixTransform>
  </Rectangle.RenderTransform>
</Rectangle>
```

Basics

- TransformGroup
  - Transform composed of other Transform objects

```
<StackPanel>
  <TextBlock FontSize="28" Text="Hello">
    <TextBlock.RenderTransform>
                <TransformGroup>
                            <RotateTransform Angle="45" />
                             <SkewTransform CenterX="0" CenterY="0" AngleX="60"/>
                </TransformGroup>
    </TextBlock.RenderTransform>
  </TextBlock>
</StackPanel>
```

Templates

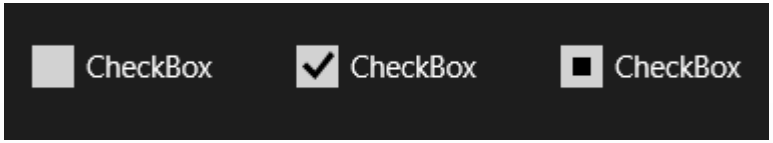
- ControlTemplate
  - It specifies the visual structure and the visual behavior of a control
- DataTemplate
  - It specifies the visualization of the data objects

Templates

- ControlTemplate
  - Controls have many properties (ex. : Background) that the developer can specify
    - But sometimes it's limited
  - *ControlTemplate* class to create a template that provides additional customization
    - The look and feel of a control can be completely customized by giving it a new ControlTemplate
    - The appearance of an existing control is changed but not its functionality

Templates

- Example
  - CheckBox by default
  - With a control template




```
<CheckBox Content="CheckBox" Template="{StaticResource CheckBoxTemplate1}"
IsThreeState="True" Margin="20"/>
```

```
<ControlTemplate x:Key="CheckBoxTemplate1" TargetType="CheckBox">
        <Border BorderBrush="{TemplateBinding BorderBrush}" …                >
            <Grid>
            </Grid>
        </Border>
</ControlTemplate>
```

http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh465374.aspx

Templates

- DataTemplate
  - It specifies the visualization of the data objects
  - DataBinding…
- Example

```
<ListBox Name="lstBoxChien" ItemsSource="{Binding}" Margin="3">
    <ListBox.ItemTemplate>
        <DataTemplate>
            <StackPanel Orientation="Horizontal" >
                <Image Source="{Binding ImageUrl}" Width="50" Margin="5"/>
                <TextBlock Text="{Binding Nom}" Margin="0,5,0,0"/>
            </StackPanel>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
```

Styles

- To format UI controls
- Example 1

```
<Button Content = "Press me!" Style ="{StaticRessource ButtonStyle} " …>

Où

<Style x:Key="ButtonStyle" TargetType="Button">
                <Setter Property="HorizontalAlignment" Value="Left"/>
</Style>
```

Styles

- Example 2

```
<Page.Resources>
    <Style x:Key="PhoneToggleSwitch" TargetType="ToggleButton">
                <Setter Property="Background" Value="{StaticResource PhoneBackgroundBrush}"/>
                <Setter Property="Foreground" Value="{StaticResource PhoneAccentBrush}"/>
                 <Setter Property="BorderBrush" Value="{StaticResource PhoneForegroundBrush}"/>
                <Setter Property="BorderThickness" Value="3"/>
                <Setter Property="MinWidth" Value="90"/>
                <Setter Property="MaxWidth" Value="90"/>
                 <Setter Property="MinHeight" Value="90"/>
                <Setter Property="MaxHeight" Value="90"/>
   </style>
</Page.Resources>
```

Styles

- Example 3

```
<Application.Resources>
    <Style x:Key="PhoneToggleSwitch" TargetType="Button">
                <Setter Property="Background" Value="{StaticResource PhoneBackgroundBrush}"/>
                <Setter Property="Foreground" Value="{StaticResource PhoneAccentBrush}"/>
                 <Setter Property="BorderBrush" Value="{StaticResource PhoneForegroundBrush}"/>
                <Setter Property="BorderThickness" Value="3"/>
                <Setter Property="MinWidth" Value="90"/>
                <Setter Property="MaxWidth" Value="90"/>
                 <Setter Property="MinHeight" Value="90"/>
                <Setter Property="MaxHeight" Value="90"/>
    </style>
</Application.Resources>
```

Resources

- StaticResource
  - Retrieved only once by the referencing element and used for entire life of the resource

```
<Grid Background="{StaticResource myColor}"></Grid>
```

- DynamicResource
  - Acquired every time the referenced object is used

```
<Grid Background="{DynamicResource myNewColor}"></Grid>
```

```
RadialGradientBrush radialGradientBrush = new RadialGradientBrush(Colors.Orange,
Colors.Blue);
this.Resources["myNewColor"] = radialGradientBrush;
```

Resources

- Repository for XAML resources such as styles
  - The developer defines the resources in XAML
  - He can retrieve them in XAML using the StaticResource markup extensions

```
<Page x:Class="Sample.MainPage"  xmlns=…>
<Page.Resources>
            <Style x:Key="ButtonStyle1" TargetType="Button"/>
 </Page.Resources>
<Grid >
            <Button Style="{StaticResource ButtonStyle1}" Text="Click Me"/> </Grid>
</Page>
```

Resources

- In a ResourceDictionary

```
<Application xmlns=…>
<Application.Resources>
    <ResourceDictionary>
                <Style x:Key="ButtonStyle1" TargetType="Button">
                    <Setter Property="Foreground" Value="…"/>
                    <Setter Property=…/>
                 …
                </Style>
                <Style x:Key= "…" TargetType= " …">
                     …
                </Style>
            ...
</ResourceDictionary>
</Application.Resources>
</Application>
```