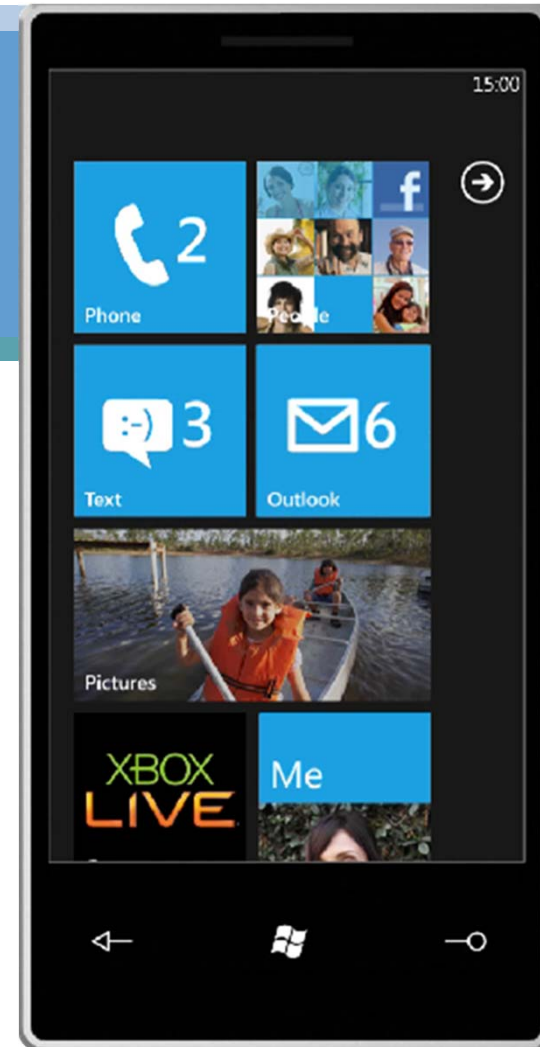


Windows Phone



Présentation

- Deux frameworks pour développer des applications windows phone:
 - Silverlight
 - XNA
- Outils
 - Microsoft Visual Studio 2010 Express
 - Windows Phone Emulator
 - Microsoft Expression Blend

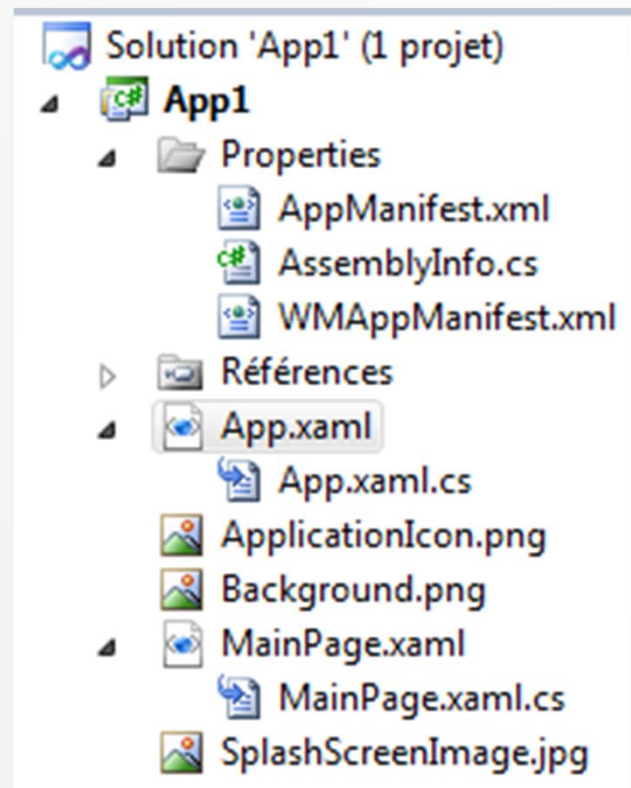
Configuration minimale d'un WP7

- Un écran tactile.
- Des capteurs tels que le GPS, l'accéléromètre, le détecteur de luminosité et la boussole.
- Une caméra de 5 mégapixels au moins avec déclenchement par bouton et flash.
- Une mémoire RAM d'au moins 256 Mo (8 Go Flash).
- Un CPU de type ARMv7 Cortex/Scorpion au minimum.
- 6 boutons : Retour, Démarrer, Recherche, Prise de photo, Volume, Mise sous tension.
- 2 résolutions d'écran : WVGA (800x480) et HVGA (480x320).
- Un GPU supportant les accélérations DirectX 9.
- Bluetooth, Wifi Radio.



Événements Tactiles

- Tap : simple touché sur l'écran d'une courte durée.
- Double Tap : double touché sur l'écran d'une courte durée.
- Pan : touché sur l'écran avec un glissé long dans une ou plusieurs directions.
- Flick : touché sur l'écran avec un glissé rapide dans une direction.
- Pinch And Stretch : déplacement et agrandissement avec deux doigts.



1-1 Structure d'une fenêtre

- L'attribut Class définit le nom complètement qualifié de la classe qui représente la fenêtre (de type System.Windows.Window).
- Les attributs xmlns définissent les schémas xml utilisés par la fenêtre.
- Les contrôles d'une fenêtre sont placés dans un conteneur de type Panel qui gère leur positionnement (Grid par exemple).

```
<phone:PhoneApplicationPage
    x:Class="PhoneApp5.MainPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-
namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
">
<Grid x:Name="LayoutRoot" Background="Transparent"></Grid>
</phone:PhoneApplicationPage>
```



La classe Binding

- La classe Binding permet d'établir une liaison entre un objet et une propriété cible.
- Propriétés:
 - ElementName: permet de définir l'élément WPF source de la liaison
 - Source: définit l'objet (dans le cas d'un élément WPF la propriété ElementName doit être utilisée au lieu de la propriété Source)
 - FallbackValue: Valeur à utiliser par défaut.
 - Mode: définit la direction, valeurs: *TwoWay*, *OneWay*, *OneWayToSource*
 - NotifyOnSourceUpdated=true: l'événement SourceUpdated est déclenché si une valeur est transférée de la cible à la source de la liaison.
 - NotifyOnTargetUpdated=true: Déclenche l'événement TargetUpdated si une valeur est transférée de la source à la cible de la liaison.

- Path: définit le chemin à la propriété source de la liaison.
- TargetNullValue: définit la valeur de la cible à utiliser si la valeur de la source est nulle.
- Liaison à un élément Silverlight

```
<Slider Name="s1" Minimum="2" Width="100" Maximum="72" Value="12" />  
<TextBlock Name="r1" Grid.Column="2" FontSize="{Binding  
ElementName=s1, Path=Value}" />
```

- La propriété DataContext
 - Définit le contexte de données pour un élément et ses éléments fils.
 - Cette propriété sera utilisée comme source de données pour les éléments liés, et dont l'instance de la classe Binding ne définit ni la propriété Source ni la propriété ElementName.

Exemple 1

```
<Grid x:Name="ContentPanel1" Grid.Row="1"
Margin="12,0,12,0">
<Grid.RowDefinitions>
<RowDefinition Height="*" />
<RowDefinition Height="*" />
<RowDefinition Height="*" /> </Grid.RowDefinitions>
<Slider Name="slider" Value="90" Grid.Row="0"
Maximum="180" Margin="24" />
<TextBlock Name="txtblk"
Text="{Binding ElementName=slider, Path=Value}"
Grid.Row="1" FontSize="48" HorizontalAlignment="Center"
VerticalAlignment="Center" />
<Rectangle Grid.Row="2"
Width="{Binding ElementName=slider, Path=Value}"
RenderTransformOrigin="0.5 0.5" Fill="Blue">
<Rectangle.RenderTransform>
<RotateTransform x:Name="rotate" Angle="{Binding
ElementName=slider, Path=Value}" />
</Rectangle.RenderTransform>
</Rectangle>
</Grid>
```

Exemple 2 (Binding à partir du code)

```
Binding binding = new Binding(); //Espace de noms: System.Windows.Data
binding.ElementName = "slider";
binding.Path = new PropertyPath("Value");
txtblk.SetBinding(TextBlock.TextProperty, binding);
```

Binding Convertisseurs

Etape 1: Créer la classe de type IValueConverter

```
public class ConvertirFormat : IValueConverter
//namespace: System.Windows.Data
    { //conversion source --> cible
        object IValueConverter.Convert(object value, Type targetType, object
parameter, System.Globalization.CultureInfo culture)
        { if (value is double)
            return Math.Round((double) value);
            return value;
        }

        //conversion cible --> source
        object IValueConverter.ConvertBack(object value, Type targetType,
object parameter, System.Globalization.CultureInfo culture)
        {
            return value;
        }
    }
```

Etape 2: Ajouter une référence au namespace de la classe dans MainPage.xaml

```
xmlns:local="clr-namespace:PhoneApp8;assembly=PhoneApp8"
```

Etape 3: déclarer le convertisseur comme une ressource de la page

```
<phone:PhoneApplicationPage.Resources>  
    <local:ConvertirFormat x:Key="formater"/>  
    ...
```

Etape 4: définir l'attribut Converter de la classe Binding dans TextBlock

```
Text="{Binding ElementName=slider, Path=Value,  
Converter={StaticResource formater}}".
```

Remarque:

Une autre possibilité d'utilisation du convertisseur sans créer une ressource:

```
TextBlock ... >  
    <Binding ElementName="slider" Path="Value">  
        <Binding.Converter>  
            <local:ConvertirFormat/>  
        </Binding.Converter>  
    </Binding>  
</TextBlock>
```

Liaison à une liste (ListBox, ComboBox)

- Propriétés
 - DisplayMemberPath: propriété de la collection liée à afficher.
 - IsSynchronizedWithCurrentItem: Détermine si l'élément sélectionné est synchronisé avec la propriété CurrentItem dans la collection Items.
 - ItemsSource: une collection qui fournit les données pour la liste.

Exemple 2 DataContext et Binding

