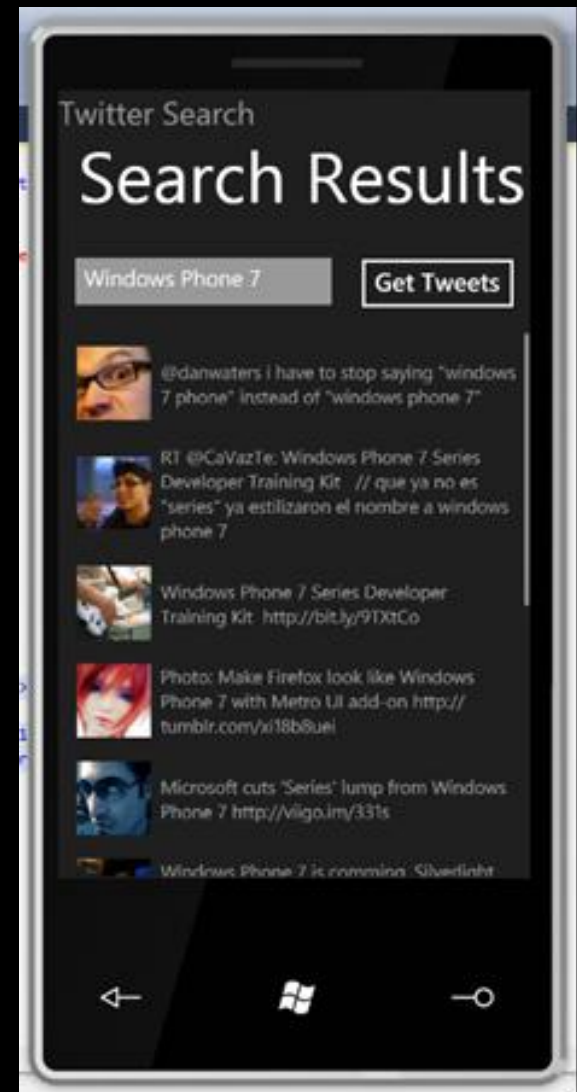




Microsoft Silverlight for Windows Phone 7

Christoph Hofmann

Microsoft Expert Student Partner
Universität Freiburg



Agenda

- Was ist Silverlight?
- Tooling
- UI-Design
- Codeeinbindung
- Data-Binding
- Funktionen & Sensoren



Was ist Silverlight?

Christoph Hofmann

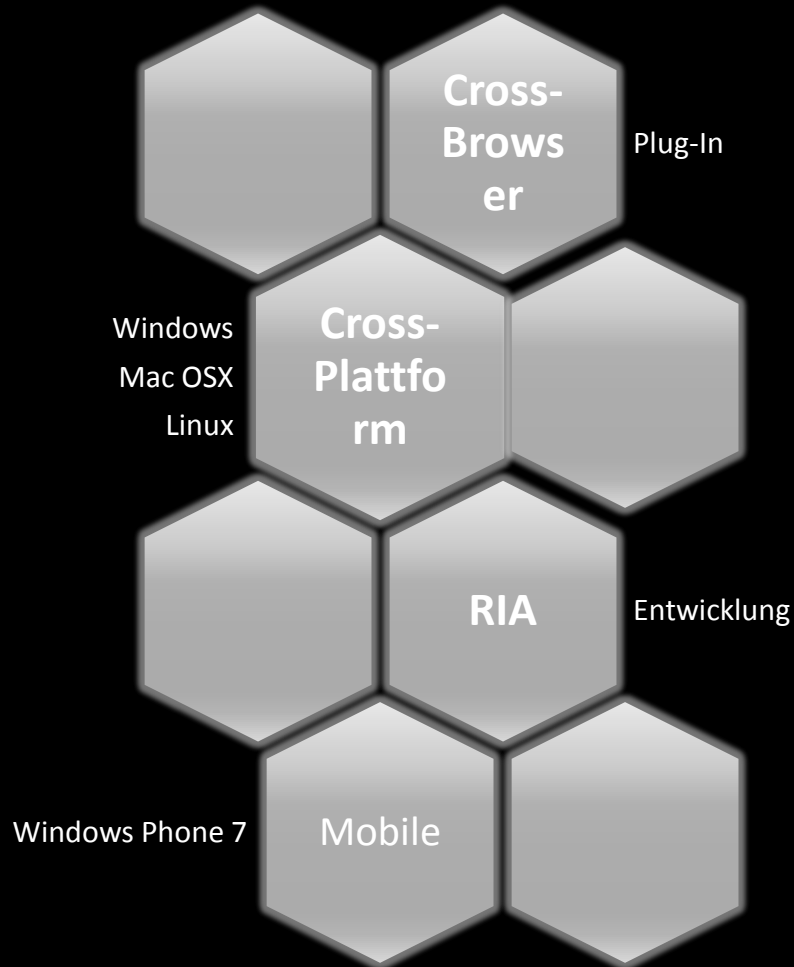
Microsoft Expert Student Partner

Universität Freiburg



Microsoft Student Partners

Was ist Silverlight?

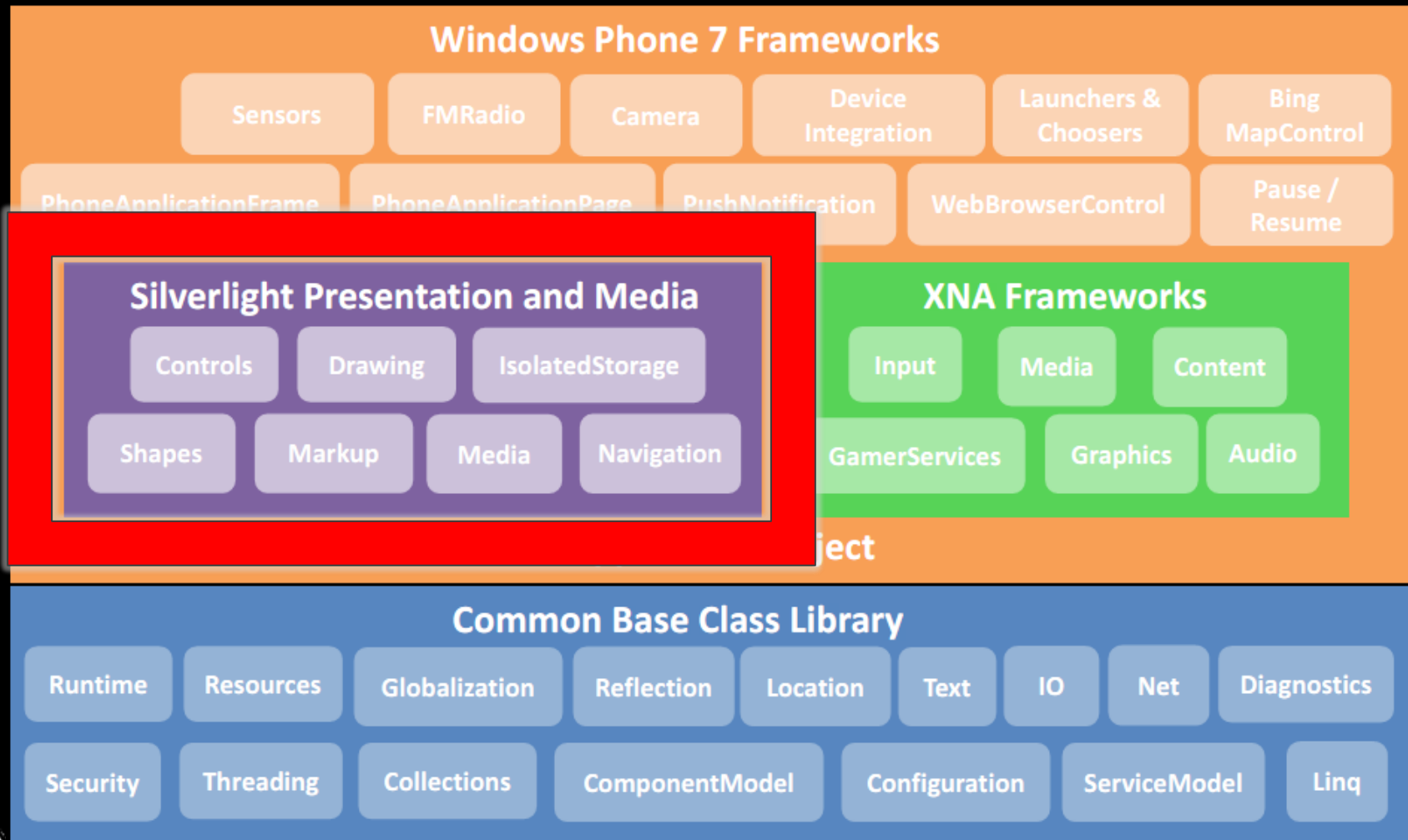


Silverlight for Windows Phone

- ◎ Silverlight 3 Plus
 - Höchste Performance
 - Input-Integration
 - HW-, Medien- & Sensor – Integration
 - Adaption für WP7-App-Modell
 - Weniger enge Sandbox



Silverlight for Windows Phone



Was ist XAML?

- ⦿ Extensible Application Markup Language
- ⦿ Deklaratives erstellen von Oberflächen in XML
- ⦿ Elemente in XAML
 - (User-) Controls
 - Ressourcen
 - Styles, Templates
 - Einfache Werte, Datenobjekte
 - Grafikelemente (Rectangle, Ellipse, Path ...)
- ⦿ Eigenschaften als Attribute oder Kindelemente (z. B. Background)
- ⦿ Markup-Erweiterungen (z. B. Binding, StaticResource, DynamicResource)



XAML - Beispiel

```
<Grid>
  <Button Width="200" Height="200">
    <StackPanel>
      <CheckBox Content="Hallo Welt"/>
      <Button Margin="10,5">
        <MediaElement Source="d:\Temp\Media.wmv"/>
      </Button>
    </StackPanel>
  </Button>
</Grid>
```



Tooling

Werkzeuge für Programmierung und Design



Microsoft Student Partners

Tools für Silverlight

- ◎ Windows Phone Developer Tools
 - Visual Studio 2010 Express for Windows Phone
 - IDE mit Codevervollständigung, Debugger, Designer und andere Tools
 - Microsoft Expression Blend for Windows Phone
 - Designertool für das Bearbeiten von XAML
 - Erleichtert Arbeiten mit Templates, Styles und Animationen
- ◎ Silverlight Toolkit for Windows Phone
 - Open Source Bibliothek von Microsoft
 - Inhalt: Controls, Themes, ...



UI-Design

Layout-Möglichkeiten und Steuerelemente



Microsoft Student Partners

Pivot Control

PivotItem.Title
FRANCE

seattle **paris** london
PivotItem.Header

City	Temperature	Condition	Humidity	Cloud Cover
Seattle	16°C	Clear/Sunny	46%	0%
Paris	16°C	Clear/Sunny	46%	0%
London	8°C	Light drizzle	81%	50%

Date	Max Temp	Min Temp	Wind	Rain	Condition
8 May 2010 (Saturday)	21°C	4°C	6 Kph	0 mm	Clear/Sunny
9 May 2010 (Sunday)	22°C	4°C	9 Kph	0 mm	Clear/Sunny
10 May 2010 (Monday)	14°C	6°C	8 Kph	1.9 mm	Light Rain
11 May 2010 (Tuesday)	20°C	5°C	6 Kph	0 mm	Clear/Sunny
12 May 2010 (Wednesday)	22°C	5°C	11 Kph	0 mm	Clear/Sunny

PivotControl.Items[0]

PivotItem.Content Powered by World Weather Online



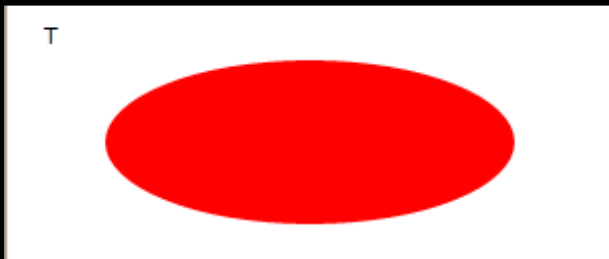
Panorama Control



Canvas

- Absolute Positionierung der Elemente
- Angabe des Abstandes zur linken oberen Ecke

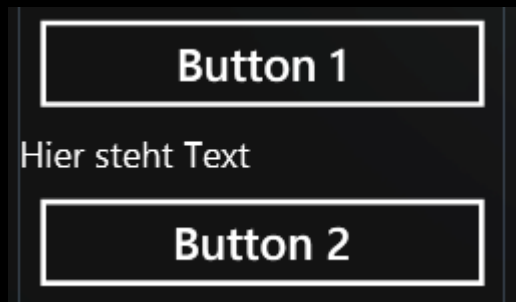
```
<Canvas Background="White">  
  <TextBlock Canvas.Left="20" Canvas.Top="10" Text="T"/>  
  <Ellipse Fill="Red" Canvas.Left="50" Canvas.Top="30"  
    Width="200" Height="80"/>  
</Canvas>
```



StackPanel

- Anordnung der Elemente horizontal (`Orientation="Horizontal"`) oder vertikal (`Orientation="Vertical"`)

```
<StackPanel Orientation="Vertical">  
  <Button>Button 1</Button>  
  <TextBlock Text="Hier steht Text"/>  
  <Button>Button 2</Button>  
</StackPanel>
```



Grid

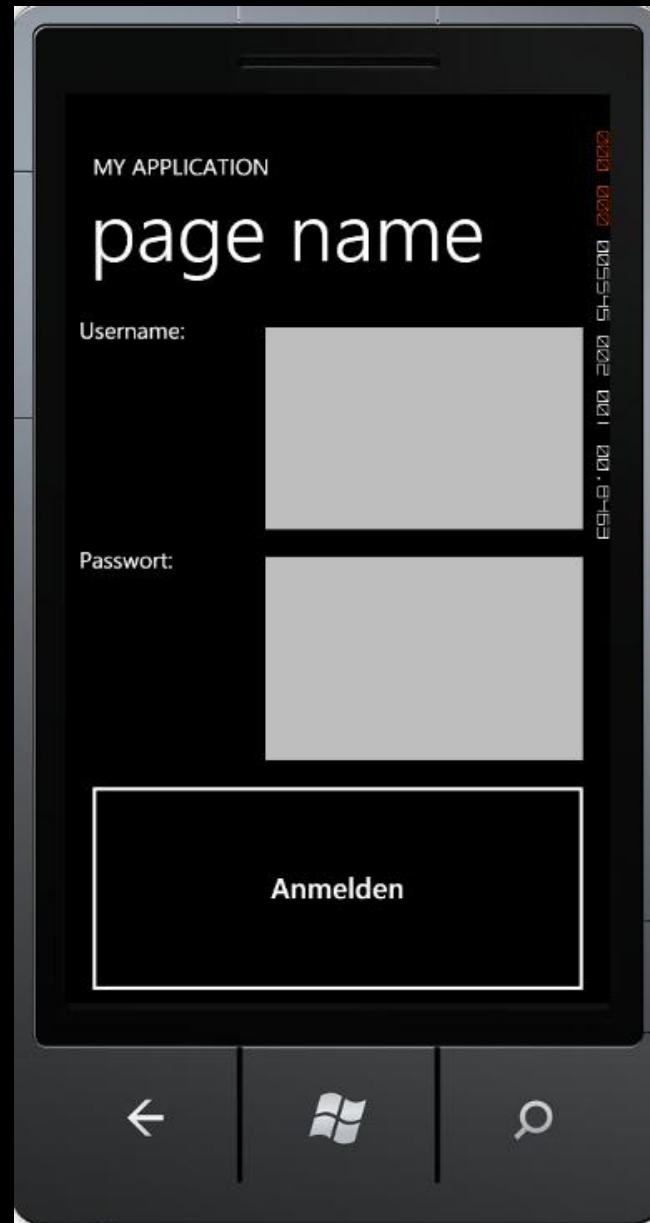
- Tabellenlayout
- Spaltendefinition durch `ColumnDefinition`, Zeilendefinition durch `RowDefinition`
- Größe für Spalten bzw. Zeilen
 - Absolut: `RowDefinition Height=„30“`
 - Relativ: `RowDefinition Height=„*“`
- Elemente können sich über mehrere Spalten (`ColumnSpan`) bzw. Zeilen (`RowSpan`) erstrecken



Grid

```
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="*" />
    <ColumnDefinition Width="2*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
    <RowDefinition Height="*" />
  </Grid.RowDefinitions>
  <TextBlock Grid.Column="0" Grid.Row="0" Text="Username:" />
  <TextBox Grid.Column="1" Grid.Row="0" />
  <TextBlock Grid.Column="0" Grid.Row="1" Text="Passwort:" />
  <TextBox Grid.Column="1" Grid.Row="1" />
  <Button Grid.Column="0" Grid.ColumnSpan="2" Grid.Row="2"
Content="Anmelden" />
</Grid>
```

Grid



Steuerelemente

Interaktionselemente eines UI



TextBlock und TextBox

- ◎ TextBlock
 - Statische Textanzeige
- ◎ TextBox
 - Editierbarer Textbereich
- ◎ Gemeinsame Eigenschaften
 - Textumbruch über TextWrapping
 - Textausrichtung über TextAlignment

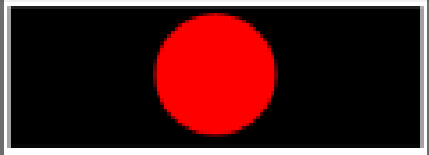


Button



Klick mich

- Klickbare Schaltfläche
- Festlegen des Buttoninhalts über Content
- Content = Ein beliebiges Control



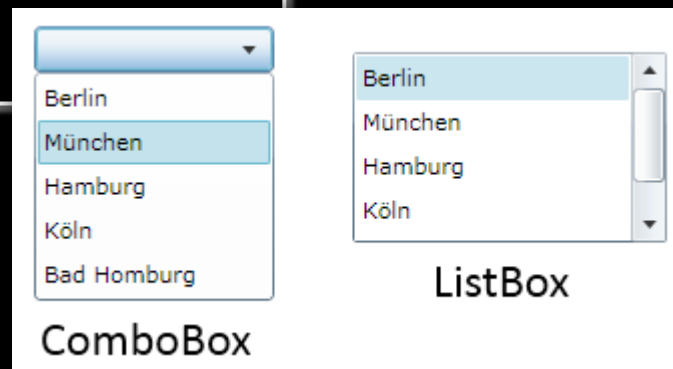
```
<Button Content="Klick mich"/>
<Button>
  <TextBlock Text="Klick mich"/>
</Button>
<Button>
  <Ellipse Width="200" Height="200" Fill="Black"/>
</Button>
```



ComboBox und ListBox

- Darstellung von Auswahllisten
- Items-Eigenschaft definiert Auswahl

```
<ComboBox>  
  <ComboBoxItem Content="Berlin"/>  
  ...  
</ComboBox>  
<ListBox>  
  <ListBoxItem Content="Berlin"/>  
  ...  
</ListBox>
```



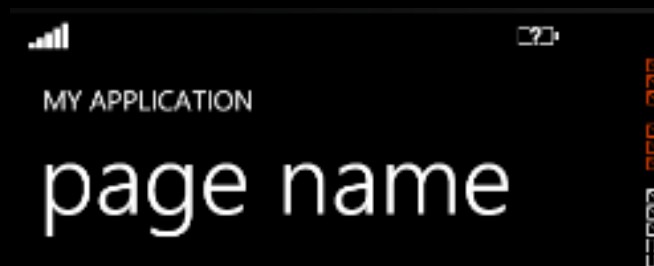
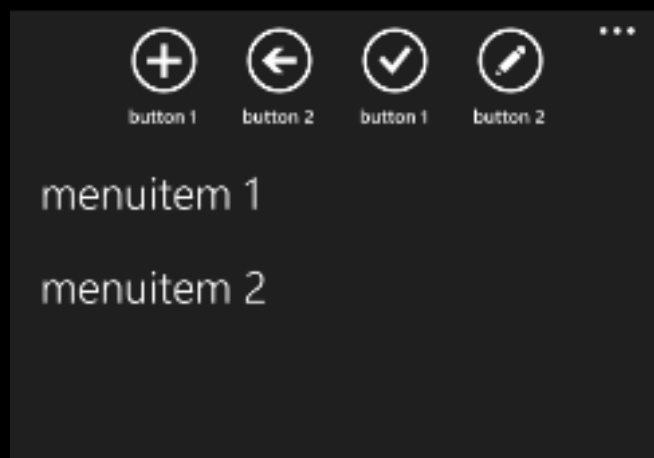
ComboBox und ListBox (2)

- ◉ Items zur Laufzeit hinzufügen via Add-Methode der Items-Eigenschaft
- ◉ ListBox erlaubt mehrere Elemente auszuwählen via SelectionMode
- ◉ Auswertung der ausgewählten Elemente:
 - SelectedIndex: Index des Elements; -1 = keine Auswahl
 - SelectedItem: Ausgewählte Objekt
 - SelectedItems (ListBox): Ausgewählten Objekte



Application Bar

- ◎ Icons
 - Maximal 5 Stück
 - Properties
 - Icon Pack zum Download
- ◎ Menü
 - Zugriff mit auf die 3 Punkte in App Bar
- ◎ Systemtray
 - W-Lan stärke, Uhr, Mobilfunkbetreiber usw.
 - `shell:SystemTray.IsVisible="False"`



Application Bar

```
<phone:PhoneApplicationPage.ApplicationBar>
  <shell:ApplicationBar IsVisible="True" IsMenuEnabled="True">
    <shell:ApplicationBarIconButton
      IconUri="Images/appbar.add.rest.png" Text="Button 1"/>
    <shell:ApplicationBarIconButton
      IconUri="Images/appbar.back.rest.png" Text="Button 2"/>
    <shell:ApplicationBarIconButton
      IconUri="Images/appbar.check.rest.png" Text="Button 1"/>
    <shell:ApplicationBarIconButton
      IconUri="Images/appbar.edit.rest.png" Text="Button 2"/>
    <shell:ApplicationBar.MenuItems>
      <shell:ApplicationBarMenuItem Text="MenuItem 1"/>
      <shell:ApplicationBarMenuItem Text="MenuItem 2"/>
    </shell:ApplicationBar.MenuItems>
  </shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationBar>
```



Allgemeine Eigenschaften

- ◎ Controlanordnung im Container:
 - HorizontalAlignment
 - VerticalAlignment
- ◎ Absolute Controlgröße: Width und Height
- ◎ Abstände: Margin und Padding
- ◎ Schrift
 - FontFamily: Schriftart
 - FontSize: Schriftgröße
 - FontStyle: normal oder kursiv
 - FontWeight: Schriftgewicht (fett)



Codeeinbindung



Überblick

- ◎ XAML beschreibt Instanzen von .NET Klassen
- ◎ Element muss Namen besitzen, um aus eigenen Code ansprechbar zu sein
 - Festlegung über Name-Eigenschaft
- ◎ Silverlight ist eine ereignisgetriebene UI-Technologie => Ereignisse behandeln, um eigenen Code auszuführen



Ereignisbehandlung

- Ereignis = Statusänderung
- Definition von Rückrufmethoden (EventHandler), um auf Statusänderung zu reagieren

```
<ListBox SelectionChanged="MyListChanged">  
    ...  
</ListBox>  
  
private void MyListChange(object sender,  
    SelectionChangedEventArgs e)  
{  
    // Hier steht der eigene Code  
}
```



Data-Binding

Datenanzeige ohne Code



Überblick

- ◎ Deklarative Datendarstellung durch Data-Binding => Einsparung von Code
- ◎ Woran kann ich binden?
 - Eigenschaften von Klassen und Controls
 - Ressourcen
 - Styles (Sammlung von Formatieranweisungen)
 - Templates (Anweisung zur Objektdarstellung)
- ◎ Verschiedene Bindungsmodi: OneTime, OneWay, TwoWay
- ◎ Elemente erben Datenquelle - DataContext



Data-Binding an Eigenschaften

```
<StackPanel Orientation="Vertical" Background="Wheat">  
  <TextBox Name="txt" Text="5" />  
  <Rectangle Width="200" Height="200" Fill="Black"  
    RadiusX="{Binding ElementName=txt, Path=Text}"  
    RadiusY="{Binding ElementName=txt, Path=Text,  
      Mode=OneTime}" />  
</StackPanel>
```

Zielelement

Eigenschaft

- ⦿ Anforderung an Zielelement für Bindungsmodi OneWay und TwoWay
 - Implementieren der Schnittstelle `INotifyPropertyChanged`



ComboBox/ListBox und Data-Binding (1)

- ◎ Datenquelle: `ItemsSource`
- ◎ `ItemsSource` kann auch an `DataContext` binden (`ItemsSource="{Binding}"`)
- ◎ Festlegung der Datendarstellung:
 - `DisplayMemberPath`: Eine Eigenschaft wird für Darstellung verwendet
 - `ItemTemplate`: Benutzerdefinierte Darstellung



ComboBox/ListBox und DataBinding (2)

```
<ListBox Name="box">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <Border CornerRadius="25" Background="LightBlue">
        <StackPanel Orientation="Horizontal" Margin="15">
          <TextBlock Text="{Binding Path=Name}"
            FontSize="24" />
          <TextBlock Text="{Binding Path=Alter}"
            FontSize="16" />
        </StackPanel>
      </Border>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```



ComboBox/ListBox und DataBinding (3)

```
public MainPage()
{
    // Hier wird die UI erzeugt
    InitializeComponent();

    List<Person> personen =
        new List<Person>();
    // Liste füllen...

    box.ItemsSource = personen;
}
```



Sensoren & Funktionen

Sensoren, Tasks, Chooser



Silverlight for Windows Phone

Windows Phone 7 Frameworks

Sensors

FMRadio

Camera

Device
Integration

Launchers &
Choosers

Bing
MapControl

PhoneApplicationFrame

PhoneApplicationPage

PushNotification

WebBrowserControl

Pause /
Resume

Controls

Drawing

IsolatedStorage

Shapes

Markup

Media

Navigation

Input

Media

Content

GamerServices

Graphics

Audio

Application Object

Common Base Class Library

Runtime

Resources

Globalization

Reflection

Location

Text

IO

Net

Diagnostics

Security

Threading

Collections

ComponentModel

Configuration

ServiceModel

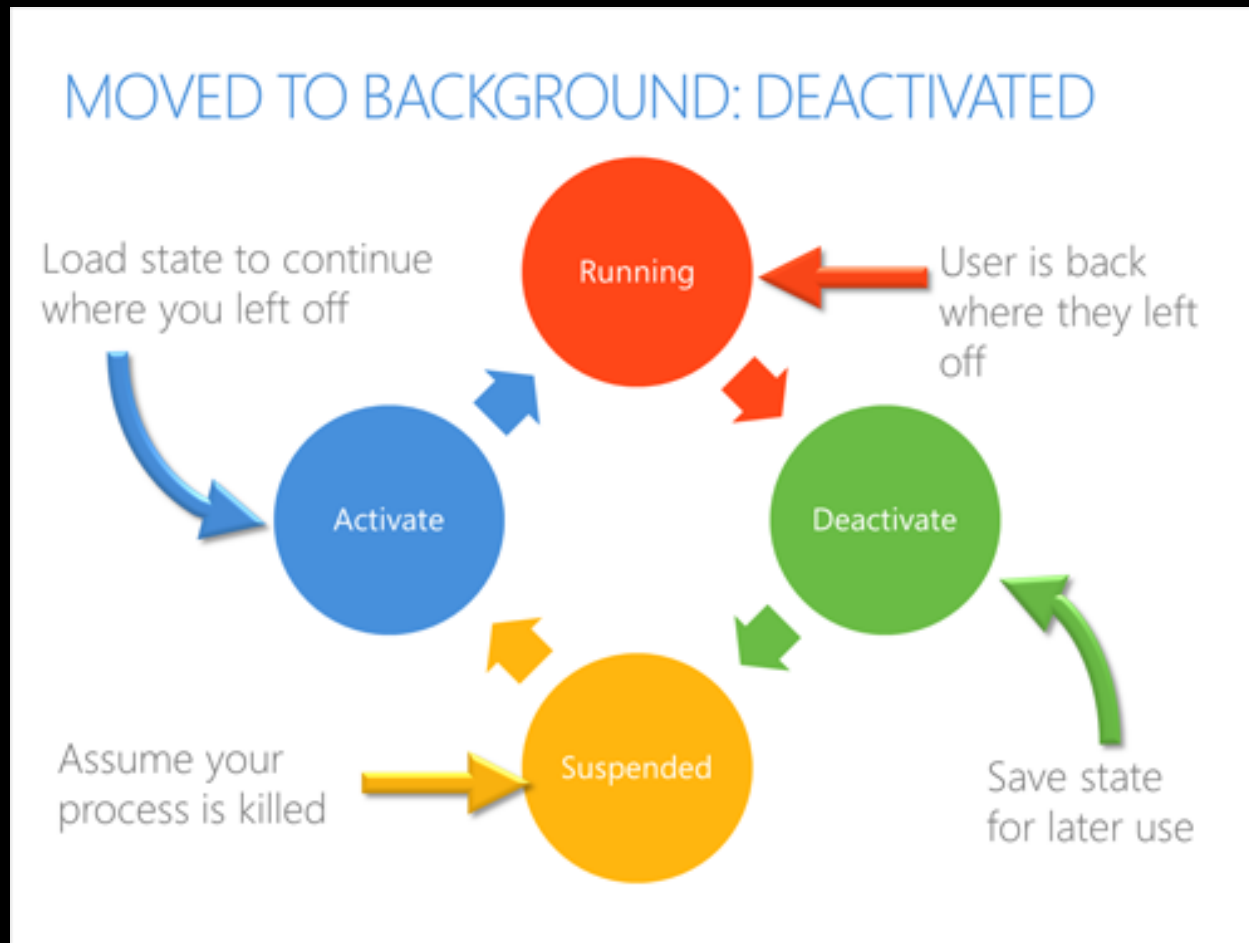
Linq

Überblick

- ◎ Application Life Cycle
- ◎ Funktionen
 - Launchers
 - Chooser
 - Push Notifications
- ◎ Sensoren
 - Location Services
 - Accelerometer (XNA Vortrag)
 - Vibration Controller



Application Life Cycle – Smart Multitasking



Application Life Cycle – Smart Multitasking

```
private void Application_Launching(object sender, LaunchingEventArgs e)
{
}

private void Application_Activated(object sender, ActivatedEventArgs e)
{
}

private void Application_Deactivated(object sender, DeactivatedEventArgs e)
{
}

private void Application_Closing(object sender, ClosingEventArgs e)
{
}
```



Launchers

- Werden benutzt um Funktionen des Telefons aufzurufen
- Namespace: using Microsoft.Phone.Tasks
- Tasks: EmailComposeTask, MarketplaceDetailTask, MarketplaceHubTask...

```
SmsComposeTask sct = new SmsComposeTask();  
sct.To = "5555555555";  
sct.Body = "Call me when you have a chance. Let's do lunch today.";  
sct.Show();
```



Chooser

- Ermöglichen Zugriff auf Telefonbuch, Bilder
- Gibt Daten an die Anwendung zurück
- Namespace: using Microsoft.Phone.Tasks;
- Choosers: CameraCaptureTask,
EmailAddressChooserTask,
PhoneNumberChooserTask,
PhotoChooserTask



Chooser

```
CameraCaptureTask cct = new CameraCaptureTask();
cct.Completed += new EventHandler<PhotoResult>(cct_Completed);
cct.Show();

void cct_Completed(object sender, PhotoResult e)
{
    BitmapImage bmp = new BitmapImage();
    bmp.SetSource(e.ChosenPhoto);
    image1.Source = bmp;
}
```

Device Orientation

- Kann in den Properties des Root Elementes eingestellt werden
- SupportedOrientation gibt an was unterstützt wird (Portrait , Landscape & PortraitOrLandscape)
- Orientation gibt die aktuelle Orientation an (Landscape, LandscapeLeft, LandscapeRight, Portrait, PortraitDown, PortraitUp)

```
<phone:PhoneApplicationPage  
x:Class="WindowsPhoneApplication1.MainPage,,  
...  
SupportedOrientations="Portrait" Orientation="Portrait,,  
shell:SystemTray.IsVisible="True">
```



Location Services

Location: Sources of Location



Location Services

- ◎ Namespace: using System.Device.Location;
- ◎ Nutzen des Location Services
 - Erstelle ein neues GeoCoordinateWatcher object
 - Erstelle einen Event Handler für Positionsänderungen
 - Hole die Daten, wenn das Event ausgelöst wird

```
GeoCoordinateWatcher gcw;  
  
public MainPage()  
{  
    gcw.PositionChanged += new  
        EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>(gcw_PositionChanged);  
  
    gcw.Start();  
}  
  
void gcw_PositionChanged(object sender,  
    GeoPositionChangedEventArgs<GeoCoordinate> e)  
{  
    Latitude.Text =  
        e.Position.Location.Latitude.ToString();  
  
    Longitude.Text =  
        e.Position.Location.Longitude.ToString();  
}
```



Vibration Controller

- Warum?
 - Game Feedback
 - Button Feedback
 - Notifications
- Wie lange?
 - Kürzeste Zeit: 100ms
 - Sollte nicht länger als 300ms sein
 - => Gefährlich fürs Handy

```
VibrateController vc = VibrateController.Default;  
vc.Start(TimeSpan.FromMilliseconds(100));
```

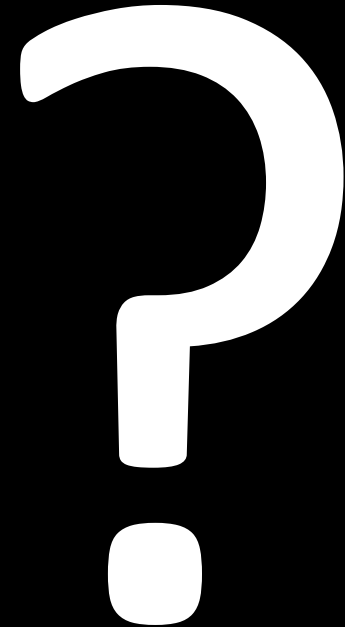


Ressourcen

- ◎ Developer Tools:
http://download.microsoft.com/download/1/7/7/177D6AF8-17FA-40E7-AB53-00B7CED31729/vm_web.exe
- ◎ Application Bar Icons:
http://windowsteamblog.com/windows_phone/b/wpdev/archives/2010/03/27/windows-phone-7-series-icon-pack.aspx
- ◎ Webcasts
 - Einführung in SL2: <http://tinyurl.com/SL2Start>
 - RIAs mit SL: <http://tinyurl.com/RIAmitsL>
 - SL2 Tipps&Tricks: <http://tinyurl.com/SL2Tipps>
 - Neuerungen SL3: <http://tinyurl.com/SL3News>
 - SL in Deep: <http://tinyurl.com/SLAdv>
 - Neuerungen SL4: <http://tinyurl.com/SL4News>



Fragen & Antworten



Vielen Dank für Ihre Aufmerksamkeit!

Email: Christoph.Hofmann@studentpartners.de

