One app to rule them all
Zero to hero with the Universal Windows Platform development

Bluefragments
Daniel Vistisen
Senior consultant

blendrocks
Deani Hansen
Developer & UI designer
Daniel
Senior developer at Bluefragments.
Architect & developer

Deani
Windows app engineer at Slack.
UI designer and developer
One hour from now..

Fundamentals of the Universal Windows Platform

Responsive UI

<RelativePanel/> <VisualStates/> <Triggers/> 

Continuum and different target devices

Q/A
Demo app

Desktop

4K video

Phone

720p video

IoT

Image
Windows 10 supports a wide range of devices including:

- Phones
- Phablets
- Small Tablets
- Large Tablets
- 2-in-1s (Tablet or Laptop)
- Classic Laptops
- Desktops & All-in-Ones
- Surface Hub
- Xbox
- Holographic
- IoT devices
Windows device families

- Universal device family
- Desktop device family
- Mobile device family
- Xbox device family
- IoT device family
- IoT headless device family
  [...]


Universal App != Universal App

**Windows 8.1**
- Full Screen
- Portable Class Libraries (PCL)
- Windows Runtime (WinRT)
- One binary per target
- Intermediate Language

**Windows 10**
- Resizable Windows
- XAML + PCL
- Core API + WinRT
- One Binary Overall
- .NET Native
Scaling algorithm

- 16"
- 20"
- 28"
- 120"

Diagram showing the scaling algorithm with measurements.
Relative resolutions

- 100 x 100
- 150 x 150
- 200 x 200
- 320 ppi
  - Phone
- 720 ppi
  - Phablet (landscape)
  - and tablet (portrait)
- 1024 ppi
  - Desktop
  - and laptop
A new XAML layout control, which arranges its children by declaring relationships between them.

<RelativePanel> ... </RelativePanel>
RelativePanel, on MSDN

```xaml
<RelativePanel BorderBrush="Gray" BorderThickness="10">
  <Rectangle x:Name="RedRect" Fill="Red" MinHeight="100" MinWidth="100"/>
  <Rectangle x:Name="BlueRect" Fill="Blue" MinHeight="100" MinWidth="100"
             RelativePanel.RightOf="RedRect"/>
  <!-- Width is not set on the green and yellow rectangles. It's determined by the RelativePanel properties. -->
  <Rectangle x:Name="GreenRect" Fill="Green" MinHeight="100" Margin="0,5,0,0"
             RelativePanel.Below="RedRect"
             RelativePanel.AlignLeftWith="RedRect"
             RelativePanel.AlignRightWith="BlueRect"/>
  <Rectangle Fill="Yellow" MinHeight="100"
             RelativePanel.Below="GreenRect"
             RelativePanel.AlignLeftWith="BlueRect"
             RelativePanel.AlignRightWithPanel="True"/>
</RelativePanel>
```

Layout and
<RelativePanel />
Visual States allow you to define different selectable layouts that can be applied to your UI.
Visual states based on WindowSize – Pre UWP

```csharp
public MainPage()
{
    this.InitializeComponent();
    this.SizeChanged += (s, e) =>
    {
        var state = "Default";
        if (e.NewSize.Width > 500)
            state = "Narrow";
        else if (e.NewSize.Width > 750)
            state = "Normal";
        else if (e.NewSize.Width > 1000)
            state = "Full";
        VisualStateManager.GoToState(this, state, true);
    }
}
```
Visual states with adaptive triggers (UWP)

<VisualState x:Name="VisualState500min">
  <VisualState.StateTriggers>
    <AdaptiveTrigger MinWindowWidth="720"/>
  </VisualState.StateTriggers>
</VisualState>

MinWindowHeight (Minimum height for state)
MinWindowWidth (Minimum width for state)
Responsive UI with `<VisualState />`
Windows 10 on Raspberry Pi 2 Model B

Low cost (less than 300 DKK)

HDMI Video Out
1 GB memory & A7 900 Mhz (ARM) running Windows 10

Runs Windows 10!

Low hardware specs

Limited resources

Can only run one app at a time

Don't do this at home

HD video

Lots of pictures

Huge lists without item virtualization
bool isHardwareButtonsAPIPresent =
Windows.Foundation.Metadata.ApiInformation
 .IsTypePresent("Windows.Phone.UI.Input.HardwareButtons");

Windows.UI.Core.SystemNavigationManager.GetForCurrentView().BackRequested +=
 TestView_BackRequested;

var qualifiers = ResourceContext.GetForCurrentView().QualifierValues;
string currentDeviceFamily;
qualifiers.TryGetValue("DeviceFamily", out currentDeviceFamily);
// currentDeviceFamily == "IoT"
Device family specific VisualState trigger

By building a simple custom trigger we can get rid of the code behind check.
More UI logic contained in XAML
Custom trigger & Raspberry Pi
Projection API & Continuum
Very powerful for product companies - high enterprise value – you’re already there with adaptive UI
Projection Manager API

Notes

One hundred percent of known life on Earth is made up of complex and varied carbon chains. Due to its unique bonding properties, carbon has the ability to form long chains of molecules. These long chains comprise the essential building blocks of life, namely amino acids and DNA. Naturally occurring pure carbon takes several forms, including coal, graphite, and diamond. Different conditions lead to each form. For instance, diamonds require a...
private async void StartProjecting_Click(object sender, RoutedEventArgs e)
{
    // Check whether there is already active connection to an external display
    if (ProjectionManager.DisPlayAvailable)
    {
        int thisViewId;
        thisViewId = ApplicationView.GetForCurrentView().Id;

        var thisDispatcher = Window.Current.Dispatcher;
        await CoreApplication.CreateNewView().Dispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
        {
            // Display the page in the view. Not visible until "StartProjectionAsync" called
            var rootFrame = new Frame();
            rootFrame.Navigate(typeof(ProjectionViewPage), initData);
            Window.Current.Content = rootFrame;
            Window.Current.Activate();
        });

        // Show the view on a second display
        await ProjectionManager.StartProjectingAsync(rootPage.Id, thisViewId);
    }
}

private async void StopProjecting_Click(object sender, RoutedEventArgs e)
{
    await ProjectionManager.StopProjectingAsync(rootPage.ProjectionViewPageControl.Id, thisViewId);
}
Wrapping up

UWP platform – one binary to rule them all feat. device families

Responsive UI is king

Really really high reusability

Bonus: One unified store for all apps with all device families
Click ‘engage’ to rate sessions and ask questions.