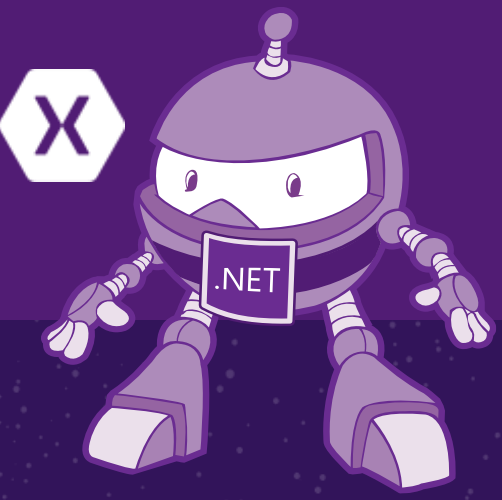

.NET Conf
"Focus on Xamarin"
.NET
focus.dotnetconf.net

Developing Dual-screen Experiences with Xamarin

Craig Dunn & Guy Merin



Surface Neo

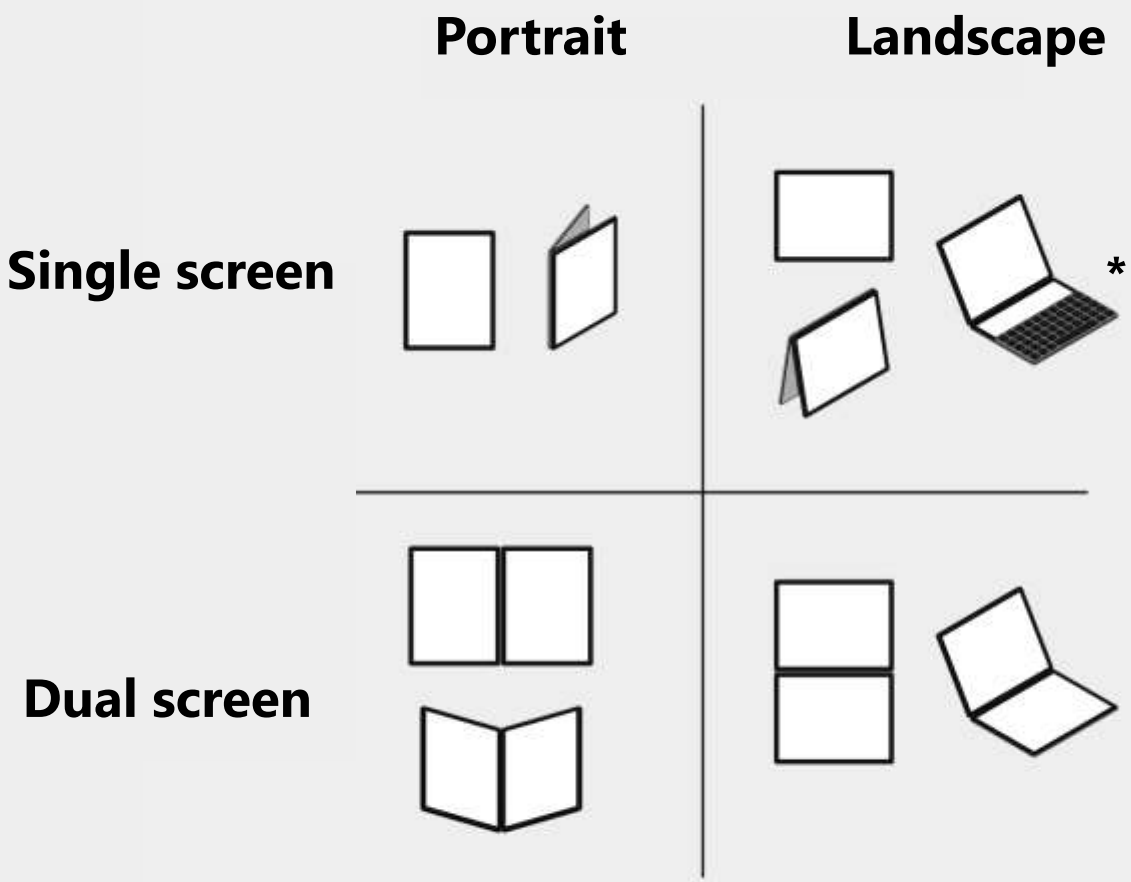
Surface Duo

Surface Neo
Windows 10X

Surface Duo
Android


10
55
Owen Small

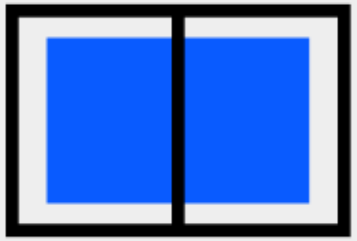
Dual screen postures



*** Only the Surface Neo has this keyboard**

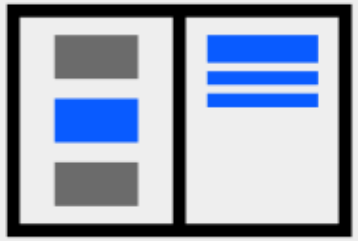
When app covers both screens, it is "spanned"

We call the space between the screens the "seam" or hinge



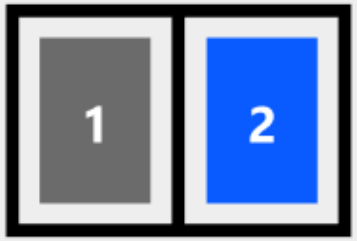
Extended Canvas

Expand content across the seam



Master-Detail

Drill down one layer deeper into content

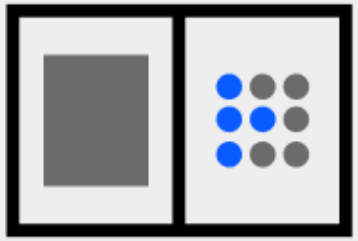


Two Page

Document-oriented, made for reading

Dual View

Alternate or transient view of the same info



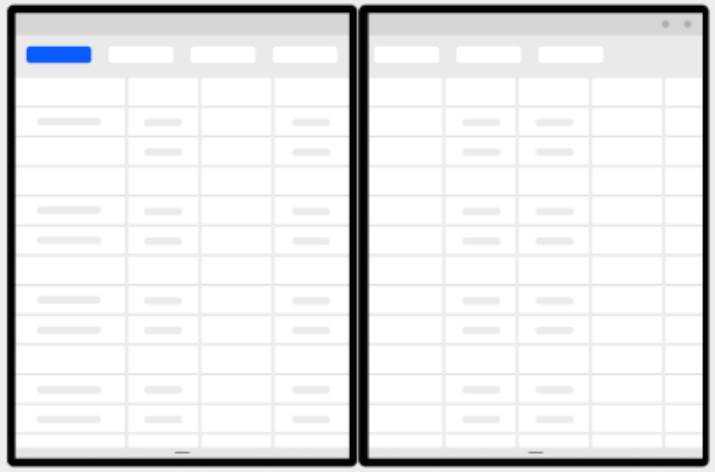
Companion Pane

Supporting content or smart suggestions

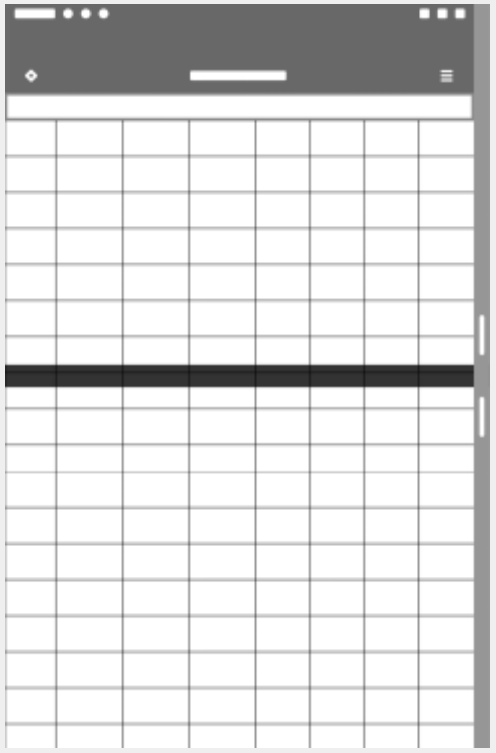
Extend Canvas

Single Screen

Spanned Wide



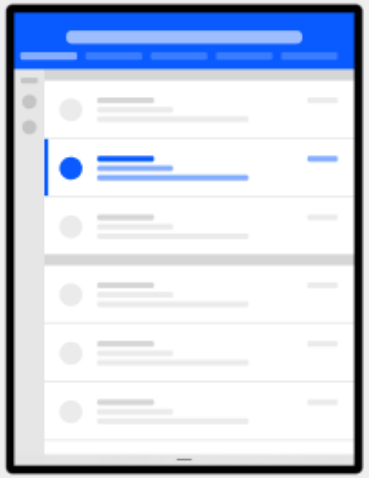
Spanned Tall

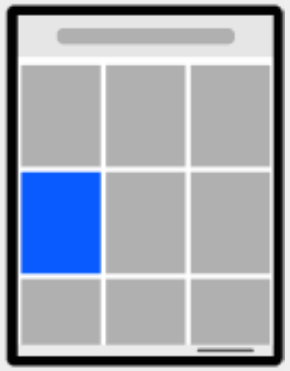


Maps is another good example

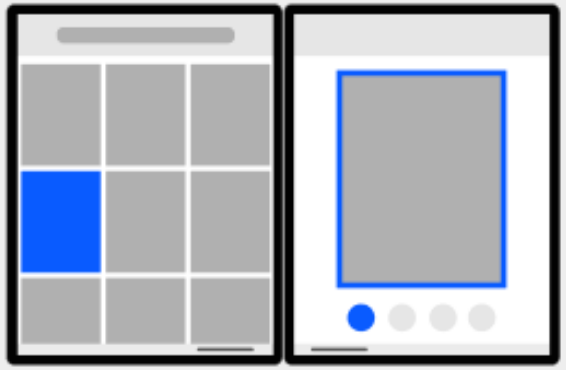
Master Detail

Single Screen





Spanned Wide

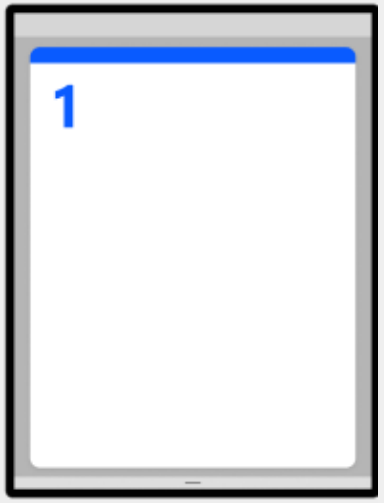


Spanned Tall

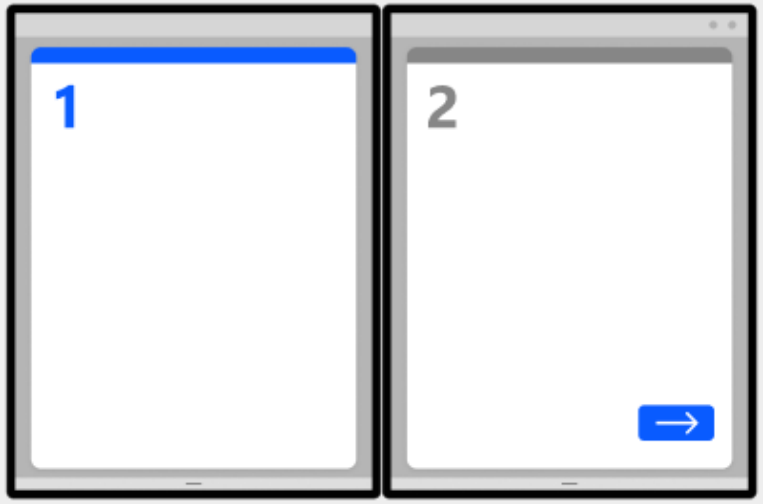
Detail only

Two Page View

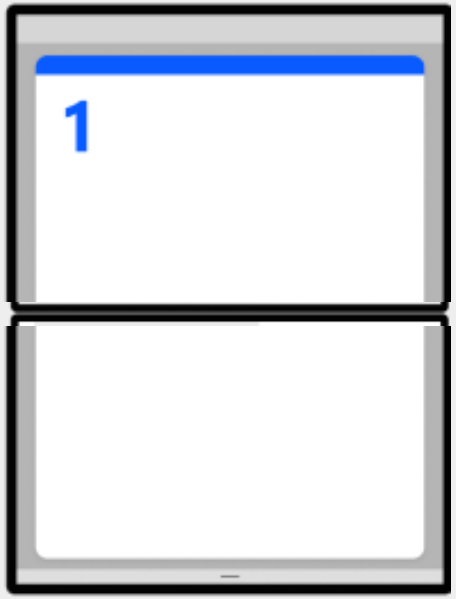
Single Screen



Spanned Wide



Spanned Tall



Dual View

Single Screen

Spanned Wide

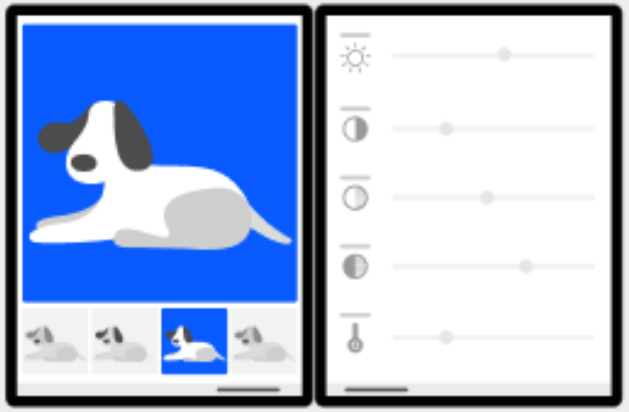
Spanned Tall

Two views of content (including editing)

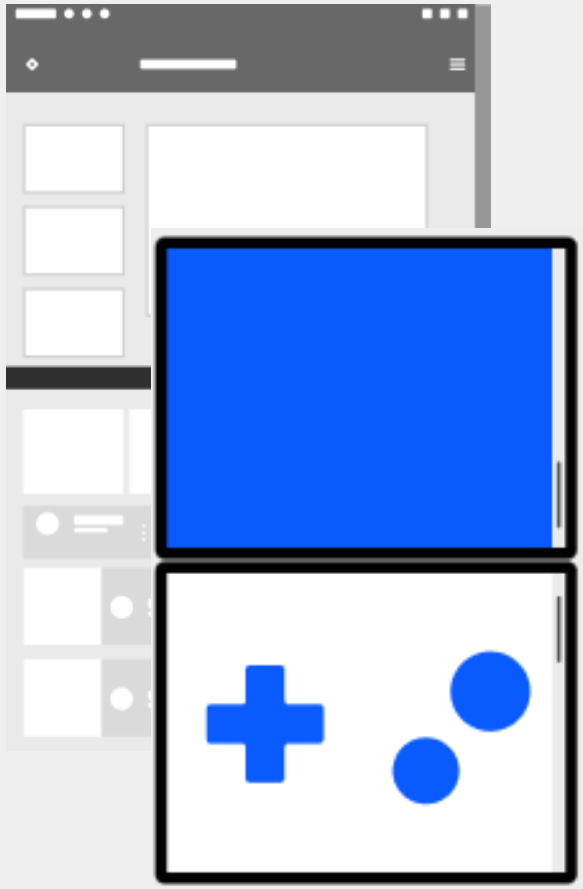
Companion Pane

Single Screen

Spanned Wide



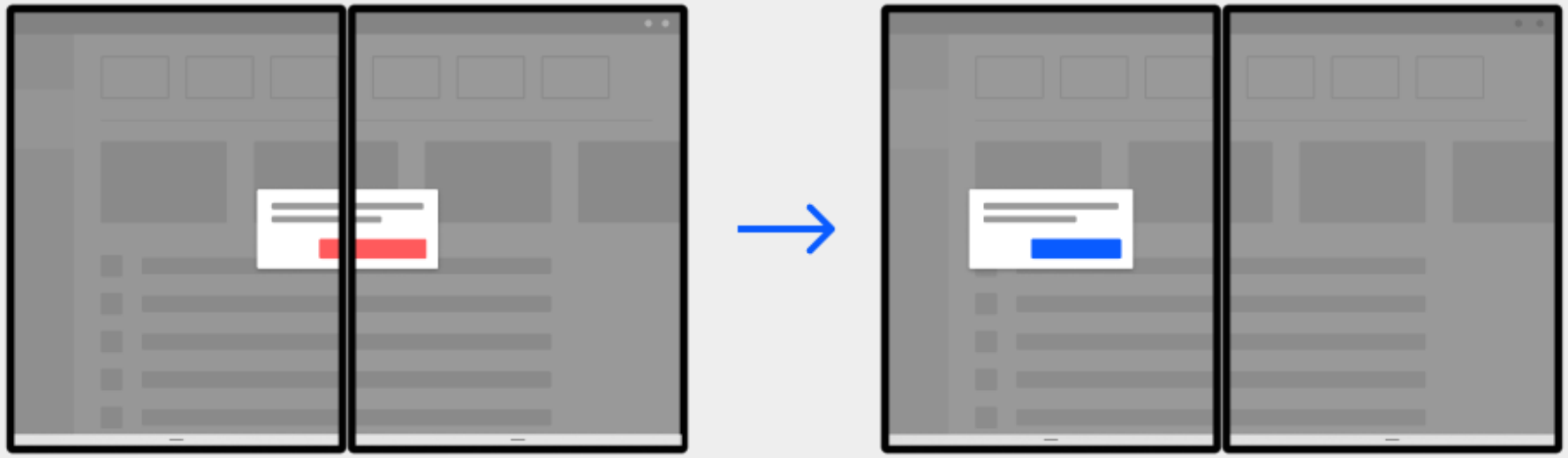
Spanned Tall



Complimentary content
(additional controls & gaming)

Don’t

Do

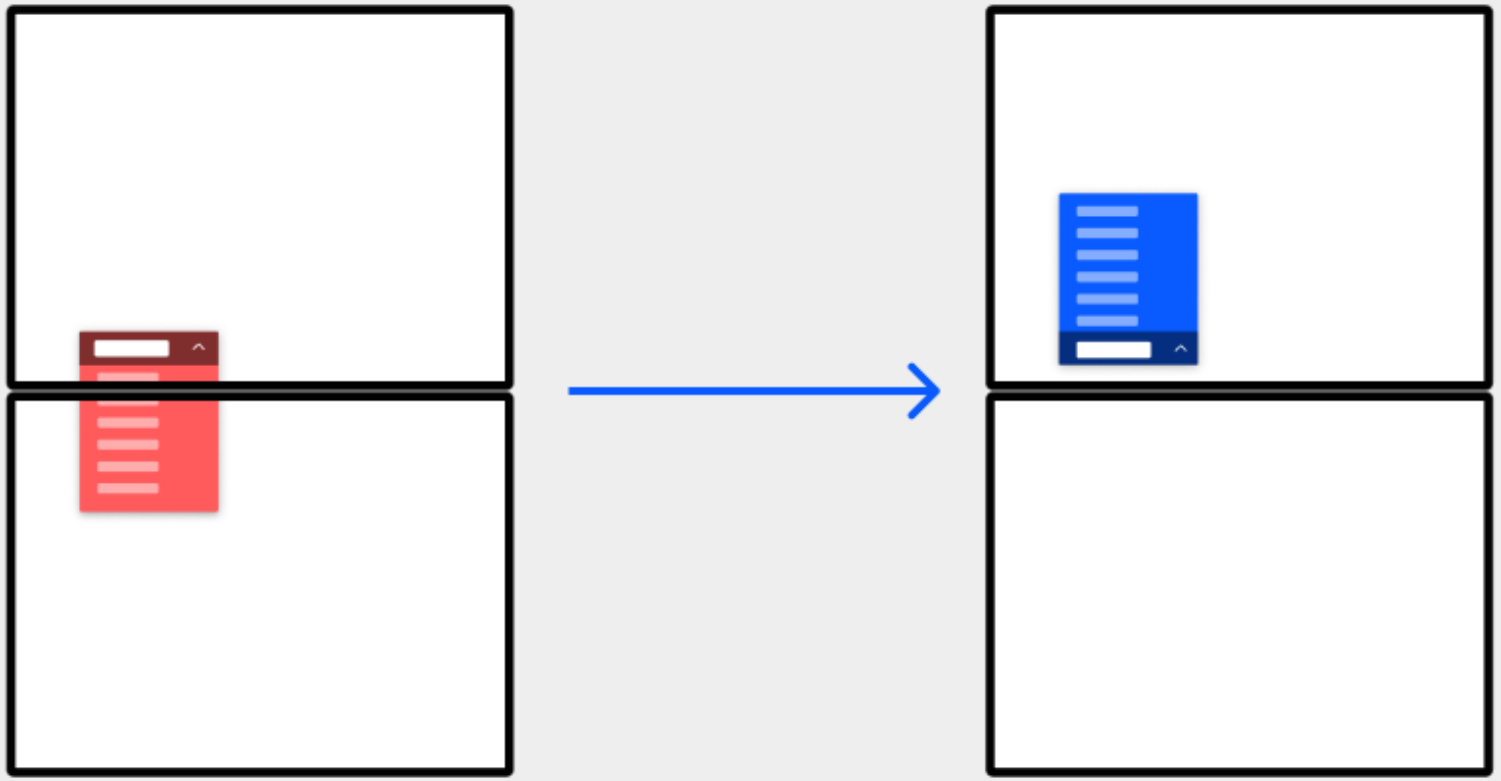


Don’t

Do

Don’t

Do



Don’t

Do

How can Xamarin help?

- Cross-platform!
  - Runs on Android and UWP for Windows 10X
- Helper methods
  - Where is the seam?
  - What angle is the device hinge, and has it changed?

  You can do all the calculations to lay out your own views
- New layout control
  - Automatically place content on two screens if they're present, otherwise easily support single screen (for folded devices and support for regular phones)
  - Update configuration depending on what the app needs

Introducing: Xamarin.Forms.DualScreen

.NET

DualScreenInfo

```
SpanningBounds // Rectangle[] of screens if spanned
HingeBounds // Mask area as a Rectangle
IsLandscape // Dual-screen aware landscape mode
SpanMode // SinglePane, Wide, or Tall

PropertyChanged // Listen to changes

GetHingeAngleAsync() // determine folded state
```

TwoPaneView

```
<dualscreen:TwoPaneView
    <dualscreen:TwoPaneView.Pane1>
        <local:SearchVideosView />
    </dualscreen:TwoPaneView.Pane1>
    <dualscreen:TwoPaneView.Pane2>
        <local:BrowseVideosView />
    </dualscreen:TwoPaneView.Pane2>
</dualscreen:TwoPaneView>
```

TwoPaneView

```
<dualscreen:TwoPaneView x:Name="twoPaneView"
        Mode="{Binding TwoPaneViewMode, Mode=OneWayToSource}"
        PanePriority="Pane1"
        TallModeConfiguration="{Binding TallModeConfiguration}"
        WideModeConfiguration="{Binding WideModeConfiguration}">
        Pane1Length="{Binding Pane1Length}"
        Pane2Length="{Binding Pane2Length}"
        MinTallModeHeight="{Binding MinTallModeHeight}"
        MinWideModeWidth="{Binding MinWideModeWidth}"
        <dualscreen:TwoPaneView.Pane1>
            <local:SearchVideosView/>
        </dualscreen:TwoPaneView.Pane1>
        <dualscreen:TwoPaneView.Pane2>
            <local:BrowseVideosView/>
        </dualscreen:TwoPaneView.Pane2>
    </dualscreen:TwoPaneView>
```

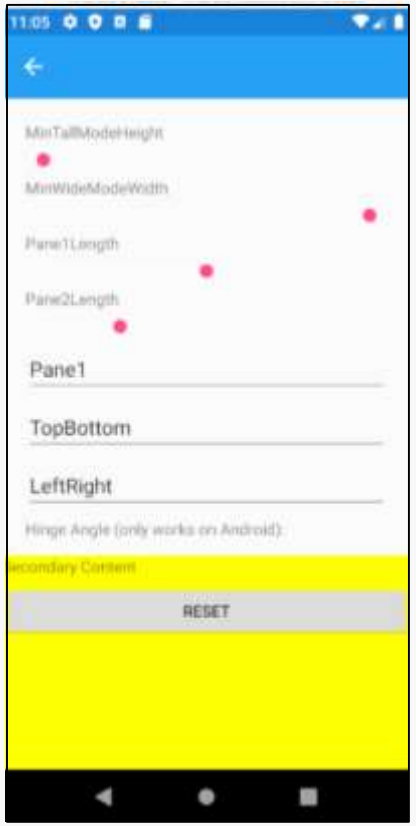
Relevant in single-screen only

Demo: DualScreenDemos

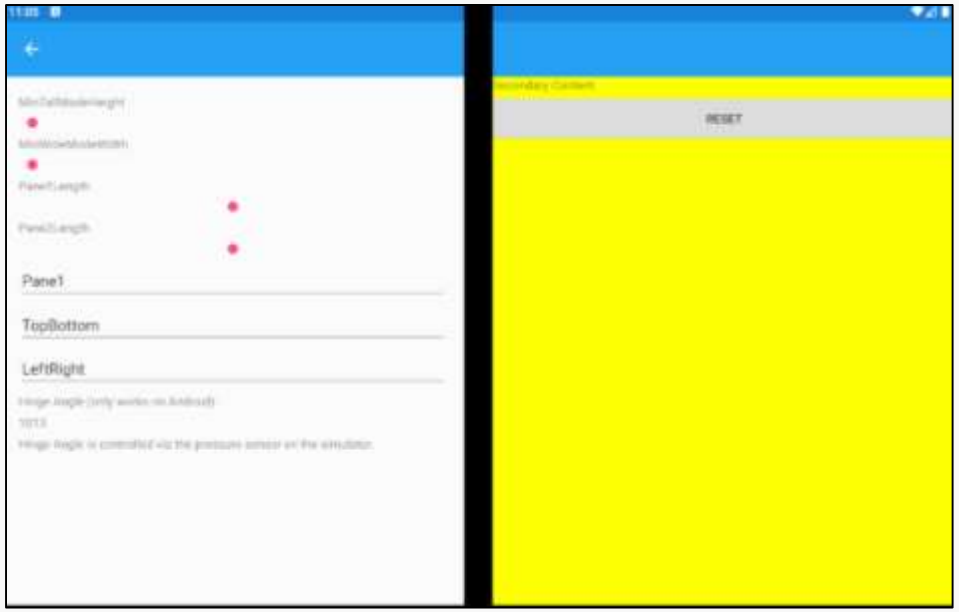
.NET

TwoPaneView Playground

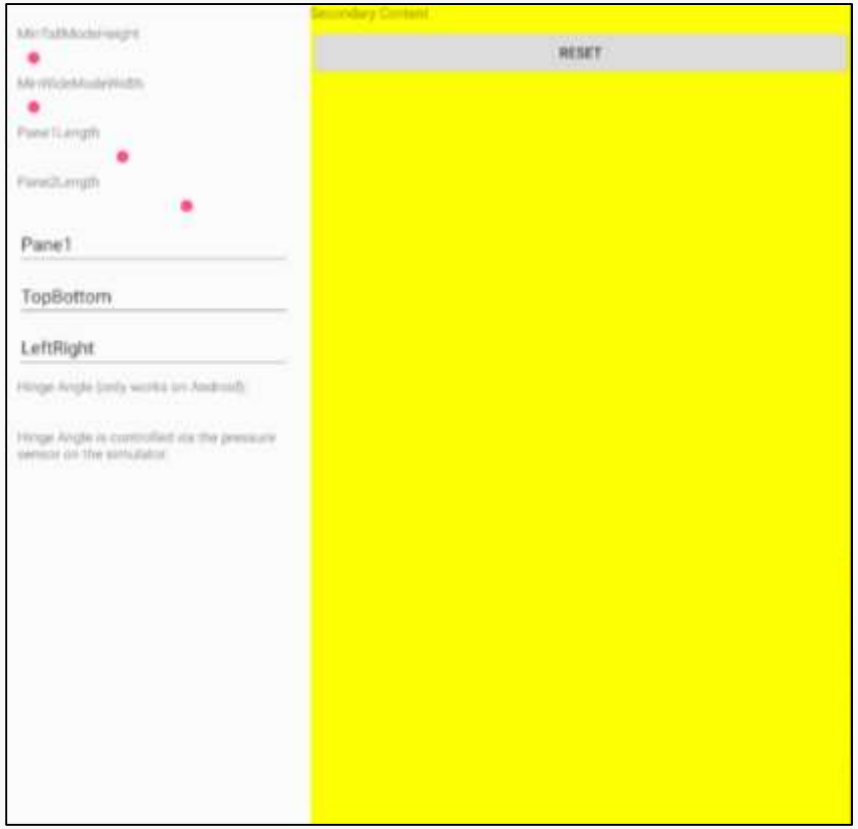
Nexus (single screen)



Surface Duo



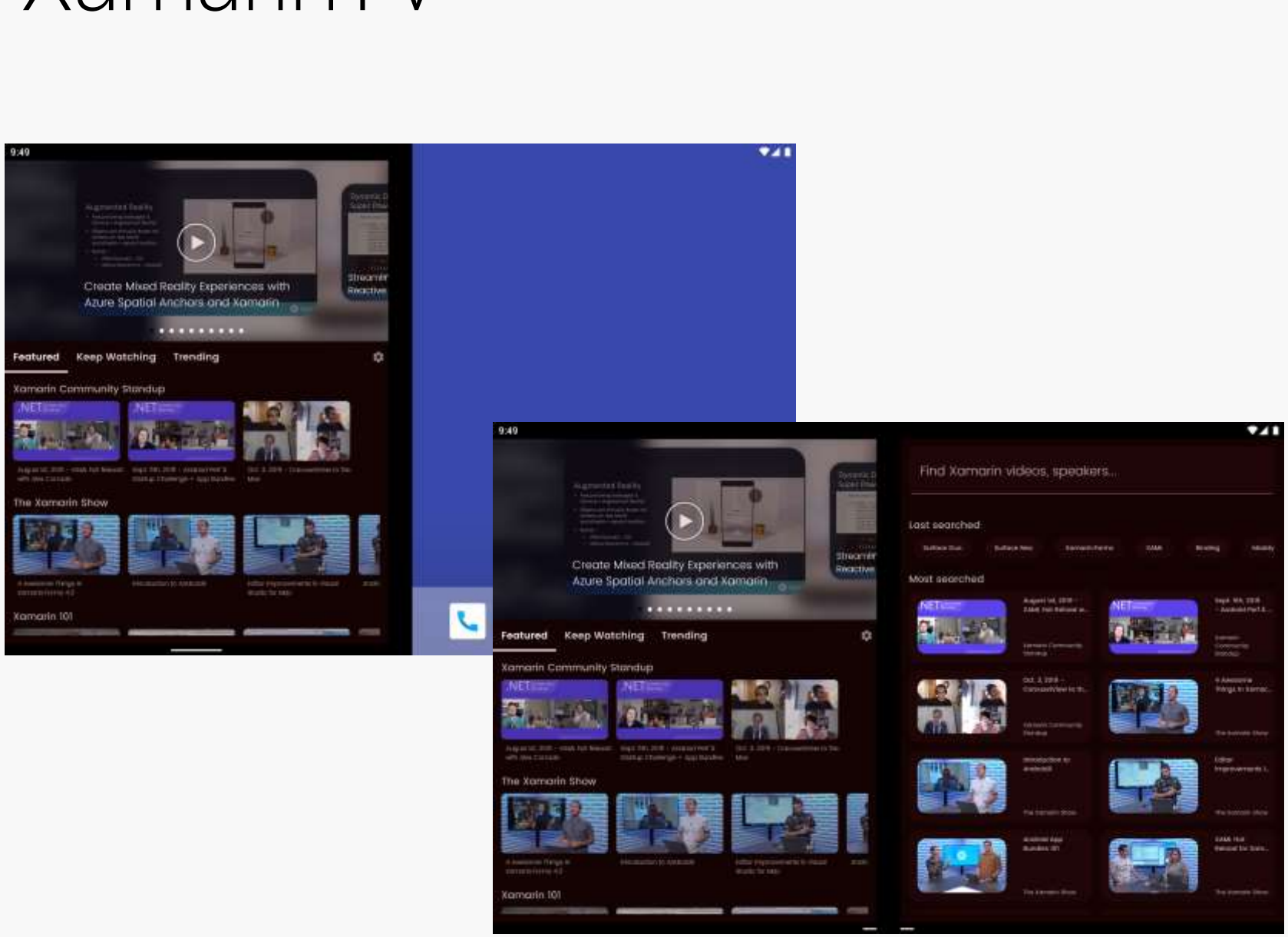
Larger screen device

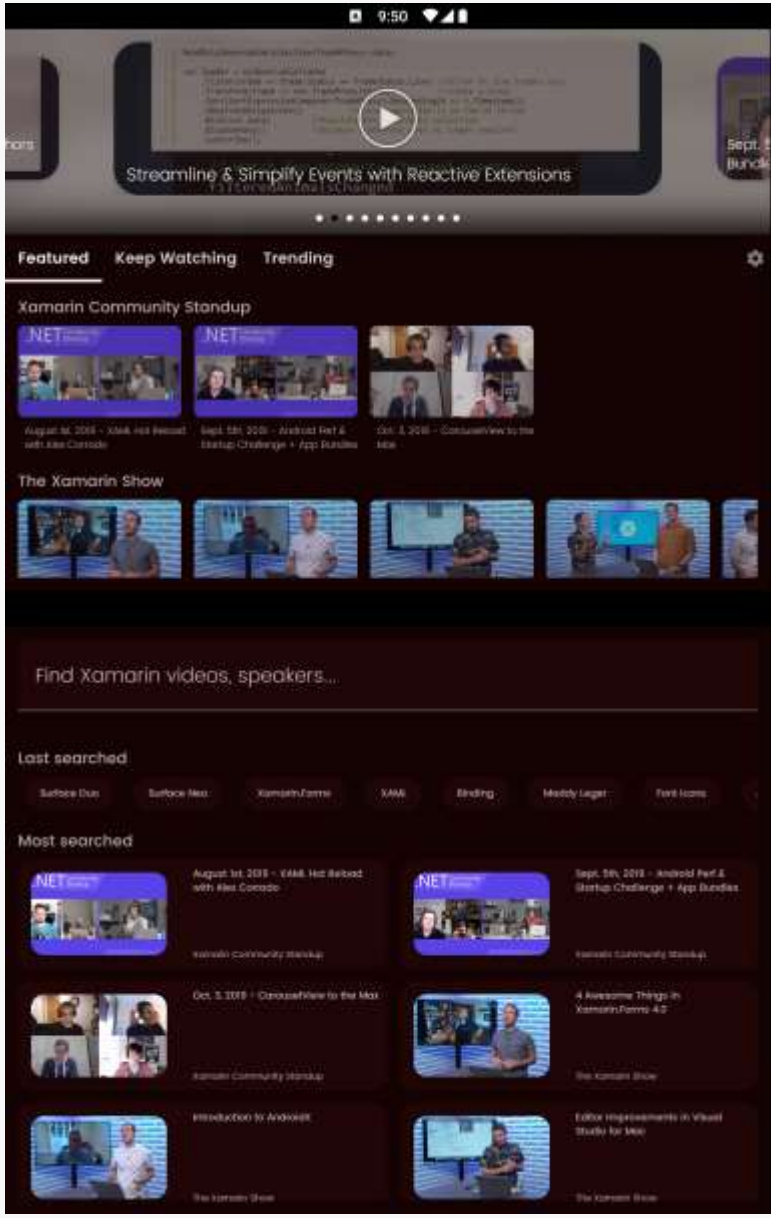


Demo: Xamarin TV

.NET

XamarinTV


9:49
Create Mixed Reality Experiences with Azure Spatial Anchors and Xamarin
Featured
Keep Watching
Trending
Xamarin Community Standup
The Xamarin Show
Xamarin 101
Find Xamarin videos, speakers...
Last searched
Most searched


9:50
Streamline & Simplify Events with Reactive Extensions
Featured
Keep Watching
Trending
Xamarin Community Standup
The Xamarin Show
Find Xamarin videos, speakers...
Last searched
Most searched

Get your app ready for dual-screen devices

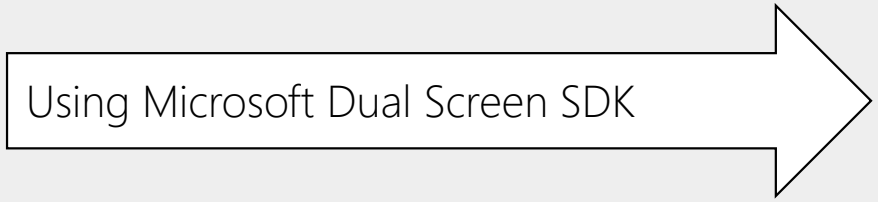
1. Test your App

It Just Works - Running on one of the screens

2. Make incremental changes

It Works Better - On dual-screens side by side

3. Embrace new features



Enhanced – Take advantage of Microsoft services and device postures and layouts

Resources

Documentation and emulator install links

- docs.microsoft.com/dual-screen
- docs.microsoft.com/xamarin/xamarin-forms/app-fundamentals/dual-screen

Samples

- github.com/microsoft/surface-duo-sdk-xamarin-samples
- github.com/xamarin/xamarin-forms-samples/tree/master/UserInterface/DualScreenDemos

Blog

- devblogs.microsoft.com/xamarin/xamarin-goes-dual-screen
- devblogs.microsoft.com/surface-duo

Reach out on Twitter

- Guy Merin - @gmerin
- Craig Dunn - @conceptdev

Ask your questions LIVE on Twitter

#dotNETConf