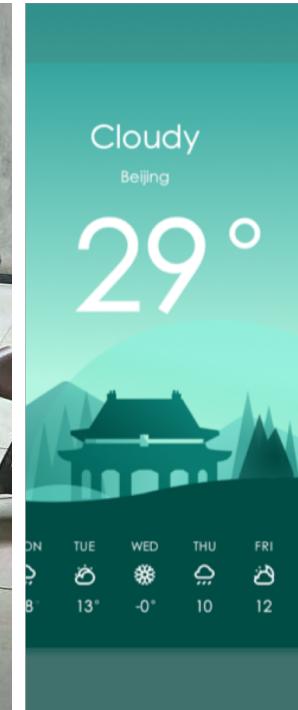


# User Interface Design

*The basics in Xamarin.Forms*

**Kym Phillipps**

Senior Content Developer - Microsoft



***Humans have an innate attraction to beautiful design...***

---

# 3.3

***Million Android apps on Google Play***

Source: <http://www.businessofapps.com/guide/app-stores-list/>

2.2

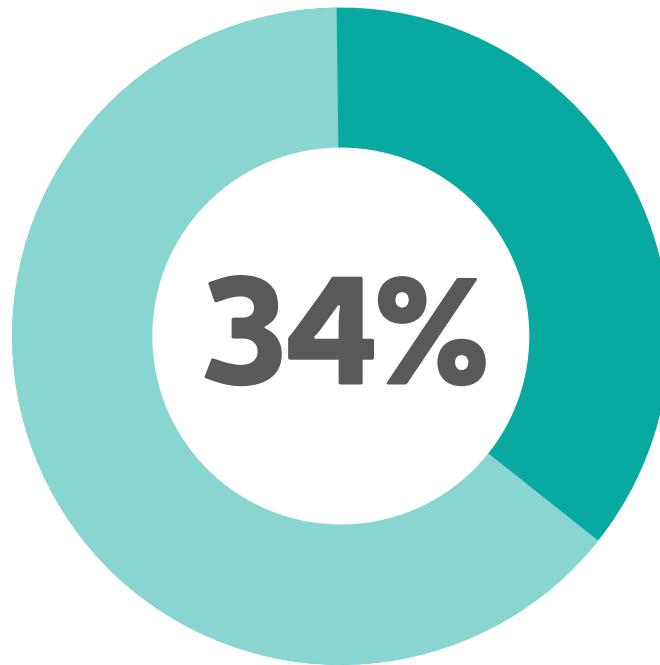
***Million iOS apps on the App Store***

Source: <http://www.businessofapps.com/guide/app-stores-list/>



***of apps aren't used more than once***

Source: Google Study “25 Principles of Mobile App Design”



***of apps aren't opened more than 11 times***

Source: Google Study “25 Principles of Mobile App Design”

## *Strive for App Elegance*

Our goal, as mobile developers, should be to build **useful, elegant** and **beautiful** applications

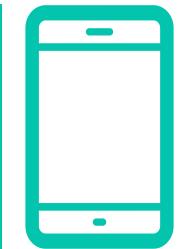
Our applications should **look and feel natural** on each platform, taking advantage of the platform's unique style and patterns

**UX deals  
with purpose**

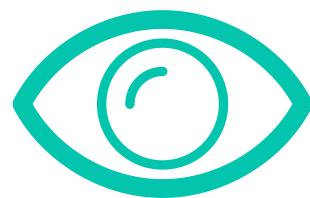
---

**UI deals with appearance  
and functionality**

# *Elements of Design*



LAYOUT



CONSISTENCY



ANIMATION



# Layouts

*Choosing the right layout*

# Layout Goals

***“Perfection is achieved when there is nothing left to take away.”***

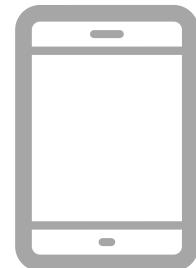
*Antoine de Saint-Exupéry*



Strive for  
Minimalism



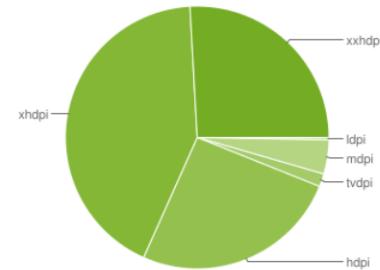
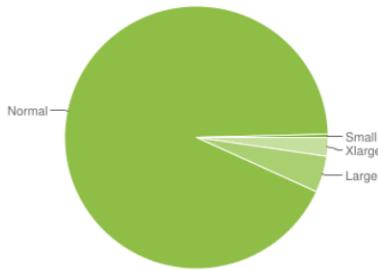
Prioritise one primary  
Action per screen



Design good  
Information  
Architecture

# Fragmentation... Oh My!

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Total
Small	0.3%				0.1%	0.4%	
Normal		0.7%	0.3%	24.7%	41.9%	25.2%	92.8%
Large	2.0%	1.3%	0.4%	0.3%	0.5%	4.5%	
Xlarge	1.5%		0.5%	0.3%		2.3%	
Total	0.3%	4.2%	1.6%	25.6%	42.5%	25.8%	

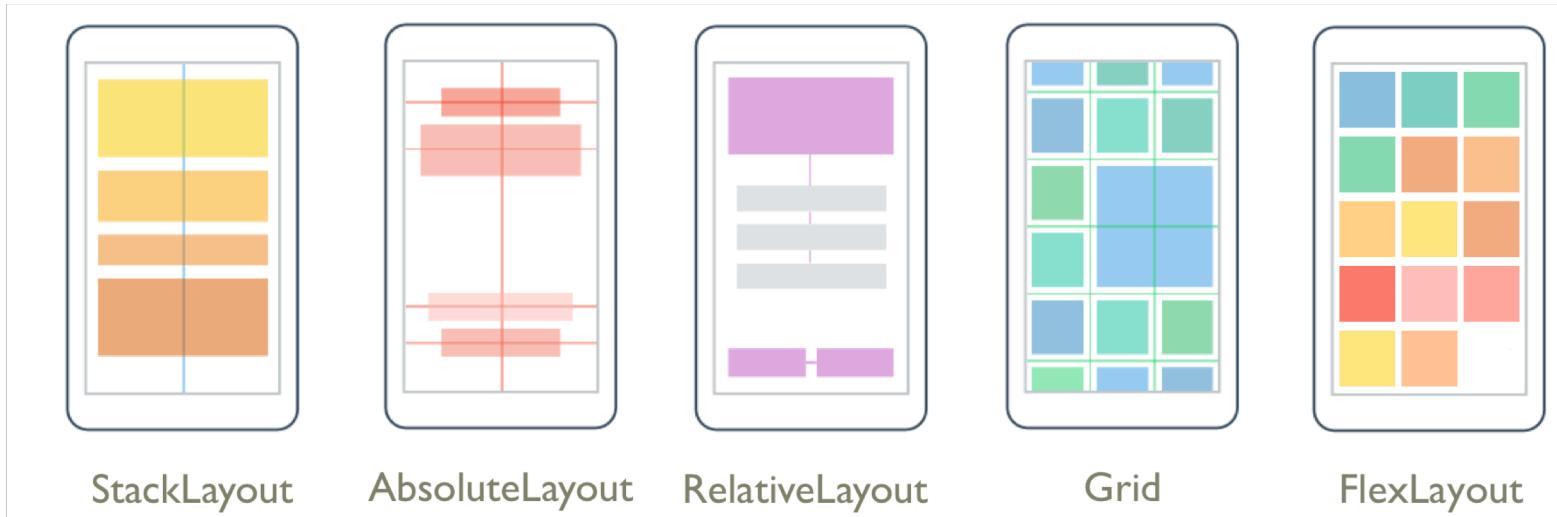


Data collected during a 7-day period ending on October 26, 2018.  
Any screen configurations with less than 0.1% distribution are not shown.

Device	Asset Resolution	PPI	Display Size
iPhone XS Max	@3x	458	6.5"
iPhone XR	@2x	326	6.1"
iPhone X, XS	@3x	458	5.8"
iPhone 6+, 6S+, 7+, 8+	@3x	401	5.5"
iPhone 6, 6S, 7, 8	@2x	326	4.7"
iPhone 5, 6SE 5, 5S, 5C, 6SE	@2x	326	4.0"
iPhone 4 4, 4S	@2x	326	3.5"
iPhone 1st, 2nd & 3rd Generation	@1x	163	3.5"
iPad Pro	@2x	264	12.9"
iPad Air / Retina iPad 1st & 2nd Generation/ 3rd & 4th	@2x	264	9.7"
iPad Mini 2nd, 3rd & 4th Generation	@2x	326	7.9"
iPad Mini 1st Generation	@1x	163	7.9"
iPad 1st & 2nd Generation	@1x	132	9.7"

DPI vs Points vs Effective Pixels?  
Xamarin.Forms just passes it down to the platform!

# *Choose the Right Layout*



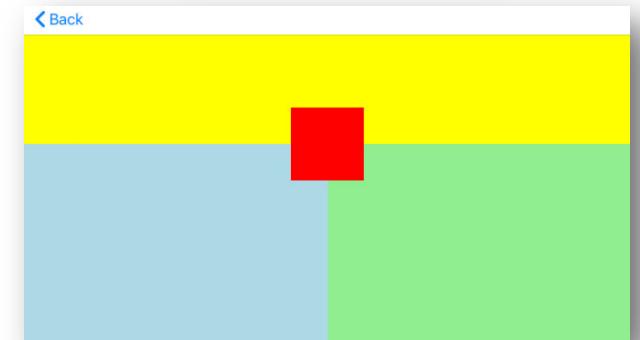
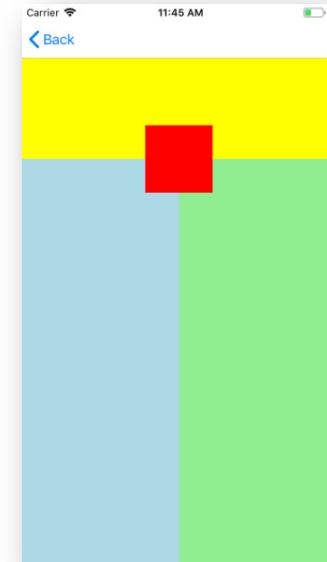
Xamarin.Forms layouts make it easier to handle responsive design

# Grids are Pretty Awesome

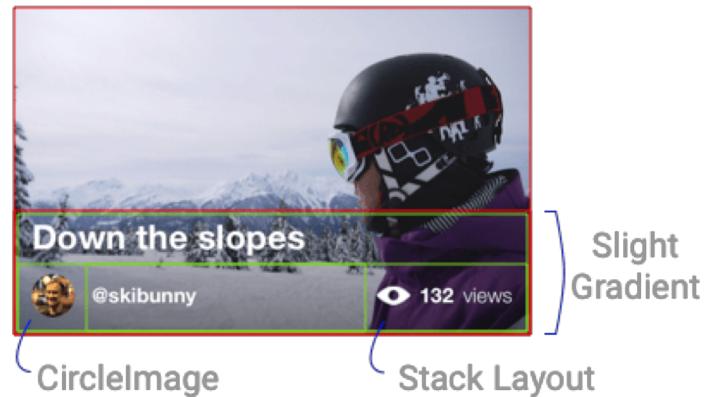
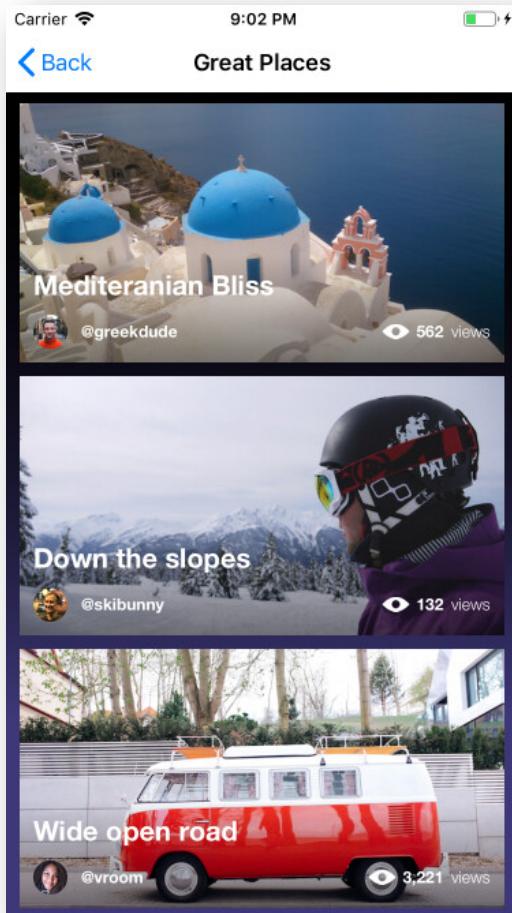
```
<Grid ColumnSpacing="0" RowSpacing="0">
    <Grid.RowDefinitions>
        <RowDefinition Height="120" />
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>

    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>

    <BoxView Grid.Row="0" Grid.ColumnSpan="2" BackgroundColor="Yellow" />
    <BoxView Grid.Row="1" Grid.Column="0" BackgroundColor="LightBlue" />
    <BoxView Grid.Row="1" Grid.Column="1" BackgroundColor="LightGreen" />
    <BoxView Grid.ColumnSpan="2" HeightRequest="80" WidthRequest="80"
            HorizontalOptions="Center" VerticalOptions="End"
            TranslationY="40" BackgroundColor="Red" />
</Grid>
```



# Overlapping Elements



# Non Rectangular Elements

```
<ContentPage  
    x:Class="Layouts.CircleImagePage"  
    xmlns="http://xamarin.com/schemas/2014/forms"  
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
    xmlns:circle="clr-namespace:ImageCircle.Forms.Plugin.Abstractions;assembly=ImageCircle.Forms.Plugin"  
    BackgroundColor="#345d89">  
  
    <ContentPage.Content>  
        <StackLayout Spacing="40" VerticalOptions="Center">  
            <Image  
                HeightRequest="200"  
                HorizontalOptions="Center"  
                Source="CookieMonster"  
                WidthRequest="200" />  
            <circle:CircleImage  
                Aspect="AspectFill"  
                BorderColor="Orange"  
                BorderThickness="5"  
                HeightRequest="200"  
                HorizontalOptions="Center"  
                Source="CookieMonster"  
                WidthRequest="200" />  
        </StackLayout>  
    </ContentPage.Content>  
</ContentPage>
```



# Non Rectangular Elements

SkiaSharp provides an excellent cross platform drawing library

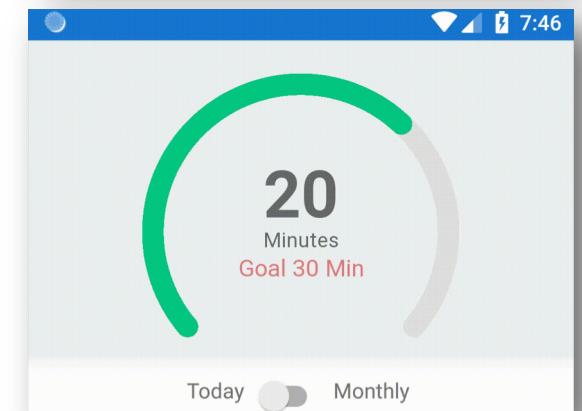
```
void OnCanvasViewPaintSurface(object sender, SKPaintSurfaceEventArgs args)
{
    // work out points for the path considering density of device
    float densityMultiplier = args.Info.Height / (float)SkiaCanvas.Height;
    var curveAmount = 40 * densityMultiplier;
    float curveStart = args.Info.Height - curveAmount;
    float curveEnd = args.Info.Height + curveAmount;

    args.Surface.Canvas.Clear();

    // create the clipping path
    SKPath path = CreatePath(curveStart, curveEnd, args.Info.Width);
    args.Surface.Canvas.ClipPath(path, antialias: true);

    // render bitmap into clipping rectangle
    using (SKPaint paint = new SKPaint())
    {
        // Create bitmap tiling
        paint.Shader = SKShader.CreateBitmap(resourceBitmap,
                                              SKShaderTileMode.Repeat,
                                              SKShaderTileMode.Repeat);

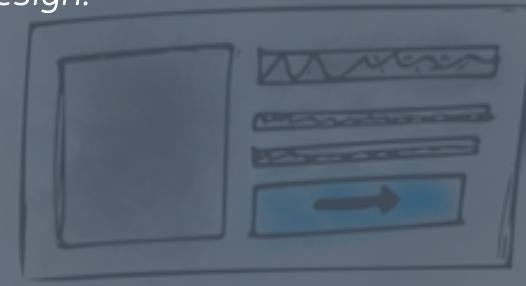
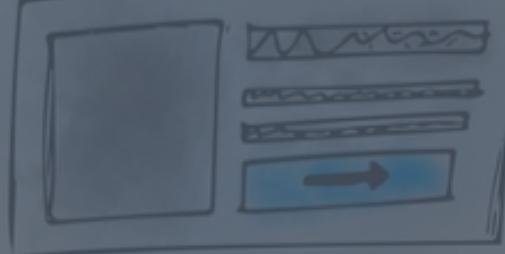
        // Draw background
        args.Surface.Canvas.DrawRect(args.Info.Rect, paint);
    }
}
```





# Consistency

*Consistent design is intuitive design.*



# *Benefits of **Consistency***

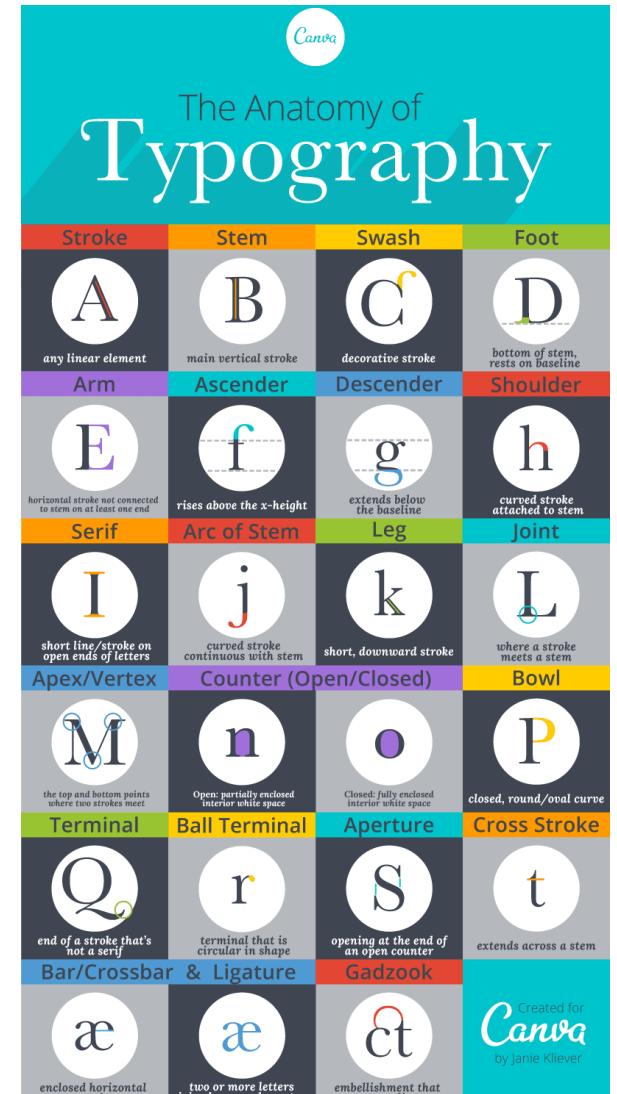
- Users will learn faster how to use your app
- Consistency eliminates confusion.
- Consistency saves money and time.
- Humans like consistency.

Fonts, sizes, buttons, labeling, colors need to be consistent across the product to keep visual consistency.

# Fonts and Typography

- Typography is a fundamental basis for design
- Fonts can convey emotion
- Critical element of branding
- Typography can make or break a design

It's both a science and an art that some people devote their entire careers to



# *Basics of Controlling Fonts*

## Key Font Properties

**FontFamily** – The font family

**FontSize** – the font size (can be a named size or a double)

**FontAttributes** – style information (e.g. Italics, Bold)

```
var about = new Label {  
    FontSize = Device.GetNamedSize (NamedSize.Medium, typeof(Label)),  
    FontAttributes = FontAttributes.Bold,  
    Text = "Medium Bold Font"  
};
```

```
<Label Text="Medium Bold Font"  
      FontSize="Medium"  
      FontAttributes="Bold"/>
```

# Fonts per Platform

```
<Label Text="Hello Forms with XAML">
    <Label.FontFamily>
        <OnPlatform x:TypeArguments="x:String">
            <On Platform="iOS" Value="MarkerFelt-Thin" />
            <On Platform="Android" Value="Lobster-Regular.ttf#Lobster-Regular" />
            <On Platform="UWP" Value="Assets/Fonts/Lobster-Regular.ttf#Lobster" />
        </OnPlatform>
    </Label.FontFamily>
</Label>
```

Or in Xamarin.Forms > 3.2

```
<Label Text="Hello Forms with XAML"
    FontFamily="{OnPlatform iOS='MarkerFelt-Thin'
                           Android='Lobster-Regular.ttf#Lobster-Regular'
                           UWP='Assets/Fonts/Lobster-Regular.ttf#Lobster'}"/>
```

# Custom Fonts

Custom fonts are platform specific

iOS

- Font goes into *Resources* folder
- BuildAction: BundleResource
- Include in *info.plist*
- Font Family: font name

Android

- Font goes into *Assets* folder
- BuildAction: AndroidResource
- fontFamily:  
fontfilename.ttf#fontfamilyname

UWP

- Font goes into *Assets* folder
- Build Action: Content
- fontFamily:  
fontfilename.ttf#fontfamilyname

Gotchas

- The name of the font is not always the same as the filename

# *Consider Icon Fonts*

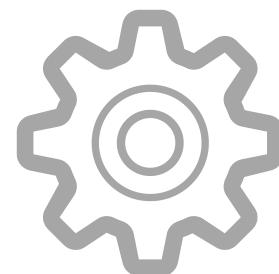
There are some great icon font libraries which web developers have been using for years. FontAwesome is one of the best known but there are others



Resizable



Can Adjust Colours



Small Footprint

A really good option is to use the **Iconize** Plugin  
<https://github.com/jsmarcus/iconize>

# *Options for Styling*

Xamarin.Forms apps are really still native apps

Styling per platform

- Android Themes
- iOS Appearance API

Consistency across platforms

- Xamarin.Forms Style API
- CSS

# *Style Structure*

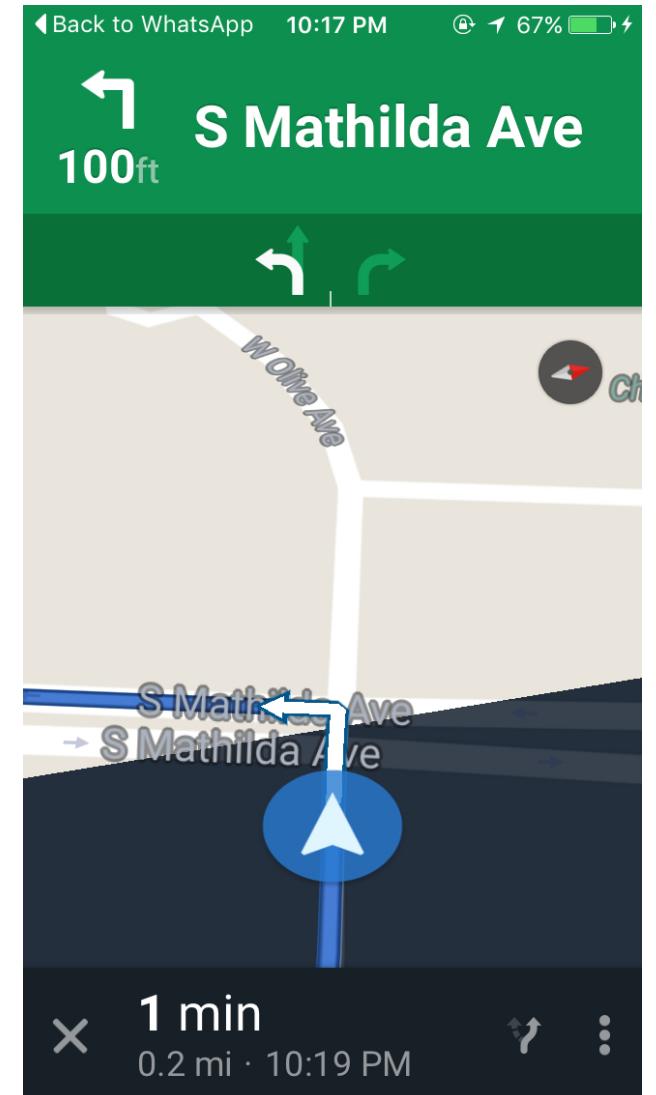
Styles get verbose. Break your XAML Styles into merged dictionaries.

- Colors
- Fonts
- Global Styles
- Literals
- Sizes
- Styles

# Selectable Themes

It is often beneficial to allow the user to select a theme they would like. Some apps even do this automatically.

Think of Google Maps which changes themes depending on the time of day





# *Animation*

*Bringing some fun to your app*

# *Animation Theory*

Animations should serve a purpose



Visual Feedback



Change Blindness



Entertain

Will the animation get annoying on the 100<sup>th</sup> use?

# *Simple Animations*

ViewExtensions has extension methods on all VisualElements

VisualElement Property	Animation Extension
TranslationX, TranslationY	TranslateTo
Scale	ScaleTo, RelScaleTo
Rotation	RotateTo, RelRotateTo,
RotationX	RotateXTo
RotationY	RotateYTo
Opacity	FadeTo

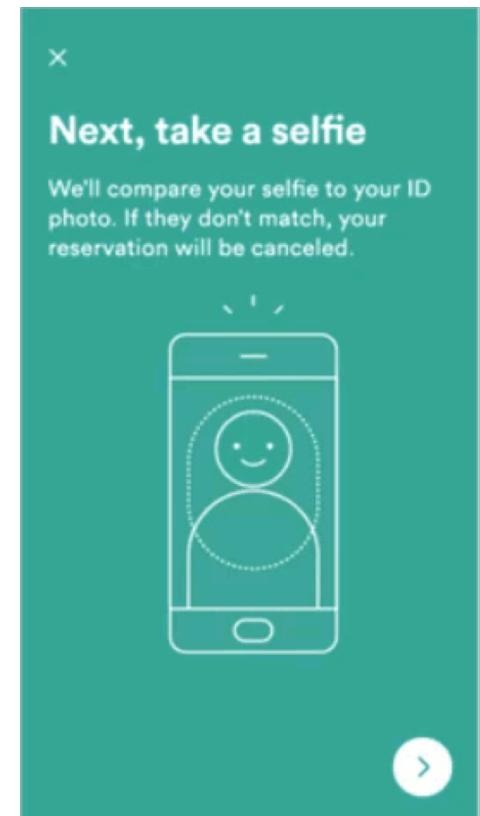
# *Advanced animation with Lottie*

For complex animations

Lightweight files

Huge community

Scalable





## Resources

*Dig into these resources for more information*

# Resources

Xamarin.Forms good looking UI Samples

<https://github.com/jsuarezruiz/xamarin-forms-goodlooking-UI>

Snppts – Xamarin Forms UI Snippets

<https://snppts.io/>

Principles of Mobile App Design – Google Study

<https://aka.ms/PrinciplesOfMobileAppDesign>

Code from this session:

<https://github.com/kphillpotts/XF-UserInterfaceTechniques>

*Thank you |  
and any questions?*

**Kym Phillipps**

kym.phillpotts@microsoft.com  
@kphillpotts