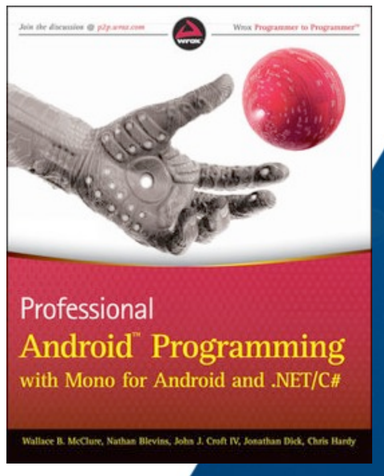
Jonathan Dick

DBA / Developer
Xamarin MVP

jondick@gmail.com
@redth

Xamarin: Native AND Cross Platform?!



NEW! Gone Mobile Podcast: ***http://GoneMobile.io***

Detroit Mobile .NET User Group
http://www.meetup.com/DetroitMobileNetUsersGroup

Wednesday, 4 September, 13

- Developing with Xamarin before Xamarin existed
- Book

Delight Developers!

Xamarin makes tools to Delight Developers
Focus on Mobile Development

Why Xamarin?

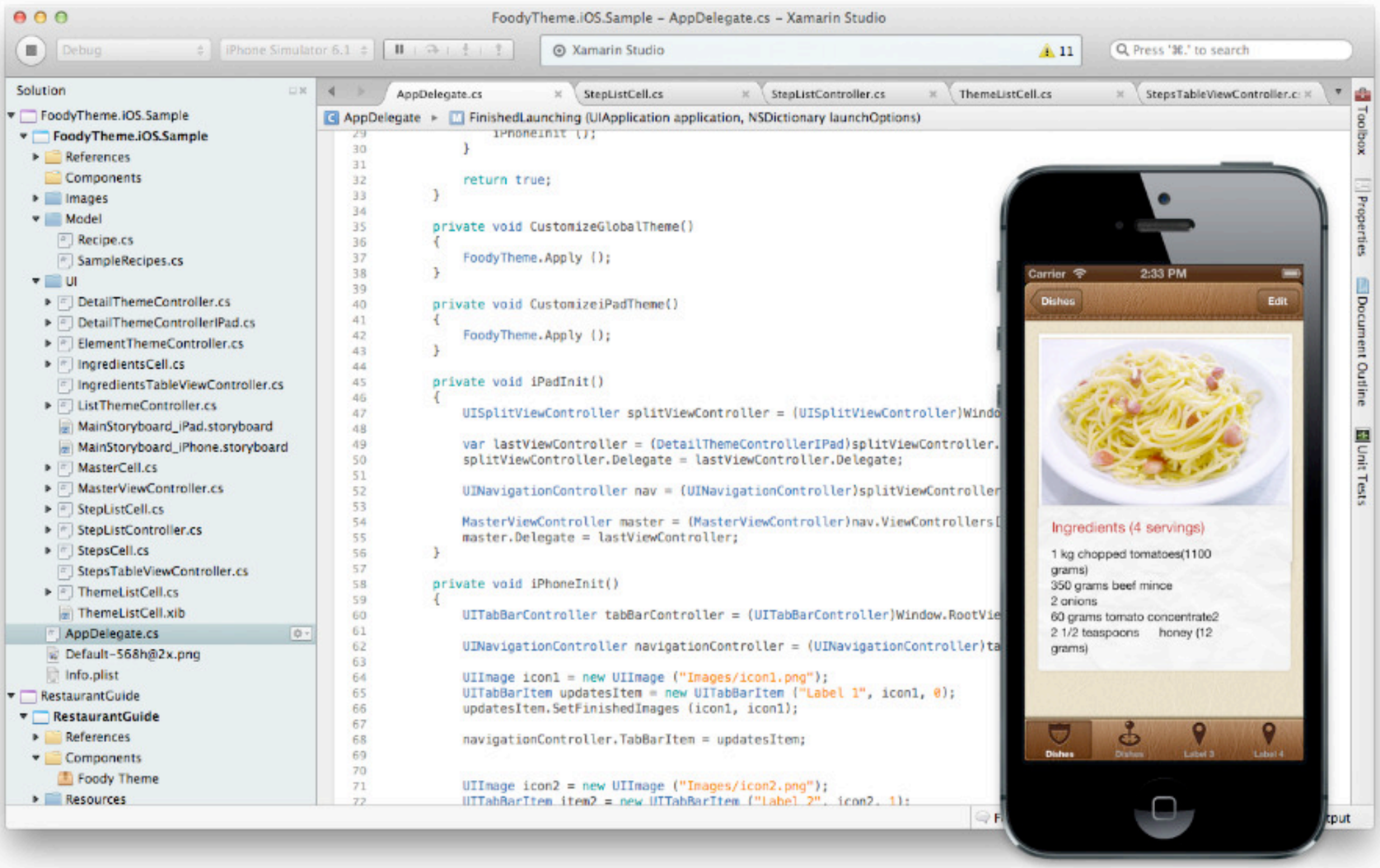
- C# all the things!
- *REAL* Native Apps
- Native UI, Native API's
- High Performance
- Code Sharing / Skill Reuse
- Debugging
- Excellent Documentation & Resources

Wednesday, 4 September, 13

You'd never know an app was made with Xamarin unless it told you

Docs - Webinars / Videos - Recipes - Sample Code

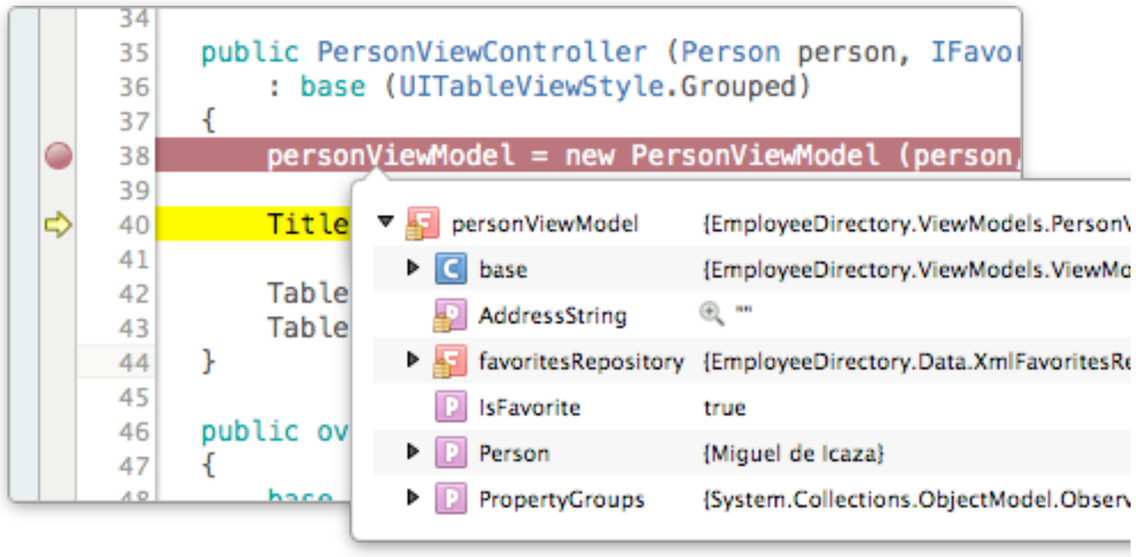
Xamarin Studio



Wednesday, 4 September, 13

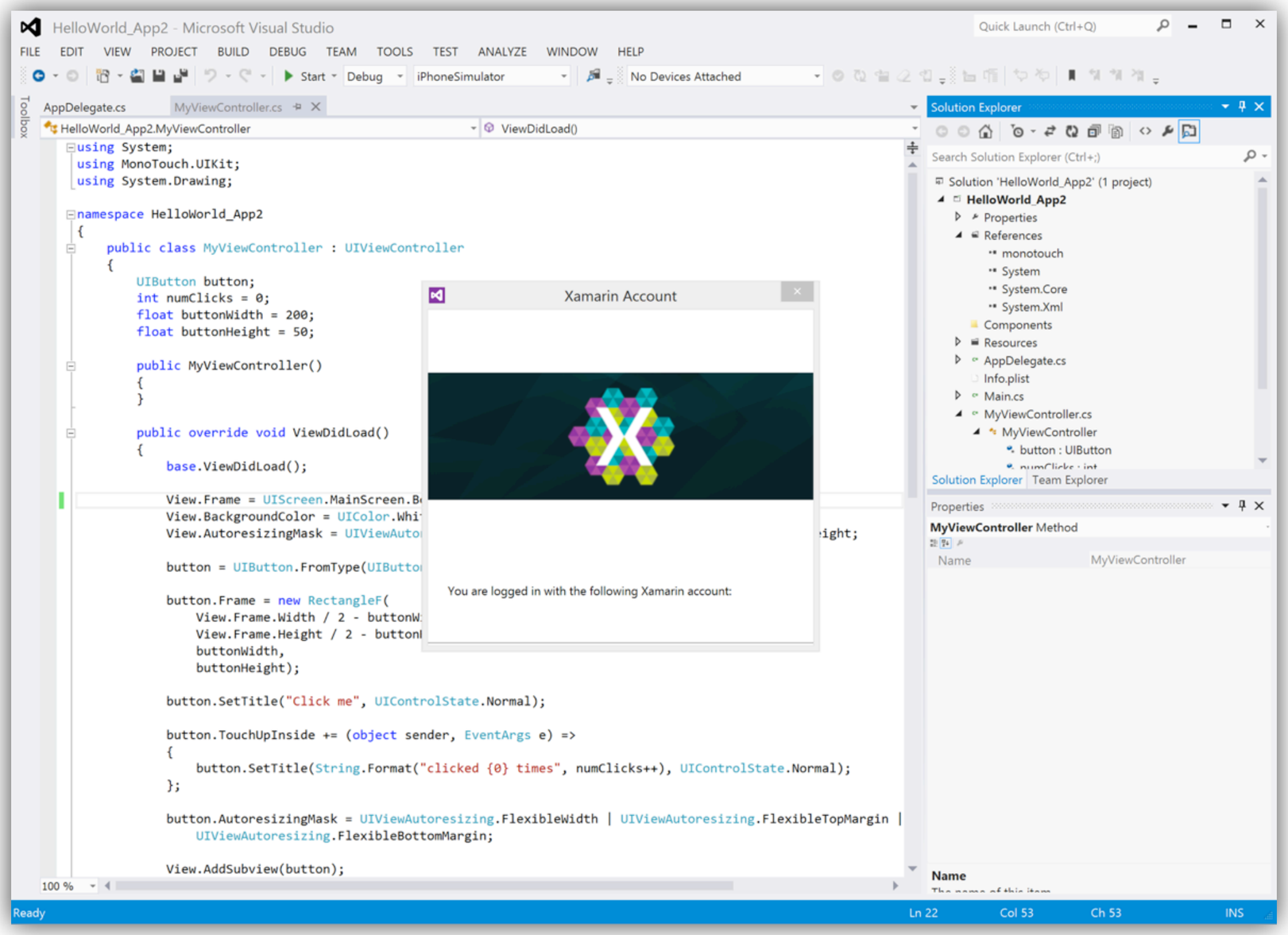
MonoDevelop++
New Design
Xamarin Sponsored huge dev effort
Shared back into open source proj
Debugging
- Simulator and real device
- breakpoints, variable inspection ON DEVICE

Xamarin Studio



MonoDevelop++
New Design
Xamarin Sponsored huge dev effort
Shared back into open source proj
Debugging
- Simulator and real device
- breakpoints, variable inspection ON DEVICE

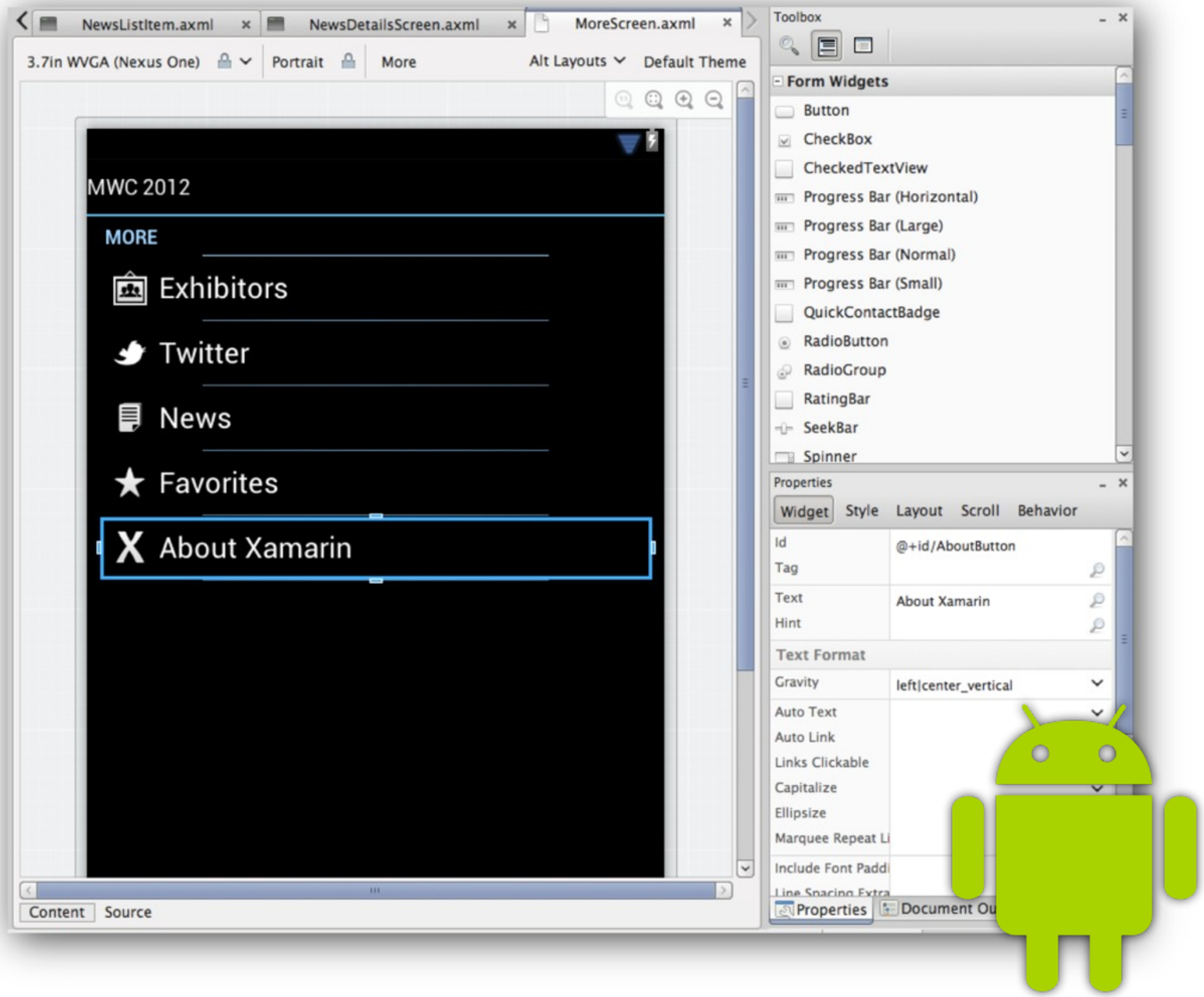
Visual Studio Support



Wednesday, 4 September, 13

Add-Ins or Android and iOS
iOS Still requires a Mac because of Apple
Android Designer (soon iOS)
All Platforms in one solution
Resharper

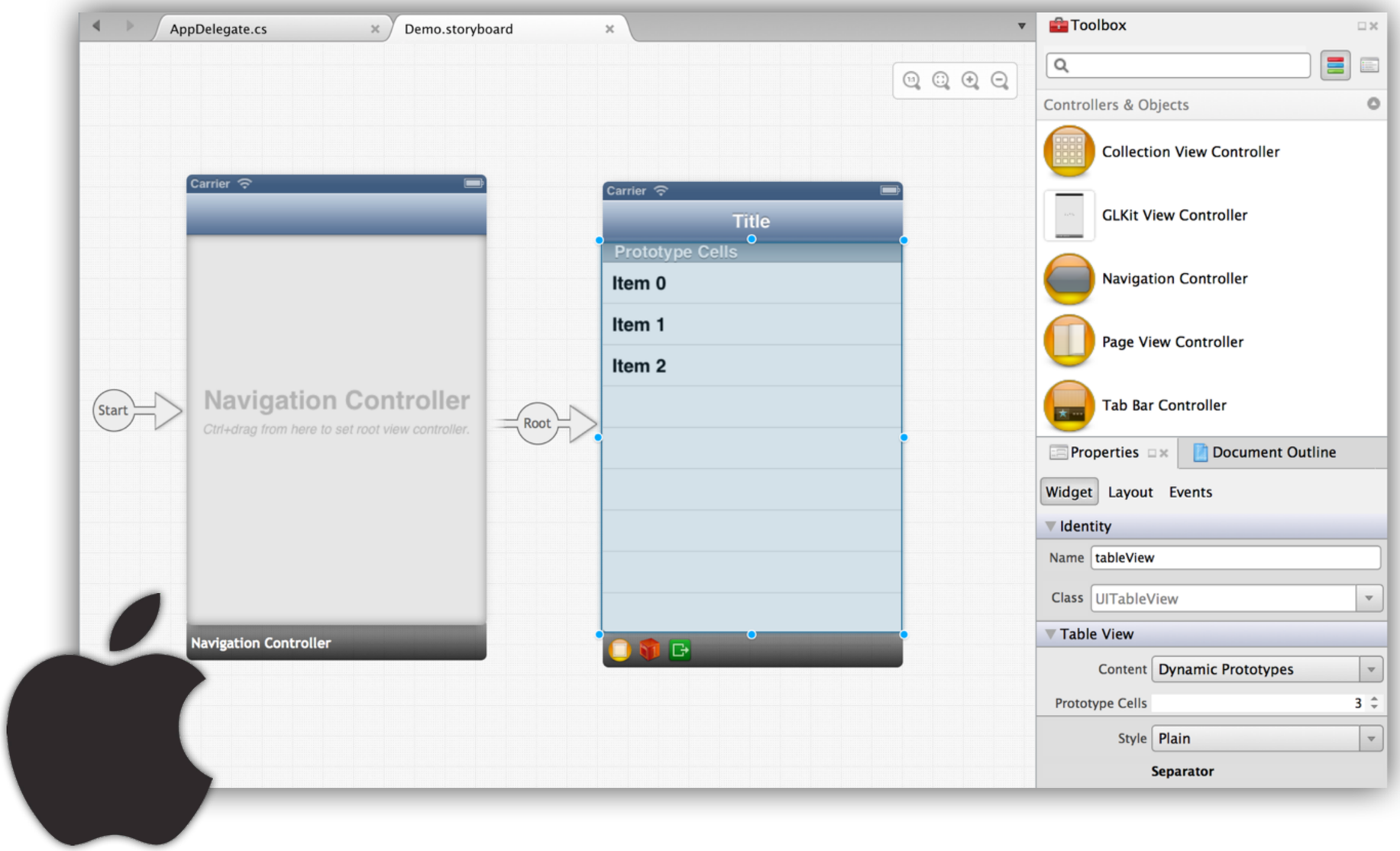
Android Designer



Best on the market

Generates pure android layout xml

Xamarin iOS Designer



Alpha
Storyboard file compatible
Renders Custom Views live designer time

Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda’s

Linq
    Query native data (collections)

Async / Wait
    Compiler organizing callbacks and completions for you
    Write synchronous-like code that executes asynchronously

Anonymous methods
Action / Funcs - Generic Delegates
Lambdas - Shorthand for anonymous methods / delegates

Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda’s

```
var hahs = from c in kittens
           where c.KittenType == KittenType.Lolcat
           && c.HazCheezburger
           orderby c.FunninessLevel
           select c;
```

Wednesday, 4 September, 13

Linq
    Query native data (collections)

Async / Wait
    Compiler organizing callbacks and completions for you
    Write synchronous-like code that executes asynchronously

Anonymous methods
Action / Funcs - Generic Delegates
Lambdas - Shorthand for anonymous methods / delegates

Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda’s

Linq
    Query native data (collections)

Async / Wait
    Compiler organizing callbacks and completions for you
    Write synchronous-like code that executes asynchronously

Anonymous methods
Action / Funcs - Generic Delegates
Lambdas - Shorthand for anonymous methods / delegates

Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda's

```
async void SpewCatz()
{
    var http = new HttpClient ();

    var resp = await http.GetAsync("http://icanhas.cheezburger.com");

    resp.EnsureSuccessStatusCode ();

    var html = await resp.Content.ReadAsStringAsync ();

    using (var stream = new StreamWriter(File.OpenWrite(@"C:\catz.html")))
        await stream.WriteAsync (html);
}
```

Wednesday, 4 September, 13

Linq
    Query native data (collections)

Async / Wait
    Compiler organizing callbacks and completions for you
    Write synchronous-like code that executes asynchronously

Anonymous methods
Action / Funcs - Generic Delegates
Lambdas - Shorthand for anonymous methods / delegates

Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda’s

Linq
    Query native data (collections)

Async / Wait
    Compiler organizing callbacks and completions for you
    Write synchronous-like code that executes asynchronously

Anonymous methods
Action / Funcs - Generic Delegates
Lambdas - Shorthand for anonymous methods / delegates

Why C# is Awesome

- Generics
- Linq
- Async / Await
- Anonymous Methods
- Action / Func
- Lambda’s

```
void buttonFindKittehs_Click(object sender, EventArgs e)
{
    FindCatz (textSearch.Text, foundCatz => {

        foreach (var kitty in foundCatz)
            Console.WriteLine(kitty.Name);

    });
}

void FindCatz(string name, Action<List<Cat>> onCatzFound)
{
    var catApi = new CatApi ();
    var results = catApi.Find (name);

    if (results != null)
    {
        var foundCatz = new List<Cat> ();

        try
        {
            foundCatz = JsonConvert.DeserializeObject<List<Cat>> (results);
        }
        catch (KittenException kEx)
        {
            Console.WriteLine (kEx);
        }

        onCatzFound (foundCatz);
    }
}
```

Wednesday, 4 September, 13

Linq
    Query native data (collections)

Async / Wait
    Compiler organizing callbacks and completions for you
    Write synchronous-like code that executes asynchronously

Anonymous methods
Action / Funcs - Generic Delegates
Lambdas - Shorthand for anonymous methods / delegates

C#... it just feels right!

Creating attributed strings

Objective-C

```
CFStringRef keys[] = {
    kCTFontAttributeName,
    kCTForegroundColorAttributeName
};

CFTypeRef bval[] = {
    cfListLineCTFontRef,
    CGColorGetConstantColor(kCGColorBlack)
};

attr = CFDictionaryCreate (kCFAllocatorDefault,
    (const void **) &keys, (const void **) &bval,
    sizeof(keys) / sizeof(keys[0]), &kCFTypeDictionaryKeyCallBac
    &kCFTypeDictionaryValueCallBacks);

astr = CFAttributedStringCreate(kCFAllocatorDefault, CFSTR("Hell
```

C# with Xamarin

```
var attrs = new CFStringAttributes {
    Font = listLineCTFont,
    ForegroundColor = UIColor.Black.CGColor
};

var astr = new NSAttributedString ("Hello World", attrs);
```


Attributed String

C#... it just feels right!

Activity Registration / Button Click

Java

```
<activity android:name=".SampleActivity" android:label="@string/app_name">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>


findViewById(R.id.button).setOnClickListener(
  new View.OnClickListener()
    {
        @Override
        public void onClick(View v)
        {
            //handle click
        }
    }
);
```

C# with Xamarin

```
[Activity(Label="App Name", MainLauncher = true)]


FindViewById<Button>(Resource.Id.button).Click += delegate
{
    //handle click
};
```

Wednesday, 4 September, 13

Java - must declare activity in separate place than activity itself

Write once, run anywhere

Xamarin's Philosophy

“The biggest mistake we made as a company was betting too much on HTML5 rather than native”

Mark Zuckerberg

- HTML5 shares some benefits - skill reuse / code reuse
- HTML5 apps should still cater to platform UX (not write once run anywhere)
- Famous Quote
- Not very informative but does speak for itself

“There are a few things that are critically missing”

“It’s not that HTML5 isn’t ready; it’s that the ecosystem doesn’t support it”

“...getting that smoothness, we felt like we needed native to really do that well.”

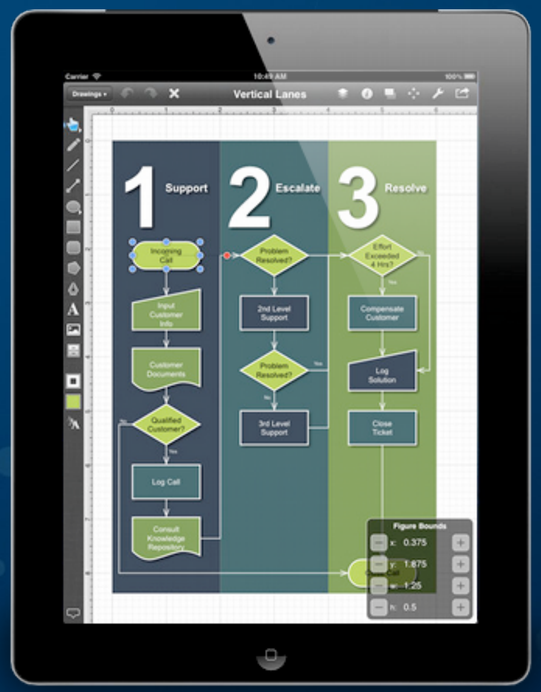
Kiran Prasad

Wednesday, 4 September, 13

- Speaks better to the reality
- What works right now?
- May be possible to use HTML5, but at what time/effort/cost compared to native
-

Do the Smart Thing

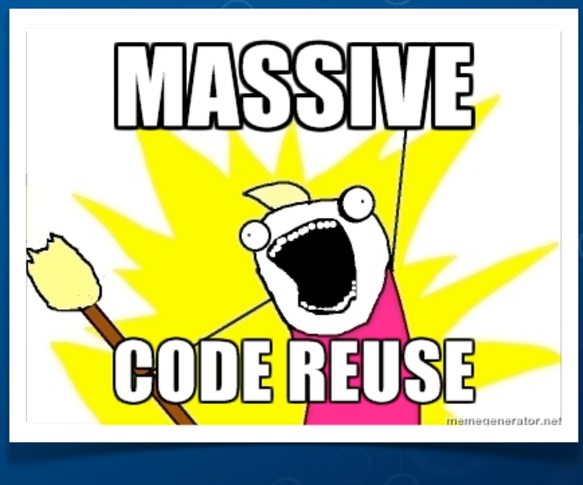
Reuse code where it makes sense, but leverage each platform's native UX



- What can be shared
  - Non UI Code
  - Core App Logic
  - Entities / Models
  - Network Access
  - File / DB Access
- Rdio
  - Share Business Logic
  - Music Syncing Engine
  - Web Services

Reuse Code!

- Typical to Reuse 40-60%
  - Web Service calls
  - Business Logic
  - SQLite / Data
- MVVM Pattern
- Games have higher reuse potential



Wednesday, 4 September, 13

Can be misleading (‘Shared code’ is so simple / concise often anyway)

Leverage Existing Awesome

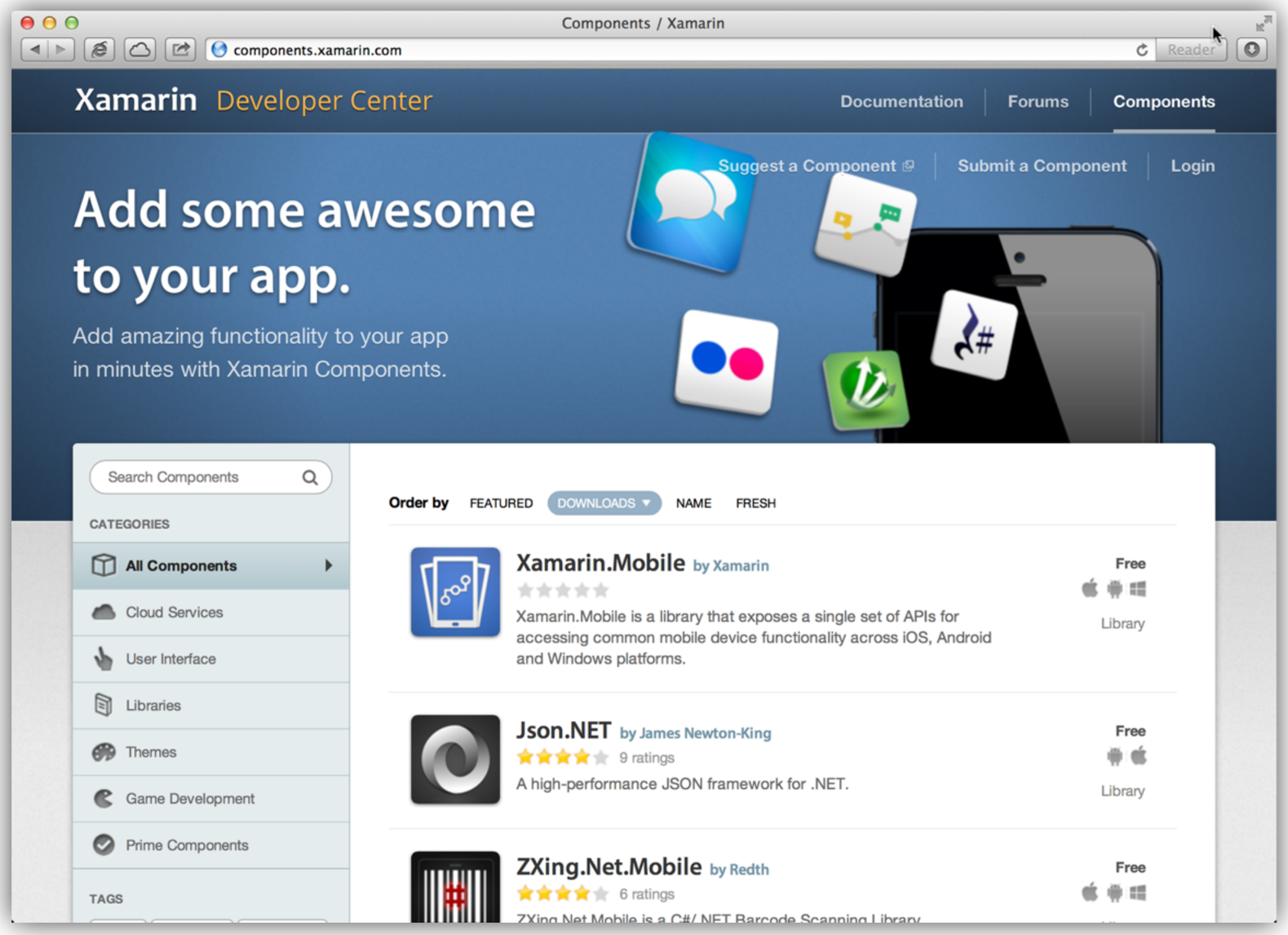
- NuGet
- Open Source

ReactiveUI

Wednesday, 4 September, 13

- NuGet - Package manager like npm or gem
- Open Source - usually very little effort to compile for Xamarin

Component Store



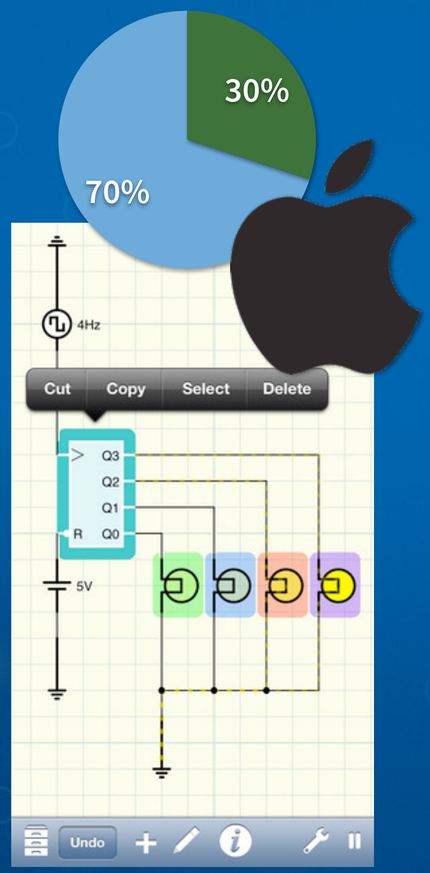
Already > 50 components

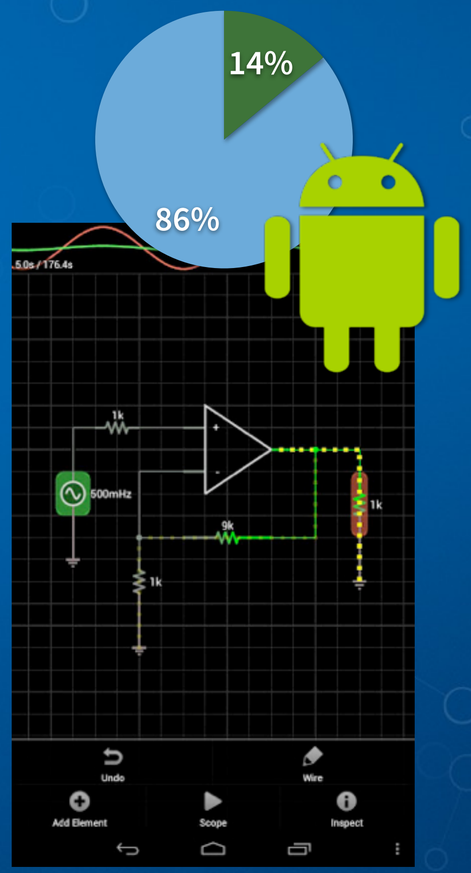
- Barcode Scanning
- Charts
- Themes
- Azure

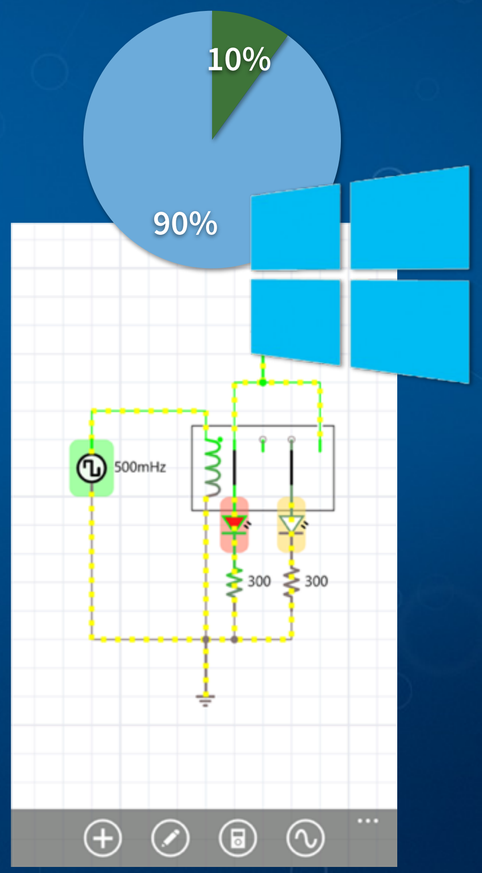
Code Sharing: iCircuit

Shared

Not Shared







http://tinyurl.com/icircuit-code-sharing

Frank Krueger
Featured on iPads in Apple Stores
Also WinRT / iPad

#MVVMCross

- Model View View Model Pattern
- Increase your code reuse
- Binding to Views
- https://github.com/slodge/MvvmCross
- Stuart Lodge

Models, View Models, become shared code

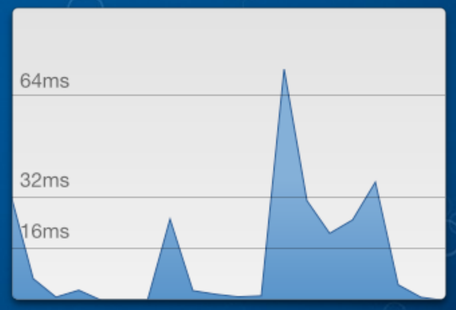
MonoGame

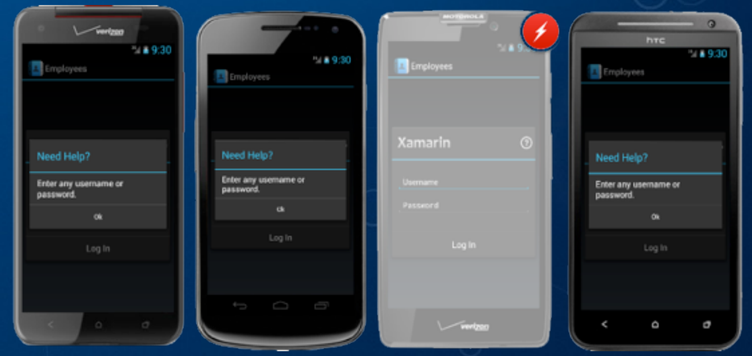
- Open Source Implementation of XNA 4.x
- OpenGL / DirectX
- Runs Everywhere. Really
  - Windows, iOS, Android
  - OUYA, Playstation Mobile
  - Xbox360
- Bastion
- Draw a Stickman Epic
- Infinite Flight

Xamarin Test Cloud

To the (Test) Cloud!

- Any iOS or Android App
- Test on 100’s of Physical Devices
- Performance Monitoring
- Visual Test Results
- App Explorer Mode
- Behavior Driven Dev
- Write Tests in C# or Ruby





Q&A

All the things

- Me - http://redth.info - http://github.com/Redth - @redth
- Xamarin - http://xamarin.com - @xamarinhq
- http://xamarin.com/test-cloud - http://components.xamarin.com
- iCircuit - http://tinyurl.com/icircuit-code-sharing - @praeclarum
- MVVMCross - http://github.com/slodge/mvvmcross - @slodge
- MonoGame - http://www.monogame.net
- Evolve Videos: http://xamarin.com/Evolve/2013/

THANK YOU