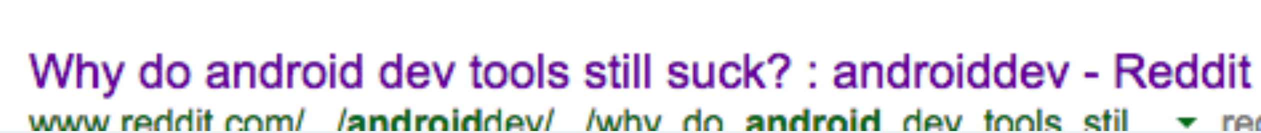
Mike Berlin
@mikeberlin
http://mike-berlin.com

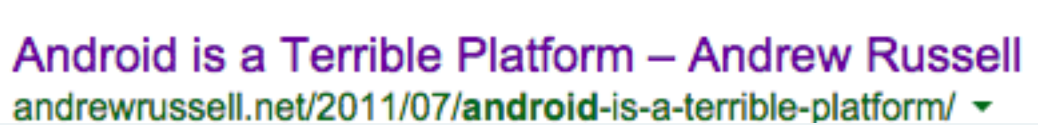
Xamarin

Kansas City Mobile .NET Developers Group

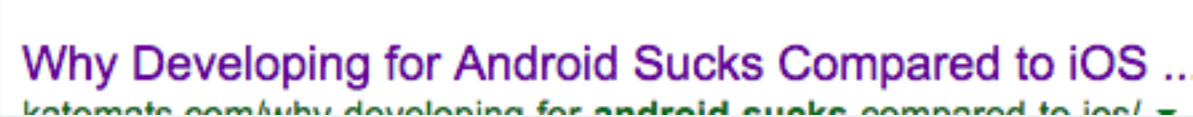
Mannnnnn…Android sucks!



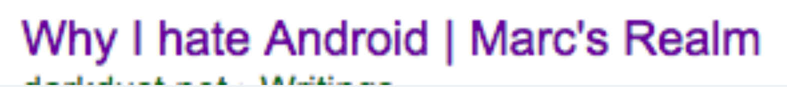
Why do android dev tools still suck? : androiddev - Reddit
www.reddit.com/.../**android**dev/.../why_do_**android**_dev_tools_stil... ▾ red



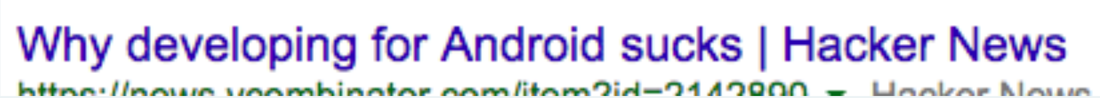
Android is a Terrible Platform – Andrew Russell
andrewrussell.net/2011/07/**android**-is-a-terrible-platform/ ▾



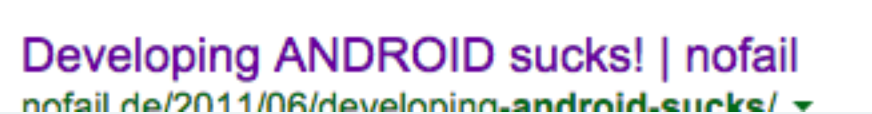
Why Developing for Android Sucks Compared to iOS ...
katemats.com/why-developing-for-**android-sucks**-compared-to-ios/ ▾



Why I hate Android | Marc's Realm
darkdust.net › Writings



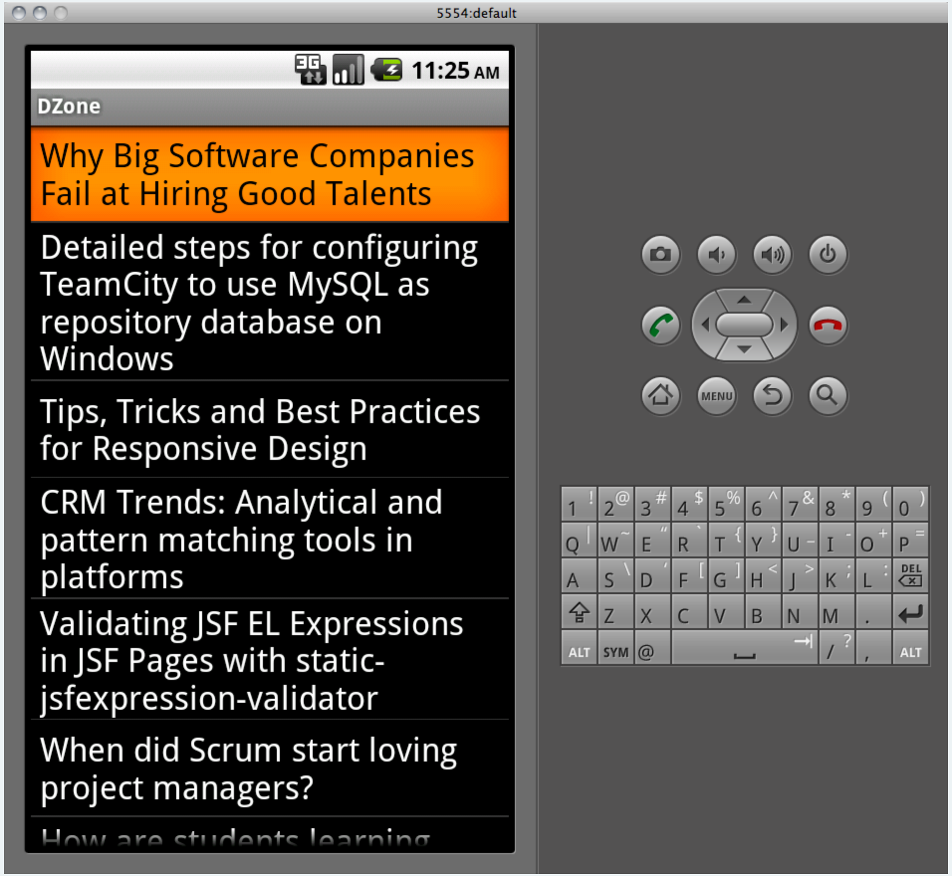
Why developing for Android sucks | Hacker News
https://news.ycombinator.com/item?id=2142890 ▾ Hacker News



Developing ANDROID sucks! | nofail
nofail.de/2011/06/developing-**android-sucks**/ ▾

Google's Android Emulator

Xamarin



5554:default

11:25 AM

DZone

Why Big Software Companies Fail at Hiring Good Talents

Detailed steps for configuring TeamCity to use MySQL as repository database on Windows

Tips, Tricks and Best Practices for Responsive Design

CRM Trends: Analytical and pattern matching tools in platforms

Validating JSF EL Expressions in JSF Pages with static-jsfexpression-validator

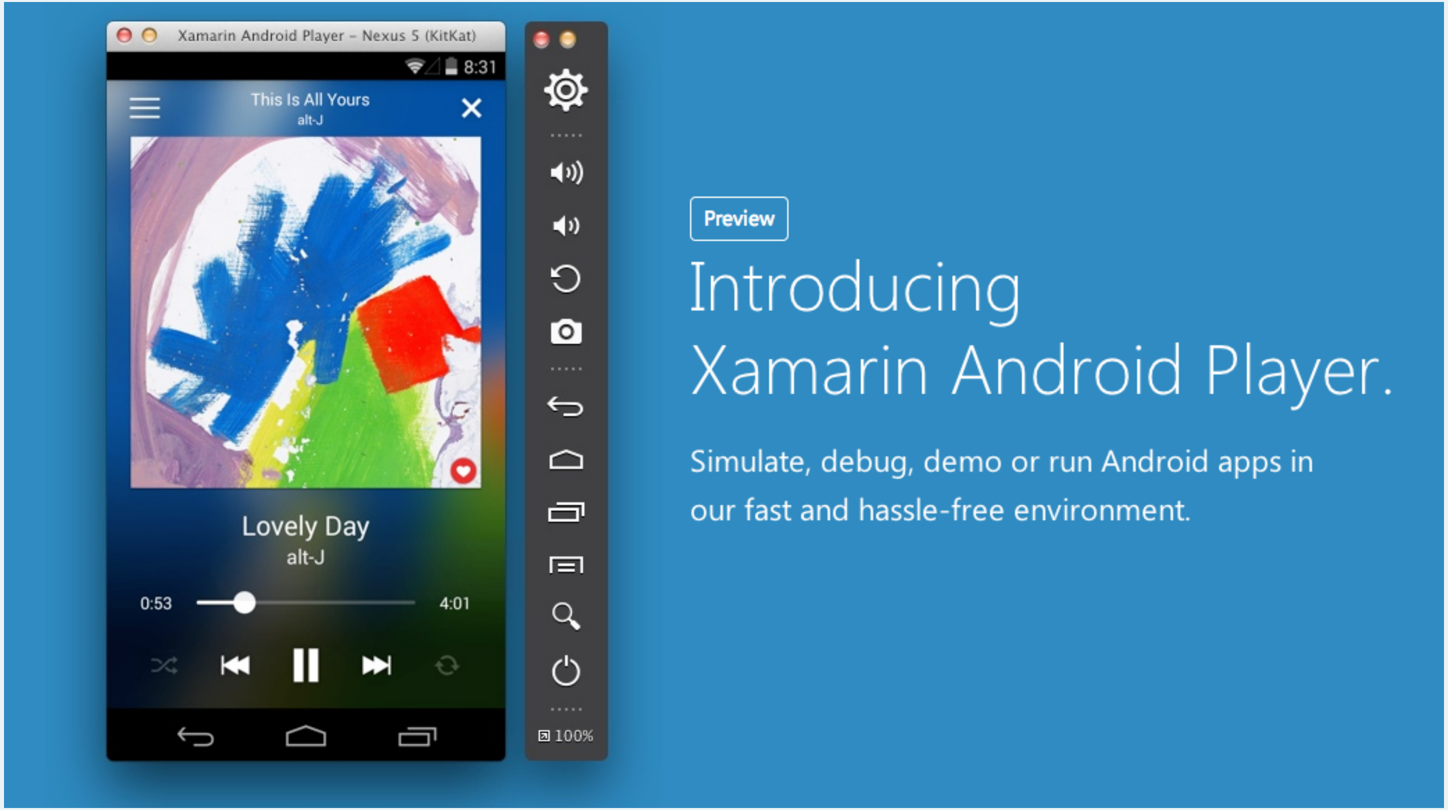
When did Scrum start loving project managers?

How are students learning

One does not simply…

The Answer?

Xamarin Android Player



Preview

Introducing Xamarin Android Player.

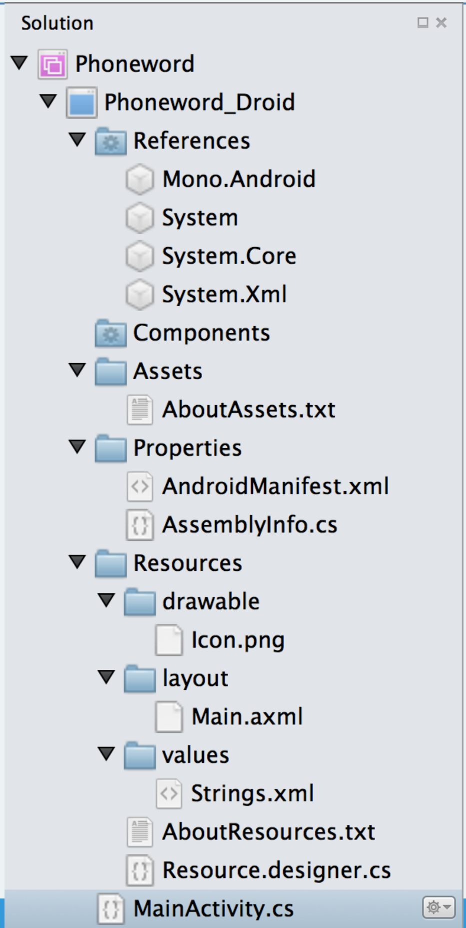
Simulate, debug, demo or run Android apps in our fast and hassle-free environment.

Xamarin

Introduction to Xamarin.Android

Android Project Structure

- References
- Components
- Assets
- Properties
- Resources
  - drawable, layout, values



Solution

- Phoneword
  - Phoneword_Droid
    - References
      - Mono.Android
      - System
      - System.Core
      - System.Xml
    - Components
    - Assets
      - AboutAssets.txt
    - Properties
      - AndroidManifest.xml
      - AssemblyInfo.cs
    - Resources
      - drawable
        - Icon.png
      - layout
        - Main.axml
      - values
        - Strings.xml
      - AboutResources.txt
      - Resource.designer.cs
    - MainActivity.cs

References

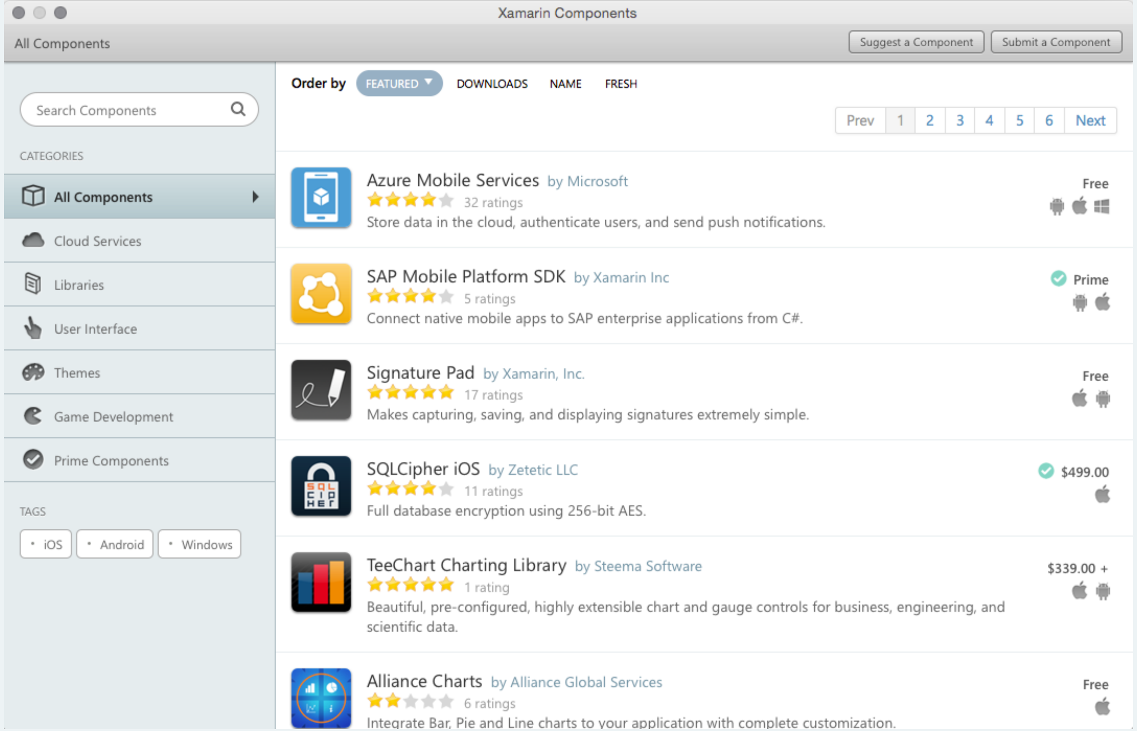
- Assemblies required to build and run app
- .NET assemblies
  - System, System.Core, System.Xml, etc.
- Mono assemblies
  - Mono.Android

Xamarin Component Store

Free and paid components guaranteed to work with Xamarin

Clearly see which platforms are supported by each component

Verified by Xamarin



Azure Mobile Services by Microsoft ★★★★☆ 32 ratings — Free
Store data in the cloud, authenticate users, and send push notifications.

SAP Mobile Platform SDK by Xamarin Inc ★★★★☆ 5 ratings — Prime
Connect native mobile apps to SAP enterprise applications from C#.

Signature Pad by Xamarin, Inc. ★★★★★ 17 ratings — Free
Makes capturing, saving, and displaying signatures extremely simple.

SQLCipher iOS by Zetetic LLC ★★★★☆ 11 ratings — $499.00
Full database encryption using 256-bit AES.

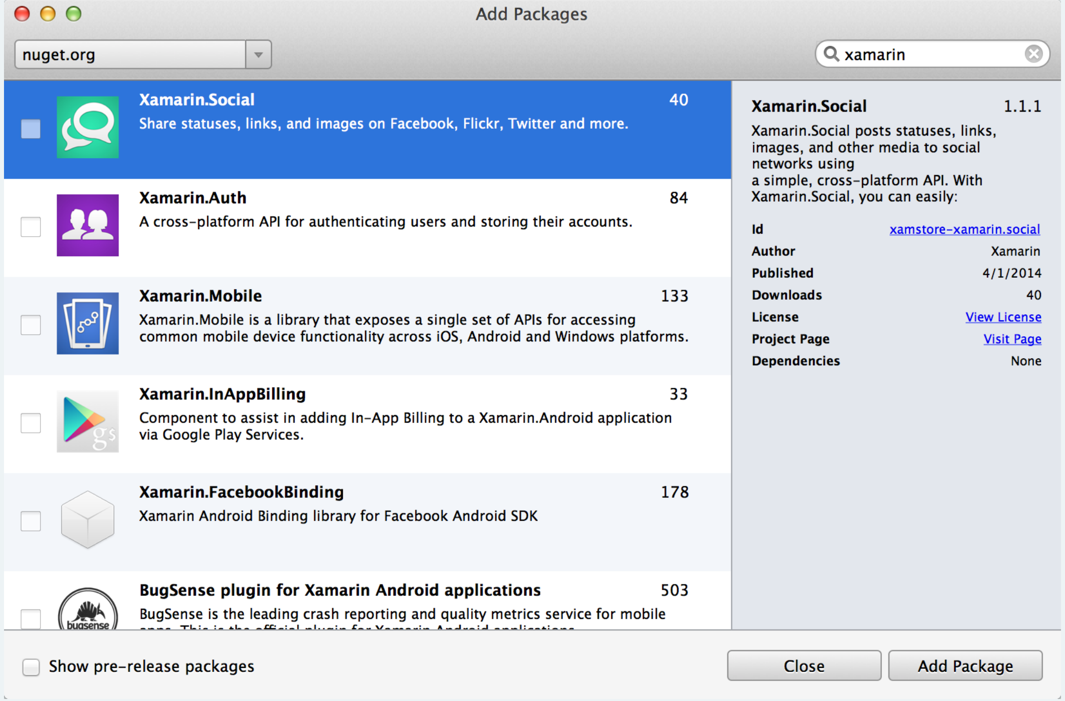
TeeChart Charting Library by Steema Software ★★★★★ 1 rating — $339.00 +
Beautiful, pre-configured, highly extensible chart and gauge controls for business, engineering, and scientific data.

Alliance Charts by Alliance Global Services ★★☆☆☆ 6 ratings — Free
Integrate Bar, Pie and Line charts to your application with complete customization.

NuGet Package Manager

Full support for using NuGet packages

Take advantage of the many NuGet packages which are are now shipping with Xamarin compatibility – in Visual Studio or Xamarin Studio



Add Packages

nuget.org

xamarin

□ **Xamarin.Social** 40
Share statuses, links, and images on Facebook, Flickr, Twitter and more.

□ **Xamarin.Auth** 84
A cross-platform API for authenticating users and storing their accounts.

□ **Xamarin.Mobile** 133
Xamarin.Mobile is a library that exposes a single set of APIs for accessing common mobile device functionality across iOS, Android and Windows platforms.

□ **Xamarin.InAppBilling** 33
Component to assist in adding In-App Billing to a Xamarin.Android application via Google Play Services.

□ **Xamarin.FacebookBinding** 178
Xamarin Android Binding library for Facebook Android SDK

□ **BugSense plugin for Xamarin Android applications** 503
BugSense is the leading crash reporting and quality metrics service for mobile apps. This is the official plugin for Xamarin Android applications.

Xamarin.Social 1.1.1

Xamarin.Social posts statuses, links, images, and other media to social networks using
a simple, cross-platform API. With Xamarin.Social, you can easily:

Id	xamstore-xamarin.social
Author	Xamarin
Published	4/1/2014
Downloads	40
License	View License
Project Page	Visit Page
Dependencies	None

□ Show pre-release packages

Assets

- Files the application *needs* to run
  - Fonts
  - Local data files
- Files are accessible through the **Assets** class

Properties

- AndroidManifest.xml
    - We’ll talk about this one later

Resources

- Contains application resources
  - Strings
  - Images
  - Layouts
- Files are accessible through the **Resource** class
- Contains subfolders that utilize a convention

drawable

- Contains drawable resources
  - Images
  - Bitmaps
- Default template contains application icon file
  - Icon.png

layout

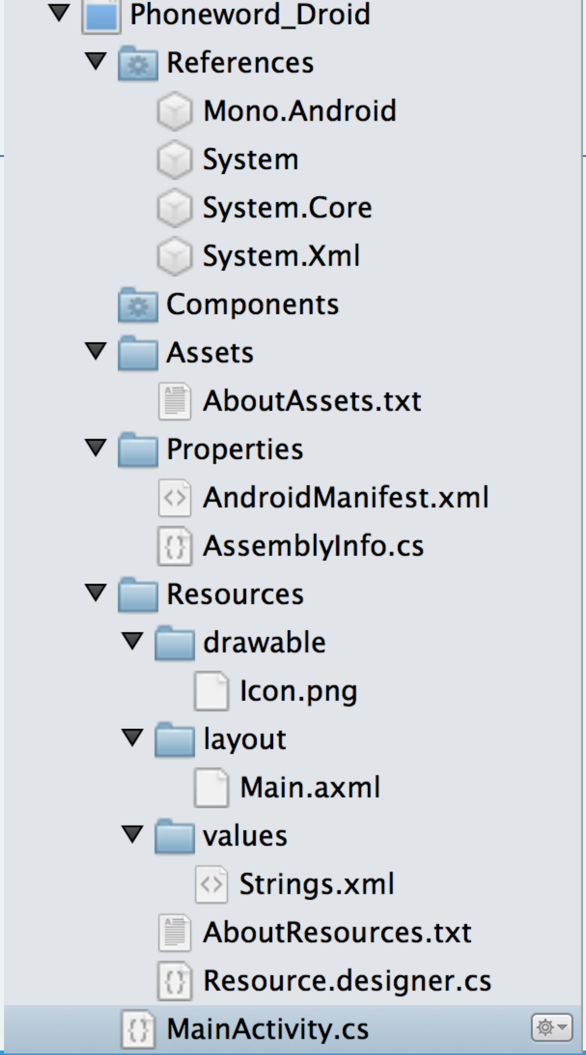
- Contains Android designer files (.axml)
  - User interfaces for each screen
  - Default template contains Main.axml

values

- Contains XML files that store simple values
  - strings
  - integers
  - colors
- Default template creates Strings.xml
- Great for default values and/or localization

Resource.designer.cs

- Commonly referred to as the Resource class
- Partial class that holds unique IDs assigned to each resource
- Automatically created by Xamarin.Android
- Do not manually edit this file



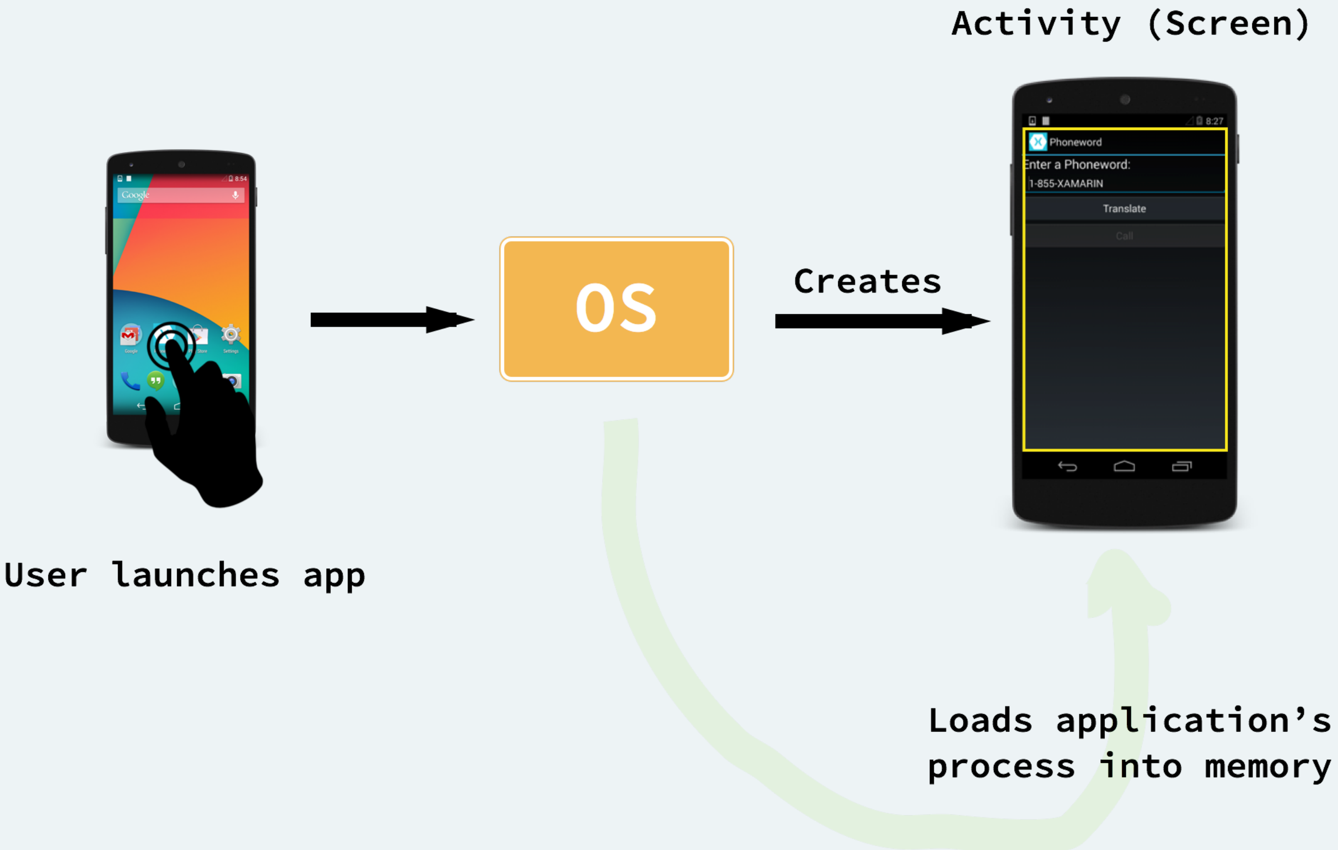
- Phoneword_Droid
  - References
    - Mono.Android
    - System
    - System.Core
    - System.Xml
  - Components
  - Assets
    - AboutAssets.txt
  - Properties
    - AndroidManifest.xml
    - AssemblyInfo.cs
  - Resources
    - drawable
      - Icon.png
    - layout
      - Main.axml
    - values
      - Strings.xml
    - AboutResources.txt
    - Resource.designer.cs
  - MainActivity.cs

Creating your User Interface

Activity

- An Android Application Block that corresponds to a single screen
- Conceptually similar to a web page in a web application
- Responsible for drawing and powering the UI

Activity



Activity (Screen)

User launches app

OS

Creates

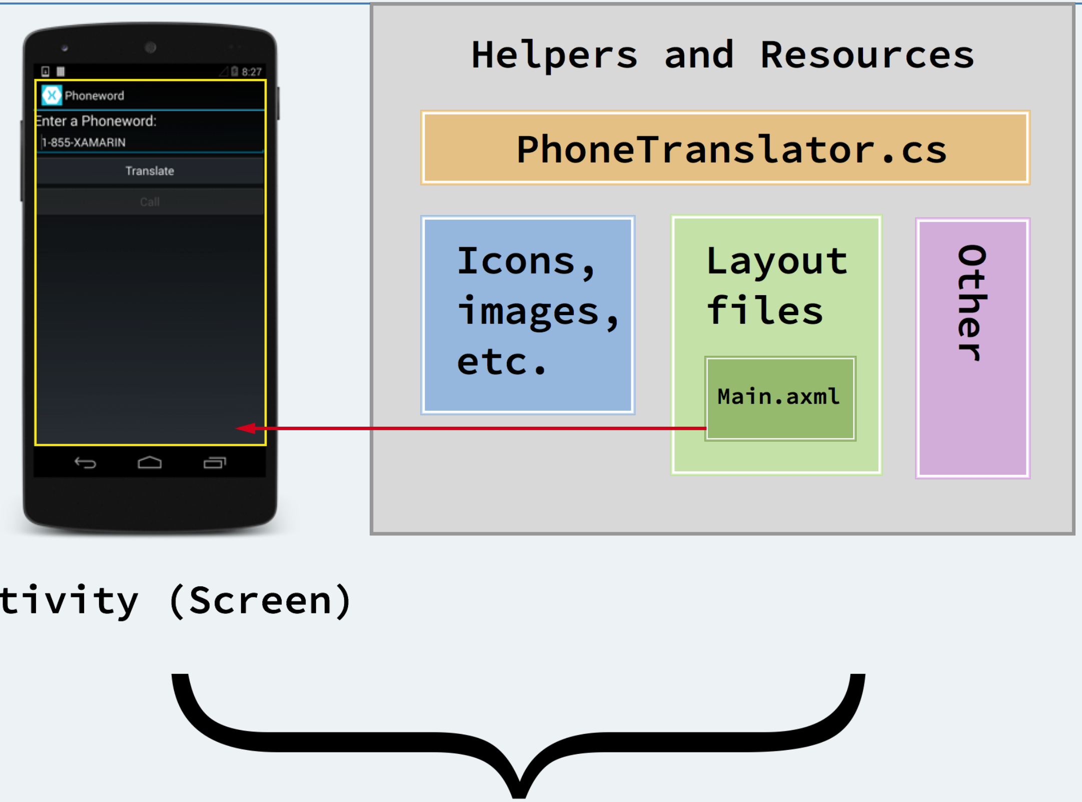
Loads application's process into memory

Android Manifest

- Middle man between Android OS and your app
- Keeps track of an application's:
  - Contents
  - Properties
  - Permisions

Android Manifest

Xamarin



Helpers and Resources

PhoneTranslator.cs

Icons, images, etc.

Layout files

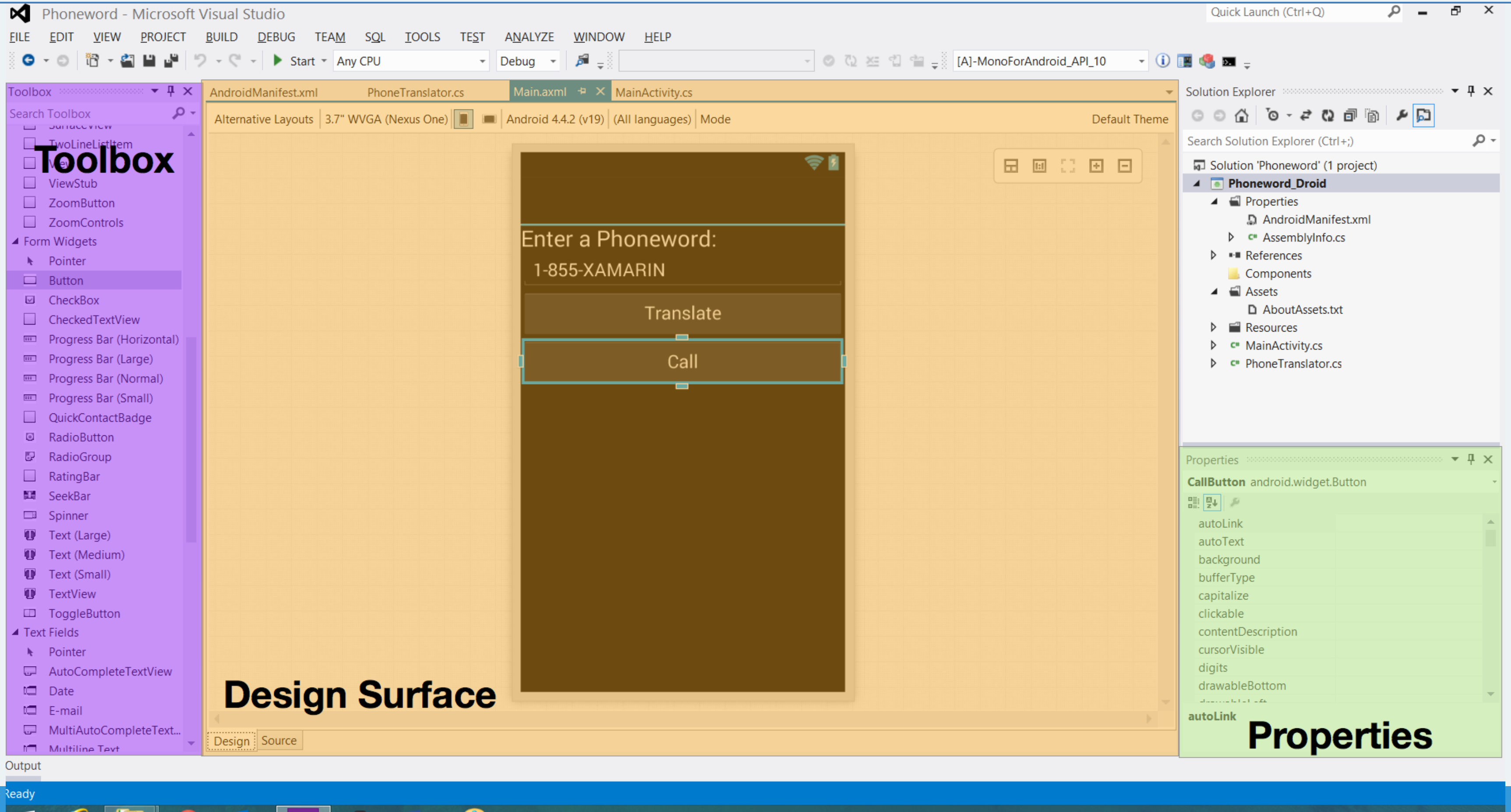
Main.axml

Other

Activity (Screen)

Android Manifest

Android Designer

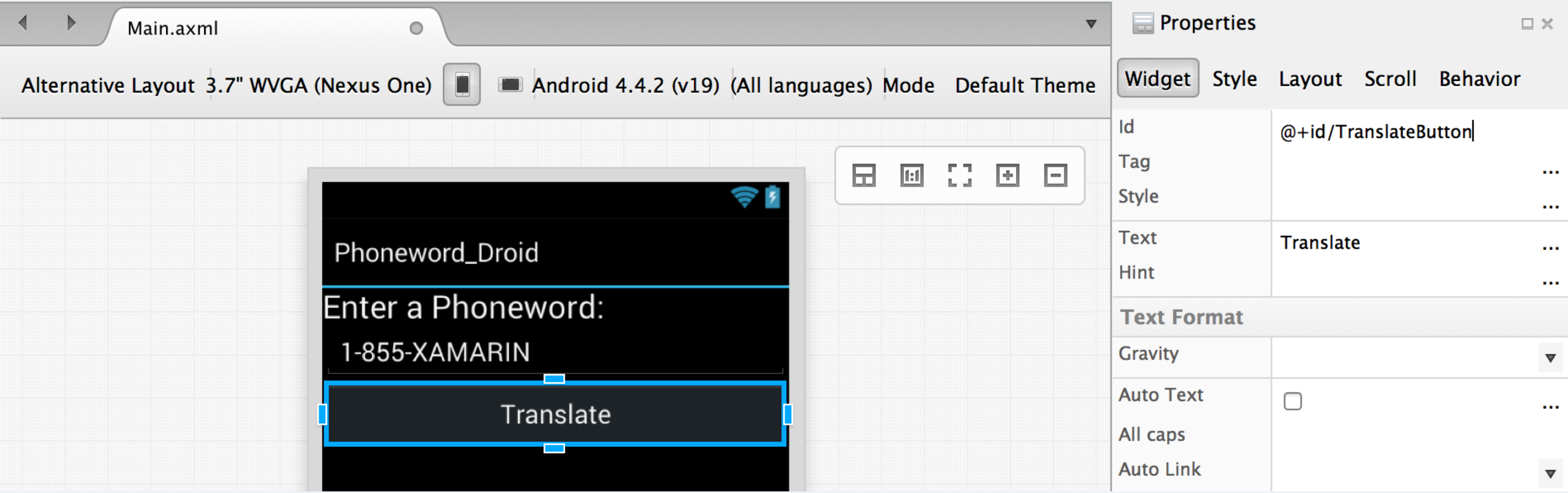


Toolbox

Design Surface

Properties

Android Designer



Main.axml

Alternative Layout 3.7" WVGA (Nexus One) Android 4.4.2 (v19) (All languages) Mode Default Theme

Properties

Widget Style Layout Scroll Behavior

Id: @+id/TranslateButton

Tag:

Style:

Text: Translate

Hint:

Text Format

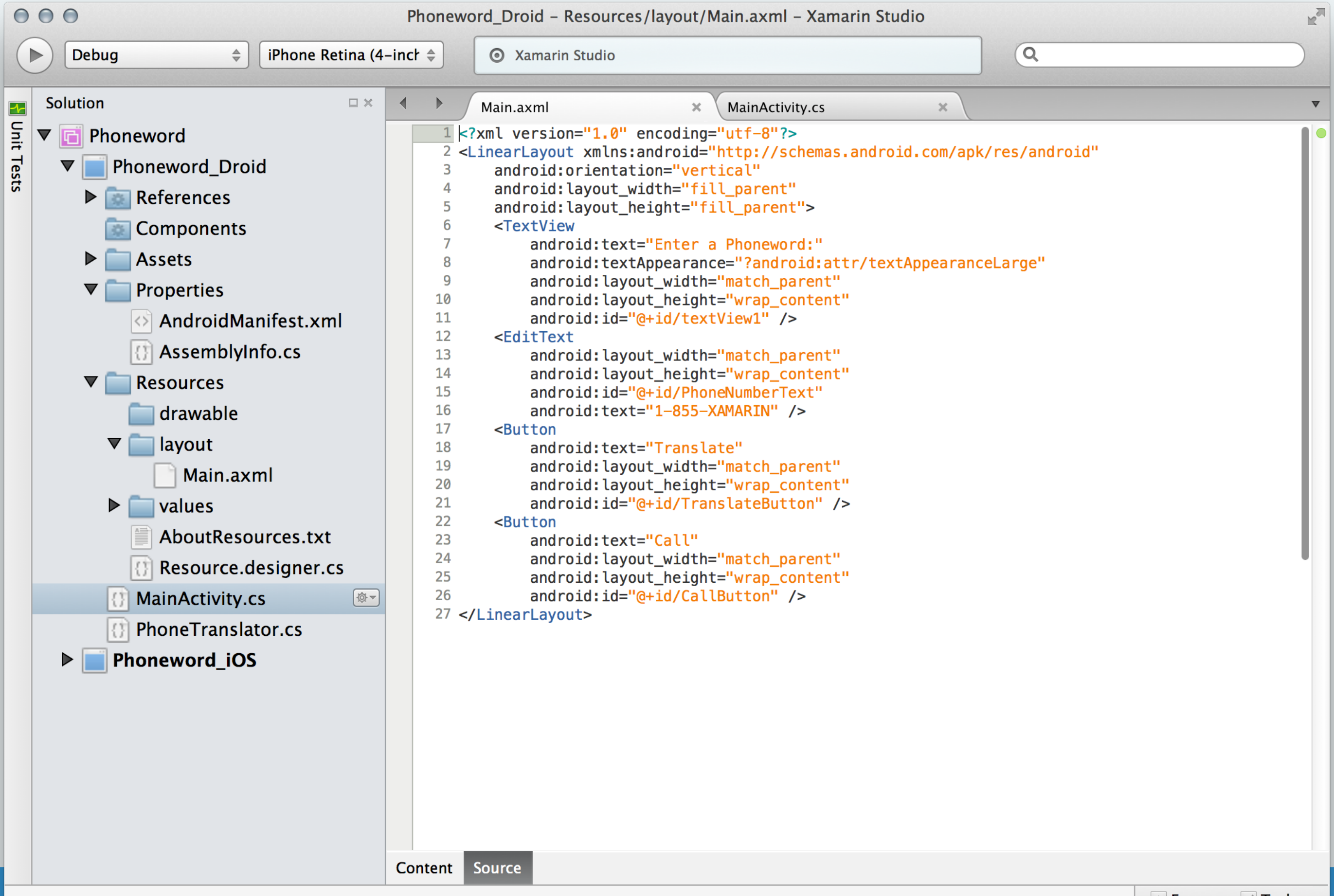
Gravity:

Auto Text: □

All caps:

Auto Link:

Android Designer

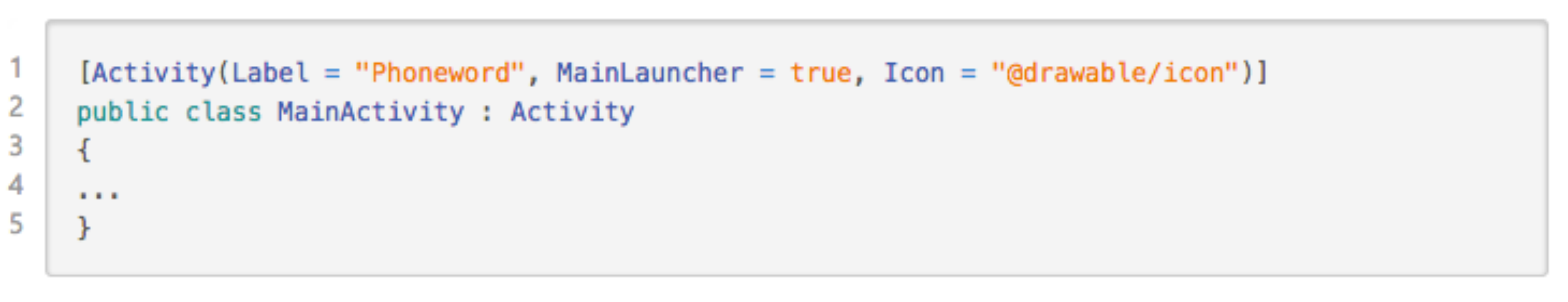


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:text="Enter a Phoneword:"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/textView1" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/PhoneNumberText"
        android:text="1-855-XAMARIN" />
    <Button
        android:text="Translate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/TranslateButton" />
    <Button
        android:text="Call"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/CallButton" />
</LinearLayout>
```

Activity and its lifecycle

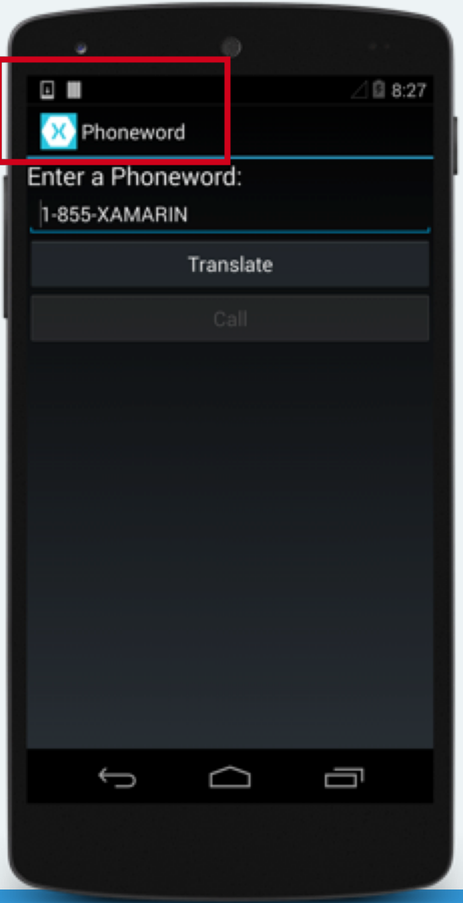
- Name of activity has no significance
- Convention is to use *MainActivity* for entry point of app
- Responsible for responding to user interaction / events
- Inherit from Activity class and adorn with ActivityAttribute

Activity and its lifecycle



```
[Activity(Label = "Phoneword", MainLauncher = true, Icon = "@drawable/icon")]
public class MainActivity : Activity
{
...
}
```

Activity and its lifecycle



Activity and its lifecycle

- Overriding Activity lifecycle methods
  - Control how the Activity loads
  - React to the user input
  - What happens after activity disappears from the screen
- OnCreate
- OnResume
- OnPause

Intents

- Android Application Blocks are designed to be loosely-coupled
  - Have little knowledge of and limited access to other Blocks (even ones that are in the same application)
- Asynchronous messages sent back and fourth between Application Blocks
  - Launch camera app to take and save photos
  - Navigate from one activity to the next
  - Place a phone call

DEMO

Xamarin

Try Xamarin Today

http://xamarin.com/shirt

Learn more

Xamarin University

Unrivaled Mobile Development Training

Live unlimited mobile development training from mobile experts, in your time-zone, on your schedule and as often as you'd like

Xamarin

Thank You

Mike Berlin
@mikeberlin
http://mike-berlin.com