Android App Development

Jordan Jozwiak
November 13, 2011
Agenda

- Why Android?
- Application framework
- Getting started
- UI and widgets
- Application distribution
- External libraries
- Demo
Why Android?
Two web comics illustrating the difference between iPhone and Android.

Android

Hey, I'd love to do X with my Android phone. Do you know if that's possible?

Hmm, that's a good question. I'm not really sure.

OK, so I've tried a few different apps from the market that come close, but none that quite do what I want. But I read on a couple of forums that there are some custom ROMs that may allow me to do it.

Well I've tried the major custom ROMs; Modaco, CyanogenMod and OpenDesire now. Man the control they give the user is amazing. Unfortunately they still couldn't quite do it how I wanted, so I'm signing up for App Inventor; I'll try to make it myself.

Ok, so I've got my head around App Inventor, and while I think it could do it, I don't think it's the best way to implement this. So I've gotten in touch with a community of like-minded open source programmers. I'm working to put something together with them, and I think if I cut back on my hours at work a bit...
Hey, I'd love to do X with my iPhone. Do you know if that's possible?

Nahh, it's not possible.

Ohh.
Why Android?

- Open-source
  - That means that it’s free!
- Easy-to-use framework based on Java
- True customization
  - Replace stock apps
  - Personalize Home Screens with widgets
- Better notifications
- Easier to publish – no review process
Why Android? - Security

- Reputation as less secure than iOS
  - True, but it is still very secure!
- Each application lives in its own security sandbox
  - System assigns each app a unique Linux user ID
- Each process has its own virtual machine, so an app’s code runs in isolation from other applications
- Permissions are approved by the user during app installation.
- Apps are signed by a developer’s private key
Application framework
Application framework

- **src** - source files where the actual coding takes place
- **res** – collection of resources for screen layouts, images, sounds, text, animations and more.
- **Manifest** – represents essential information about the application to the Android system
Application framework

- src code example

```java
package com.example.helloandroid;

import android.app.Activity;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Object o = null;
        o.toString();
        setContentView(R.layout.main);
    }
}
```
Application framework

- res
  - res/drawable – logo and all images
    - Customizable for different pixel densities
  - res/raw – sounds and music
  - res/layout – the XML layout for each activity
    - Customizable for different screen sizes and orientations
  - res/values – saves all strings and object styles

- Not as stream-lined or standardized as iOS
Application framework

- Layout example

```xml
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/textview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:text="@string/hello"/>
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="button"/>
</LinearLayout>
```
Application framework

- Manifest
  - Permissions
    - e.g. access internet, access SMS messages
  - Minimum API
  - Hardware and software features required
  - You must also declare every activity you create!
Application framework

- Manifest example

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloandroid"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="7" />
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".HelloAndroid"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Application framework – Activity Lifecycle
Application Framework

- **Activity**
  - Visible screens

- **Service**
  - Background services

- **Content provider**
  - Shared data

- **Broadcast receivers**
  - Receive and react to broadcast events

- **Intent**
  - Launch new activities
Getting started
Getting started - installation

- Get the latest Java Development Kit (JDK)
- Get the latest version of Eclipse Classic (recommended)
- Install the Android Software Development Kit (SDK)
- Install the Android Development Tools (ADT) plugin for Eclipse
  - Makes your life easier: lets you quickly set up new Android projects, create an application UI, debug your apps, and export APKs for distribution!
Getting started – Eclipse + ADT
Getting started - Emulator

- Eclipse + ADT + Emulator = 😊
Getting started - debugging

- DDMS debugger - not fun, but you should do it.
UI and widgets
UI and widgets

- Android user interface is composed of hierarchies of objects called Views
  - View: a drawable object such as a button, image, or text label
  - Widget: like form types in HTML, ways that the user can interact with the UI such as TextView, ListView, ScrollView, Spinner, TabWidget, Button
  - Layout: a container for widgets
UI and widgets

- Linear Layout, Relative Layout, and Table Layout
Application distribution
Application distribution

- Publish in the Android Market
  - Purchase a developer account for $25
  - Export APK from Eclipse with ADT
  - Upload APK, images, and description
External Libraries

- Google Analytics
- admob
- Facebook
- Twitter
- Dropbox
- OpenFeint
External Libraries

- Easily add external libraries by importing 3rd party JARs (.jar files)
  - Allows you to easily integrate cool features into your app
  - Make use of SDKs from big corporations
  - Selling point
  - Less coding
External Libraries - Analytics

- Google Analytics (recommended)

- Understand how users interact with your app
  - Number of visitors per day
  - Time on page (in activity)
  - Demographics and location
  - Track app version
  - App interaction (button presses, conversions, etc.)
External Libraries - Monetization

Download distribution of Android apps by price category, November 12, 2011

Price

- Free (N=211817)
- < $1 (N=33905)
- $1 - $2.5 (N=43004)
- $2.5 - $5 (N=17248)
- $5 - $10 (N=8349)
- > $10 (N=3891)

Percentage of apps per download category

- < 500
- 500 - 5,000
- 5,000 - 50,000
- > 50,000

AppBrain
External Libraries - Monetization

- Open-source philosophy
  - People are less willing to pay for apps, so developers may make money through advertisements

- AdMob (recommended)
  - Bought by Google in 2009 for $750 million
  - Paid by click, not by impression
  - Easily transfer funds to advertise your own app (20% bonus)
Many large companies have their own SDKs in the form of JARs
- Facebook, Twitter, Dropbox, OpenFeint, etc.
- Search online for JARs for other functions (advanced math operations, etc.)
Demo

- Tutorial for Hello, world!

- Tutorials for Views

- Other tutorials
Thanks for coming!

Well, it depends what you want. The iPhone wins on speed and polish, but the Droid has that gorgeous screen and physical keyboard.

What if I want something more than the pale facsimile of fulfillment brought by a parade of ever-fancier toys? To spend my life restlessly producing instead of sedately consuming?

Is there an app for that?

Yeah, on both. / Wait, no, looks like it was rejected from the iPhone store.

Droid it is, then.