

Intro to Android Development



By Philip Peng, 2011-01-12

For PennApps Mobile 2011, University of Pennsylvania

All images used in this belong to their rightful owners (not me).

Intro to Android Development

Why mobile?

- smartphones = “new thing”
- portable, powerful



Why Android? (over iOS)

- Linux → Open Source
- Consumer-driven apps
- Hardware/software choice
- Google <3 Devs

vs.

iOS 4

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'ma try to make it myself.

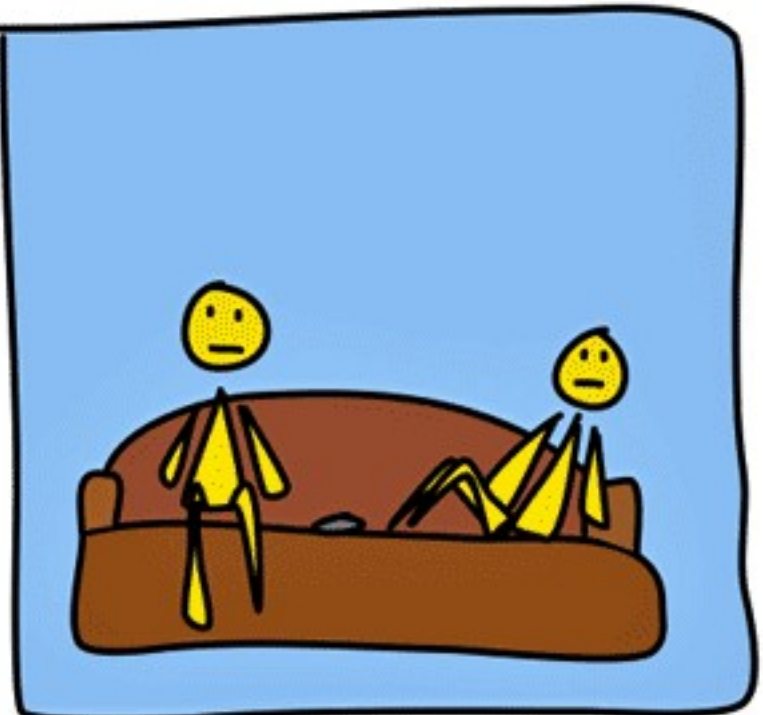
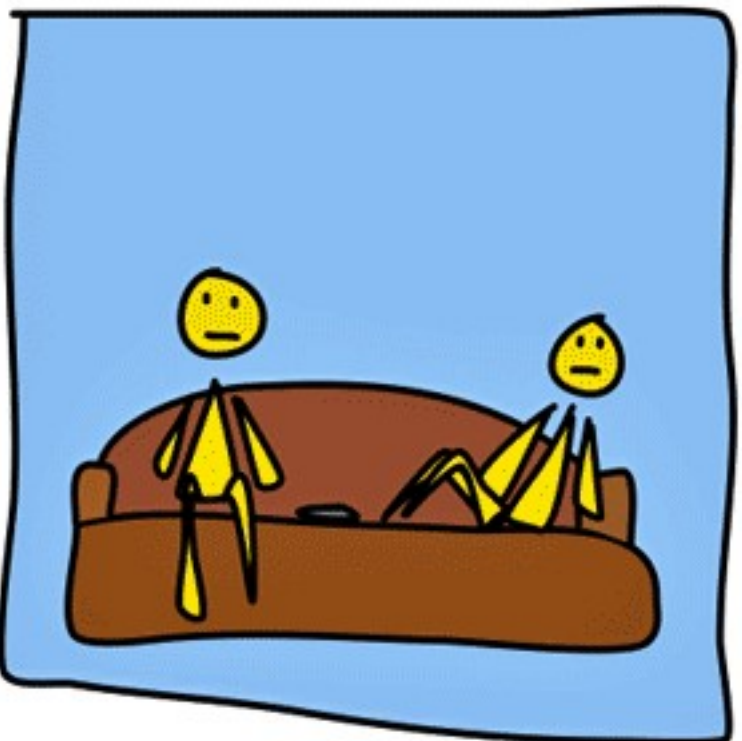
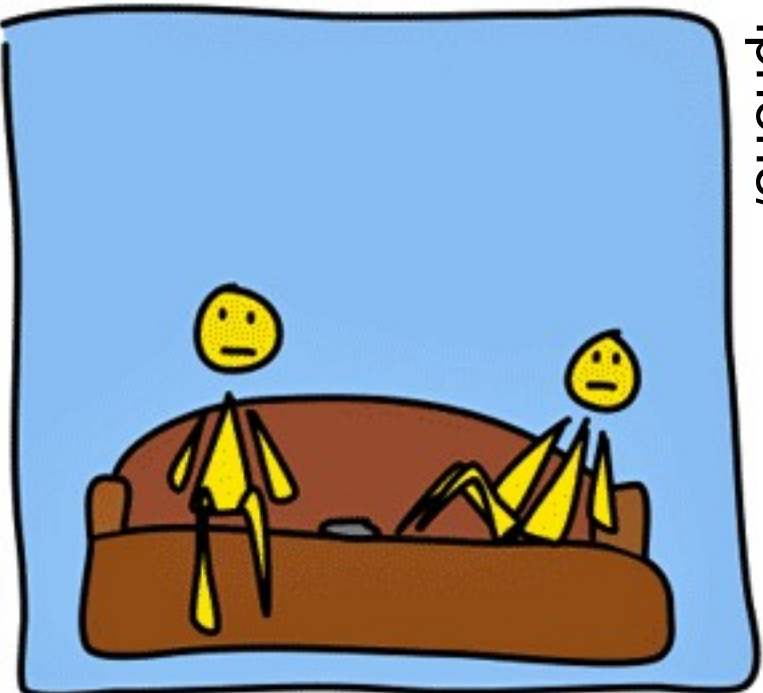
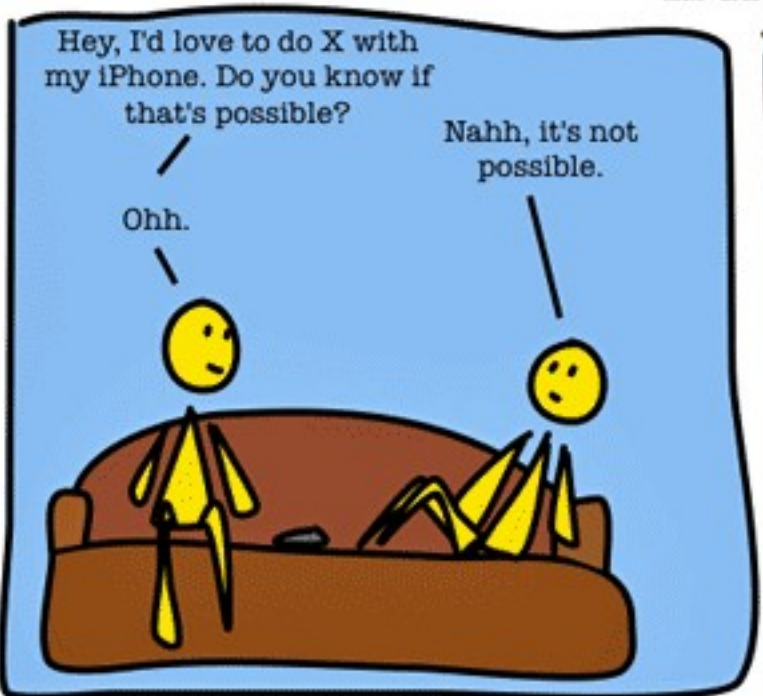


Ok, so I've got my head around App Inventor, and while I think it could do it, I don't think it's the best way to implement this. So I've gotten in touch with a community of like-minded open source programmers. I'm working to put something together with them, and I think if I cut back on my hours at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



<http://icantdrawfeet.com/2011/08/02/android-vs-iphone/>

iPhone



Intro to Android Development

Overview

- Java
- Eclipse
- Hello World
- Activity Cycle
- Try API Demos
- Stopwatch Example
- Logcat/DDMS Debugging
- Market Publishing and more...
- Beats and Useful Resource Links



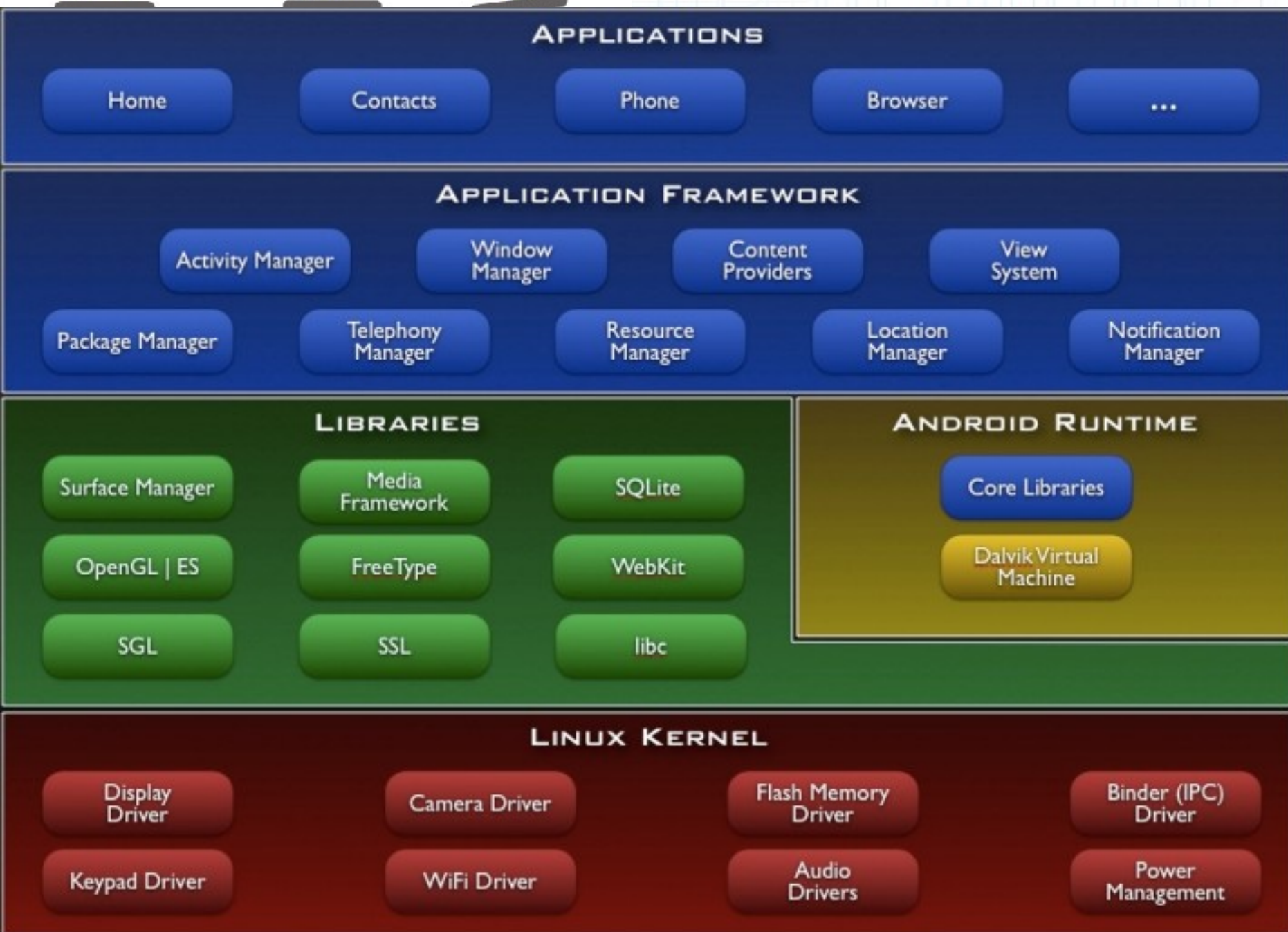
Intro to Android Development

Android Java

- Linux, but apps in Java
- Dalvik Virtual Machine – DVM
- .apk format ← Eclipse
- Standard java.* packages
- Custom graphics/UI code
- NDK – C/C++ for ARM
(OpenGL ES 2.0, libraries, etc.)



Intro to Android Development



Intro to Android Development

Eclipse

- FOSS, customizable, industry-standard
- Android-ready

Download these (if you haven't already)

- Eclipse Classic 3.6.1

<http://www.eclipse.org/downloads/packages/eclipse-classic-361>

- Android SDK R8 (get the .zip)

<http://developer.android.com/sdk/index.html>

- Java SE Development Kit 6u23 (JDK)

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- Code Samples used

<http://www.stwing.upenn.edu/~pengp/Files/PennApps/PennApps.html>

Intro to Android Development

Setup (Live Demo!)

- Install JDK (run installer)
- Extract Eclipse (e.g. *C:/Android/eclipse*)
- Install Android SDK
 - Extract Android SDK (e.g. *C:/Android/android-sdk-windows*)
 - Run SDK Manager
 - Select packages
(only need Platform-tools, API 9, document, and samples)
 - Download and wait (long)
 - Virtual Devices > New
 - Name: "TestPhone2.3", Target: Android 2.3 – API Level 9, SD Card Size: 64 MiB
 - Create AVD > Start... > Launch
 - A virtual Android 2.3 phone for testing! Yay lol/

Intro to Android Development

Setup (Live Demo! Continued)

- Install ADT Plugin for Eclipse:
 - <http://developer.android.com/sdk/eclipse-adt.html#installing>
 - Help > Install New Software.... > Add
 - <https://dl-ssl.google.com/android/eclipse/> → “ADT”
 - Select All > Next > Next > Accept Terms > Finish > Restart
 - Windows > Preferences... > Android
 - Browse (find “*android-sdk-windows*” folder) > Apply > OK
 - Eclipse is now ready for Android development!

Ready for your Android “Hello World”?

Intro to Android Development

Hello World

File > New > Project >
Android > Android
Project

New Android Project

Creates a new Android Project resource.

Project name: HelloWorld

Contents

- Create new project in workspace
- Create project from existing source
- Use default location

Location: C:/Android/workspace/HelloWorld

Create project from existing sample

Samples: This target has no samples. Please select another target.

Build Target

Target Name	Vendor	Platform	API ...
<input checked="" type="checkbox"/> Android 2.3	Android Open Source Project	2.3	9

Properties

Application name: HelloAndroid

Package name: com.example.helloandroid

Create Activity: HelloAndroid

Min SDK Version:

Intro to Android Development

Hello World

Add the following lines marked by “// <-----”

```
HelloAndroid.java X
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView; // <-----

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TextView tv = new TextView(this); // <-----
        tv.setText("Hello, Android"); // <-----
        setContentView(tv); // <-----
    }
}
```

Intro to Android Development

Hello World

Run > Run > Android Application



Intro to Android Development

Hello World Inspection

- Package naming convention?
- Activity, Layout, TextView?
- R, src, res, AndroidManifest?

```
HelloAndroid.java X
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView; // <-----

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TextView tv = new TextView(this); // <-----
        tv.setText("Hello, Android"); // <-----
        setContentView(tv); // <-----
    }
}
```

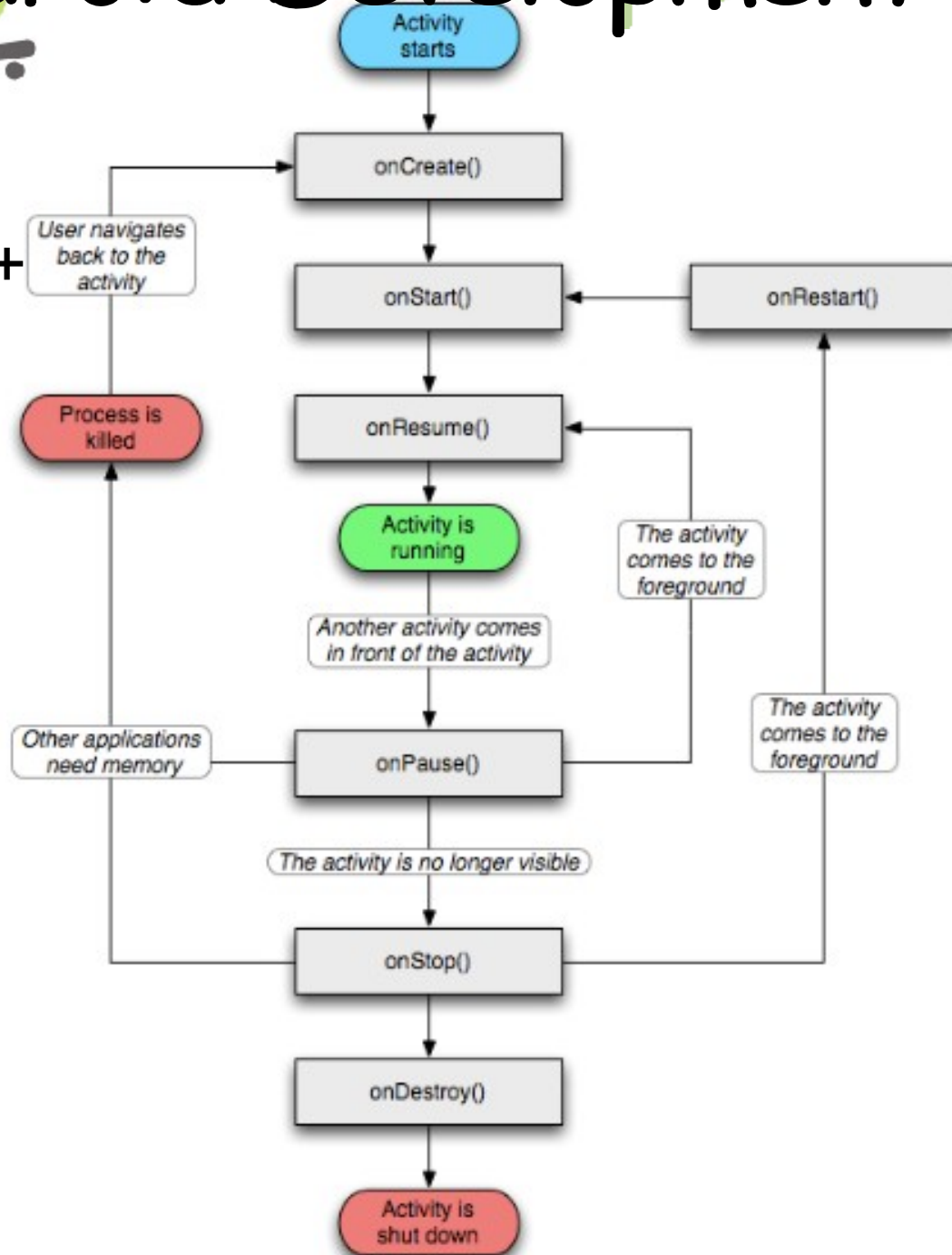
```

HelloWorld
├── src
│   ├── com.example.helloandroid
│   │   ├── HelloAndroid.java
│   │   │   └── onCreate(Bundle):
│   └── gen [Generated Java Files]
│       ├── com.example.helloandroid
│       │   └── R.java
│       │       └── R
│       │           ├── attr
│       │           ├── drawable
│       │           ├── layout
│       │           └── string
├── Android 2.3
│   └── android.jar - C:\Android\andr
├── assets
├── res
│   ├── drawable-hdpi
│   ├── drawable-ldpi
│   ├── drawable-mdpi
│   ├── layout
│   │   └── main.xml
│   ├── values
│   │   └── strings.xml
├── AndroidManifest.xml
├── default.properties
└── proguard.cfg
```

Intro to Android Development

Activity Cycle

- Activity = *methods* + *Layout* + *Dialogs*
- Layout = *Views*, *Buttons*, etc.
- App = *many Activities*
- Task = *stack of Activities from one app*
- Also: *Services*, *Broadcast Receivers*, *Content Providers*



Intro to Android Development

API Demos

- File > New > Projects > Android Project
- Create project from existing source
(*C:\Android\android-sdk-windows\samples\android-9\ApiDemos*)
- Right click ApiDemos project > Refresh
- src > com.examples.android.apis > ApiDemosApplication
- Run > Run > Android Application (or Ctrl+F5)
- Explore the demos!

Ready to try it yourself?

Intro to Android Development

Stopwatch 1

- Hello World, but need time and screen updating
- *FormattedTime.java* and *RefreshHandler.java*

FormattedTime

- *start()* - starts timer, returns "00:00:00"
- *stop()* - stops timer, returns time "MM:SS:mm"
- *reset()* - resets timer, returns "00:00:00"
- *update()* - returns time "MM:SS:mm"

RefreshHandler

- *start(delay)* - starts calling *update()* every *delay* ms
- *stop()* - stops calling *update()*
- *update()* - override this with your code

Intro to Android Development

Stopwatch A

- Hello World, with *FormattedTime* and *RefreshHandler*
- But this just keeps running, what about stopping or resetting?

```
1 package com.pennapps.stopwatch;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.widget.TextView;
6
7 public class Stopwatch extends Activity {
8
9     private FormattedTime timer;
10    private TextView tv;
11
12    private RefreshTime refresh;
13    class RefreshTime extends RefreshHandler {
14        public void update() {
15            tv.setText("Time is: " + timer.update());
16        }
17    };
18
19    /** Called when the activity is first created. */
20    @Override
21    public void onCreate(Bundle savedInstanceState) {
22        super.onCreate(savedInstanceState);
23        setContentView(R.layout.main);
24        tv = new TextView(this);
25        tv.setText("Hello, Android");
26        setContentView(tv);
27
28        timer = new FormattedTime();
29        timer.start();
30        refresh = new RefreshTime();
31        refresh.start();
32    }
33 }
```

Intro to Android Development

Views, Images and Buttons, Oh My!

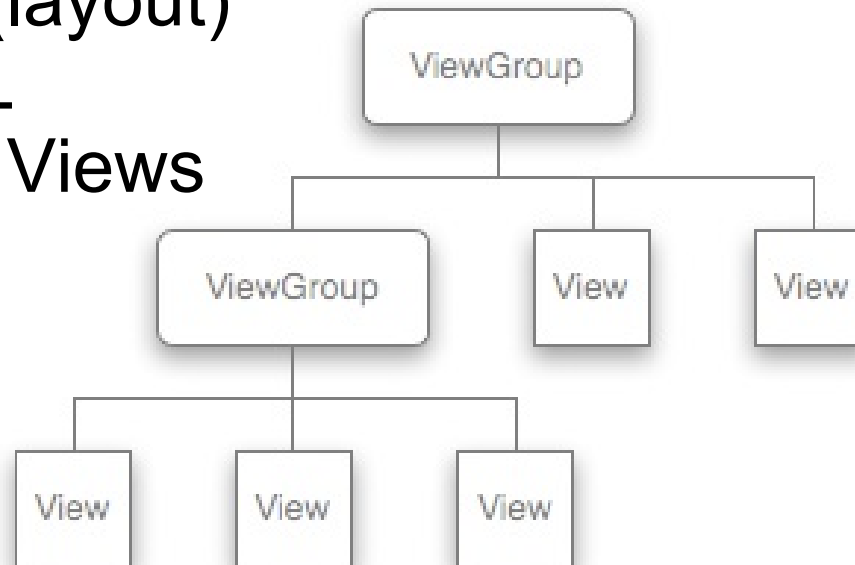
- `Activity.setView(layout)`
- Graphical layout → XML
- Add in Widgets or more Views

ViewGroups:

*LinearLayout, TableLayout,
RelativeLayout, ScrollView,
ListView, GridView*

Widgets/Content Views:

*TextView, WebView, ImageView
ImageButton, Button, Checkbox, EditText, RadioButton*



See Android documentation for details and more

<http://developer.android.com/guide/topics/ui/index.html>

Intro to Android Development

Stopwatch B

- Buttons for Start, Stop, Reset
- Give *TextView* an *id* for reference
- Good practice: use *strings.xml*
- *onClick* → method
- Use *GridView* or other *ViewGroups* for nicer layout

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:orientation="vertical"
4     android:layout_width="fill_parent"
5     android:layout_height="fill_parent"
6     >
7     <TextView
8         android:id="@+id/time"
9         android:layout_width="fill_parent"
10        android:layout_height="wrap_content"
11        android:text="@string/hello"
12    />
13    <Button
14        android:layout_height="wrap_content"
15        android:layout_width="wrap_content"
16        android:text="@string/start"
17        android:onClick="start"
18    />
19    <Button
20        android:layout_height="wrap_content"
21        android:layout_width="wrap_content"
22        android:text="@string/stop"
23        android:onClick="stop"
24    />
25    <Button
26        android:layout_height="wrap_content"
27        android:layout_width="wrap_content"
28        android:text="@string/reset"
29        android:onClick="reset"
30    />
31 </LinearLayout>
32
```

Intro to Android Development

Stopwatch B

- Cross-reference TextView (recast)
- *onClick* must be *public void foo(View view)*

```
20     /** Called when the activity is first created. */
21     @Override
22     public void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.main);
25         tv = (TextView) findViewById(R.id.time);
26
27         timer = new FormattedTime();
28         timer.start();
29         refresh = new RefreshTime();
30         refresh.start();
31     }
32
33     public void start(View view) { timer.start(); }
34     public void stop(View view) { timer.stop(); }
35     public void reset(View view) { timer.reset(); }
36 }
```

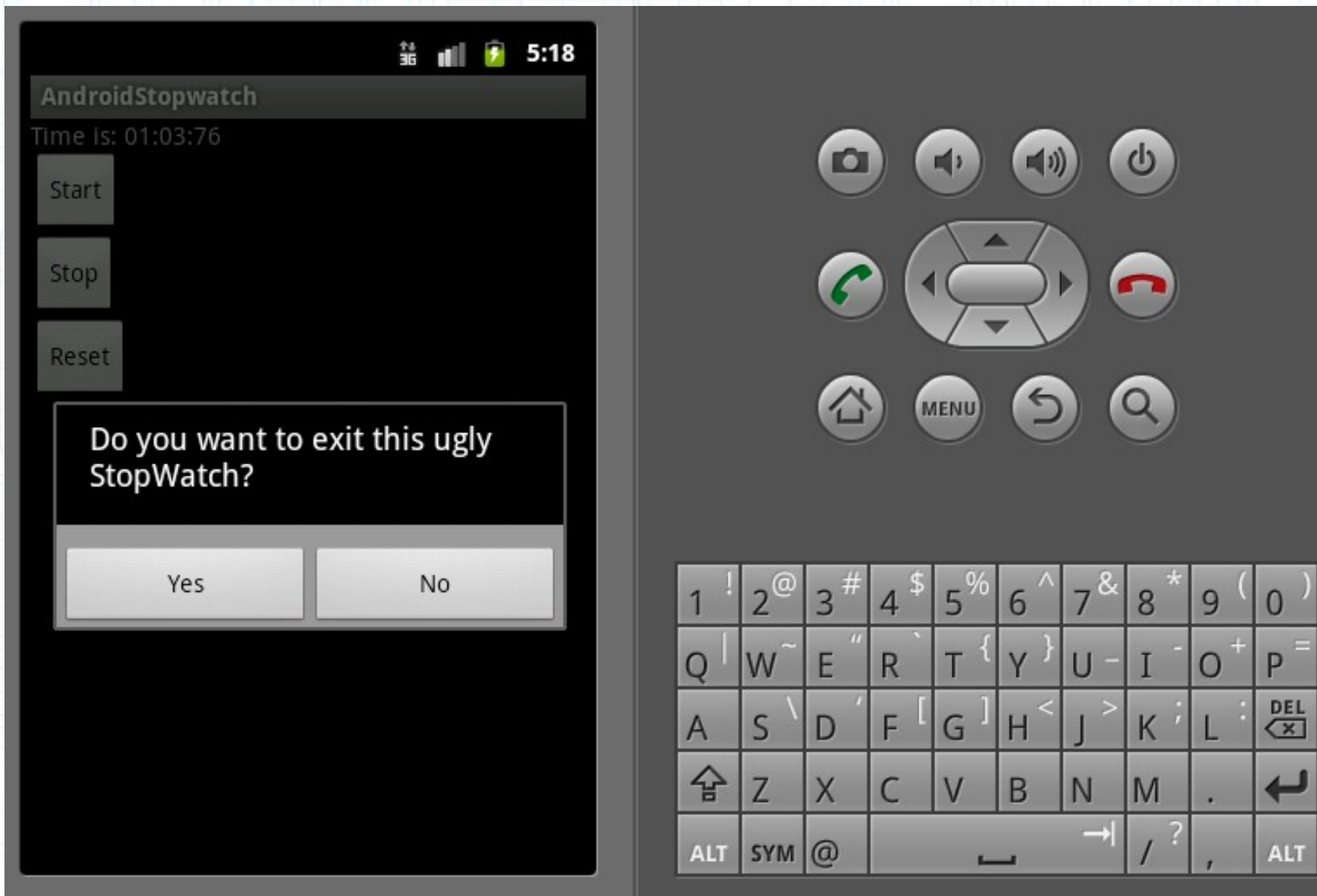
Intro to Android Development

Stopwatch C

- Lets add unnecessary stuff!

```
41 public boolean onKeyDown(int keyCode, KeyEvent event) {
42     switch (keyCode) {
43         case KeyEvent.KEYCODE_MENU:
44             AlertDialog.Builder builder = new AlertDialog.Builder(this);
45             builder.setMessage("Do you want to exit this ugly StopWatch?")
46                 .setCancelable(false)
47                 .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
48                     public void onClick(DialogInterface dialog, int id) {
49                         Toast.makeText(
50                             Stopwatch.this, "Goodbye!", Toast.LENGTH_SHORT).show();
51                         Stopwatch.this.finish();
52                     }
53                 })
54                 .setNegativeButton("No", new DialogInterface.OnClickListener() {
55                     public void onClick(DialogInterface dialog, int id) {
56                         dialog.cancel();
57                     }
58                 });
59             AlertDialog alert = builder.create();
60             alert.show();
61             return true;
62         default:
63             return super.onKeyDown(keyCode, event);
64     }
65 }
66 }
```

Intro to Android Development



Intro to Android Development

More Common Stuff

Graphics:

<http://developer.android.com/> →

- Static → XML layouts
- Active → Canvas
- 3D → OpenGL ES 1.x or 2.0 (NDK)

Audio/Video:

- Playback → MediaPlayer
- Recording → MediaRecorder

Data:

- Settings → PreferenceActivity & PreferenceManager
- Storage → External Storage, SQLite

Network:

- Connectivity → ConnectivityManager
- Locations → Uri*

Interactivity:

- Multiple activities → Intents
- Other apps/Content handling → Intent Filters

Blog

Videos

Resources

Reference

Dev Guide

SDK

Home

sdolevnap
CIOFCUB

Intro to Android Development

Debugging!

Logcat!!! (and DDMS later)

- `Log.v(String tag, String msg)`
- Also `Log.v`, `.w`, `.d`, `.wtf`

ADB

- `android-sdk-*/platform-tools/adb`
- `adb push/pull/install`
- `adb shell`
- `adb logcat`



Intro to Android Development

DDMS (Dalvik Debug Monitor Server)

- Eclipse integrated Android debugger
- Windows > Open Perspective > Other... > DDMS
- Select device, select process
- Debug, Update Heap, Update Threads
- Allows inspection of threads, heap, allocation, files, etc.
- Also displays Logcat and allows for screenshots!

Intro to Android Development

The screenshot displays the Android Studio interface with three main panels:

- Devices:** A table listing running processes on the emulator.
- Threads:** A table showing the current thread list for the selected process.
- LogCat:** A log viewer showing system messages from the emulator.

Devices Panel:

Name	State	Android Version
emulator-5554	Online	Android2...
system_process	61	8600
com.android.phone	121	8602
jp.co.omronsoft.openwnn	117	8603
com.android.launcher	126	8601
com.android.systemui	125	8604
android.process.media	243	8626
com.android.email	268	8628
com.android.protips	284	8630
com.android.quicksearchb	307	8634
com.svox.pico	316	8636
com.pennapps.stopwatch	336	8638 / 8700

Threads Panel:

ID	Tid	Status	utime	stime	Name
1	336	running	20309	3886	main
*2	338	vmwait	9	278	HeapWorker
*3	339	vmwait	157	18	GC
*4	340	vmwait	0	0	Signal Catcher
*5	341	running	119	105	JDWP
*6	342	vmwait	79	54	Compiler
7	343	native	0	0	Binder Thread #1
8	344	native	0	0	Binder Thread #2

LogCat Panel:

Time	pid	tag	Message
01-12 05:29...	I 336	dalvikvm	Debugger is active
01-12 05:30...	I 336	dalvikvm	dvmDdmHandleHpsgChunk(when 1, wha...
01-12 05:30...	D 336	dalvikvm	+++ active profiler count now 1
01-12 05:30...	I 336	dalvikvm	TRACE STARTED: '[DDMS]' 8192KB
01-12 05:30...	I 61	Activ...	Process com.android.mms (pid 228)...

Intro to Android Development

Market and Actual Testing?

- Windows > Android SDK & AVD Manager
- Available packages > Third party Add-ons
- Google Usb Driver and Market Licensing

On your actual phone:

- MENU > Settings > Applications > Development > USB debugging ON
- Plug in, compile and run your Android app, select your actual phone (or might auto-select)

Intro to Android Development

Publishing on the Market

- <http://market.android.com/publish/Home>
- <http://developer.android.com/guide/publishing/publishing.html>
- \$25 one-time fee for a developer account
- Make money back via either paid app or advertisements (AdMob)
- Compare to iOS's \$99/YEAR and content filtering/approval-based system
- Need to make sure to check support for different screen sizes, Android OS (current is 2.3 but lots still run 1.5/1.6), etc. in *AndroidManifest.xml* file

Intro to Android Development

Beats, Advanced Rhythm Game

- Website: <http://beatsportable.com>
- Try it: <http://www.tinyurl.com/beatspre14b>
- Examples of: graphics, multi-touch, synchronized audio, menus, settings, dialogs, intent-filters, multiple activities, persistent data, multi-threading, file browser, and more...

k.darktiger@gmail.com | [Home](#) | [Help](#) | [Android.com](#) | [Sign out](#)



Keripo

k.darktiger@gmail.com
[Edit profile »](#)

All Android Market listings



[Beats Advanced Rhythm Game](#)

v1.3b

Games: Arcade & Action

(929) ★★★★★☆

[Comments](#)

116189 total
43330 active installs (37%)

Free

[Errors \(8\)](#)

✓ Published

(screenshot taken Jan 12, 2011, prior to 1.4b release)



Upload Application

Intro to Android Development

• Useful Links/Resources

- *#android-dev* at freenode.irc.net
- Android Developers
<http://developer.android.com/index.html>
 - Dev Guide = fundamentals
 - Reference = Google's Javadocs
 - Resources = Tips and conventions
 - Blog = Newest updates, code examples
- *android-developers@googlegroups.com*
<http://www.mail-archive.com/android-developers@googlegroups.com>
- StackOverflow – public Q&A
<http://stackoverflow.com/questions/tagged/android>
- anddev.org – forums with lots of tutorials
<http://www.anddev.org/>
- App Inventor Beta – Google's test project
<http://appinventor.googlelabs.com/about/index.html>



ANDROID
developer lab

Intro to Android Development

Questions? Office Hours 8-10pm, Sat. Jan 16

The Joy of Tech™

by Nitrozac & Snaggy



©2008 Geek Culture

joyoftech.com