

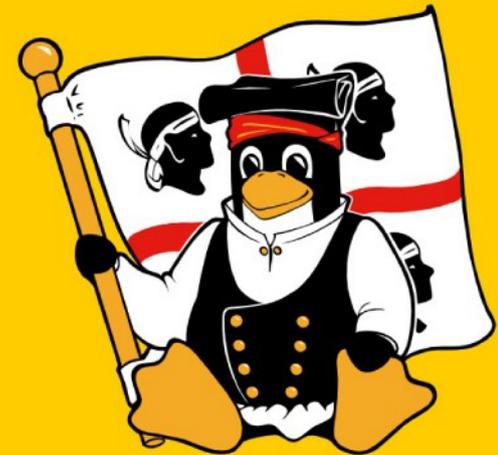
Sviluppare app per Android

di Paolo Cortis



GULCh

Gruppo Utenti Linux Cagliari h...?



Android world

- Un **insieme di software** per dispositivi mobili (smartphone, tablet, portatili...)
- Contiene un **Sistema Operativo**, software middleware e applicazioni varie
- Sviluppato da **Open Handset Alliance**, un insieme di grandi aziende guidate da Google
- Annunciato nel 2007
- Prima release 23 settembre 2008



Android 1.5 - Cupcake



Android 1.6 - Donut



Android 2.0 - Eclair



Android 2.2 - Froyo



Android 2.3 - Gingerbread



Android 3.0 - Honeycomb



Android 4.0 – Ice Cream Sandwich



Android 4.1 – Jelly Bean



Android 4.4 - KitKat



Perché scegliere Android?

	Android 4.3 	Apple iOS 7 	MS Windows Phone 8 
Device compatibili	Grande varietà di dispositivi (smartphone, tablet, pc) Multi produttori	iPhone, iPad, iPod touch Unico produttore	Smartphone Produttori: Nokia, Samsung, HTC, Huawei
Personalizzazioni	Widget, alternative per varie applicazioni di sistema (SMS, launcher, tastiera, etc.) Alcune ROM supportano i temi	Modifica dello sfondo.	Cambiare i colori delle tiles
Widgets	✓	✗	Riquadri nella home screen
Multitasking	✓	Da iOS 4 in poi (limitato)	Da Windows Phone 8 (limitato)
Tethering (Condivisione connessione)	✓	✓	✓
Storage espandibile	✓ Non sui dispositivi Nexus di Google	✗	✓
Trasferimento file	Cavo usb (mini, micro) in modalità MSC, MTP e PTP, bluetooth, wireless, etc	Tramite cavo proprietario, solo coniTunes. Bluetooth non utilizzabile per trasferimento file	Cavo usb (mini, micro), tramite Zune
Storage rimovibile	✓	✗	✗
NFC	✓	✗	✓
App nel market	1.000.000+ (Luglio 2013)	900.000+ (Giugno 2013)	145.000+ (Maggio 2013)
Market alternativi	✓ Amazon, AppBrain, etc.	✗	✗
Assistente vocale	✓ Google Now	✓ Siri	✗



Perché sviluppare per Android?

	Android 	Apple iOS 7 	MS Windows Phone 8 
Linguaggi	Java, C/C++, Adobe Air, Python, Perl, Javascript, ...	Objective-C, C/C++	C# (.NET), C++ (.NET)
Documentazione API	http://developer.android.com/ Completa, stile Javadoc. Presenti parecchi esempi.	http://developer.apple.com/library/ios Completa, disponibili parecchi esempi	http://msdn.microsoft.com/en-us/library/ff626516%28v%3Dvs.92%29.aspx Completa, documentazione caotica: oggetti delle singole classi separati in pagine differenti. Pochi esempi disponibili.
SDK+IDE	Android Studio, Eclipse (gratuiti e supportati ufficialmente) IntelliJ IDEA, Netbeans (gratuiti) Qualsiasi sistema operativo, Android compreso	Xcode (gratis), IDE di terze parti Richiesto Mountain Lion o superiore	Visual Studio .NET (Express: gratis, Professional: 1,477€, Premium: 7,534€, Ultimate: 16,374€), IDE di terze parti Limitazioni sulla versione Express Richiesto Windows 8
Costo market	25\$ alla registrazione	Gratuito per test solo sul pc con emulatore altrimenti 80€ all'anno, necessari anche per la pubblicazione.	\$19 all'anno per massimo 100 applicazioni gratuite, 19.99\$ per ogni applicazione gratuita successiva. App a pagamento illimitate.
Licenza	Apache Licence 2.0	Proprietario	Proprietario

<http://developer.android.com>
<http://developer.android.com/sdk/installing/studio.html>
<http://robtiffany.com/windows-phone-marketplace-free-app-submission-limit-increased-to-100/>
<http://dev.windowsphone.com/en-us/join>
http://www.microsoftstore.com/store/mseea/it_IT/list/Visual_Studio/categoryID.66235600
<https://developer.apple.com/programs/>
<https://itunes.apple.com/it/app/xcode/id497799835?mt=12>



Java, ma non Java: Dalvik

- Le app sono scritte in linguaggio Java, ma non è Java;
- Dalvik è l'equivalente della Java Virtual Machine;
- Architettura register-based (fino al 30% di operazioni in meno rispetto alla JVM);
- Codice compilato in bytecode Java;
- Conversione da bytecode Java (.class) a bytecode Dalvik (.dex);
- Ottimizzazioni per l'esecuzione su dispositivi mobili (meno memoria, cpu meno potente, presenza di batteria);



Concetti essenziali: AndroidManifest.xml

- File in formato XML contenente informazioni essenziali sulle applicazioni.
- Fornisce, tra le altre, le seguenti informazioni al sistema:
 - Identificatore dell'applicazione (Java package);
 - I componenti forniti dall'applicazione (Activity, Service, Receivers, etc);
 - I permessi necessari per il funzionamento;
 - Versioni compatibili di Android.



Concetti essenziali: AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="it.gulch.android.test_manifest"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="4"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

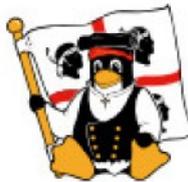
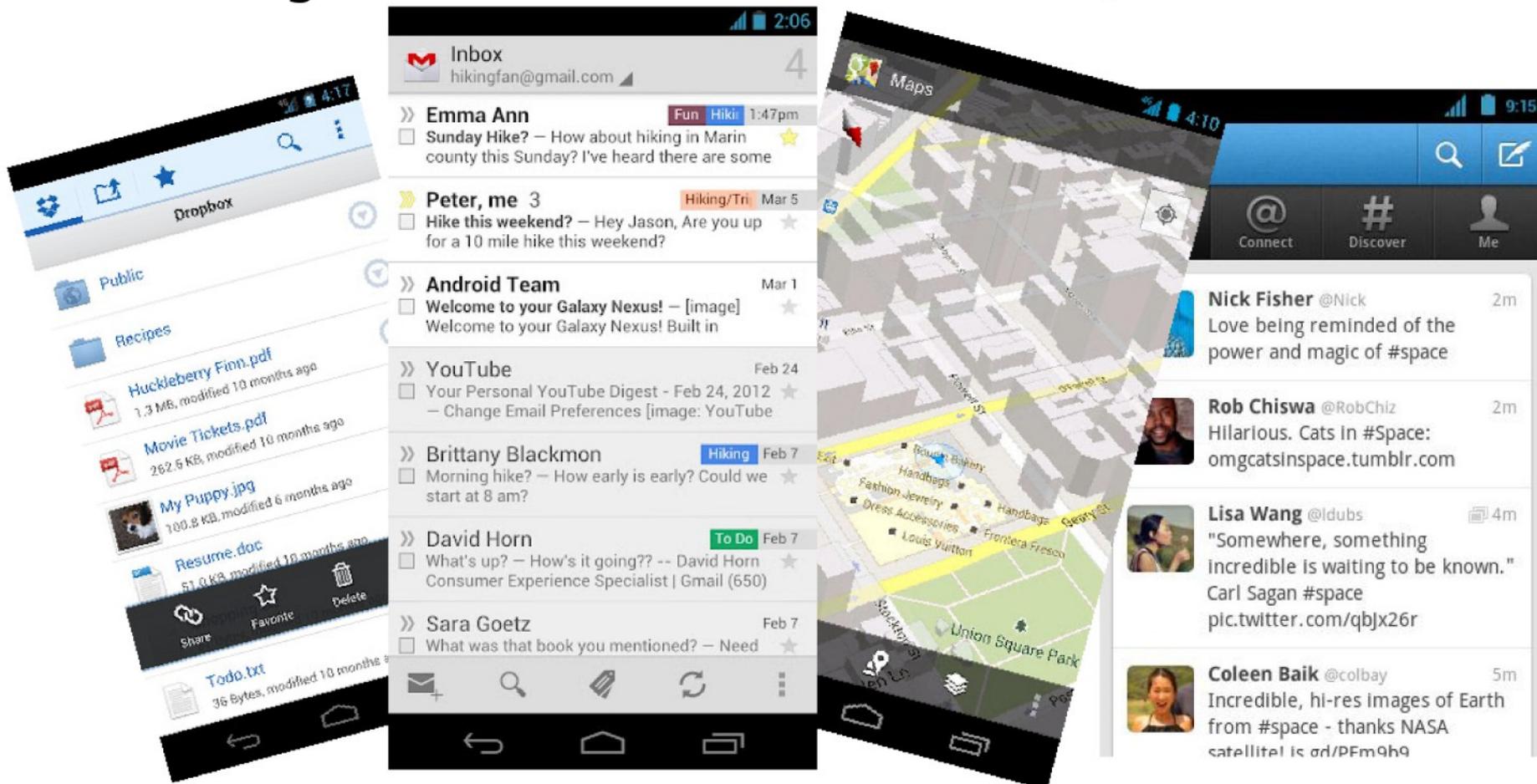
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

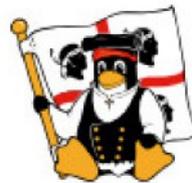
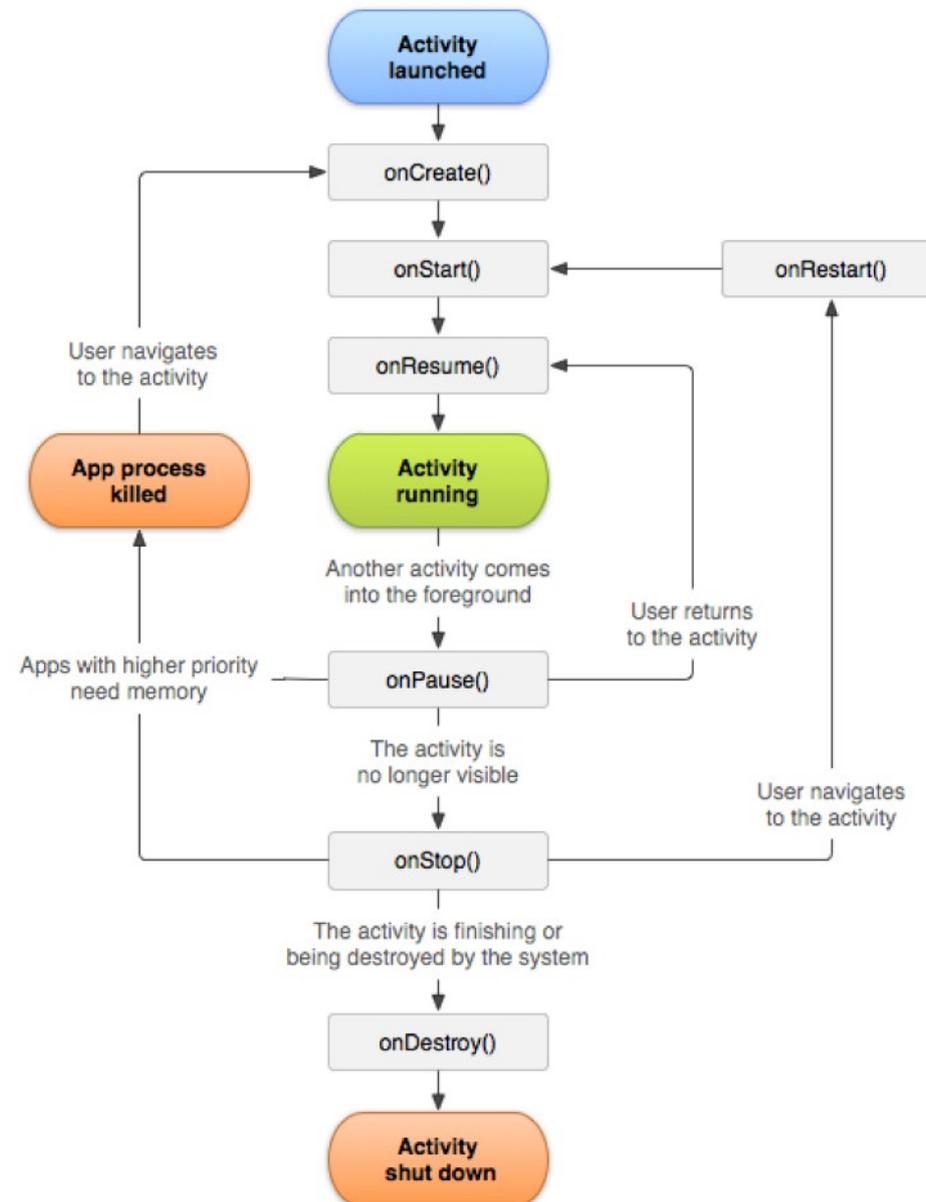


Concetti essenziali: Activity

- Rappresenta le interazioni con l'utente, associabile al concetto di schermata;
- Un'applicazione è formata da una o più Activity collegate assieme tramite Intent;

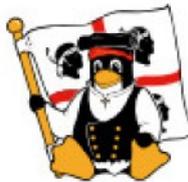


Ciclo di vita di una Activity



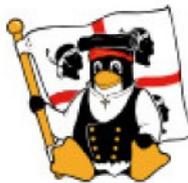
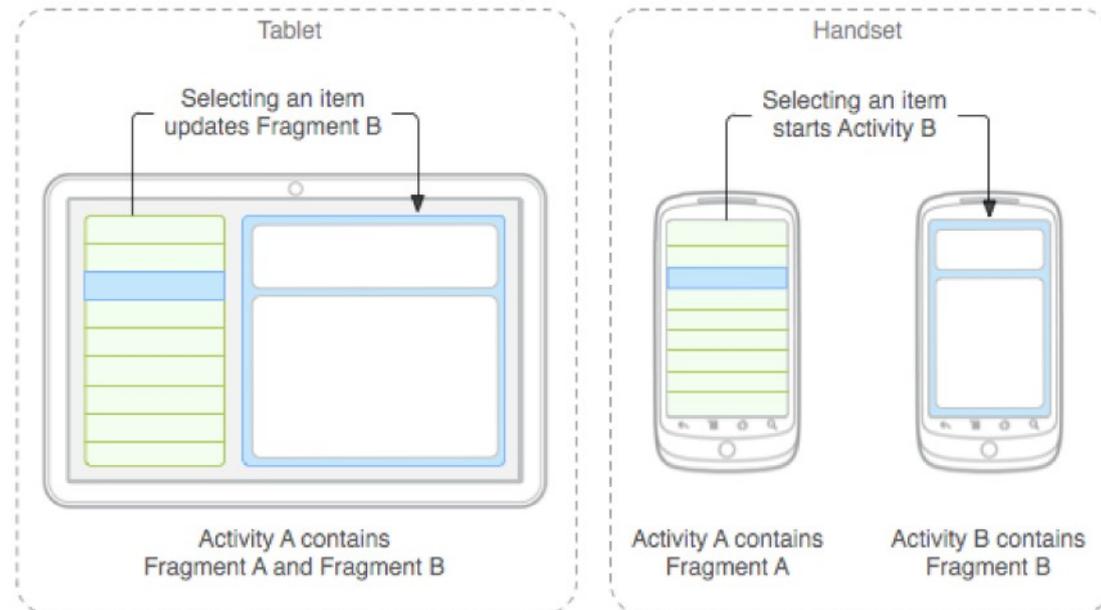
Concetti essenziali: Intents e BroadcastReceiver

- Un Intent è una descrizione astratta di una certa operazione da eseguire (lancio Activity, invio di un “messaggio” ad un componente dell'applicazione, ...)
- I BroadcastReceiver sono i ricevitori dei messaggi inviati tramite Intent.



Concetti essenziali: Fragment

- Una porzione di interfaccia utente all'interno di una particolare activity.
- Permette il riuso ed evita la duplicazione di codice.
- Ogni fragment può essere aggiunto, rimosso, modificato a runtime.
- Possiede un suo particolare ciclo di vita, simile a quello delle Activity.



Concetti essenziali: Multithreading e Service

- Alcune applicazioni o parti di esse non hanno bisogno di interfacce grafiche.
- Il multithreading permette di eseguire azioni (anche lunghe) in background
- I Service hanno priorità più elevata rispetto ad altri processi: il sistema non li eliminerà in caso di scarse risorse.
- Servizi “locali” (Started service) e “remoti” (Bound service)



Concetti essenziali: le risorse

- Fondamentalmente si tratta di immagini, testi, layouts, drawables e assets.
- Sono accessibili tramite una particolare classe autogenerata: R
- Layout
 - File XML che rappresentano l'interfaccia grafica
- Drawable
 - Qualsiasi componente “visualizzabile”
- Assets
 - File a disposizione dell'applicazione. Non sono accessibili tramite la classe R, ma tramite path.

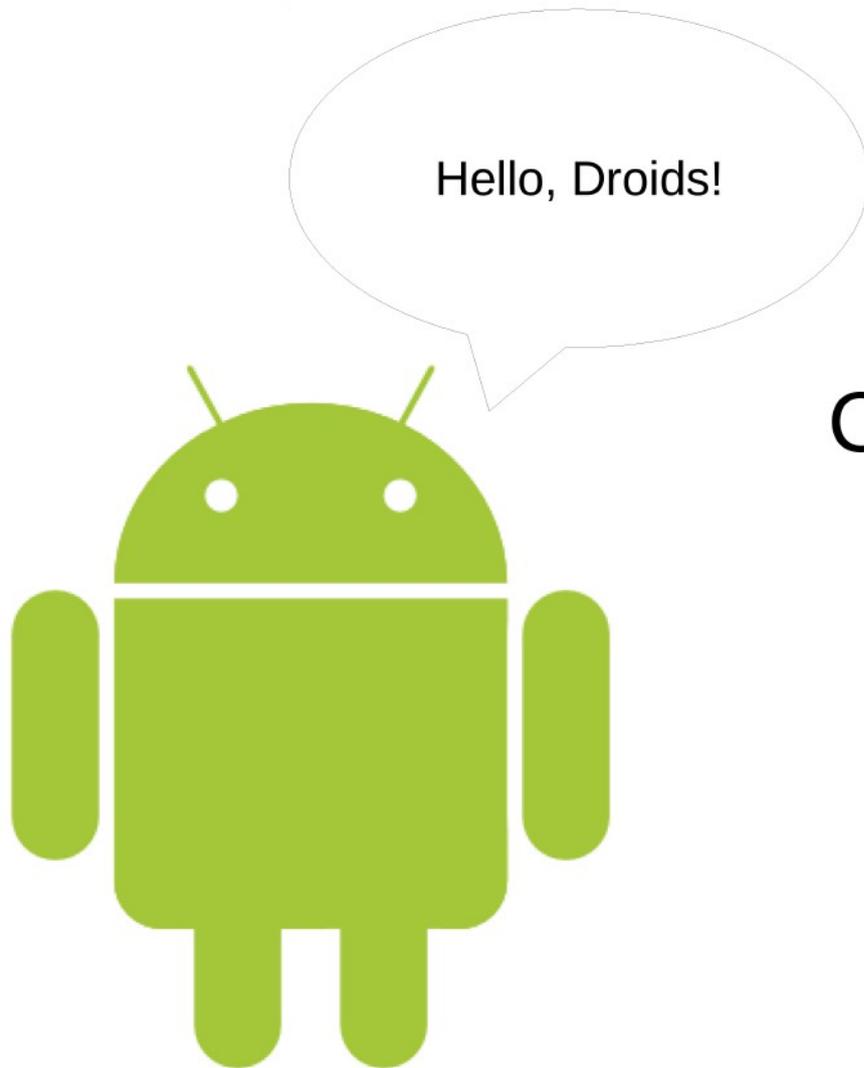


Android e i permessi

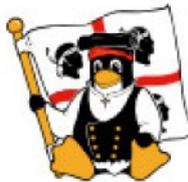
- Ogni applicazione viene eseguita in una sandbox.
- Normalmente non può accedere a nessuna risorsa esterna (rete, sensori, fotocamera, etc)
- Si dichiarano nel file `AndroidManifest.xml` i permessi a cui un'app può accedere.
- Permette all'utente di sapere cosa può fare un'app e decidere se installarla o meno.
- È buona norma non usare troppi permessi in una sola app.



Benvenuti su Android: Hello, world



Code time!



Applicazioni multi-dispositivo

- Una app deve poter girare su dispositivi molto differenti
 - Risoluzione dello schermo
 - Hardware a disposizione (WiFi, 3G, Sensori...)
- e su versioni di Android diverse
 - Nuove API disponibili ad ogni nuova versione
 - Nuovi widget grafici
 - Alcune API vengono deprecate



La classe `android.os.Build`

- Contiene le classi **VERSION** e **VERSION_CODES**
 - **VERSION**: Stringhe della versione corrente
 - **VERSION_CODES**: Elenco delle versioni di Android
- Elenca alcune informazioni sull'hardware e sulla build del sistema operativo

- Esempio:

```
if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN) {  
    ...  
} else {  
    ...  
}
```



Connessioni disponibili: ConnectivityManager

- Usiamo il ConnectivityManager per conoscere quali tipi di connessione possiede il dispositivo
- Verifichiamo le reti 3G, LTE, WiFi, WiMax, Bluetooth
- Esempio:

```
public boolean isConnectedToWifi(Context context)
{
    ConnectivityManager connManager = (ConnectivityManager)
        context.getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo info;
    info = connManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI);
    if(info == null) {
        return false;
    }

    return info.isConnected();
}
```



I sensori: la classe `SensorManager`

- Indica se esiste un particolare sensore
- Recupera una istanza di una classe `Sensor` per ottenere informazioni dal sensore che ci interessa
- Esempio:

```
SensorManager sensorManager;  
List<Sensor> sensors;
```

```
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);  
sensors = sensorManager.getSensorList(Sensor.TYPE_ACCELEROMETER);
```



Orientamento

- Gestiamo l'orientamento tramite le directory di resource (res/layout-port, res/layout-land)
- Non sempre è necessario specificare più layout per le diverse orientazioni
- Possiamo recuperare l'orientazione attuale via codice (`getResources().getConfiguration().orientation` nella classe `Context`)



Tablet

- I parametri per le directory resource sono:
 - sw600dp (tablet 7")
 - sw720dp (tablet 10")
- Usiamo le risorse booleane per indicare se l'app sta usando un layout da tablet o meno
- Esempio:
 - In values/booleans.xml

```
<resources>
  <bool name="isTablet">false</bool>
</resources>
```
 - In values-sw600dp/booleans.xml

```
<resources>
  <bool name="isTablet">>true</bool>
</resources>
```



Tablet

- Sfruttiamo appieno lo spazio aggiuntivo a disposizione
 - Layout multi-pane
 - Diversa disposizione degli elementi nel layout
 - Riutilizziamo il codice sfruttando i Fragment
- Facciamo attenzione ai diversi sensori e hardware a disposizione
 - Non tutti i tablet hanno connessione 3G
 - Alcuni dispongono di tastiera fisica



Tablet (Twitter)

Home Connetti Scopri Account

Slashdot @slashdot 15h
IE 11 Breaks Rendering For Google Products, and Outlook Too bit.ly/GXh6cl

XDA Developers @xdadevelopers 18h
Return to the Top in BlinkFeed with a Single Tap goo.gl/IVH1jl

Slashdot @slashdot 18h
NSA App Ideas To Popularize Spying and Big Data bit.ly/1d9l0Ng

Android Police @AndroidPolice 19h
[New Game] Namco Takes Air Combat Intergalactic With Sky Gamblers: Air Supremacy andp.lc/16lvJ4S

Android Police @AndroidPolice 19h
AT&T And Verizon Versions Of The One Max Press Images Leaked, With Annoying Carrier Logos Where HTC's Should Be andp.lc/16lrPsZ

MobileSyrup @MobileSyrup 21h
Top trending stories this past week:
1: Nexus 5 press images goo.gl/bGg4tP
2: BlackBerry Z30 now available goo.gl/7ZQ9gH

Slashdot @slashdot 21h
Your Next Network Operating System Is Linux bit.ly/17D97YE

Android Police @AndroidPolice 22h
[The Android Police Podcast] Episode 82: Foiled By The G Flex andp.lc/16leHnr

23:39



Tablet (Twitter)

Paolo Cortis
@paolinux86
Cagliari

Home

Connetti

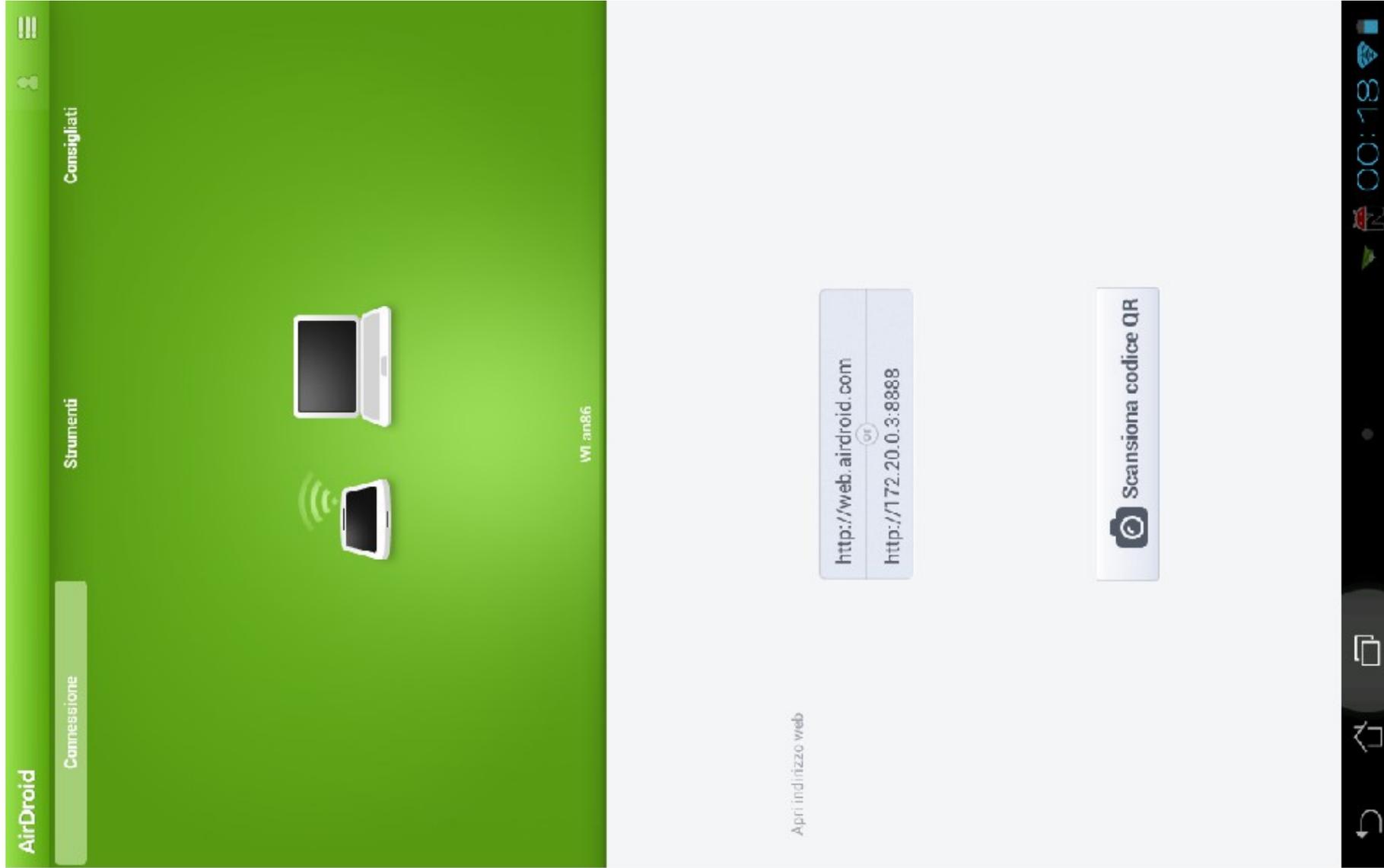
Scopri

- Slashdot** @slashdot 15h
IE 11 Breaks Rendering For Google Products, and Outlook Too bit.ly/GXh6cl
- XDA Developers** @xdadevelopers 18h
Return to the Top in BlinkFeed with a Single Tap goo.gl/lVH1jl
- Slashdot** @slashdot 18h
NSA App Ideas To Popularize Spying and Big Data bit.ly/1d9l0Ng
- Android Police** @AndroidPolice 19h
[New Game] Namco Takes Air Combat Intergalactic With Sky Gamblers: Air Supremacy andp.lc/16lvJ4S
- Android Police** @AndroidPolice 19h
AT&T And Verizon Versions Of The One Max Press Images Leaked, With Annoying Carrier Logos Where HTC's Should
- MobileSyrup** @MobileSyrup 21h
Top trending stories this past week:
1: Nexus 5 press images goo.gl/bGg4tP
2: BlackBerry Z30 now available goo.gl/7ZQ9gH
- Slashdot** @slashdot 21h
Your Next Network Operating System Is Linux bit.ly/17D97YE
- Android Police** @AndroidPolice 22h
[The Android Police Podcast] Episode 82: Foiled By The G Flex andp.lc/16leHnr
- Android Police** @AndroidPolice 22h
[The Android Police Podcast] Episode 82: Foiled By The G Flex andp.lc/16leHnr

23:39



Tablet (AirDroid)



Domande?

