



Sviluppare applicazioni per Android

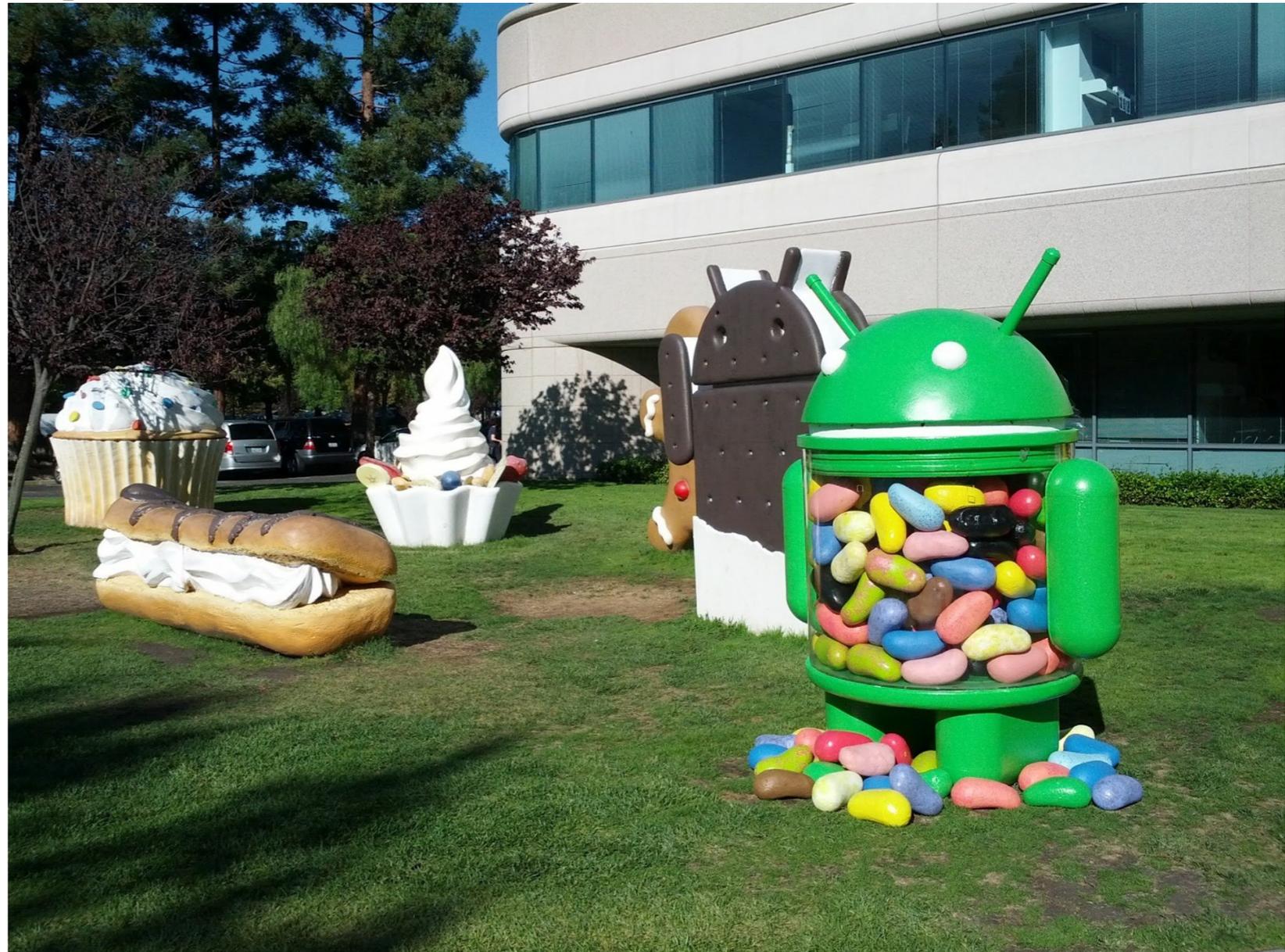
Paolo Cortis





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

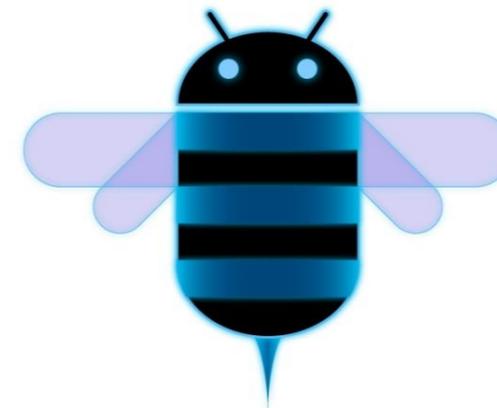




Versioni di Android

- I nomi seguono un ordine alfabetico, si ispirano a nomi di dolci:

- 1.0 (?)
- 1.1 (?)
- 1.5 **Cupcake**
- 1.6 **Donut**
- 2.0, 2.1 **Eclair**
- 2.2 **Froyo**
- 2.3 **Gingerbread**
- 3.0, 3.1, 3.2 **Honeycomb**
- 4.0 **Icecream sandwich**
- 4.1 **Jelly bean**
- 4.2 **Key Lime Pie (?)**





Perché scegliere Android?

| | Android 4.1  | Apple iOS 6  | MS Windows Phone 8  |
|--------------------------------------|--|---|--|
| Device compatibili | Grande varietà di dispositivi (smartphone, tablet, pc) | iPhone, iPad, iPod touch | Solo alcuni dispositivi |
| Personalizzazioni | Quasi tutti gli elementi della GUI sono sostituibili Alternative per varie applicazioni di sistema (SMS, rubrica, etc.) | ✗ | Sulla versione 8 è possibile cambiare i colori delle tiles e scegliere suonerie custom |
| Widgets | ✓ | ✗ | Riquadri nella home screen |
| Multitasking | ✓ | Da iOS 4 in poi (limitato) | Da Windows Phone 8 (limitato) |
| Tethering (Condivisione connessione) | ✓ | ✓ | ✓ |
| Storage espandibile | ✓ | ✗ | ✓ |
| Trasferimento file | Cavo usb (mini, micro) in modalità MSC, MTP e PTP, bluetooth, wireless, etc | Tramite cavo proprietario, solo attraverso iTunes | Cavo usb (mini, micro), tramite Zune |
| Storage rimovibile | ✓ | ✗ | ✗ |
| App nel market | 600.000+ | 650.000+ | 100.000+ |
| Assistente vocale | ✓ Google Now | ✓ Siri | ✗ |



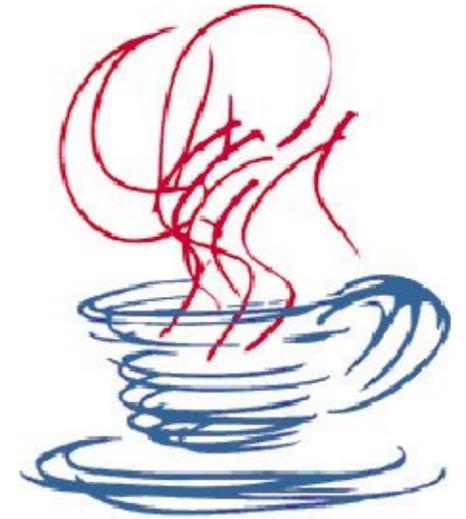
Perché sviluppare per Android?

| | Android  | Apple iOS  | MS Windows Phone  |
|--------------------|--|--|---|
| Linguaggi | Java, C/C++, Adobe Air, Python, Perl, Javascript, ... | Objective-C, C/C++ | 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=vs.92%29.aspx Completa, documentazione caotica: oggetti delle singole classi separati in pagine differenti. Pochi esempi disponibili. |
| SDK+IDE | Eclipse, IntelliJ IDEA, Netbeans (gratuiti) | Xcode (gratis), IDE di terze parti | Visual Studio .NET (Express: gratis, Professional: 1.599€, Premium: 6.999€, Ultimate: 15.299 €), IDE di terze parti |
| Costo market | 25\$ alla registrazione | Gratuito per test solo sul pc con emulatore altrimenti, da 99\$ fino a 299\$ all'anno | \$99 all'anno per massimo 100 applicazioni gratuite, 19.99\$ per ogni applicazione gratuita successiva |
| Licenza | Apache Licence 2.0 | Proprietario | Proprietario |



Java, ma non Java: Dalvik

- Applicazioni solitamente scritte in 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" />

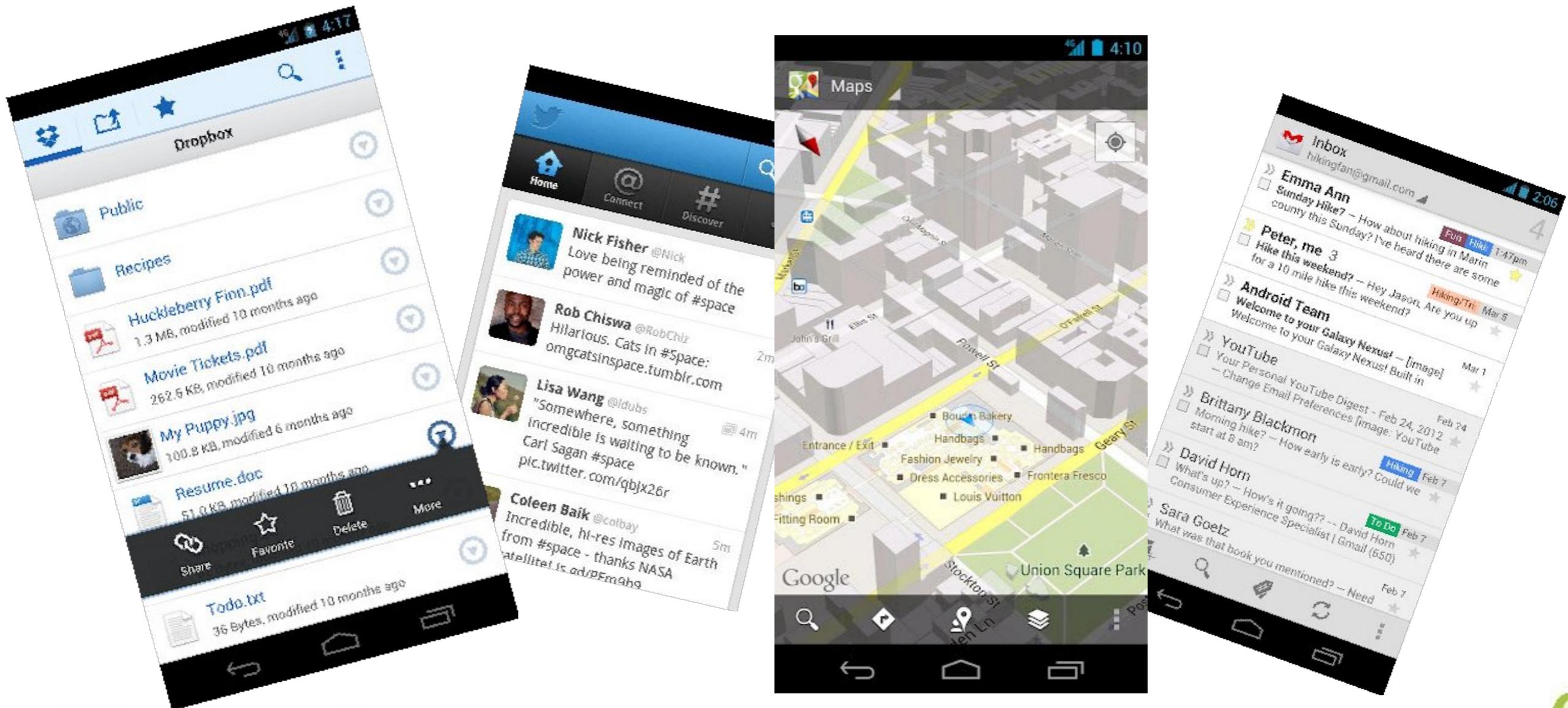
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

</manifest>
```



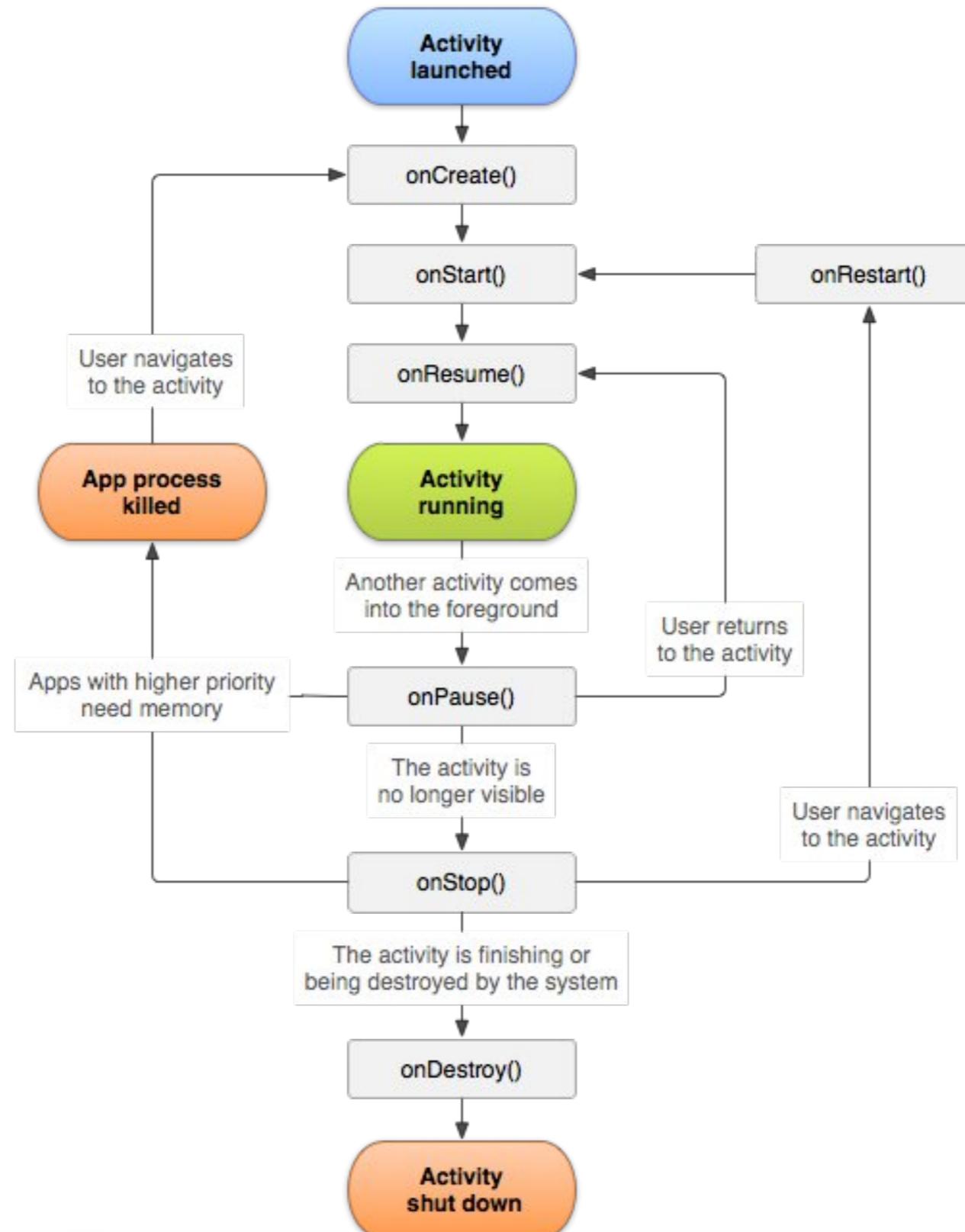
Concetti essenziali: Activity

- 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 BroadcastReceivers

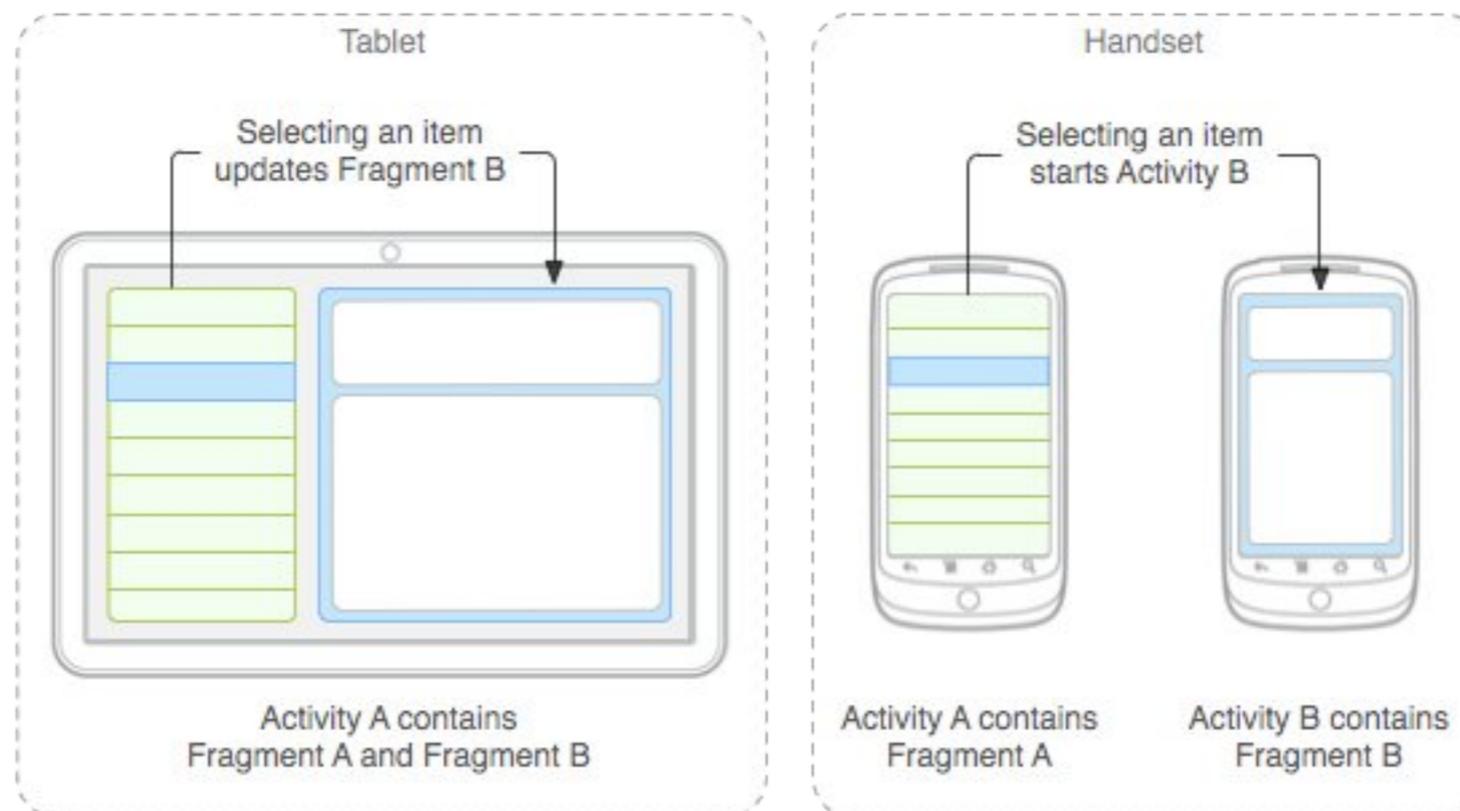
- Un Intent è una descrizione astratta di una certa operazione da eseguire (lancio Activity, invio di un “messaggio” ad un componente dell'applicazione, ...)
- I BroadcastReceivers sono i ricevitori dei messaggi inviati tramite Intent.





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 una 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



Hello, Droids!

Code time!



Anche gli androidi sanno navigare

- Le app android possono “uscire” dalla sandbox ed effettuare comunicazioni di rete.
- Serve almeno il permesso “android.permission.INTERNET”
- Le comunicazioni di rete non possono essere eseguite sul thread principale (della User Interface)
- Utilizziamo le API fornite da Apache HttpClient



Anche gli androidi sanno navigare

Code time!



Android e i sensori

- I dispositivi Android sono dotati di sensori per misurare il movimento, l'orientazione e alcune condizioni ambientali.
- Android gestisce 3 categorie di sensori:
 - Sensori di movimento
Misurano le forze di accelerazione e rotazione. Comprende accelerometro e giroscopio
 - Sensori ambientali
Misurano diversi parametri ambientali (temperatura dell'aria, pressione, illuminazione, etc). Esempio: barometro e termometro.
 - Sensori di posizionamento
Misurano la posizione del dispositivo. Esempio: bussola e magnetometro.



Android e i sensori: Accelerometro

- L'accelerometro misura l'accelerazione impressa al dispositivo.
- Misura anche la forza di gravità
- Per misurare l'accelerazione reale è necessario rimuovere la componente fornita dalla gravità
- Spesso utilizzato nei giochi
- Proviamo ad utilizzarlo per eseguire un'azione quando giriamo il dispositivo



Android e i sensori: Accelerometro

Code time!



Guadagnare con la nostra applicazione

- Upload dell'applicazione sui market.
 - 30% del guadagno va a Google/Amazon
 - vendita di applicazioni Freemium
- Servizi In-App / DLC
 - Servizi o funzionalità aggiungibili dopo il download dell'applicazione "core".
- Banner pubblicitari in-App
 - Poco graditi dagli utenti
 - L'utente può disattivarli (Rom custom o con rooting)



Domande?



GRAZIE PER L'ATTENZIONE