Android e Appinventor
lo stato dell'arte

LIKE THAT, BUT BETTE

LINUX DAY 2011 - Ivrea - 22 OTTOBRE

Alessandro Orlandi & Ettore Bartoccetti

- **Origini di android**
- **Le maggiori piattaforme su cui gira: tablet, cellulare....**
- **Le quote di mercato**
- **Gli ambienti di sviluppo professionali**
- **Appinventor: piattaforma didattica di sviluppo applicazioni**
- **Android una piattaforma open o chiusa?**

Android e Appinventor
lo stato dell'arte

COS'E' ANDROID

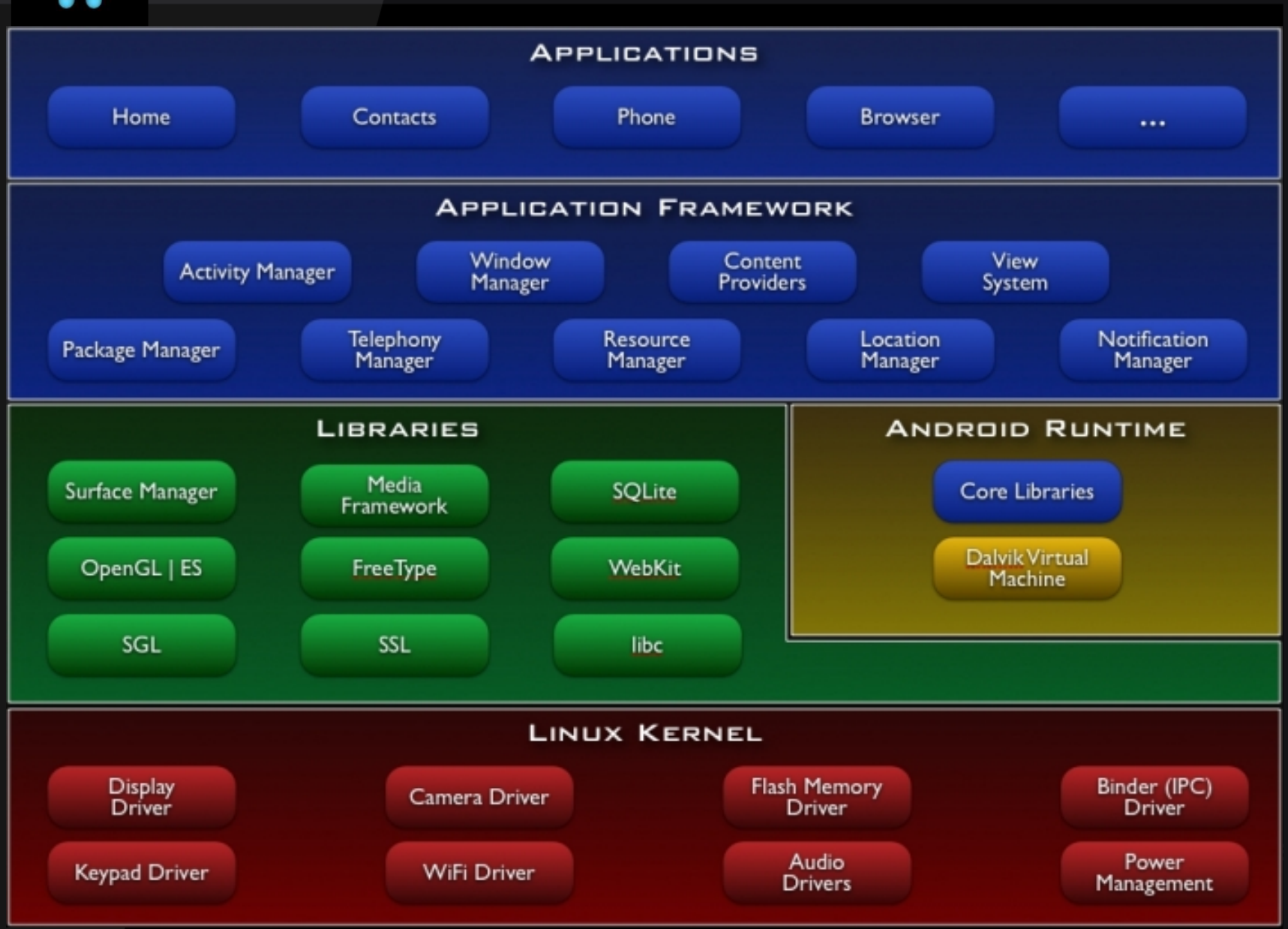
Android
è un
sistema operativo
per
dispositivi mobili
basato su kernel
Linux

COMPONENTI OPEN

Altri componenti open di Android

- Open GL - *SGI Free Software License B*
- SGL (Scene Graph Library) - *LGPL*
- SQLite - *Public domain*
- WebKit - *LGPL*

Com'è fatto...


Applications
Home
Contacts
Phone
Browser
...
Application Framework
Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager
Libraries
Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc
Android Runtime
Core Libraries
Dalvik Virtual Machine
Linux Kernel
Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

LE ORIGINI

2003 - California - Android Inc.

A. Rubin, R. Miner, C. White e N. Sears fondano la **Android Inc.** con l'obiettivo di creare un sistema operativo mobile open source basato sul **kernel Linux**

Rubin è ora Senior Vice President della Google Mobile Division

2005 - "Big G" acquisisce Android Inc.

.... e fonda la **Open Handset Alliance**

Open Handset Alliance

OHA è un ente, capeggiato da Google, per la definizione di standard aperti per dispositivi mobili; è costituito da:

- Operatori mobili

Vodafone, Telecom Italia, China Mobile, ...

- Produttori di semiconduttori

Texas Instrument, Nvidia, Synaptics, ...

- Produttori di dispositivi mobili

HTC, Samsung, LG, Sony, ...

- Produttori di software

Google, eBay, PacketVideo, Ascender, ...

- Compagnie di commercializzazione

Aplix, Wind River Systems, ...

LE ORIGINI

2007 - Prima versione di Android (SO + SDK)
2008 - HTC Dream
il primo smartphone Android

- una **tastiera fisica QWERTY** a scomparsa
- schermo da **3.2 pollici**
- 192 MB di **Ram**
- 256 MB di **memoria flash**

...ma **Android** è ancora "piccolo":

- manca il supporto per il BT
- manca il touch-screen
- ...

LE ORIGINI

2009 - HTC Magic
il secondo smartphone Android

- **scompare** la **tastiera fisica (touchscreen)**
- supporto per **BT**
- **API per Gtalk**

Samsung segue a ruota con il Galaxy i7500

Esplode il fenomeno Android.

Innumerevoli produttori lanciano sul mercato dispositivi mobile basati su Android.

UN S.O. DOLCISSIMO...

A ciascuna release di **Android** viene assegnato il nome di un **dessert** (in ordine alfabetico crescente)

...e ad ogni rilascio, davanti alla sede di Mountain View, viene installato un pupazzo rappresentante il dessert

TABLET

Intanto sul mercato impazza la "corsa al tablet" in cui la Apple la fa da padrone con il suo iPad.

Le versioni di **Android dalla 1.0 alla 2.3** si sono evolute moltissimo dimostrandosi sempre più adeguate e complete per l'uso su dispositivi mobili ma

INADEGUATE per i TABLET!!

All'inizio del 2011 Google corre ai ripari
e tira fuori dal cilindro...

Honeycomb

Android 3.0 "Honeycomb"
(Gennaio 2011)

E' una versione di Android ottimizzata per i tablet.

In meno di un anno, escono altre 2 versioni evolutive.

- Android 3.1
  - miglioramento della gestione del multitasking
  - resize delle finestre
  - gestione USB Host
- Android 3.2
  - ottimizzazione per display a 7"
  - supporto per i processori Qualcomm
- Android 4.0 Ice cream sandwitch- 19 ottobre -
  - un sistema unico per tablet & smarthphone

Honeycomb

Il primo tablet con Honeycomb 3.0 è stato il **Motorola XOOM**

Open...

Android nasce come progetto "open"
e sulla carta lo è:

- molti dei componenti sono aperti

kernel, webkit, sqlite, ...

- i sorgenti sono disponibili sotto **licenza Apache**

...tuttavia...

LIKE THAT. BUT BETT

...and close

Uno studio **comparativo**[1], evidenzia che l'indice di Openess **effettiva** è molto basso:

- Eclipse 84%
- Linux 71%
- Meego 61%
- Android **23%**

Le aree su cui è valutata l'Openess sono:

1. Accesso: disponibilità dei sorgenti, supporto agli sviluppatori, trasparenza nelle decisioni
2. Sviluppo: possibilità degli sviluppatori di influenzare l'andamento del progetto
3. Derivati: possibilità di creare opere (sw) derivate
4. Community: una community che non discrimina tra gli sviluppatori

[1] A new way of measuring Openness, from Android to WebKit: The Open Governance Index - *L. Laffan*

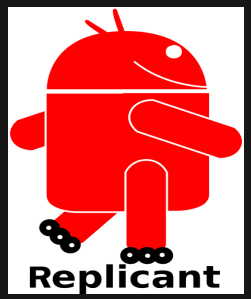
LIKE THAT. BUT BETT

"Big G" chiude Android

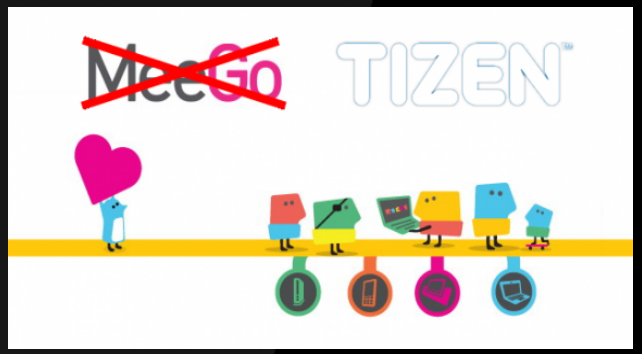
Google decide di mettere un accesso ristretto
su Android 3.0 "Honeycomb"
(la versione tablet-oriented di Android)

Questa versione, di fatto, è "closed source" e
l'accesso ai sorgenti è garantito solo ad un
gruppo ristretto di sviluppatori.

Alcune alternative



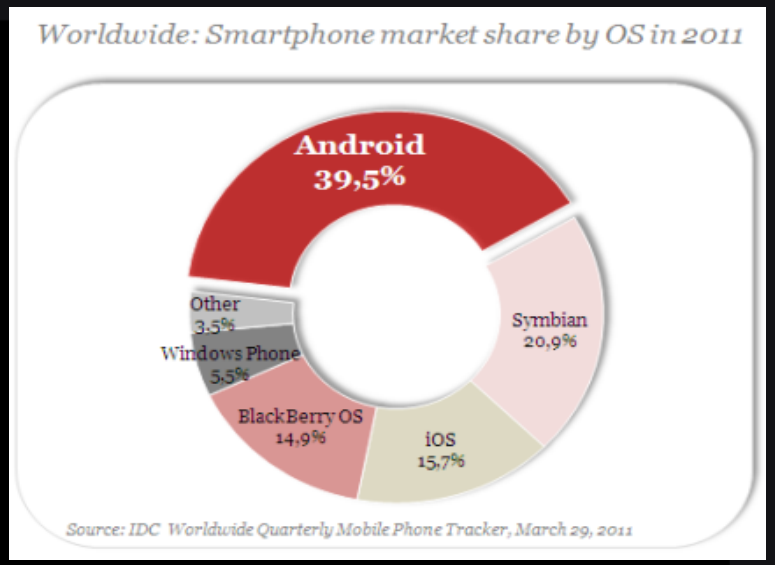
Replicant
è una distribuzione di Android completamente "free".
Purtroppo, al momento, i dispositivi supportati sono pochi e non recentissimi.



Tizen
è l'*erede* di MeeGo.
Si tratta di un OS per dispositivi mobili e tablet.

Il progetto, che vede coinvolta Samsung in prima linea, è completamente open ed ospitato dalla **Linux Foundation.**
Il primo SDK è atteso per inizio 2012.

LE QUOTE DI MERCATO PER IDC



Worldwide Smartphone Operating System 2011 and 2015 Market Share and 2011-2015 CAGR

Operating System	2011 Market Share	2015 Market Share	2011-2015 CAGR
Android	39.5%	45.4%	23.8%
BlackBerry	14.9%	13.7%	17.1%
iOS	15.7%	15.3%	18.8%
Symbian	20.9%	0.2%	-65.0%
Windows Phone 7/Windows Mobile	5.5%	20.9%	67.1%
Others	3.5%	4.6%	28.0%
Total	100.0%	100.0%	19.6%

LE QUOTE DI MERCATO PER GARTNER

ANDROID

Worldwide Mobile Communications Device Open OS Sales to End Users by OS (thousands of units)				
OS	2010	2011	2012	2015
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research in Motion	47,452	62,600	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,560	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	296.647	467,701	630,476	1,104,898

AMBIENTI DI SVILUPPO PROFESSIONALI

Gli strumenti essenziali per sviluppare applicazioni per Android sono:

Android SDK
(Android Software Development Kit)

Java Development Kit (JDK)

Questi possono essere utilizzati direttamente da CLI ma è molto più comodo farli pilotare a

Eclipse
dotandolo dei seguenti plug-in
ADT (Android Development Tools)
JDT (Java Development Tools)

AMBIENTI DI SVILUPPO PROFESSIONALI

Le piattaforme per la combo di sviluppo

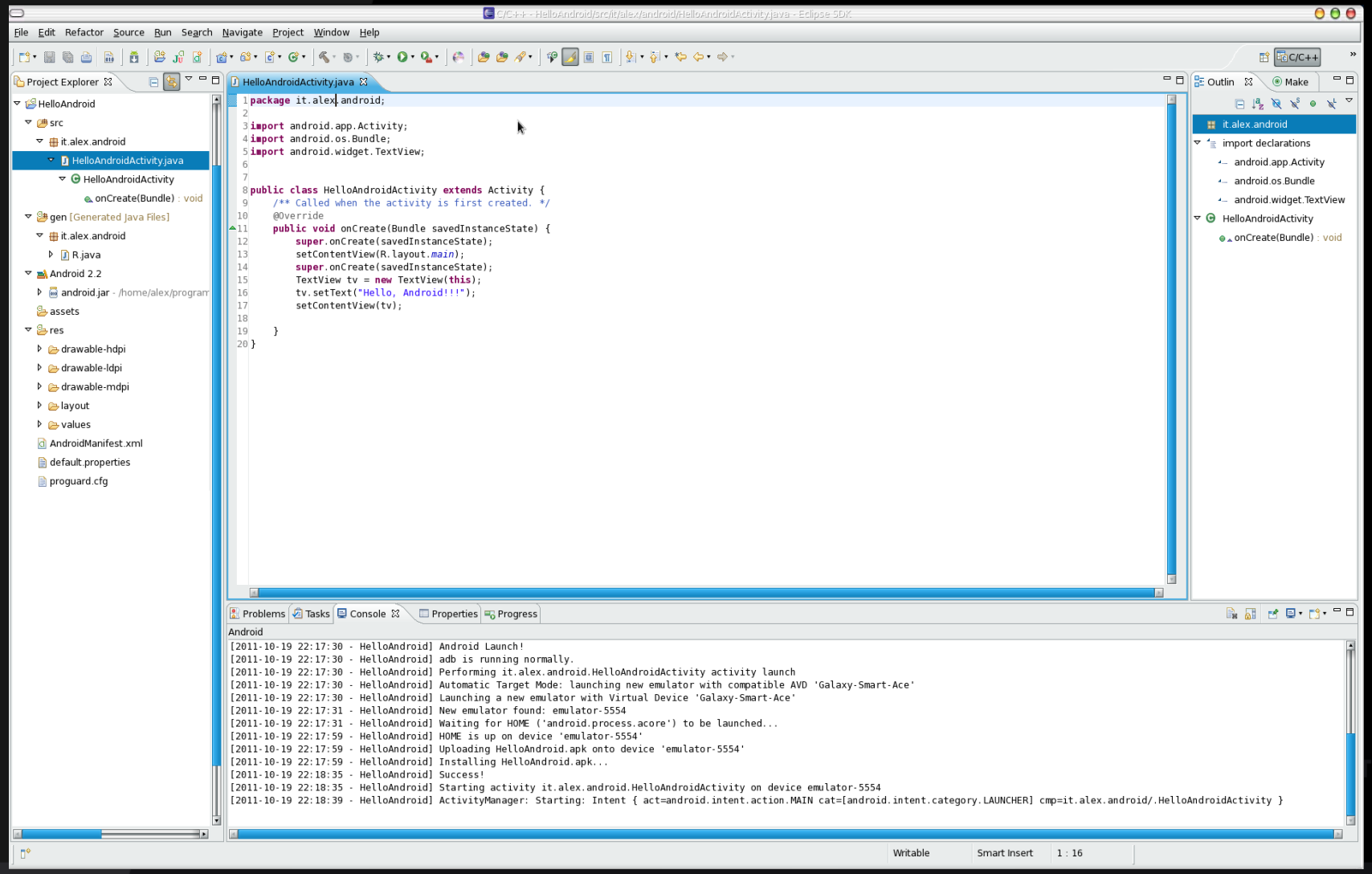
Android SDK + JDK + Eclipse

sono

AMBIENTI DI SVILUPPO PROFESSIONALI

Eclipse + Android SDK + JDK

=



Eclipse + Android SDK + JDK

Scrivere un'applicazione per Android

di fatto significa

scrivere codice in Java
(utilizzando le API della versione di Android prescelta)

e

realizzare dei layout basati su XML

...ma è Java???

Ah, quindi le applicazioni per Android
sono applicazioni in Java?! NO!!!

Il programma
viene
scritto con la **sintassi Java**
ma
l' "eseguibile" prodotto
NON è **bytecode per JVM**
(i classici file *.class*)
bensì
bytecode per Dalvik
(file con estensione *.dex*)

Dall'Islanda con furore...

Cos'è Dalvik?

Dalvik è una macchina virtuale (VM)
ottimizzata per sfruttare la poca memoria
presente nei dispositivi mobili.

La logica di base
è simile a quella della Java VM ma
il bytecode (*.dex*) è differente.

Il nome **Dalvik** deriva dal villaggio Islandese di pescatori **Dalvíkurbyggð** di cui è originaria la famiglia di **Bornstein** (il progettista di Dalvik).

Ricapitolando...

La ricetta per realizzare un programma

Sorgente Java

```
package it.alex.android;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

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

    }
}
```

Layout (XML)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" android:orientation="vertical">
    <TextView android:id="@+id/textview"
      android:layout_width="fill_parent"
      android:text="@string/hello" android:layout_height="42dp"/>
    <RatingBar android:id="@+id/ratingBar1" android:layout_width="wrap_c
</LinearLayout>
```

Android SDK

Il **layout** può essere *disegnato* servendosi della comoda ***GUI WYSIWYG*** in Eclipse

file **.apk**

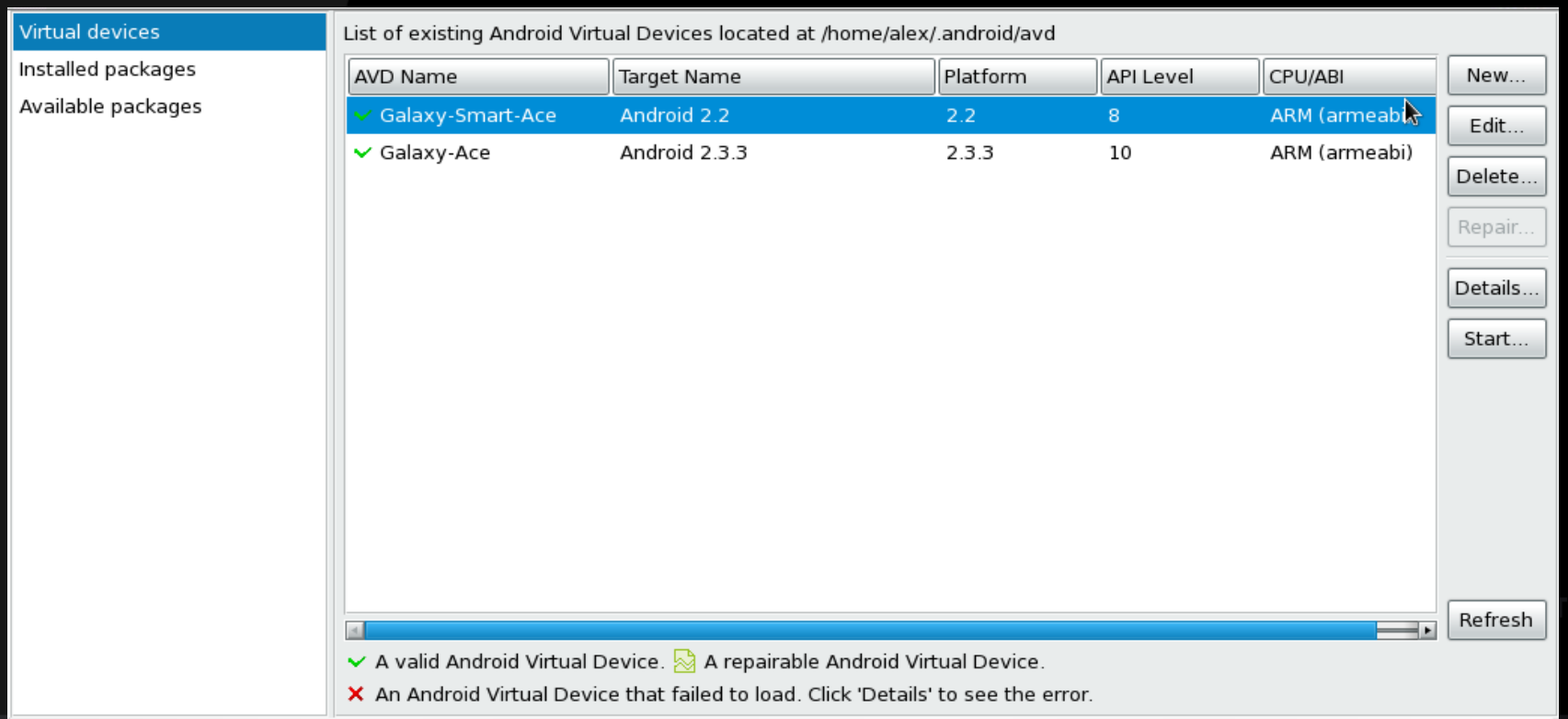
Cos'è un .apk?

Un file con estensione **.apk** (**A**ndroid **P**ac**K**age)
è un **archivio** compresso (simile a .jar e .zip)
che **contiene tutti i file necessari**
per far girare il programma su Android.

In particolare contiene
il file *classes.dex*
che è il bytecode binario
che viene dato in pasto a Dalvik.

Android Virtual Device

Attraverso l'**SDK** ed eventualmente con l'ausilio della **GUI di Eclipse** è possibile creare degli **AVD (Android Virtual Device)** che consentono di emulare le diverse versioni di Android su cui è possibile testare l'applicazione.

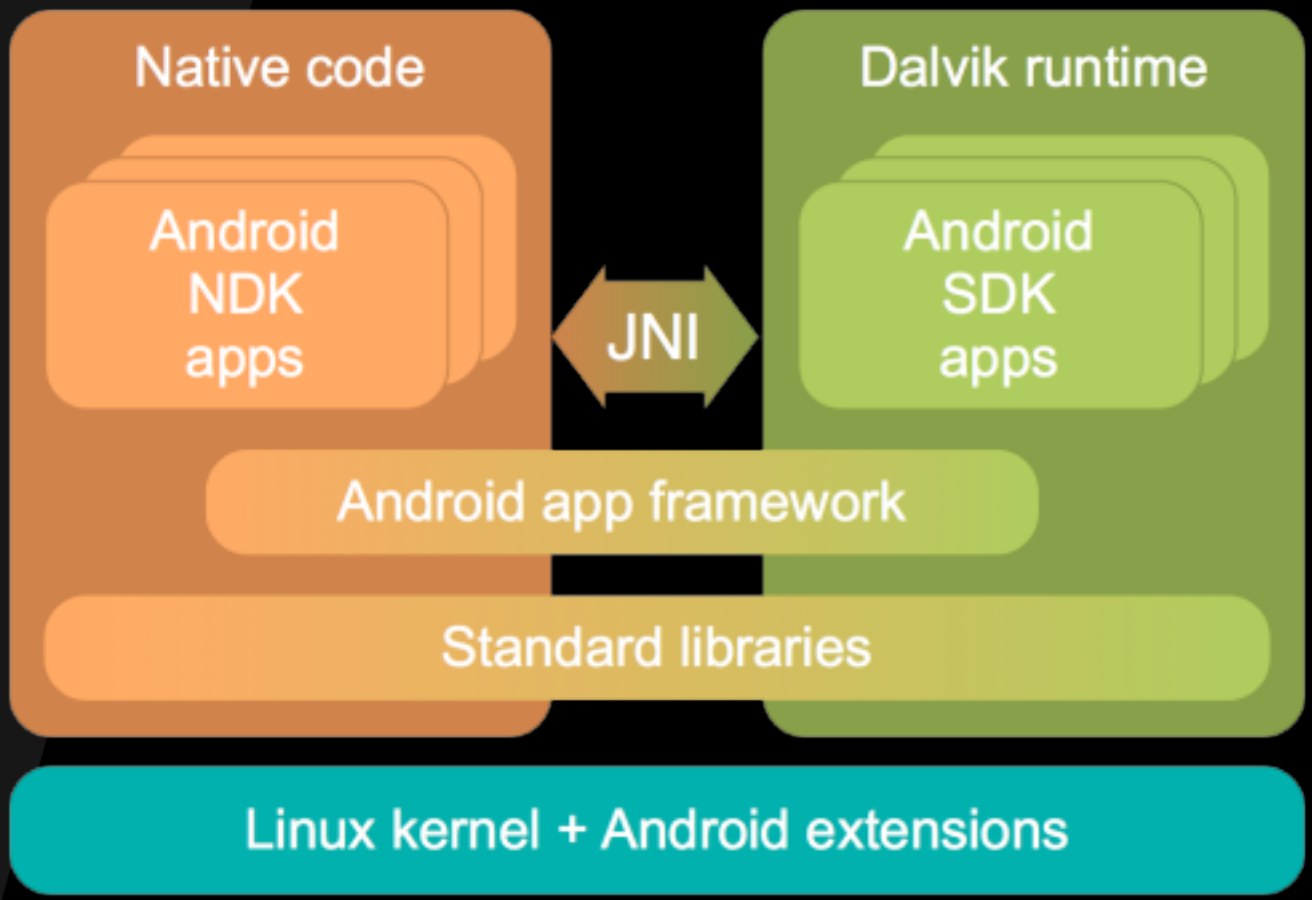


Dove me lo metto l'apk???

L'apk può essere testato su uno (o più) **AVD**...

...e successivamente caricato su
un dispositivo *mobile* Android
(smartphone, tablet, etc.)

I layer sw di Android



APPINVENTOR
...per i comuni mortali

Google App Inventor è un'applicazione fornita da Google che permette a chiunque di creare applicazioni software per il sistema operativo Android.

L'applicazione si rivolge a persone che NON hanno familiarità con la programmazione del computer .

Il ragionamento è molto semplice: se le persone hanno la possibilità di sviluppare applicazioni per soddisfare i propri bisogni e di installarle semplicemente sul proprio telefono cellulare, saranno incentivate a passare al sistema operativo Android.

Le applicazioni potranno essere facilmente distribuite sia per canali privati che, con poche modifiche, attraverso l'android market.

Che cosa ci offre APPINVENTOR ?

Oltre ad un’ampia dotazione di strumenti per la grafica, la registrazione e riproduzione di suoni e la navigazione Web, tra i molti componenti di rilievo si evidenziano:

- Gestione della comunicazione bluetooth in modalità client e server,
- Lettura dei sensori presenti all’interno di cellulari e tablet, quali accelerometri, bussole, GPS etc
- Barcode scanner, utilizzando le webcam in dotazione ai cellulari o ai tablet
- Speech recognition
- Text to Speech
- Set comandi per Lego Mindstorm
- Connessione ad API di Social Network come Twitter

APPINVENTOR: che cosa occorre ?

- Un computer collegato ad internet
  - Macintosh (con processore Intel): Mac OS X 10.5, 10.6
  - Windows: Windows XP, Windows Vista, Windows 7
  - GNU/Linux: Ubuntu 8+, Debian 5+
  - Java 6 (conosciuto anche come Java 1.6)
- un browser per www.appinventorbeta.com
- un account google (es. mario.rossi@gmail.com)
- un software da installare: App Inventor Setup (disponibile per Mac, Windows e naturalmente Linux)
- il software fornisce anche un emulatore sw di un telefono cellulare Android per provare i nostri programmi
- ...infine per utilizzare le nostre creazioni...
- l'ideale è avere un cellulare con sistema operativo Android o un tablet sempre con s.o. Android....ma se non li abbiamo possiamo sempre provare i nostri programmi sul telefono di un amico :-)

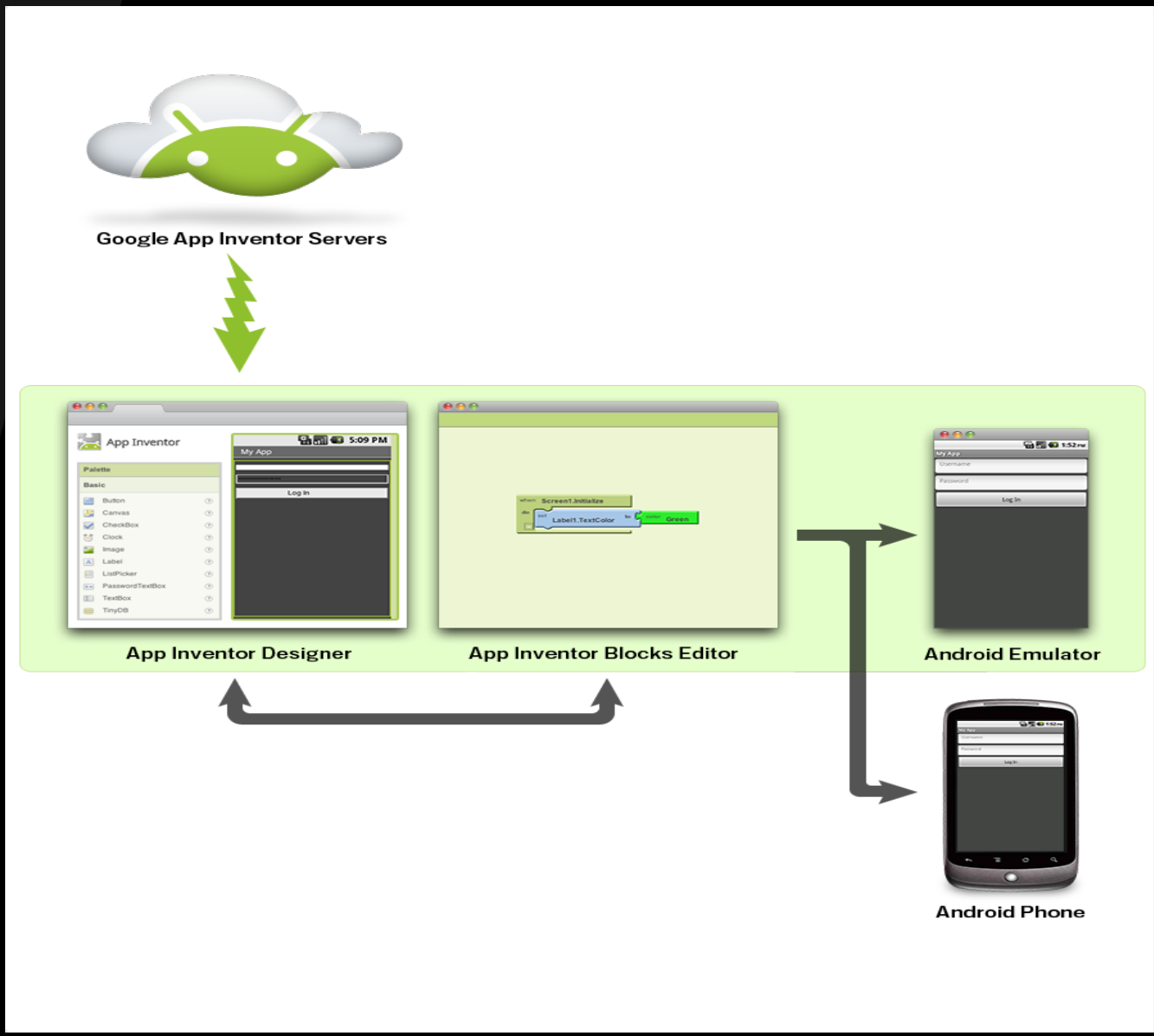
Cosa possiamo fare con APPINVENTOR ?

Con la varietà di componenti messi a disposizione e' possibile utilizzare App Inventor per sviluppare gran parte delle applicazioni Android, ed il tutto senza scrivere il codice: in pratica si progetta visivamente l'aspetto dell'applicazione e si utilizzano i blocchi logici per specificarne il comportamento.

Come abbiamo detto, insieme al sistema vengono forniti esempi funzionanti naturalmente in formato Open Source.

E' solo necessario aggiungere la nostra curiosità, fantasia e voglia di provare per ideare e realizzare nuovi programmi e nuove applicazioni.

ARCHITETTURA



APPINVENTOR, sulle spalle dei giganti

"Nel creare App Inventor per Android, siamo fortunati ad essere in grado di attingere significativi ***precedenti ricerche nel campo dell'informatica educativa****, e il* ***lavoro svolto in Google su ambienti di sviluppo on-line****. L'editor a blocchi utilizza la Open Block Java Library per la creazione di linguaggi di programmazione basati su blocchi visuali distribuita dal Massachusetts Institute of Technology - MIT "*

La programmazione a Blocchi Visuali è strettamente correlata al ***linguaggio di programmazione Scratch*** *, un progetto del MIT Media Laboratory del Lifelong Kindergarten Group .*

Il compilatore che traduce il linguaggio visivo a blocchi per l'implementazione su Android utilizza il framework del ***linguaggio Kawa****, un derivato del linguaggio di programmazione* ***Scheme****, sviluppato da Per Bothner e distribuito come parte del* ***sistema operativo Gnu dalla Free Software Foundation*** *.*

La prospettiva educativa che motiva App Inventor sostiene che la programmazione può essere un veicolo potente per coinvolgere le idee attraverso l'apprendimento attivo. Come tale, essa è parte di un movimento in corso nel computer e l'educazione che ***ha avuto inizio con il Logo Gruppo MIT nel 1960****.*

,

LIKE THAT. BUT BETT

DESIGNER

E' la GUI Web attraverso la quale selezionare i componenti necessari alla realizzazione dell'App:

- Componenti base:

Button, Canvas, Checkbox, Clock, Image, Label, ListPicker, PasswordTextBox, TextBox,TinyDB

- Contributi multimediali:

Camera, ImagePicker, Player, Sound, VideoPlayer

- Animazioni:

Ball, ImageSprite

- Social:

ContactPicker, EmailPicker, PhoneCall, PhoneNumberPicker, Texting, Twitter

- Controllo Sensori:

AccelerometerSensor, LocationSensor, OrientationSensor

segue....

DESIGNER

...altre componenti del Designer

- Gestione schermo:

HorizontalArrangement, TableArrangement, VerticalArrangement

- Lego (R) Mindstorms (R)

Lego Mindstorms è una linea di prodotti LEGO che combinano mattoncini programmabili con motori elettrici, sensori, mattoncini LEGO, pezzi di LEGO Technic (come ingranaggi, assi e parti pneumatiche) per costruire robot e altri sistemi automatici e/o interattivi

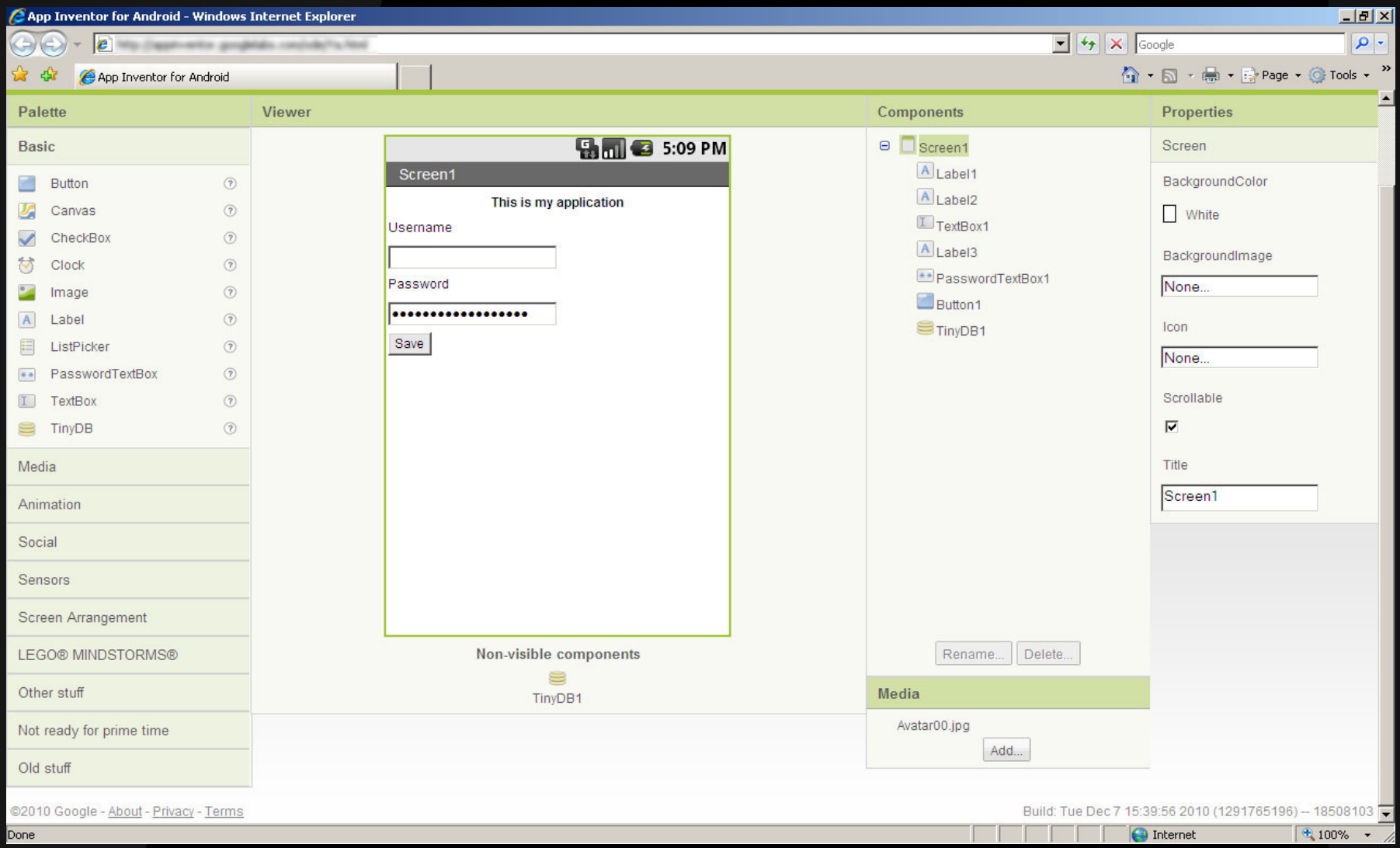
- Altri componenti:

ActivityStarter, BarcodeScanner, BluetoothClient, BluetoothServer, Notifier, SpeechRecognizer, TextToSpeech, TinyWebDB, Web

- e per finire...

GameClient, SoundRecorder, Voting, WebViewer

DESIGNER

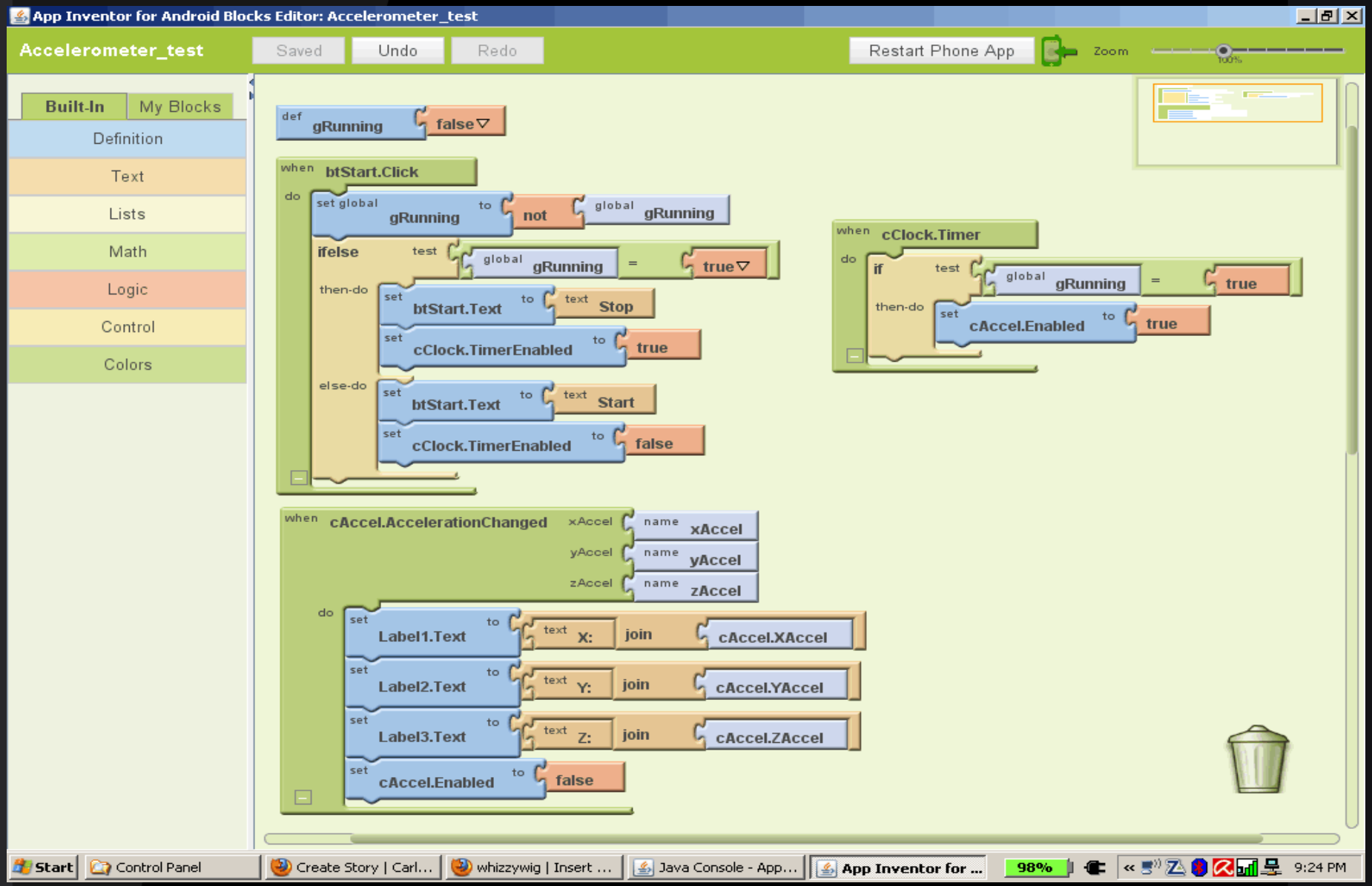

App Inventor for Android - Windows Internet Explorer
App Inventor for Android
Palette
Basic
Button
Canvas
CheckBox
Clock
Image
Label
ListPicker
PasswordTextBox
TextBox
TinyDB
Media
Animation
Social
Sensors
Screen Arrangement
LEGO® MINDSTORMS®
Other stuff
Not ready for prime time
Old stuff
Viewer
5:09 PM
Screen1
This is my application
Username
Password
Save
Non-visible components
TinyDB1
Components
Screen1
Label1
Label2
TextBox1
Label3
PasswordTextBox1
Button1
TinyDB1
Rename...
Delete...
Media
Avatar00.jpg
Add...
Properties
Screen
BackgroundColor
White
BackgroundImage
None...
Icon
None...
Scrollable
Title
Screen1
©2010 Google - About - Privacy - Terms
Build: Tue Dec 7 15:39:56 2010 (1291765196) -- 18508103
Done
Internet
100%

BLOCKS EDITOR

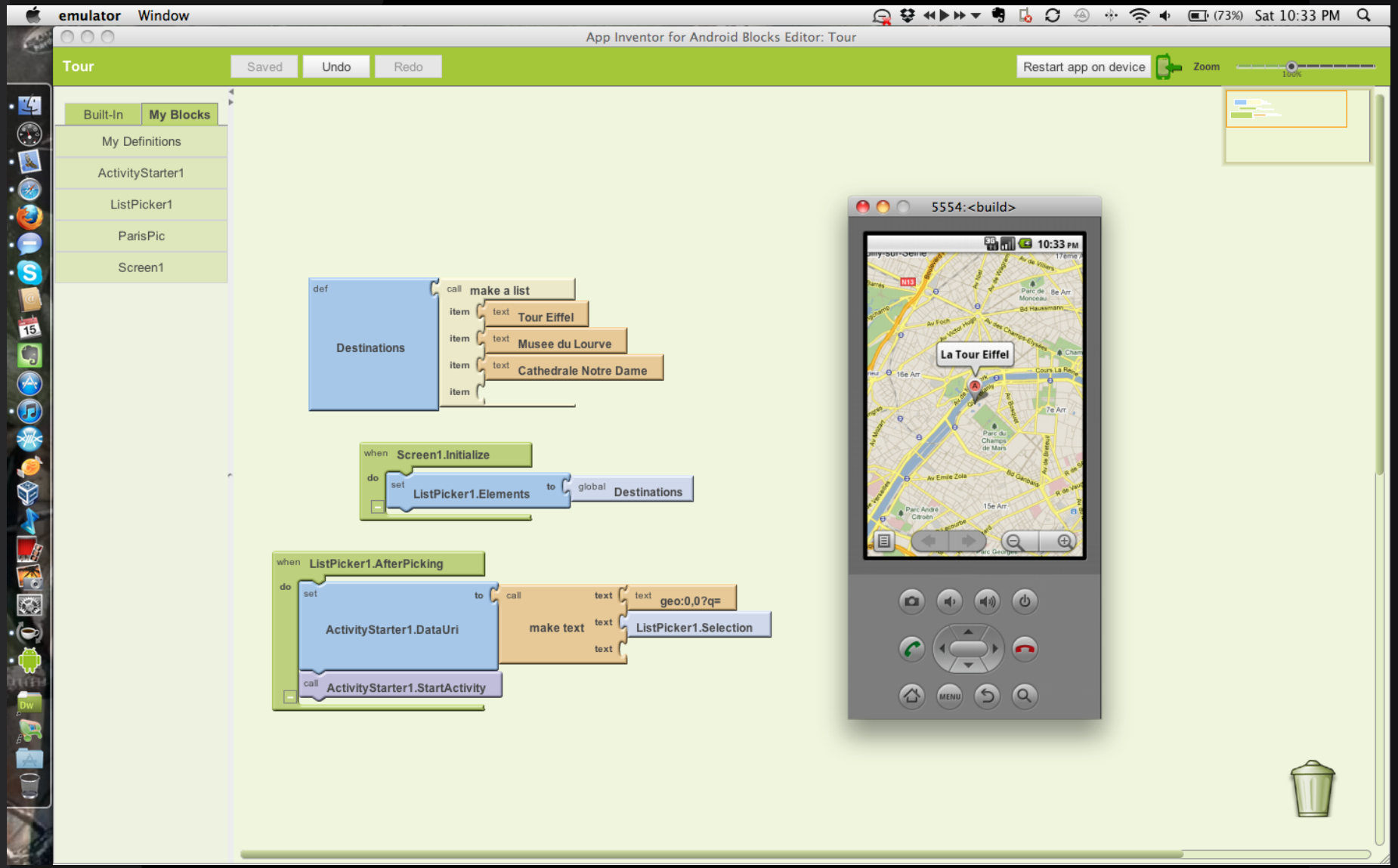
E' una applicazione Java lanciata dal Designer con al quale vengono assemblati I blocchi di programma specificandone il comportamento.

La cosa simpatica e’ che i blocchi logici vengono interconnessi tra loro come se si stesse componendo un puzzle: i pezzi se sono corretti devono incastrarsi, se "non si incastrano" c'è un errore di sintassi

BLOCKS EDITOR



ANDROID EMULATOR



ANDROID EMULATOR

E' l'applicazione java che emula uno smartphone Android sul Pc per verificare passo passo come l’ app si comporta.
Una volta soddisfatti dell’app costruita si attiva la funzione che impacchetta il tutto e rende il contenuto installabile su un vero terminale Android (scaricandola sul terminale vervo con il cavetto di collegamento o inviandolo via mail, o condividendo il file .apk da installare)

Come pubblicare sull'Android Market le APP di APPINVENTOR

Grazie al Tool Marketizer di Gene Kupter "quasi" con un singolo click i nostri lavor creato un account per il Market (costo una tantum di 25 $).
Con questo tool (che richiede Microsoft .NET framework 3.5 per funzionare, e gira solo sotto Windows) è possibile infatti:

- Creare il certificato digitale richiesto dal Google Market
- Decomprimere il file .apk
- Modificare il file Androidmanifest.xml
- Aggiungere e riscalare l'icona personalizzata per la nostra app
- Ricomprimere il file .apk
- Eseguire lo zipalign e la firma del file .apk

Tutto il processo richiede pochi minuti di lavoro, possiamo ad ogni modo modificare manualmente AndroidManifest.xml nel caso ce ne fosse bisogno oppure accettare le scelte di default che si adattano alla maggior parte dei casi.

Sempre da questo tool possiamo inoltre installare comodamente la nostra app (preparata per il market) sul nostro smartphone per testarla a dovere prima dell'upload sul market.

Dove trovare esempi di applicazioni

da provare, modificare

Gli esempi ufficiali forniti da App Inventor

- http://www.appinventorbeta.com/learn/tutorials/index.html

Applicazioni realizzate all'University of San Francisco dagli studenti e dal Prof.Wolber

- http://www.appinventorbeta.com/learn/tutorials/index.html

Altri esempi di App Inventor...

- http://ai.kittywolf.net/index.php/Main_Page

tAIR - the AI Repository .info

- http://www.theairepository.info/

App Inventor Blocks

- http://www.appinventorblocks.com

A BREVE:1 LIBRO + 1 EBOOK

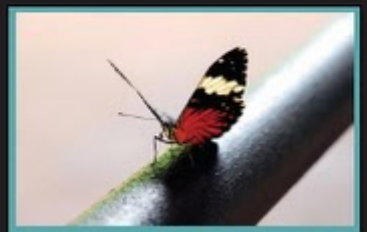
Google App Inventor *Beginner's Guide*

A beginner's guide that teaches you all the basics of App Inventor and how you can create apps with lots of simple and fun examples and tips

Google App Inventor Beginner's Guide eliminates the mystery around programming. This book shows you how to use a visual language, where we simply drag-and-drop blocks (graphic elements representing blocks of code) in various combinations to give us applications that run on our phones or other Android-based devices. No programming background is required. Playing with blocks has never been more fun! By an author with over 100 books, who keeps it entertaining, informative, and memorable. You'll be inventing apps from the first day.

What this book will do for you...

- Install App Inventor (did we mention it's free?), and use an emulator and your phone as test devices
- Package apps, install them on your phone, share with friends, and publish them to Market and other places
- Design apps by choosing the right components and formatting the screens so that everything looks as good as it works!
- Learn how the blocks click together and smoothly perform the tasks your app needs to accomplish, and understand which blocks do what
- Explore the advanced concepts of App Inventor, such as the ActivityStarter, which lets AI communicate with and manipulate other apps
- Create apps that e-mail, text, tweet, use databases, find GPS coordinates, play games, and a great deal more!



The Beginner's Guide approach...

- Clear step-by-step instructions for the most useful tasks
- Learn by doing, start working right away
- Leave out the boring bits
- Inspiring, realistic examples give you ideas for your own work
- Tasks and challenges encourage experimentation

$ 39.99 US
£ 24.99 UK

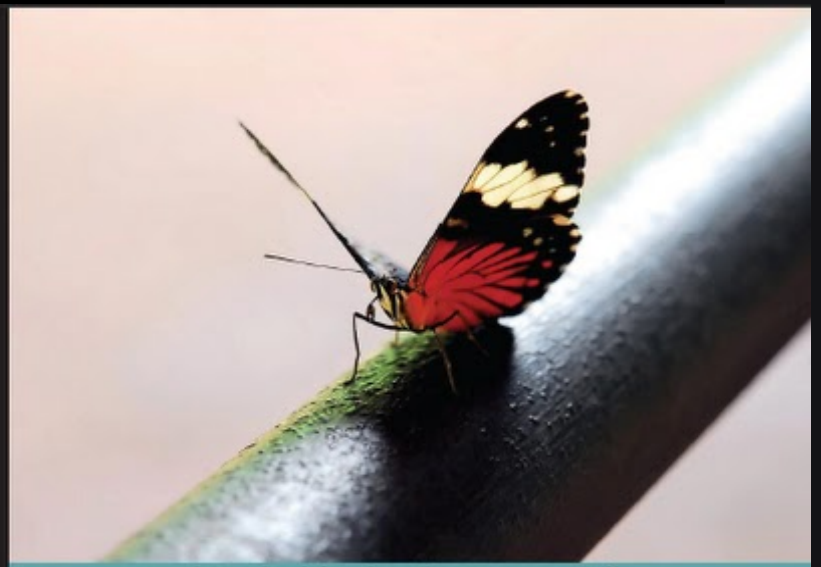
Prices do not include local sales tax or VAT where applicable

ISBN 978-1-849690-20-1

[PACKT] PUBLISHING "Community Experience Distilled"

Visit www.PacktPub.com for books, eBooks, code, downloads, and PacktLib.

Google App Inventor *Beginner's Guide* Ralph Roberts [PACKT]



Learn by doing: less theory, more results

Google App Inventor

Create powerful Android apps the easy all-visual way with Google App Inventor

Beginner's Guide

Ralph Roberts

[PACKT] PUBLISHING