Accelerating Information Technology

http://aiti.mit.edu

Ghana Summer 2012
Android Intro and Layouts

What is Android?

- Google's operating system for mobile devices
  - Open-source
  - Free
  - International
  - Java-based
  - Hardware cross-compatible (HTC, Motorolla, LG, Samsung, Huawei, …)

What can Android do?

Example applications

- GPS based weather
- Barcode scanner
- Google maps
- Games– Angry Birds
- gmote – media remote control for computer

Hardware capabilities

- GPS
- Wifi + 3G, Bluetooth (Comms)
- Gyroscopes and accelerometers
- Sensors (e.g. light)
- Still and video cameras
- External Storage (SD card)

Software Capabilities

- 2D/3D Graphics
- Built-in SQLite Database
- Gesture detection, Multi-touch
- Text-to-speech, speech recognition

Monetization

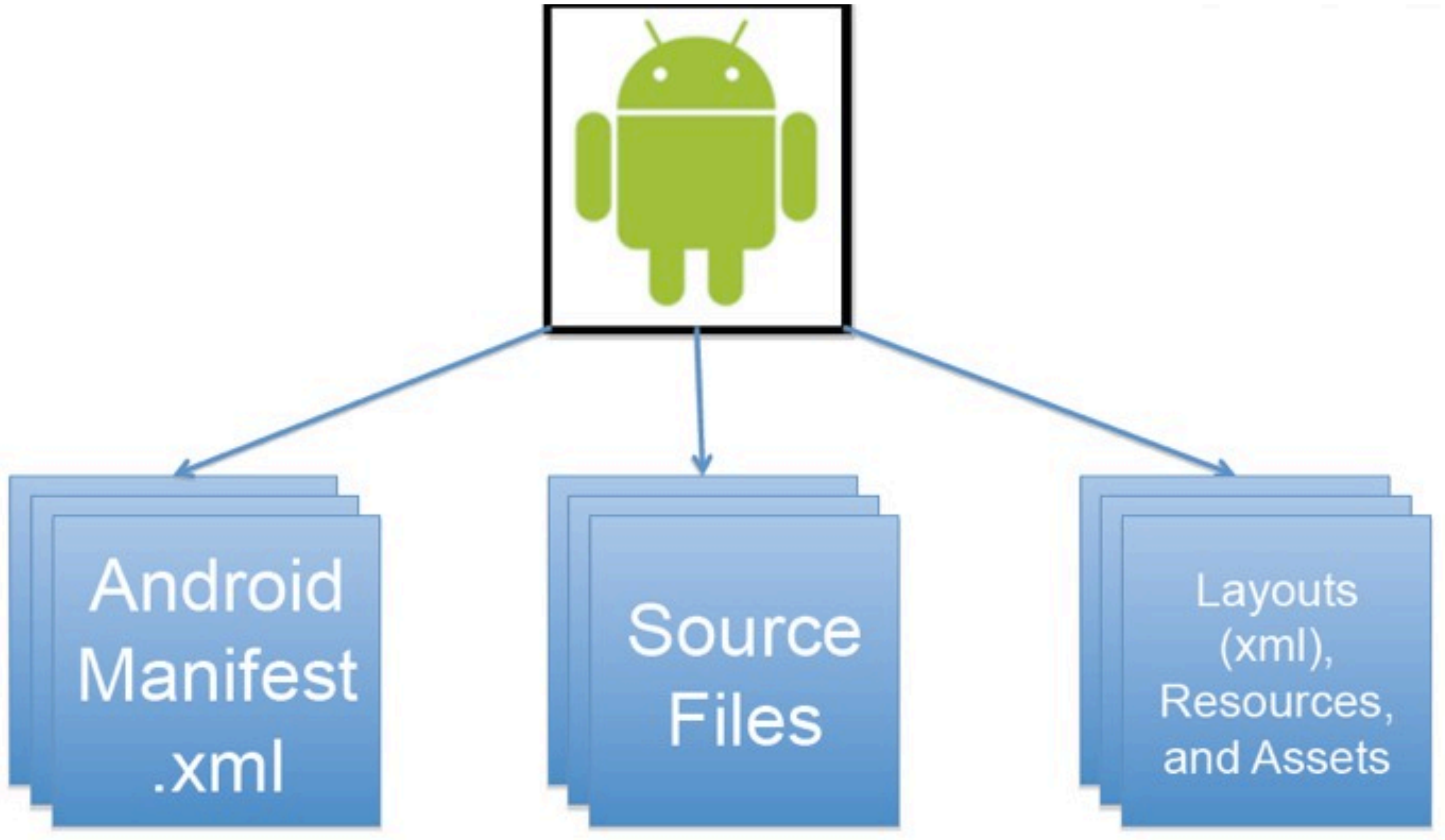
- Three ways to make money with Android Apps:
  - Sell the application on the Android Markets(Google Play, Amazon Market, …)
  - Offer the application for free on the Android Market and include small ads
  - Sell through your own website, outside of the Android Market
- Very competitive: over 600,000 apps in June 2012

Android Platform

Android Platform

- Linux-based OS
- Dalvik VM vs. JVM
- Platform components: GPS, WiFi, Camera, Audio/Video recording + playback, Sensors (acceleration, temperature, proximity, gyroscope, magnetic, …)
- SQLite Local Database Storage
- Built-in Applications (Home, Contacts, Phone, Browser, Voice Recognition, Camera, …)

Structure of an Android App

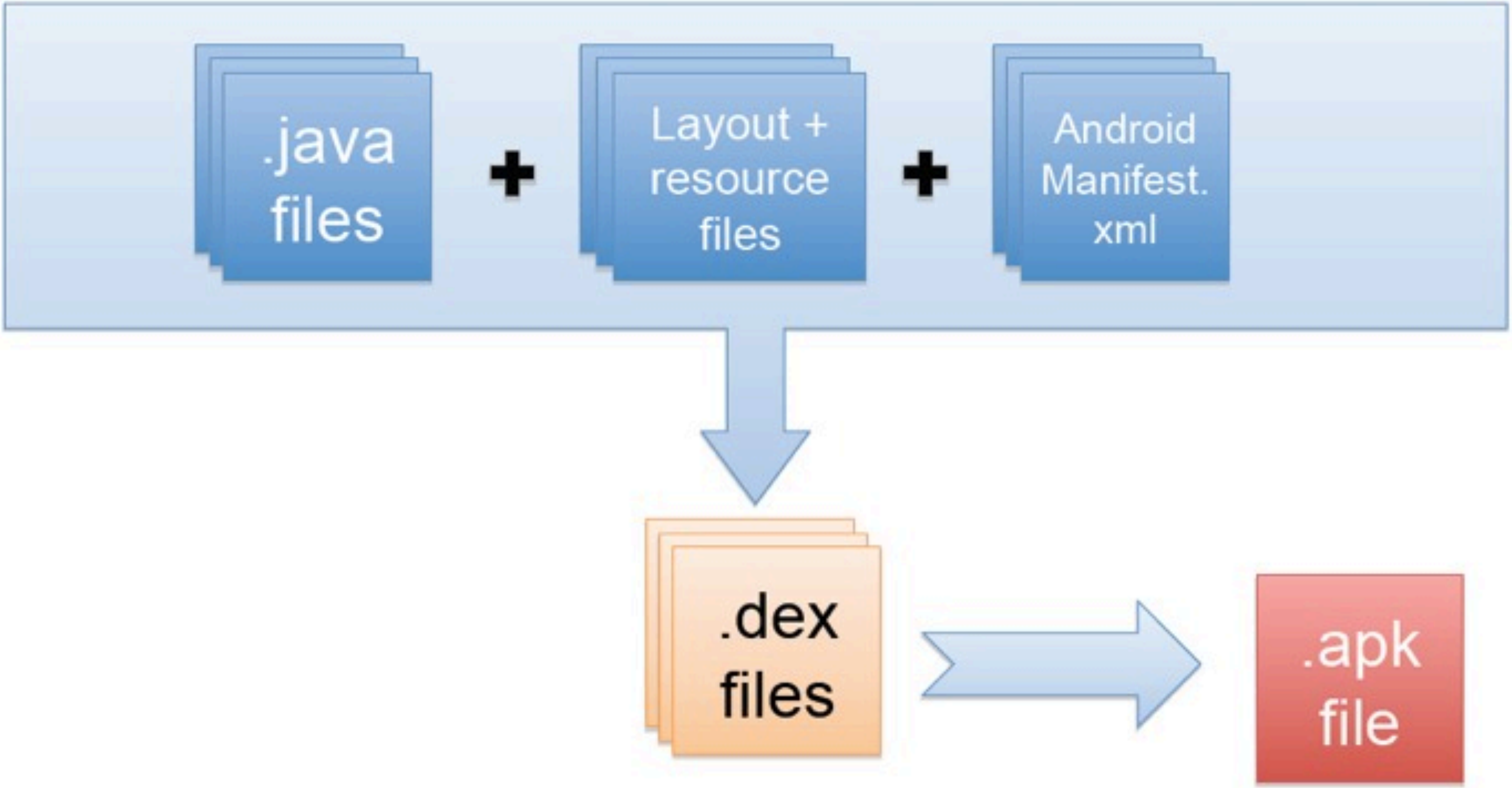


Android Manifest .xml

Source Files

Layouts (xml), Resources, and Assets

Android App Build Process



Android Programming

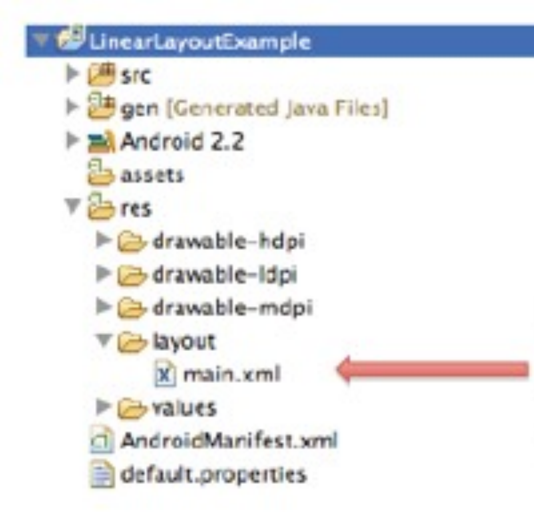
What you need:

- Eclipse
  http://www.eclipse.org/downloads/
- Android SDK
  http://developer.android.com/sdk/index.html
- Eclipse ADT Plug-in
  http://developer.android.com/sdk/eclipse-adt.html
- Android API (Android 2.3, 3.2, 4.0.3, etc.)
  - Latest version is Android v4.1 (API 16)
- Emulator (Android Virtual Device)

Layouts

Layouts

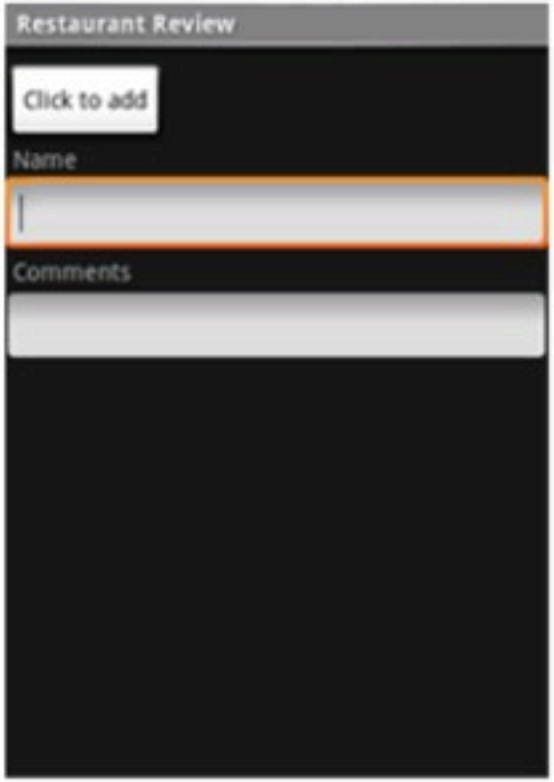
- Defined in two ways
  - XML layout files



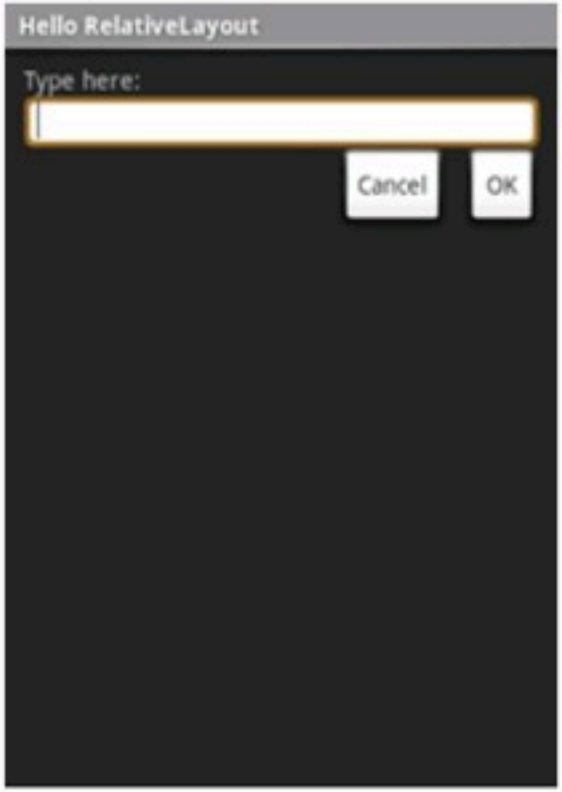
Layout file main.xml is auto-generated when an Android project is created in Eclipse. App layout can be defined in this file in XML.

  - using code (e.g. in the onCreate() method)

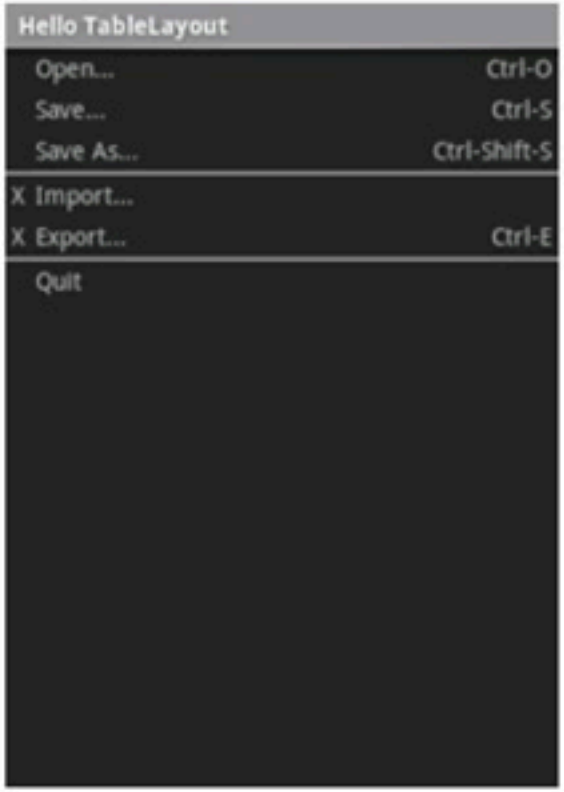
Some Layouts



LinearLayout



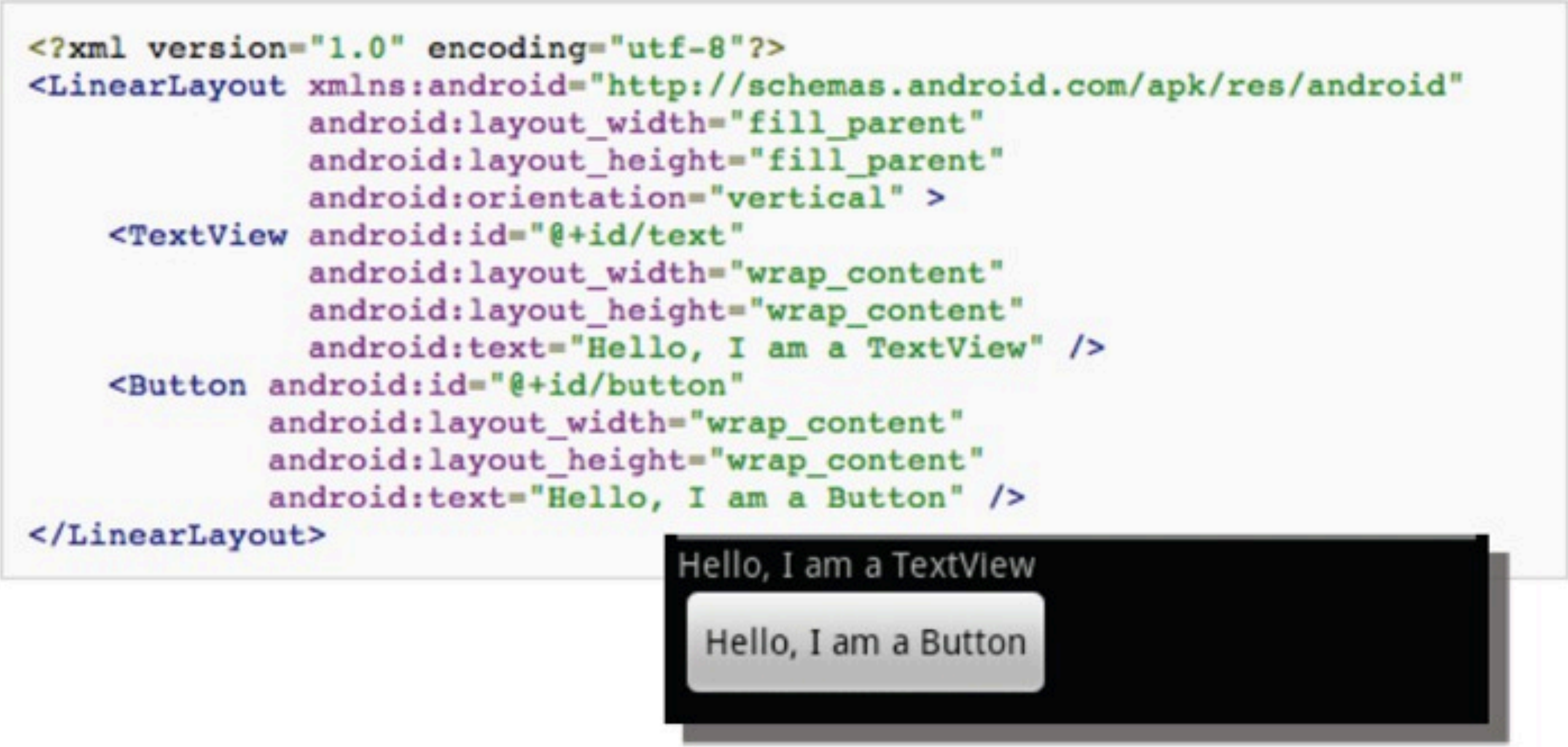
RelativeLayout



TableLayout

LinearLayout

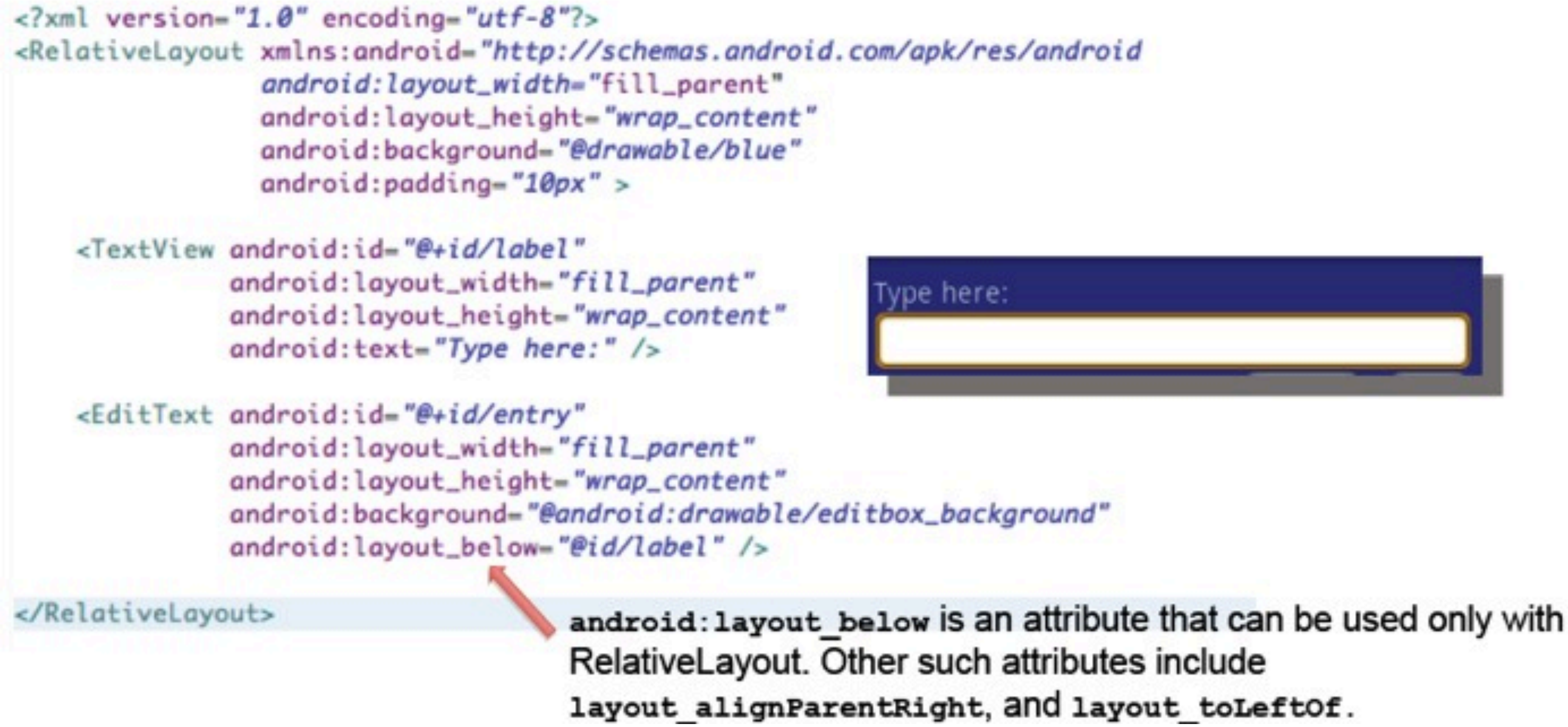
- Arrange components one after another, left-to-right, top-to-bottom:



```xml
<?xml version="1.0" encoding="utf-8"?>
<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/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

RelativeLayout

- Position and align components relative to other components:

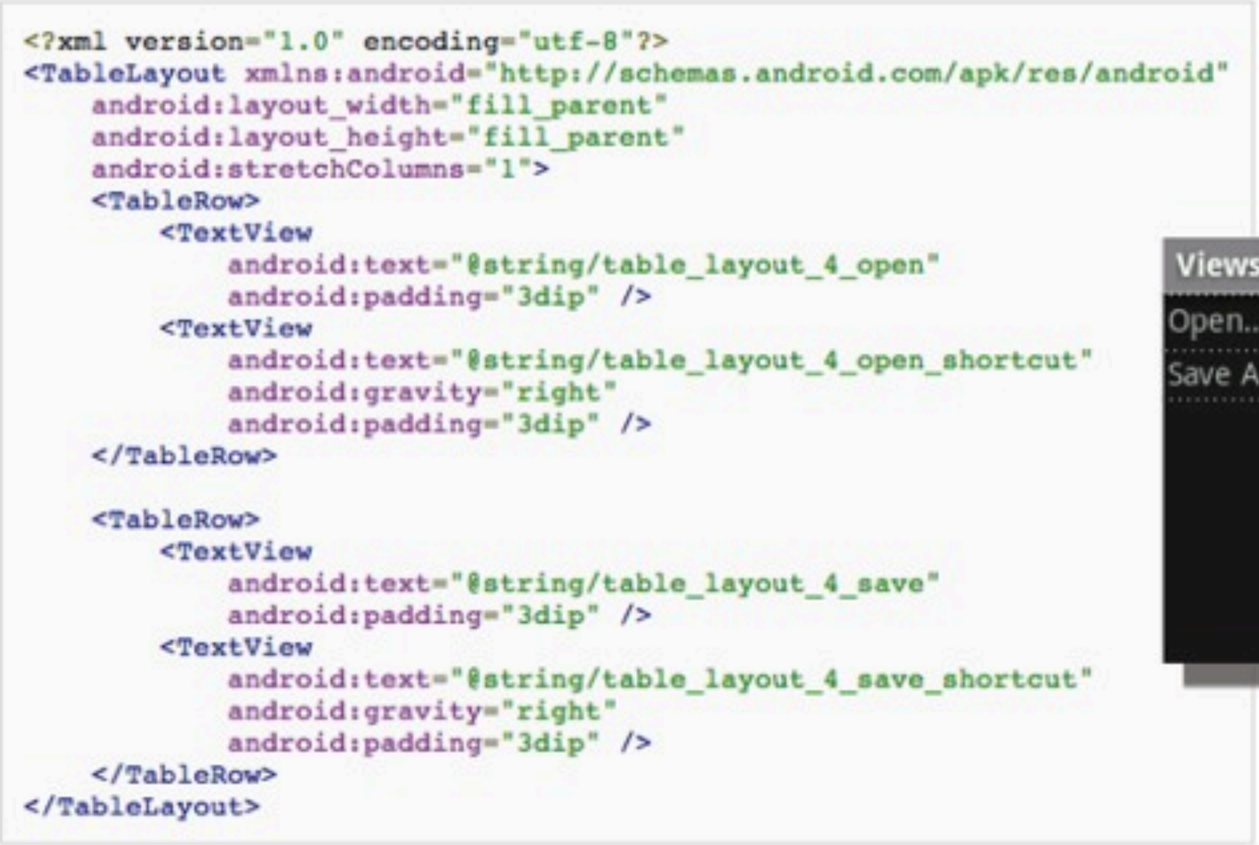


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:background="@drawable/blue"
                android:padding="10px" >

    <TextView android:id="@+id/label"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:text="Type here:" />

    <EditText android:id="@+id/entry"
              android:layout_width="fill_parent"
              android:layout_height="wrap_content"
              android:background="@android:drawable/editbox_background"
              android:layout_below="@id/label" />

</RelativeLayout>
```

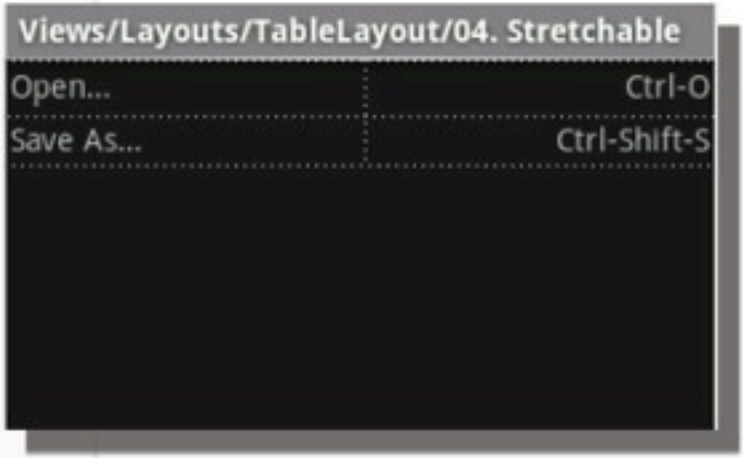
`android:layout_below` is an attribute that can be used only with RelativeLayout. Other such attributes include **`layout_alignParentRight`**, and **`layout_toLeftOf`**.

TableLayout

- Position components in rows and columns:



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1">
    <TableRow>
        <TextView
            android:text="@string/table_layout_4_open"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_open_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>

    <TableRow>
        <TextView
            android:text="@string/table_layout_4_save"
            android:padding="3dip" />
        <TextView
            android:text="@string/table_layout_4_save_shortcut"
            android:gravity="right"
            android:padding="3dip" />
    </TableRow>
</TableLayout>
```

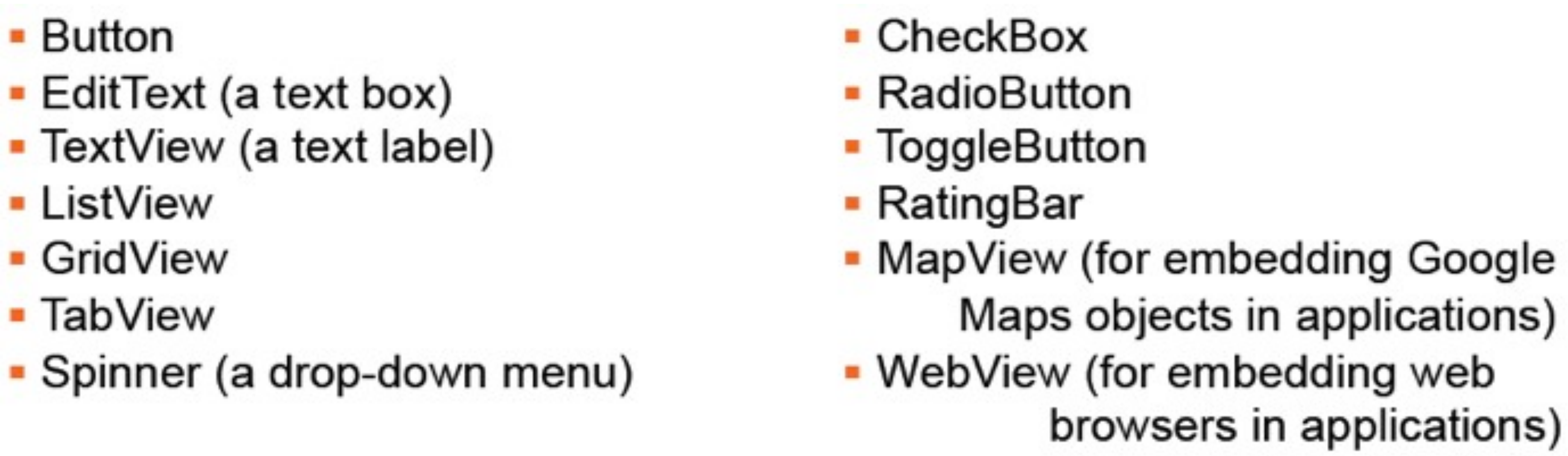


Views/Layouts/TableLayout/04. Stretchable

Open...	Ctrl-O
Save As...	Ctrl-Shift-S

Views

- What they are: UI components
- Some common views and widgets:

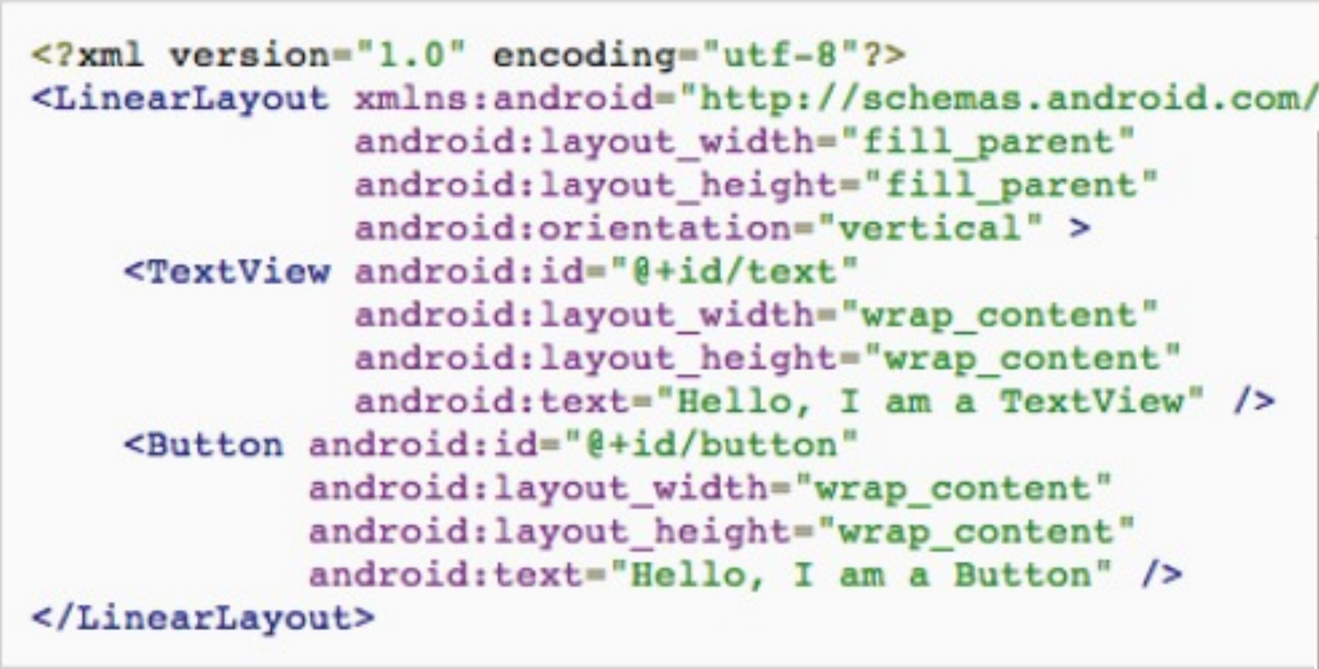


- Button
- EditText (a text box)
- TextView (a text label)
- ListView
- GridView
- TabView
- Spinner (a drop-down menu)

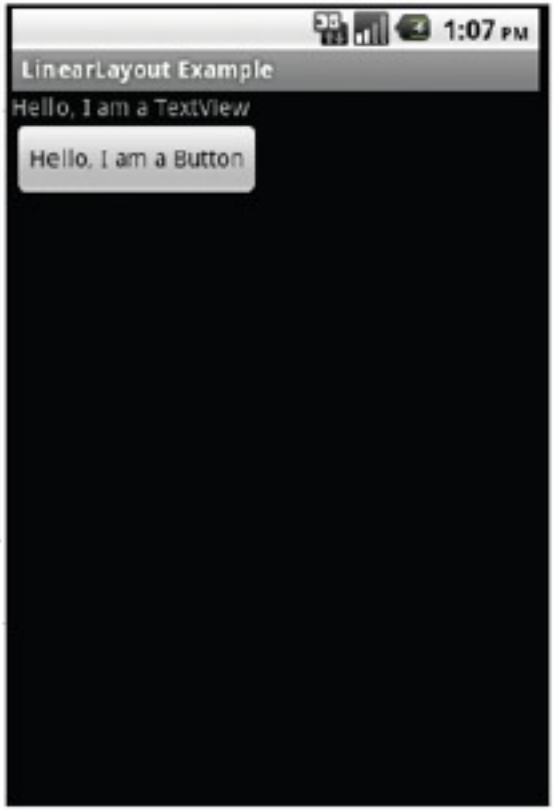
- CheckBox
- RadioButton
- ToggleButton
- RatingBar
- MapView (for embedding Google Maps objects in applications)
- WebView (for embedding web browsers in applications)

Adding Views to Layouts

- Example: adding a button and text label to a LinearLayout:

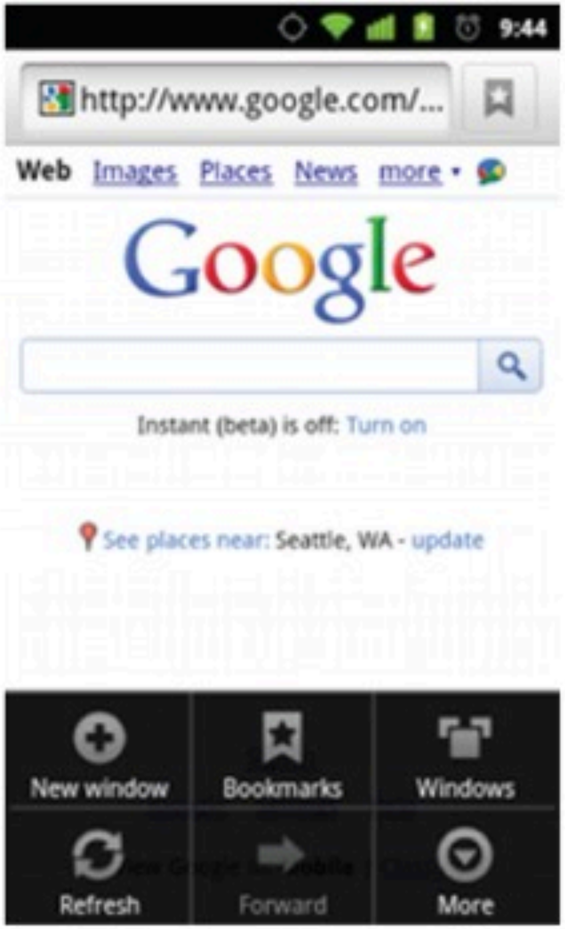


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

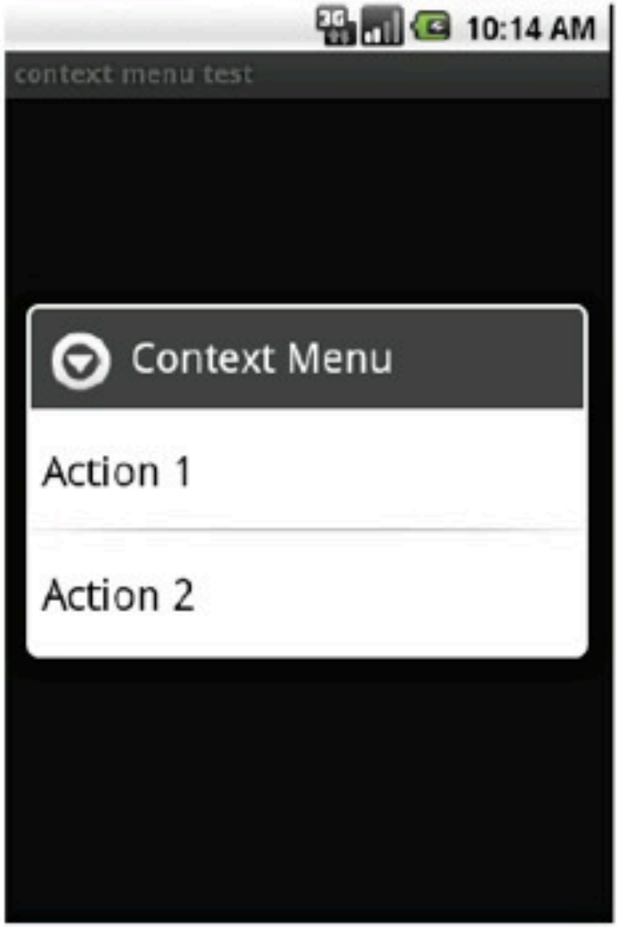


Menus

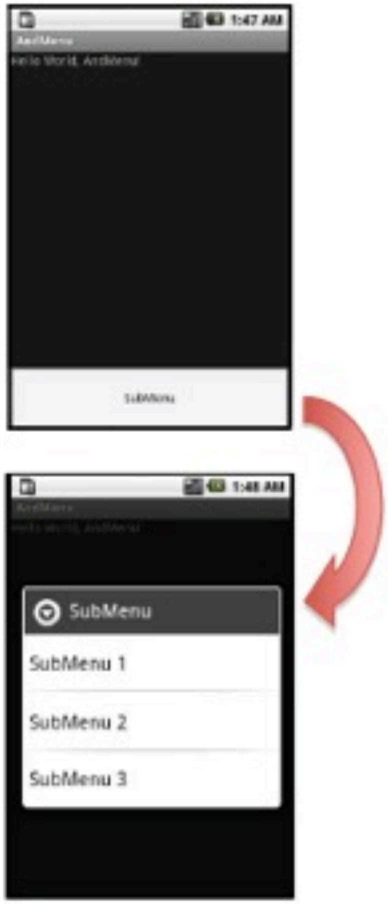
Options Menu


9:44
http://www.google.com/...
Web
Images
Places
News
more
Google
Instant (beta) is off: Turn on
See places near: Seattle, WA - update
New window
Bookmarks
Windows
Refresh
Forward
More

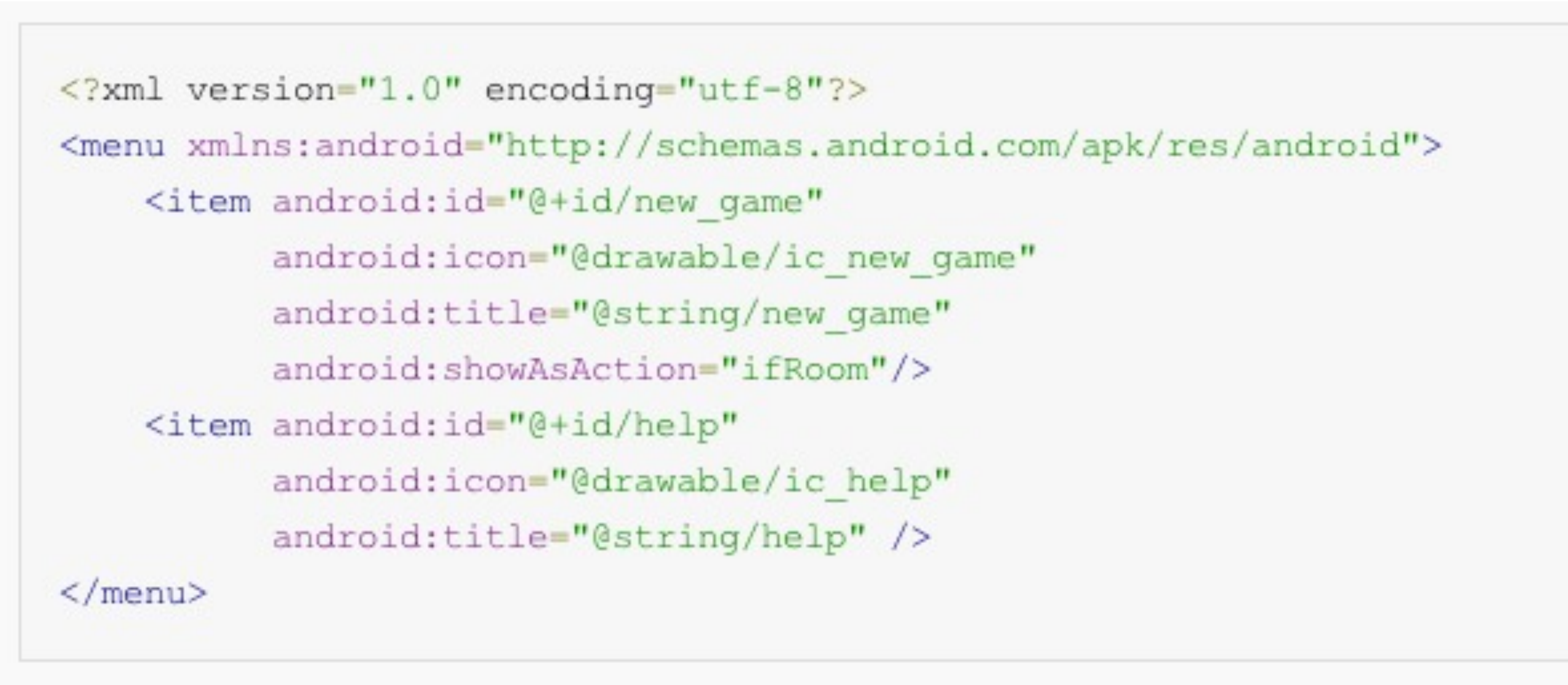
Context Menu



SubMenu



Menus



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
          android:icon="@drawable/ic_new_game"
          android:title="@string/new_game"
          android:showAsAction="ifRoom"/>
    <item android:id="@+id/help"
          android:icon="@drawable/ic_help"
          android:title="@string/help" />
</menu>
```

Homework

- Develop a “Hello world” android app using instructions from:
  - http://developer.android.com/training/basics/firstapp/index.html
- Prototype layout(s) for your Project ideas on Android
- http://developer.android.com/guide/topics/ui/index.html

Collaborative programming

- Divide your work meaningfully! (but collaborate and keep in touch)
- Pair programming
- Comment your code!
- Git commit messages
- Pull before you push
- Resolve conflicts