# Understanding Heart and Body Status from a Smartphone

## – Introduction to programming on Android–

Guillaume Lopez
Living Environment Laboratory
The University of Tokyo, School of Engineering

# Smartphone Sensor Web #3

「Table of Contents」


🗁. **Class #3**

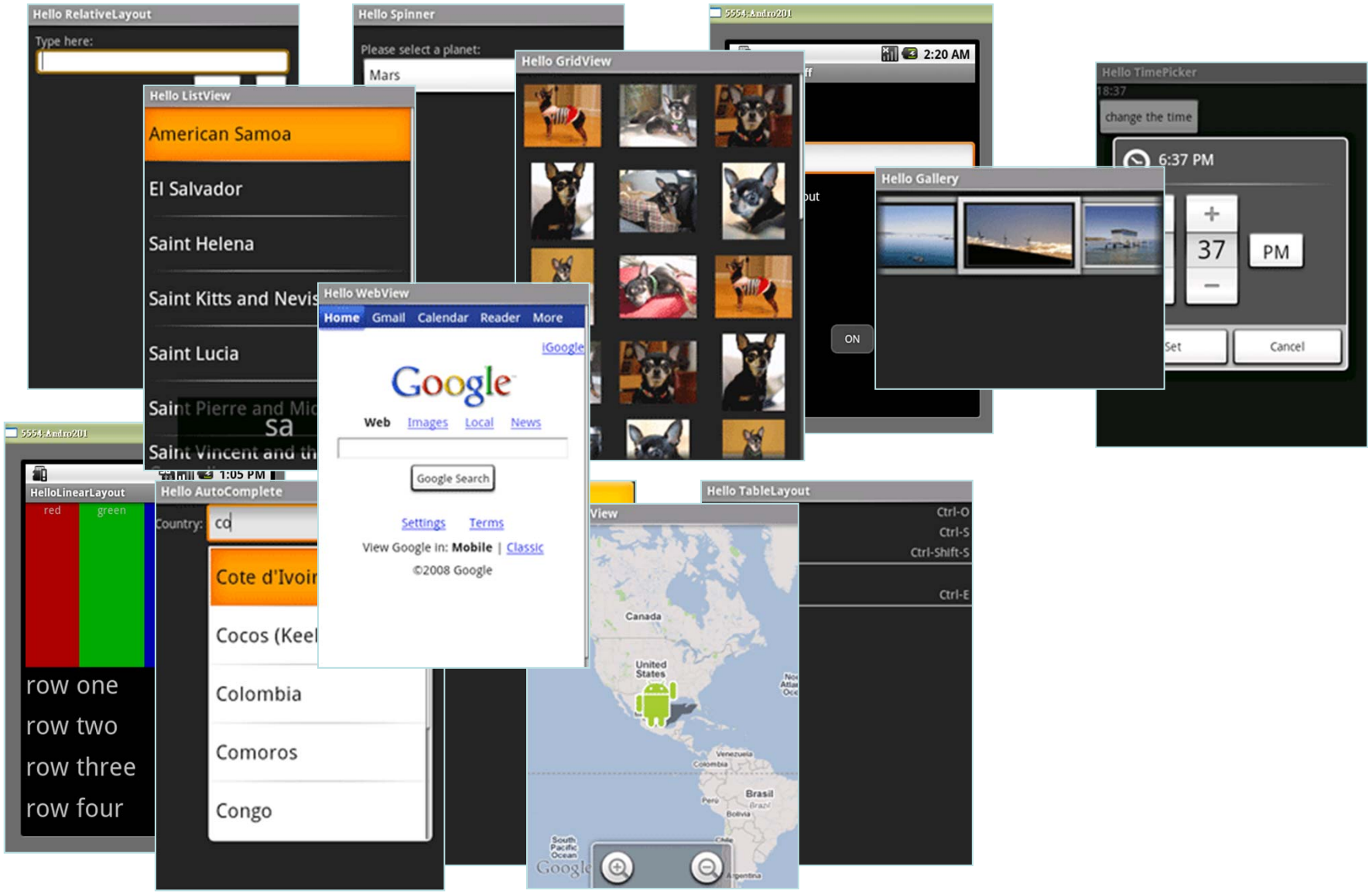
★. **Introduction to Java Programming** 🙂

★. **Hello World application programming (tutorial provided)** 🙂

★. **Introduction to Android GUI** 🙂

★. **Consultation and decision of individual objectives (targeted application/demo…)**

# User Interface



# Life cycle of Applications

- Life cycle not directly controlled by application
- System can kill an application to free up memory
- Controll through onCreate(), onPause(), onStop() ... methods
- Android has different types of processes, visible processes, service processes, background processes ...
- Services can be used for long-lived background processes

# Android Application Anatomy

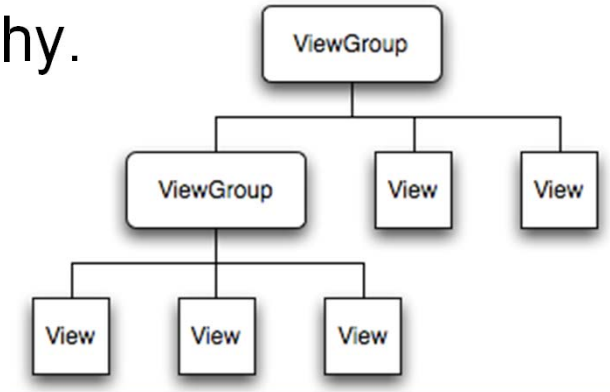
- **Activity**
- **View**
- **Intent, IntentFilter, IntertReceiver**
- **Service**
- **Content provider**

# Android components: Activities

- Displays a **user interface** component and responds to system/user.
- When an application has a user interface, it contains one or more “Activities”.
- An existing “Activity” *can be replaced* with a new one that fulfill the same contract (intent).
- Each “Activity” *can be invoked* from other applications.
- Adding a new “Activity” in an Android project
  - The new Java class must extend the framework “Activity” class.
  - Created “Activity” must be defined into the application’s Manifest.xml.

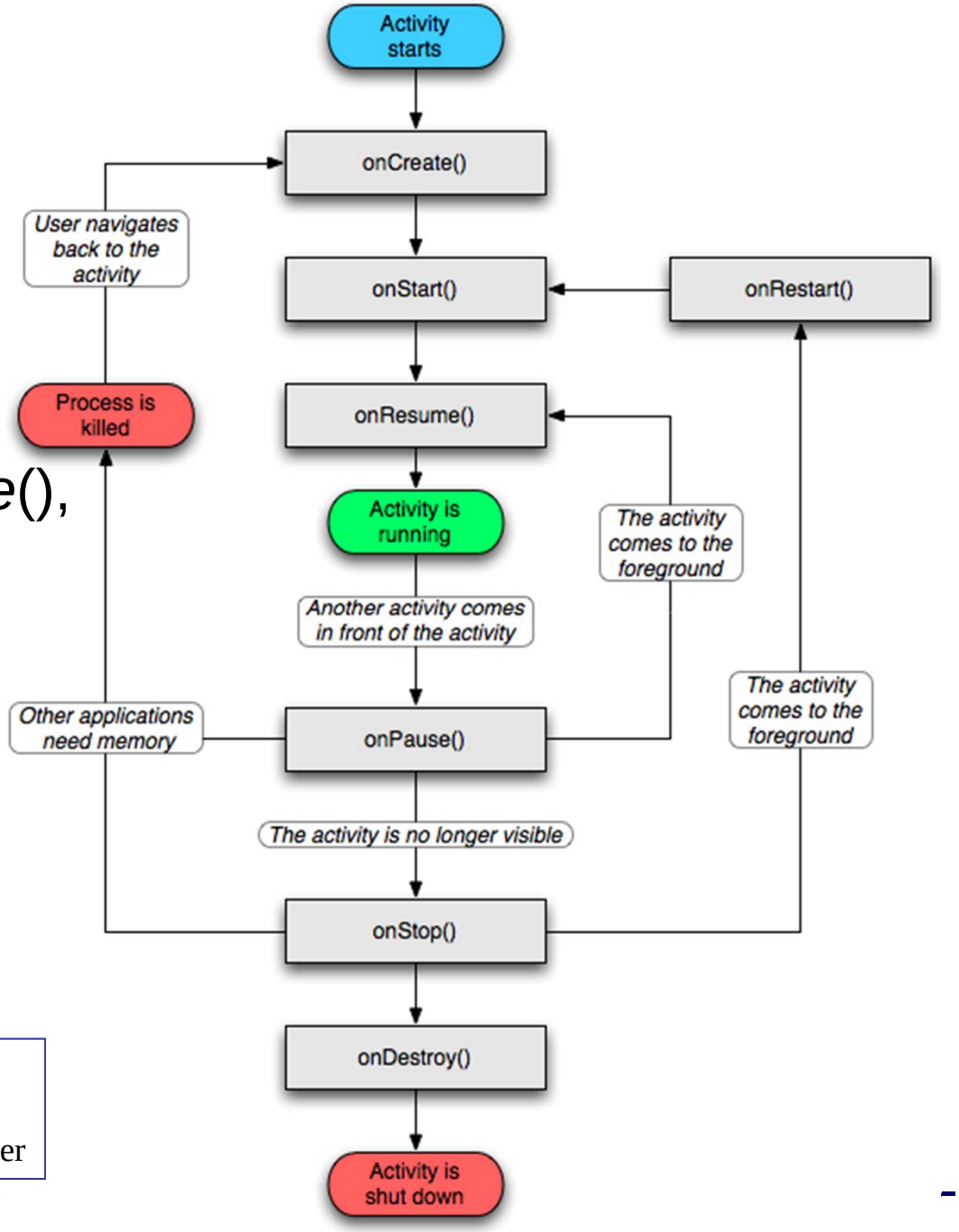
# Android components: Activity (cont'd)

- Each **Activity** is given a default window to draw in. Typically, the window fills the screen, but it might be smaller than the screen and float on top of other windows.
- The visual content (a rectangular window) of is provided and controlled by a hierarchy of **Views**
  - Objects derived from the base View class.
  - Ready-made views: buttons, text fields, scroll bars, menu items, check boxes, images and more.
- A **View** hierarchy is placed within an activity's window by the *Activity.setContentView()* method. The content view is the View object at the root of the hierarchy.



# Activity control loop

- Life cycle not directly controlled by application
- System can kill an application to free up memory.
- Controll through *onCreate*(), *onPause*(), *onStop*() ... methods



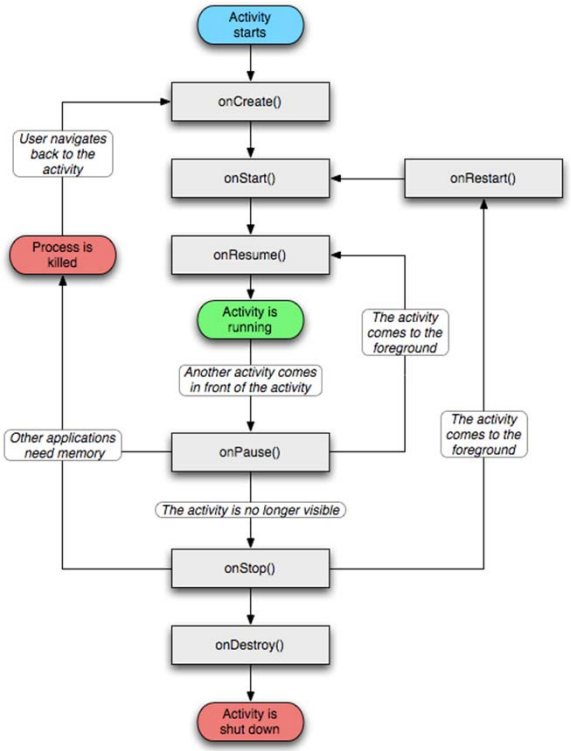

Colored ovals: states of the activity

Grey rectangles: callback methods written by developer

# An Example

```
public class LifeCycleTest extends Activity {
    private static final String TAG = "ActivityLifeTest";

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Log.v(TAG, "onCreate");

        Uri uri = Uri.parse("http://maps.google.com/maps?f=d&saddr" +
                "=startLat%20startLng&daddr=endLat%20endLng&hl=en");
        Intent it = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(it);
    }

    public void onStart() {
        super.onStart();
        Log.v(TAG, "onStart");
    }

    public void onResume() {
        super.onResume();
        Log.v(TAG, "onResume");
    }

    public void onPause() {
        super.onPause();
        Log.v(TAG, "onPause");
    }
```

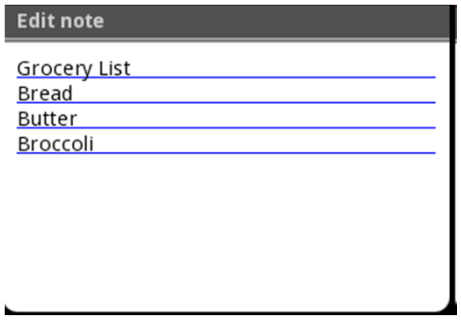
```
    public void onPause() {
        super.onPause();
        Log.v(TAG, "onPause");
    }

    public void onStop() {
        super.onStop();
        Log.v(TAG, "onStop");
    }

    public void onRestart() {
        super.onRestart();
        Log.v(TAG, "onReStart");
    }

    public void onDestroy() {
        super.onDestroy();
        Log.v(TAG, "onDestroy");
    }
}
```



# Views in Android

- Most activities will present a ‘view’ to the user, either to display some graphics, or to get some user-input.
- Each activity can create (instances) of one or more views.
- Each view has some graphical objects that either fill the complete screen, or a part of the screen.
- Each object in a view, i.e. the layout, is also described in XML

Notepad app
(layout of Note-Editor)




```
<?xml version="1.0" encoding="utf-8"?>
<view xmlns:android="http://schemas.android.com/apk/res/android"
    class="com.example.android.notepad.NoteEditor$LinedEditText"
    android:id="@+id/note"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@android:color/transparent"
    android:padding="5dip"
    android:scrollbars="vertical"
    android:fadingEdge="vertical"
    android:gravity="top"
    android:textSize="22sp"
    android:capitalize="sentences"
/>
```

# Intents/Intent Filters

- Provide a **late runtime (asynchronous) binding** between the code in different applications (for activities, services, and broadcast receivers)
- **Intents**: Simple message objects that represent an intention to do something
- **Intent Filters**: A declaration of capacity and interest in offering assistance to those in need
- An "intent" is made up a number of pieces of information describing the action or the service. (Specifically for activities and services)
  - **action** -- The general action to be performed, such as ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, etc.
  - **data** -- The data to operate on, such as a person record in the contacts database, expressed as a URI (Universe Resource Identifier).


**Example:**
*Intent newActivity = new **Intent**(MyActivity.this, OtherActivity.class);*
***startActivity(**newActivity);*

*REF: Slides from "Android Anatomy and Physiology," Patrick Brady ©*

# Android components: Services

- Service is an activity that runs in the background

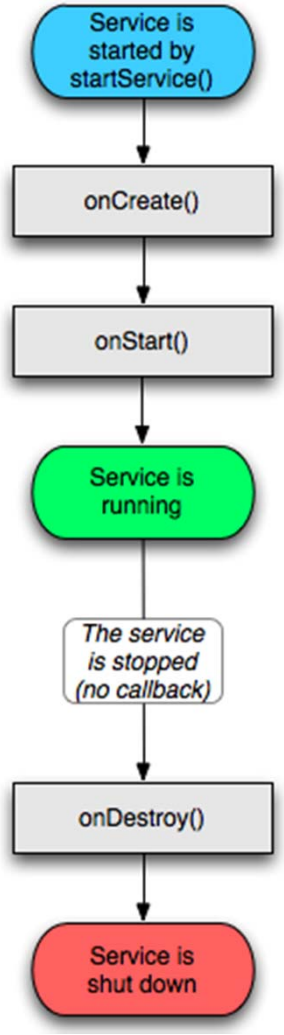
  ➔ no visual interface
- Each activity is derived from base class **Service**
- For long-running background tasks
- Example:

  A common example of a service is an mp3 player that may run in the background as the user may be involved with some activity of another (e.g. web browser.)

  - Core Services
    - Activity Manager
    - Package Manager
    - Window Manager
    - Resource Manager
    - Content Providers
    - View System
  - Hardware Services
    - Telephony Service
    - Location Service
    - Bluetooth Service
    - WiFi Service
    - USB Service
    - Sensor Service

# Service control loop

NOTE:
Typically, a service may be created, say, by an activity;
Alternatively, a service may be started and running in some other context, and can announce its interface to other activities – in this case, the activity may just connect itself to the service, in Android, this is called “bind”-ing to the service.




Colored ovals: states of the service

Grey rectangles: callback methods written by developer

# Android components: Broadcast receivers

- Broadcast receivers are similar to interrupt handlers in normal OS
- BRs run in the background, listening for interrupts generated by other apps
- An application may have one or more BR’s to handle interrupts.

- Examples of interrupts:
  - Incoming phone call
  - User changed language setting
  - Battery is low
  - User has transited from one time zone to different one

# Android components: Content providers

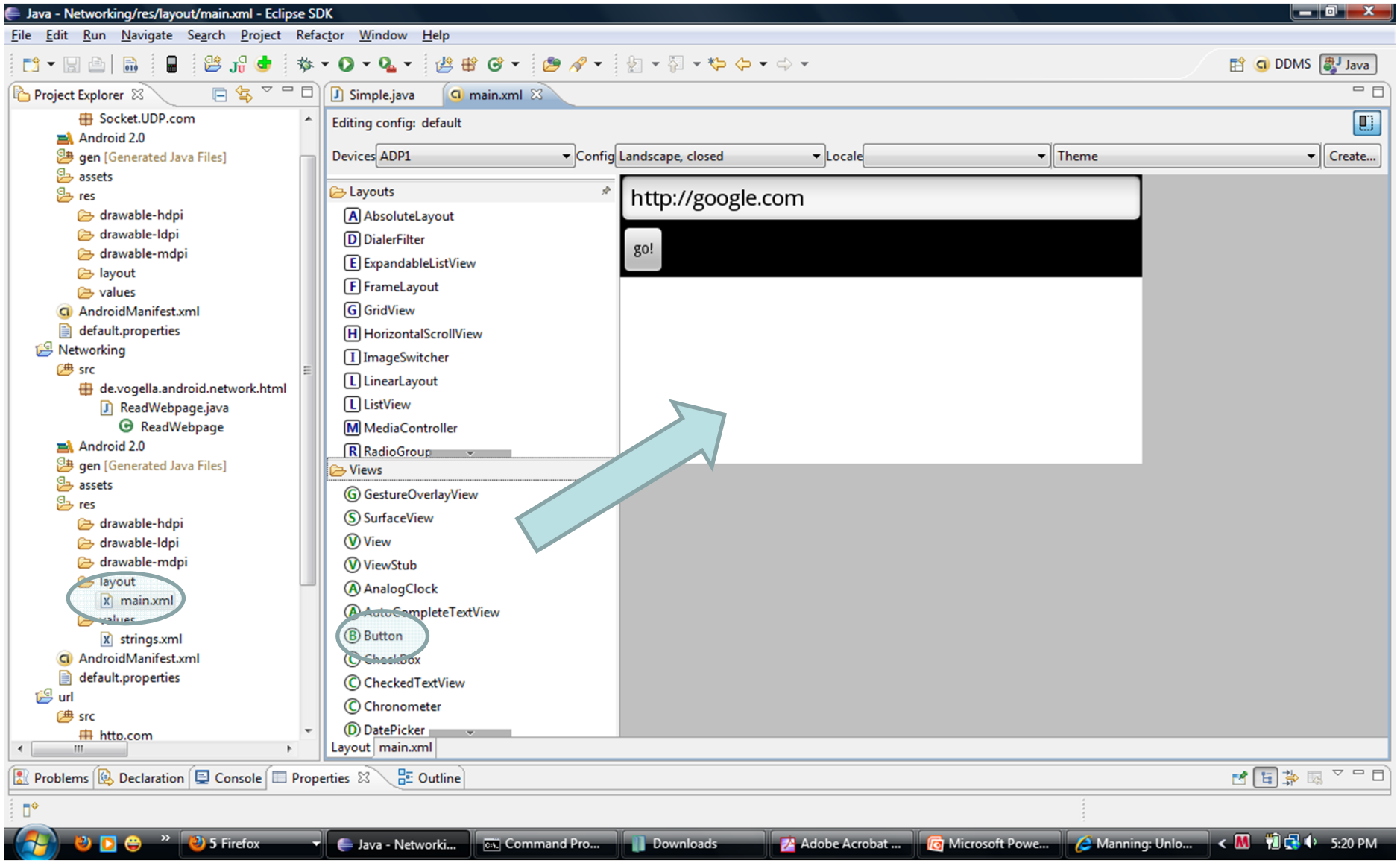
- Content providers make some subset of an application's data available to other apps when requested

- Content providers are the only mechanism for apps to share data.

# Android application process

- Process can be multi-threaded
  - Android apps do not have a C-style “main”.
  - Activities, services, broadcast receivers: activated by messages called intents.
- Depending on the state of the application, and the user’s actions, the app may start (or terminate) some activity, or service, etc.
  - Before Android can start an application component, it must know
  - the name, location, and input types of the component are defined in the manifest

# Layout Edition with Eclipse



# Link Activity and View

- View object may have an integer ID associated with it

```
android:id="@+id/my_button“
```

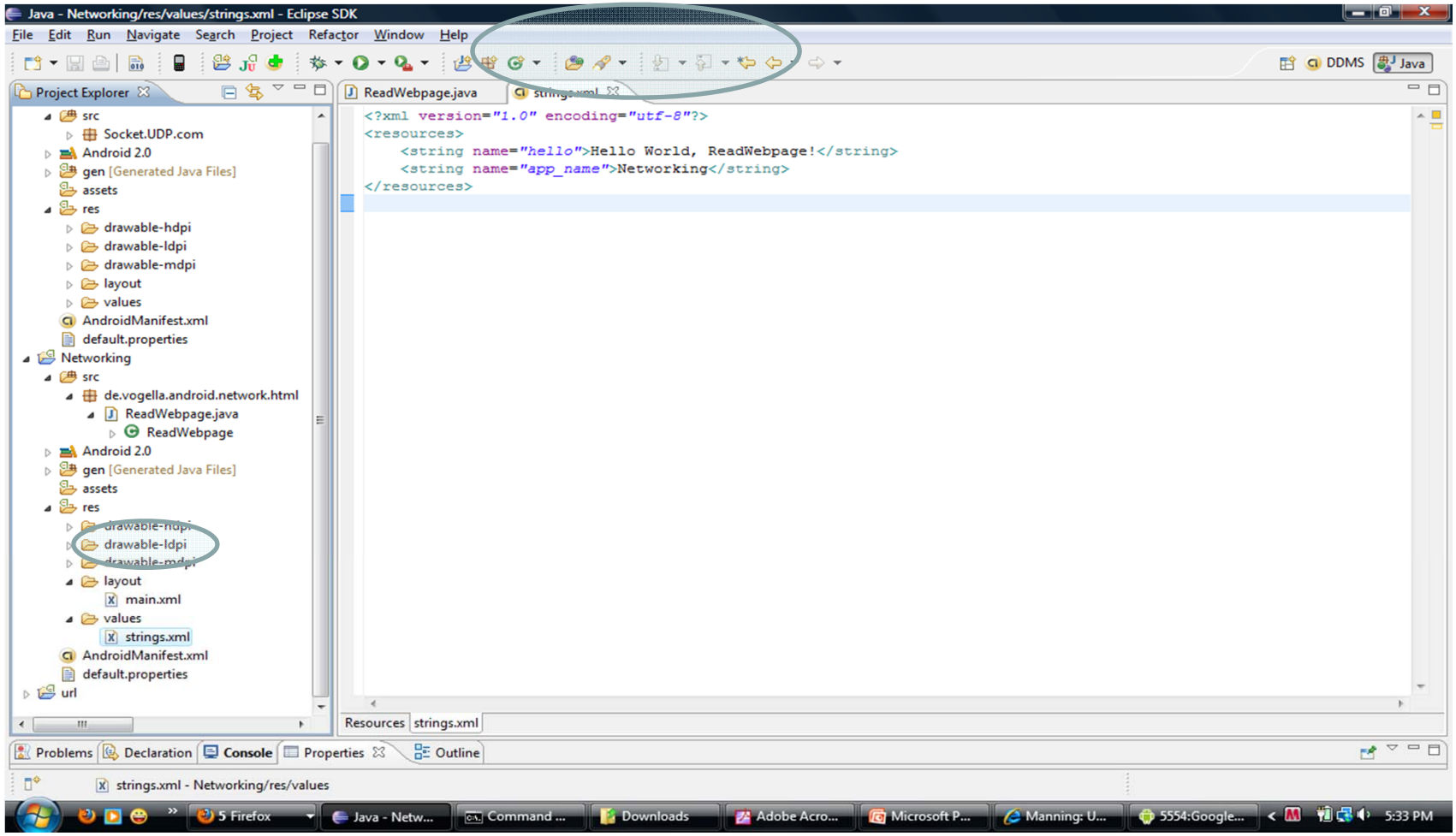
- To get the reference of the view object in activity

```
Button myButton =
  (Button)findViewById(R.id.my_button);
```

# Adding Event to View Object

- View.OnClickListener()
  - Interface definition for a callback to be invoked when a view is clicked.
- onClick(View v)
  - Called when a view has been clicked. Inside this function you can specify what actions to perform on a click.

# Strings.xml

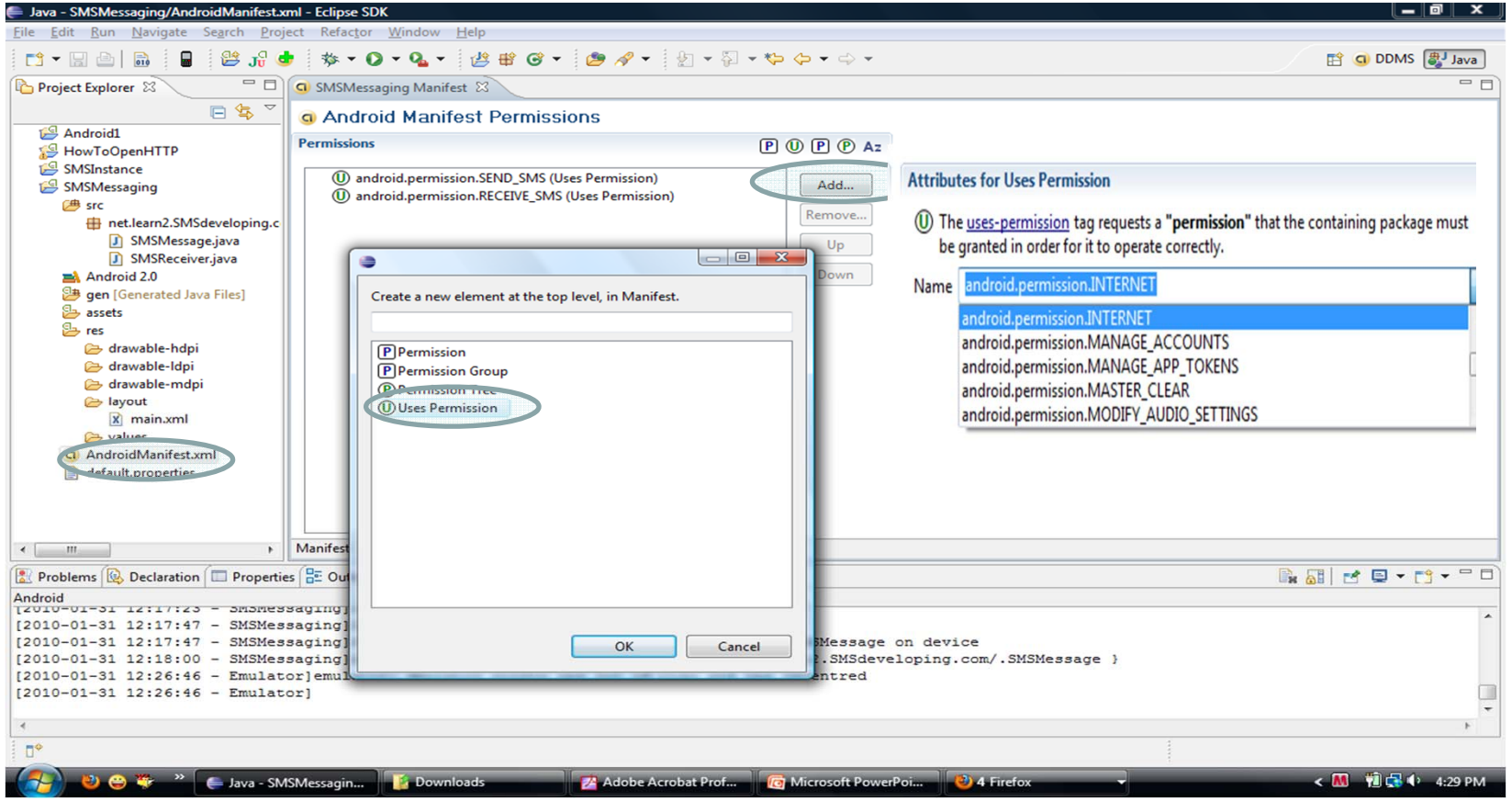

Java - Networking/res/values/strings.xml - Eclipse SDK
File Edit Run Navigate Search Project Refactor Window Help
DDMS Java
Project Explorer
src
Socket.UDP.com
Android 2.0
gen [Generated Java Files]
assets
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
layout
values
AndroidManifest.xml
default.properties
Networking
src
de.vogella.android.network.html
ReadWebpage.java
ReadWebpage
Android 2.0
gen [Generated Java Files]
assets
res
drawable-ldpi
layout
main.xml
values
strings.xml
AndroidManifest.xml
default.properties
url
ReadWebpage.java
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="hello">Hello World, ReadWebpage!</string>
<string name="app_name">Networking</string>
</resources>
Resources strings.xml
Problems Declaration Console Properties Outline
strings.xml - Networking/res/values
5 Firefox
Java - Netw...
Command ...
Downloads
Adobe Acro...
Microsoft P...
Manning: U...
5554:Google...
5:33 PM

# AndroidManifest.xml

- **Used for security**
- **Define permissions, e. g.**

**<uses-permission android:name="android.permission.RECEIVE_SMS" />**

- **Give other Activities access**



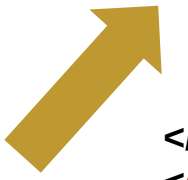
# A real word example I

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.greenliff.translator">
   <application android:icon="@drawable/logo">
      <activity android:label="@string/settings"
android:name="Settings">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
         </intent-filter>
      </activity>
      <activity android:label="@string/app_name"
android:name="Translate">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category
android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
      <activity android:label="@string/ocr" android:name="OCR">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
         </intent-filter>
      </activity>
   </application>
</manifest>
```

# A real word example II


```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.greenliff.translator">
   <application android:icon="@drawable/logo">
      <activity android:label="@string/settings" android:name="Settings">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
         </intent-filter>
      </activity>
      <activity android:label="@string/app_name"
android:name="Translate">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
      <activity android:label="@string/ocr" android:name="OCR">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
         </intent-filter>
      </activity>
   </application>
</manifest>
```

# A real word example III



Activity

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.greenliff.translator">
    <application android:icon="@drawable/logo">
        <activity android:label="@string/settings" android:name="Settings">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity android:label="@string/app_name" android:name="Translate">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:label="@string/ocr" android:name="OCR">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Launch

# A real word example IV

**An XML snipped of the main Activity**

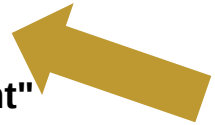
```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:scrollbars="vertical">
   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            android:id="@+id/linLayout"
            android:orientation="vertical"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">
<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/blue"
        android:text="@string/translate_to_1"/>
<EditText
        android:id="@+id/toTranslate"
             android:layout_width="fill_parent"
                  android:layout_height="wrap_content"
                  android:background="@android:drawable/editbox_background"
                  android:layout_below="@id/linLayout“
                 android:hint="Type here..." />
.....
```

Text reference

# A real word example V

**An XML snipped of the main Activity**

```
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:scrollbars="vertical">
   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
            android:id="@+id/linLayout"
            android:orientation="vertical"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content">
<TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/blue"
        android:text="@string/translate_to_1"/>
<EditText
        android:id="@+id/toTranslate"
              android:layout_width="fill_parent"
                    android:layout_height="wrap_content"
                    android:background="@android:drawable/editbox_background"
                    android:layout_below="@id/linLayout“
                   android:hint="Type here..." />
.....
```



Id

Text reference

# GUI Development

- **Could also be developed purely in Java**
- **XML cannot be debugged**
- **Not all the attributes can be defined in XML**

# A real word example VII

**A code snipped of the Translate Activity**

```
@Override
  public void onCreate(Bundle icicle) {
     super.onCreate(icicle);
     Window wp = getWindow();
     mContext = wp.getContext();
     setTheme(android.R.style.Theme_Light);
     setContentView(R.layout.main);
     mLayout = (LinearLayout) this.findViewById(R.id.linLayout);
     mToTranslate = (EditText) this.findViewById(R.id.toTranslate);
     setShowLanguages();

     mEnge = (LinearLayout) this.findViewById(R.id.enge);
     LANGUAGE_LAYOUT[0] = mEnge;
     de2en = (Button) this.findViewById(R.id.de2en);
     de2en.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
           if(!connect()) {
             notLoggedInAlert();
           } else {
             doConnect("de2en@bot.talk.google.com");
             rearrange(mEnge);
           }
        }
     });
....
```

# A real word example VIII

**A code snipped of the Translate Activity**

Set layout

```
@Override
  public void onCreate(Bundle icicle) {
     super.onCreate(icicle);
     Window wp = getWindow();
     mContext = wp.getContext();
     setTheme(android.R.style.Theme_Light);
     setContentView(R.layout.main);
     mLayout = (LinearLayout) this.findViewById(R.id.linLayout);
     mToTranslate = (EditText) this.findViewById(R.id.toTranslate);
     setShowLanguages();

     mEnge = (LinearLayout) this.findViewById(R.id.enge);
     LANGUAGE_LAYOUT[0] = mEnge;
     de2en = (Button) this.findViewById(R.id.de2en);
     de2en.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
           if(!connect()) {
             notLoggedInAlert();
           } else {
             doConnect("de2en@bot.talk.google.com");
             rearrange(mEnge);
           }
        }
     });
....
```

# A real word example IX

**A code snipped of the Translate Activity**

Set layout

Find elements

Add behavior

```
@Override
  public void onCreate(Bundle icicle) {
     super.onCreate(icicle);
     Window wp = getWindow();
     mContext = wp.getContext();
     setTheme(android.R.style.Theme_Light);
     setContentView(R.layout.main);
     mLayout = (LinearLayout) this.findViewById(R.id.linLayout);
     mToTranslate = (EditText) this.findViewById(R.id.toTranslate);
     setShowLanguages();

     mEnge = (LinearLayout) this.findViewById(R.id.enge);
     LANGUAGE_LAYOUT[0] = mEnge;
     de2en = (Button) this.findViewById(R.id.de2en);
     de2en.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
           if(!connect()) {
             notLoggedInAlert();
           } else {
             doConnect("de2en@bot.talk.google.com");
             rearrange(mEnge);
           }
        }
     });
....
```

# Trends

- **Speech recognition**
- **Accelerometer**
- **Magnetic compass**
- **Location self-awareness**
- **Locating others**
- **Sensing the environment**

# Smartphone Sensor Web

「Introduction to Android Platform/Smartphone」

## 🗁. References


★. Dr. Frank McCown,Harding University, ”A Developer’s Introduction to Google Android”, Spring 2010.

★. Android Introduction by Marko Gargenta, http://www.lecturemaker.com/2009/10/android-software-platform/

★. Android Dev Guide http://developer.android.com/guide/topics/fundamentals.html

★. Pro Android by Hashimi & Komatineni (2009)

★. Patrick Brady, “Android Anatomy and Physiology”

★. William W.-Y. Liang, National Taipei University of Technology, “System Integration for the Android Operating System”

★. Mihail L. Sichitiu, “Android Introduction - Hello Views Part 1 –”

# Questions?


*REF: http://www.pocket-lint.com/news/30712/android-powered-microwave-cooking-google*

# Questions?

1.

# Smartphone Sensor Web #3

「Table of Contents」


🗁. Class #3 Smartphone Sensor Web practice

★. Sensor-based Twit using a smartphone

# Smartphone Sensor Web #4~#11

「Table of Contents」

🗁. **Class #4~#10 Individual projects**

★. **Sensor-based Twit using a smartphone**

🗁. **Class #? Industrial Visit**

★.

🗁. **Class #11 Final Demonstration**

# What can be done ?

スマートフォン搭載センサを活用してみる

他の無線センサを接続する

例：心電センサからの心拍数と血圧計算
加速度センサの信号と体動による変化

- Embedded sensors
  - ✓ Automatic motion detection from accelerometers (sit, walk, run)
  - ✓ Speaking/Listening time count from microphone
  - ✓ Location sensitive (GPS) sensing application
  - ✓ . . .
- Outside sensors (wireless)
  - ✓ Simple stress checker from ECG and/or Pulse
  - ✓ Chewing real-time counting from bone-conduction microphone
  - ✓ . . .