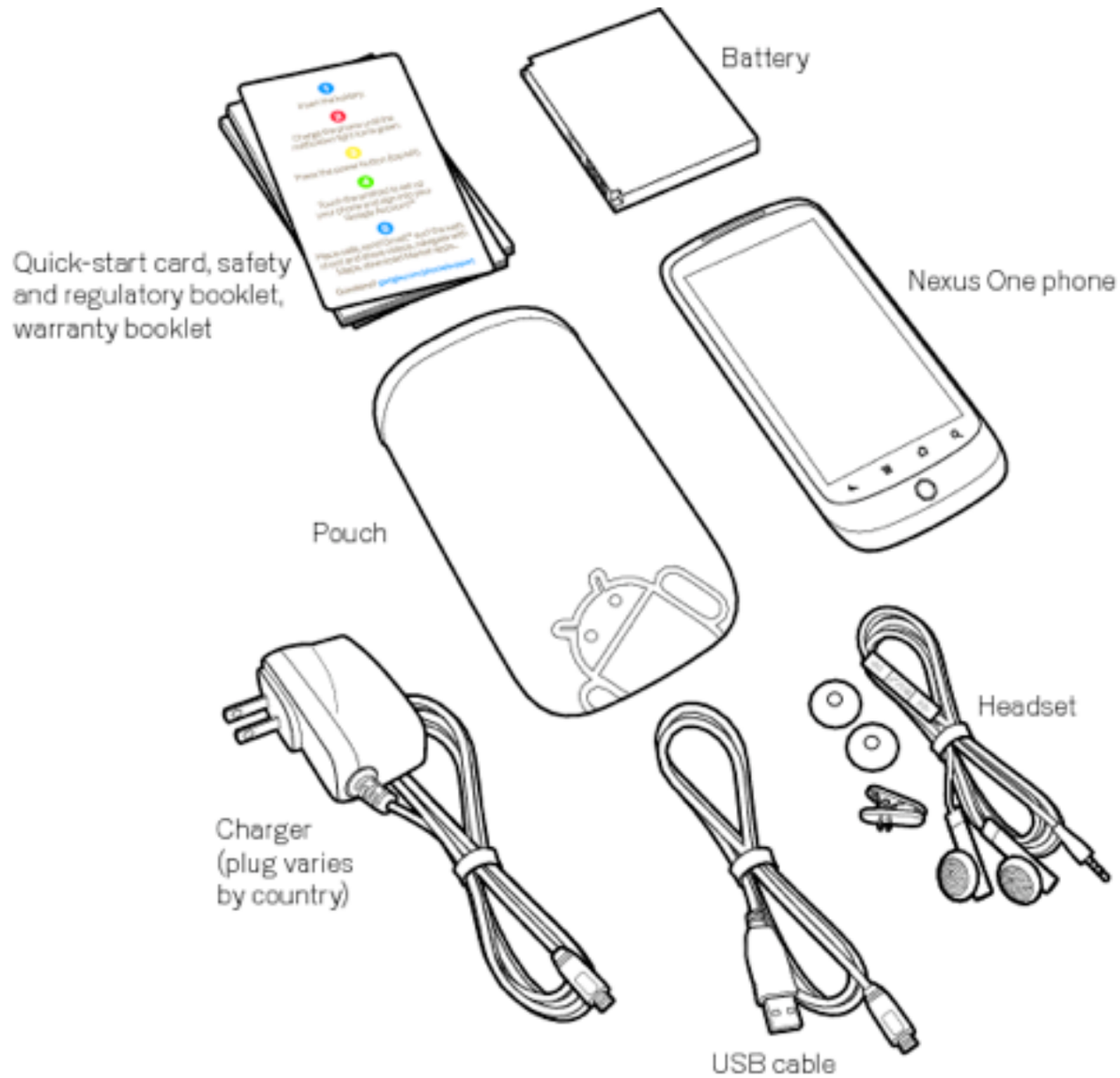
User Interaction: Intro to Android

Assoc. Professor Donald J. Patterson
INF 133 Fall 2012

Checking out the phone

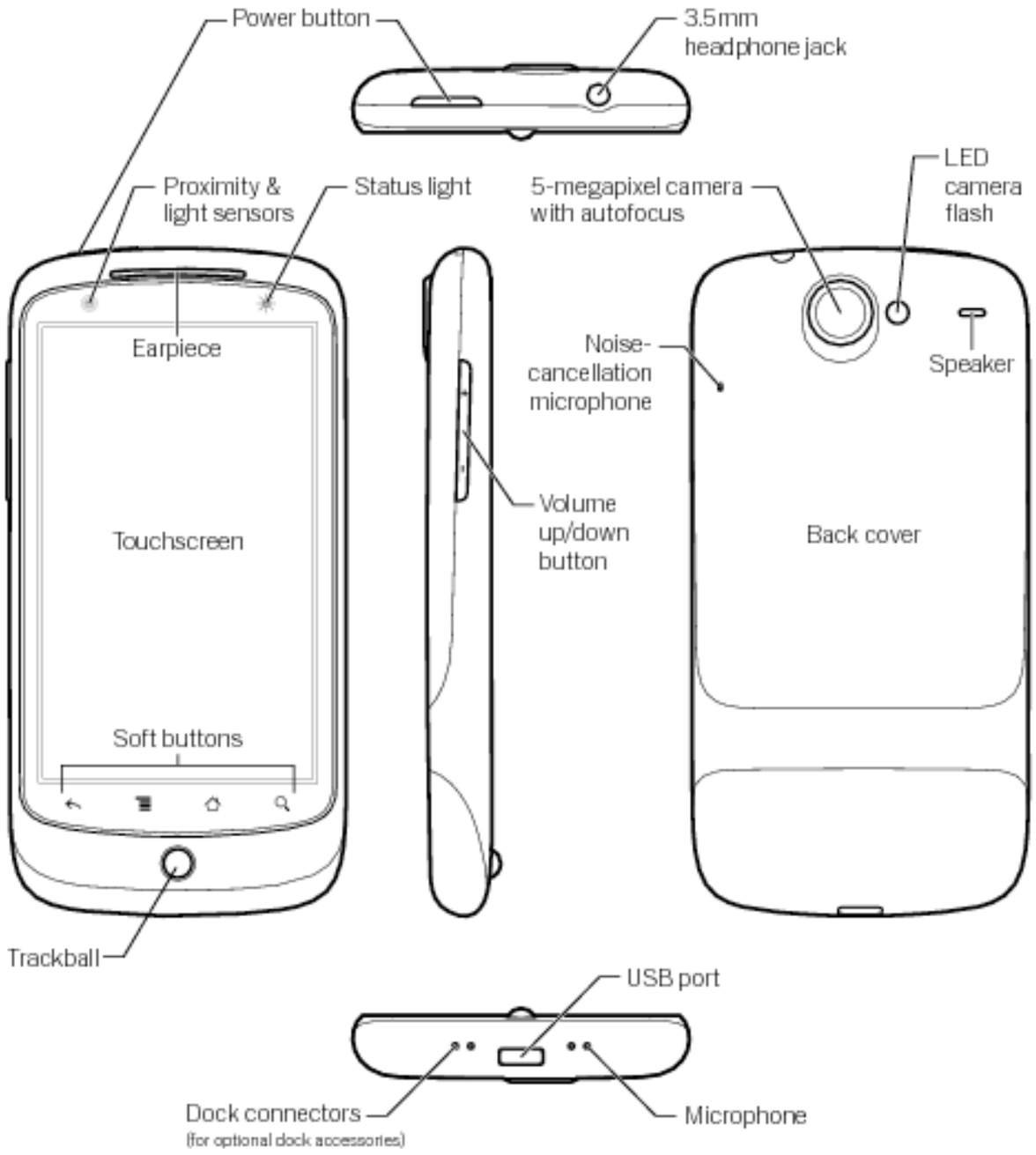
- Unpack the phone



http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

Checking out the phone

- Take a look at the sensors



http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

Making the phone work

http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

Making the phone work

- Charge the phone to 100%
  - USB to computer
  - USB to wall plug
- Wipe the phone
  - “menu” -> “settings” -> “privacy” -> “factory data reset”
    - erase the SD card too
- If necessary, go through on-phone tutorial
  - Do not sync to your Google
  - Enable Location reporting

http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

Making the phone work

- Go through on-phone tutorial
  - You do not need a SIM card (no phone calls)
  - You should connect to WiFi
  - Do not sync to your Google (skip it)
  - Enable Location reporting
  - Set Date and Time to automatic

http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

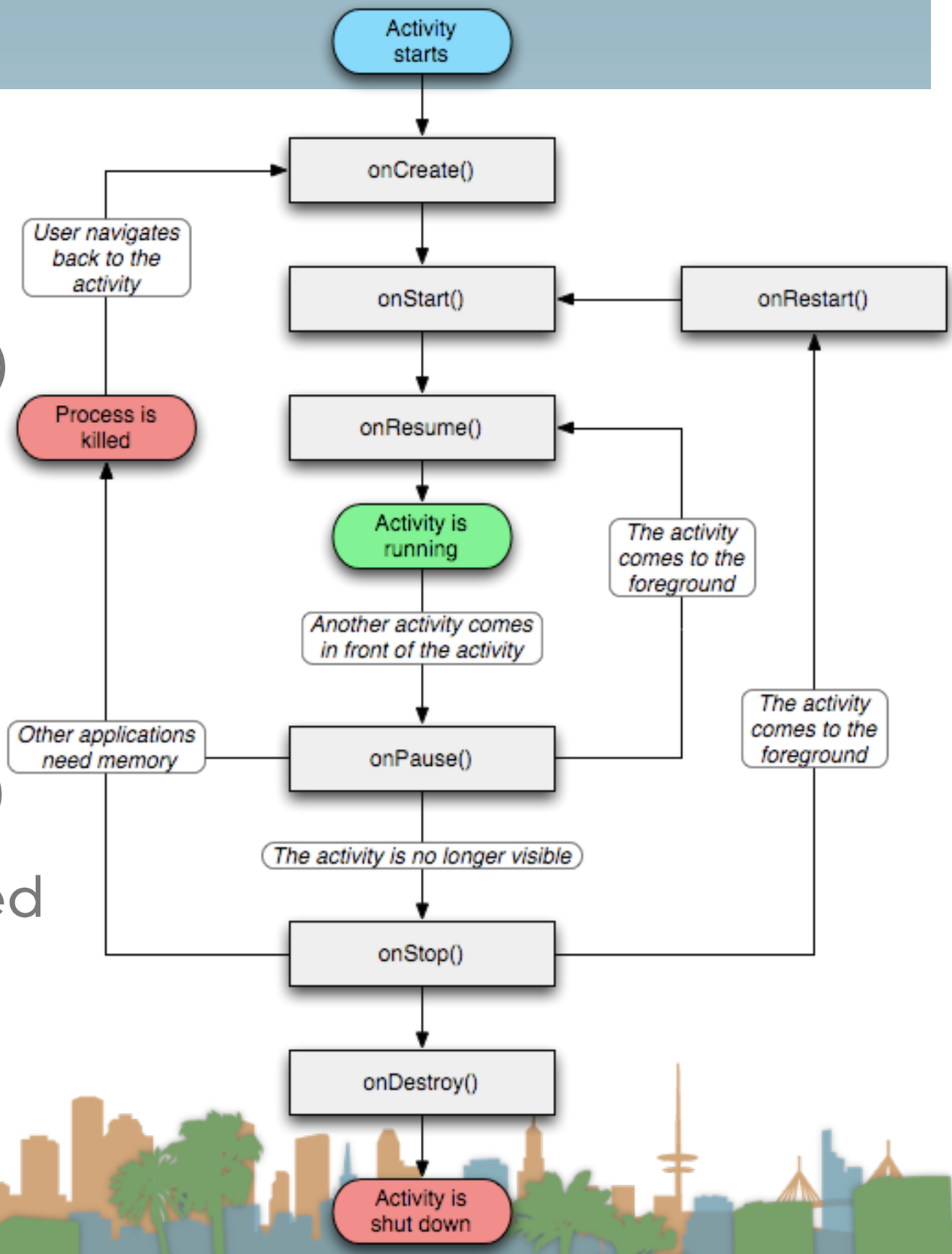
Making the phone work

- Turn on developer mode
  - "home"->"menu"->"settings"->"applications" -> "Development"
    - "USB debugging" on
    - "Stay awake" on
    - "Allow mock locations" on
- Dial *#*#CHECKIN#*#*
  - to update phone software

http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

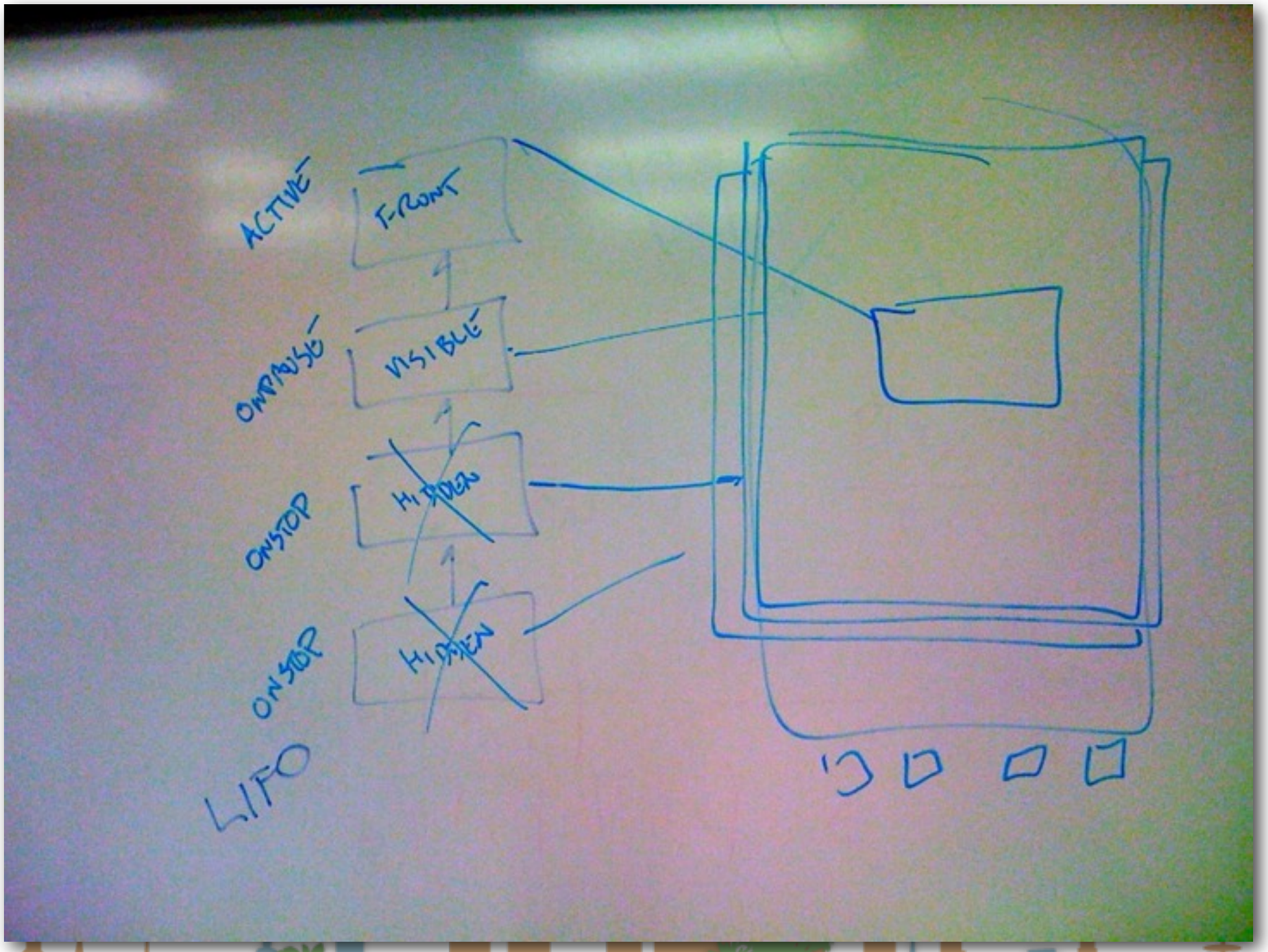
Activity Lifecycle

- Key loops
  - Entire Lifetime
    - onCreate()- onDestroy()
  - Visible Lifetime
    - onStart() - onStop()
  - Foreground Lifetime
    - onResume() - onPause()
- onPause() may be followed by kill

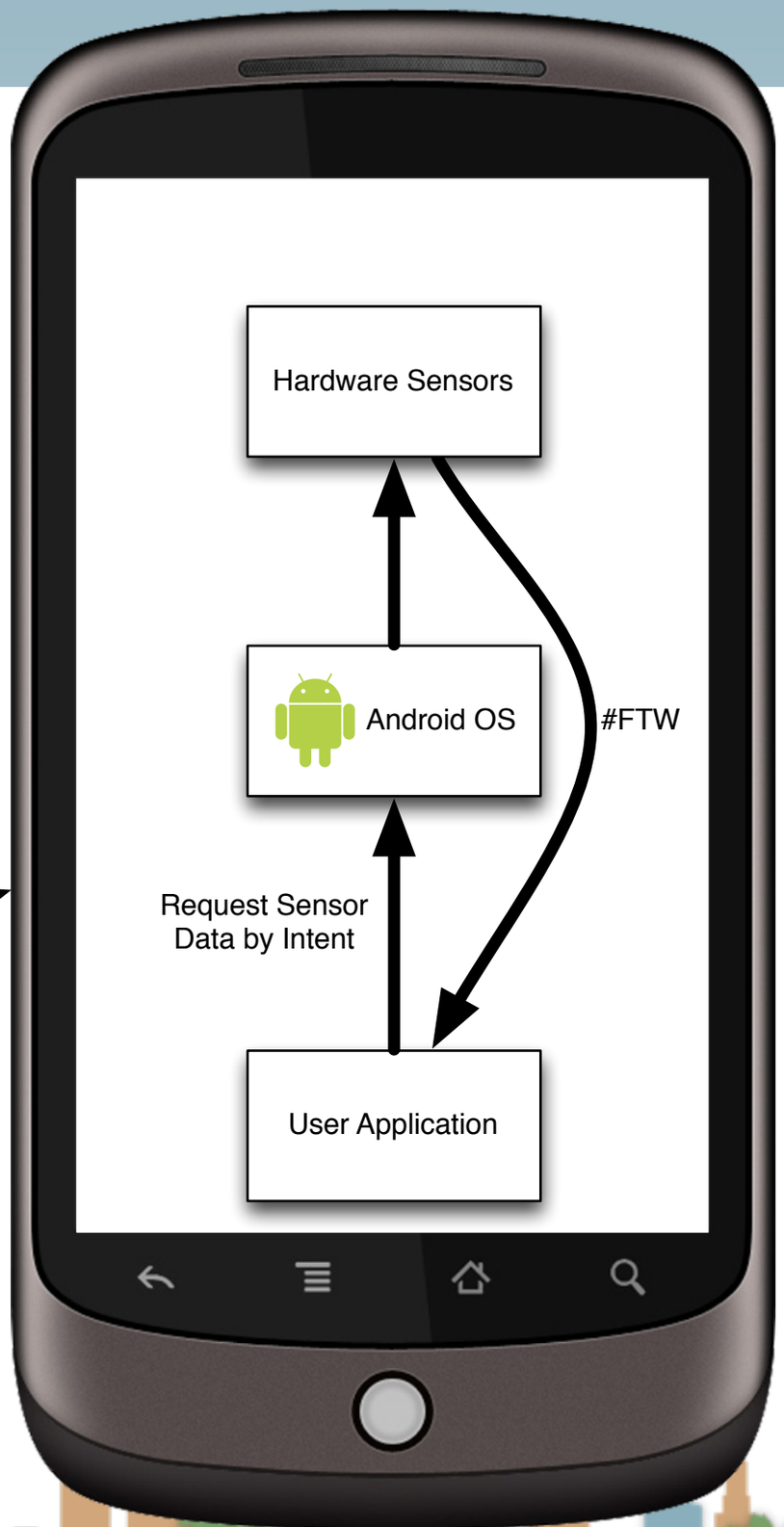


http://developer.android.com/reference/android/app/Activity.html

Activity Stack



How to handle the assignment



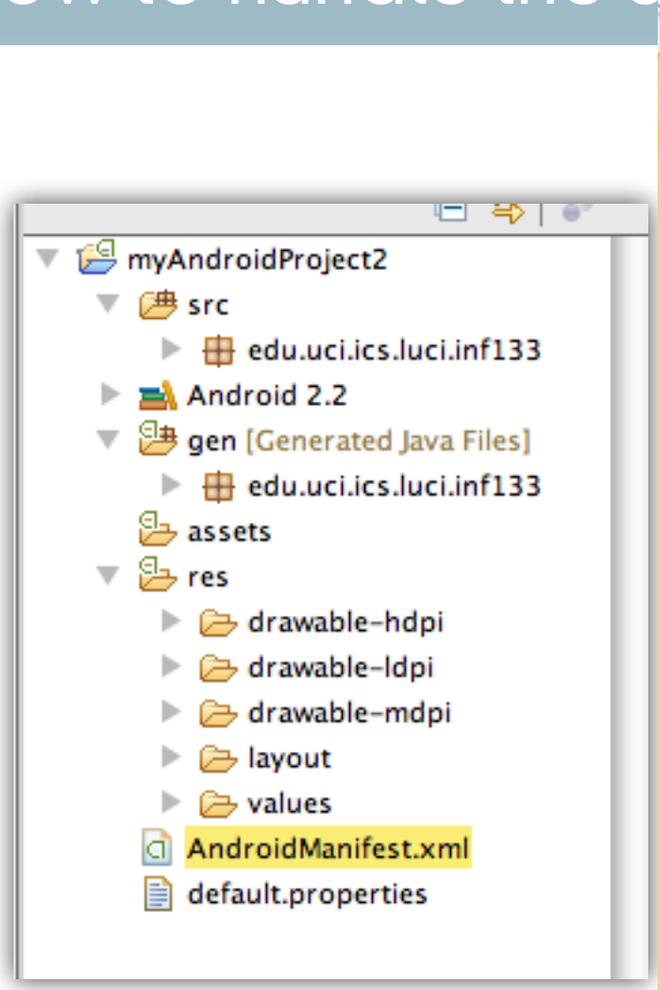
Real Phone

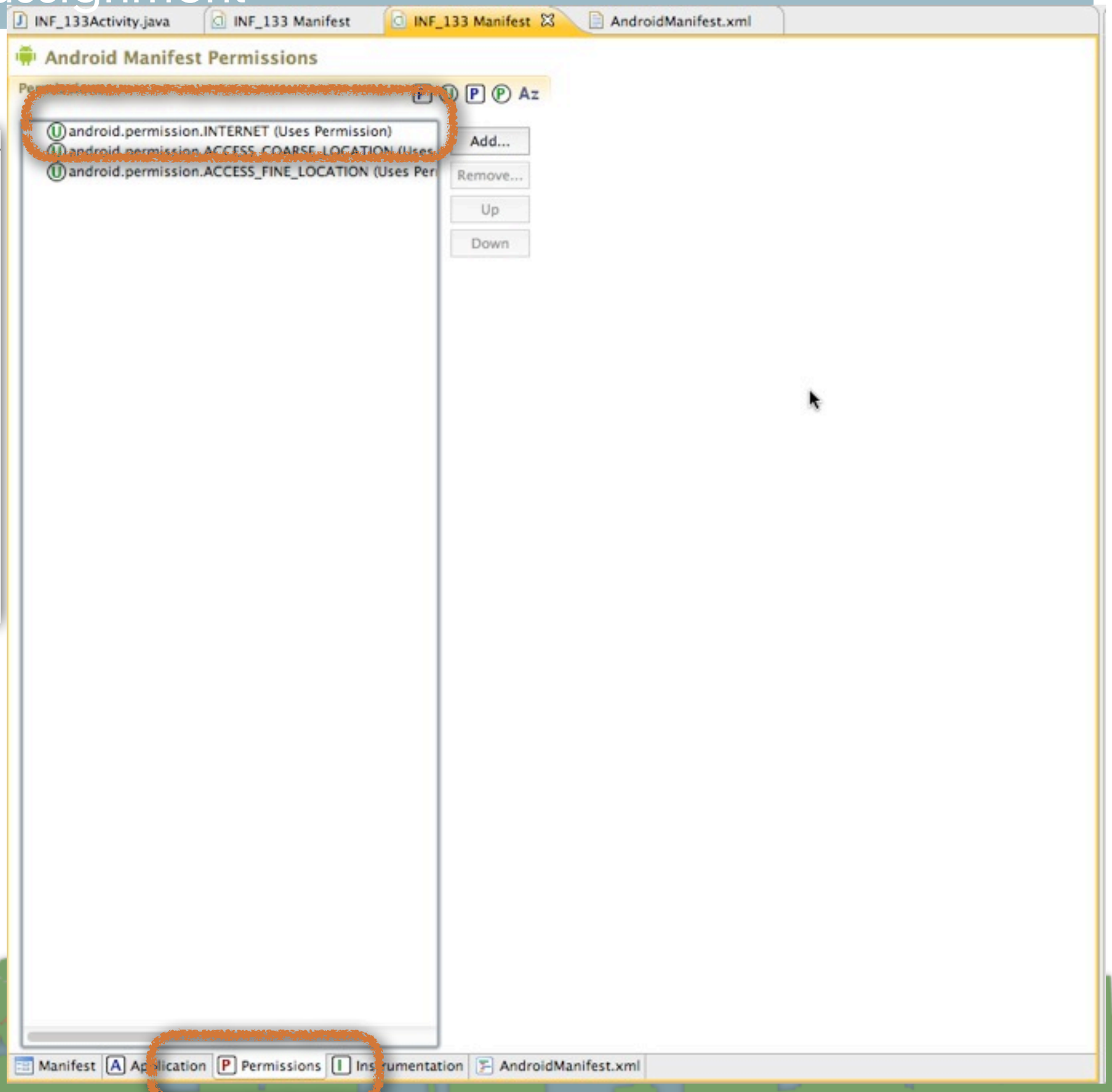
How to handle the assignment

- Start a new Android Application project for Android 2.3.3
  - Give your application permissions in AndroidManifest.xml
    - Add a "Uses Permission"
      - to use the Internet
        - android.permission.INTERNET
      - to use location
        - android.permission.ACCESS_FINE_LOCATION
        - android.permission.ACCESS_COARSE_LOCATION
  - Make your application debuggable

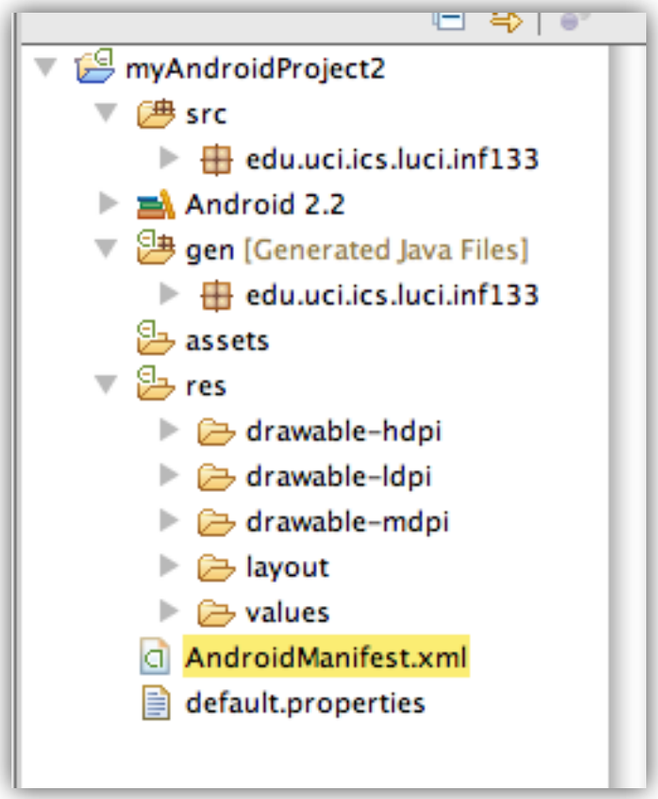
http://developer.android.com/guide/developing/device.html

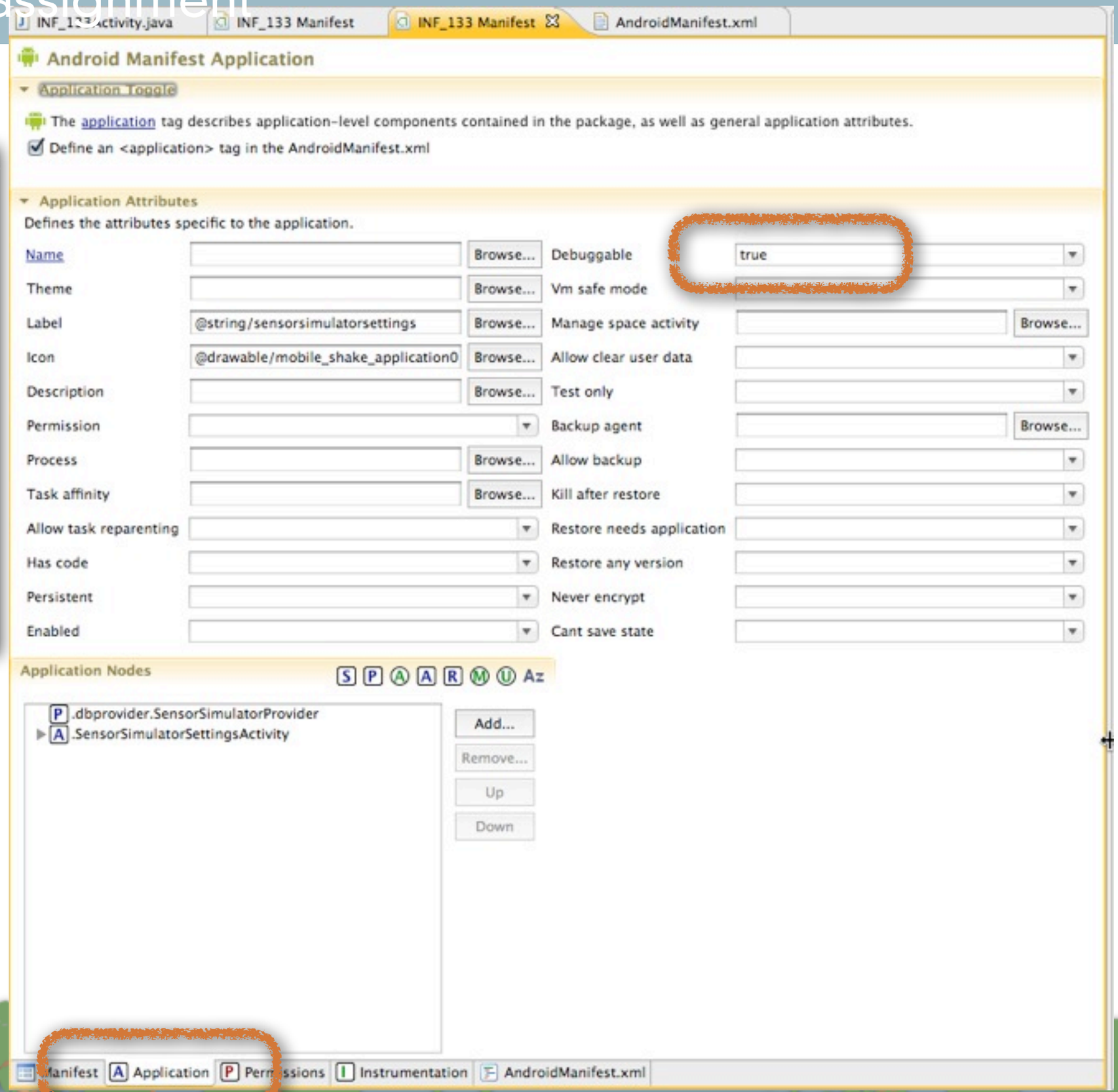
How to handle the assignment





How to handle the assignment





How to handle the assignment

- High-Level
  - You are going to ask Android to give you information about the phone's orientation
  - You are going to do something in response to the information (with U/I and audio)

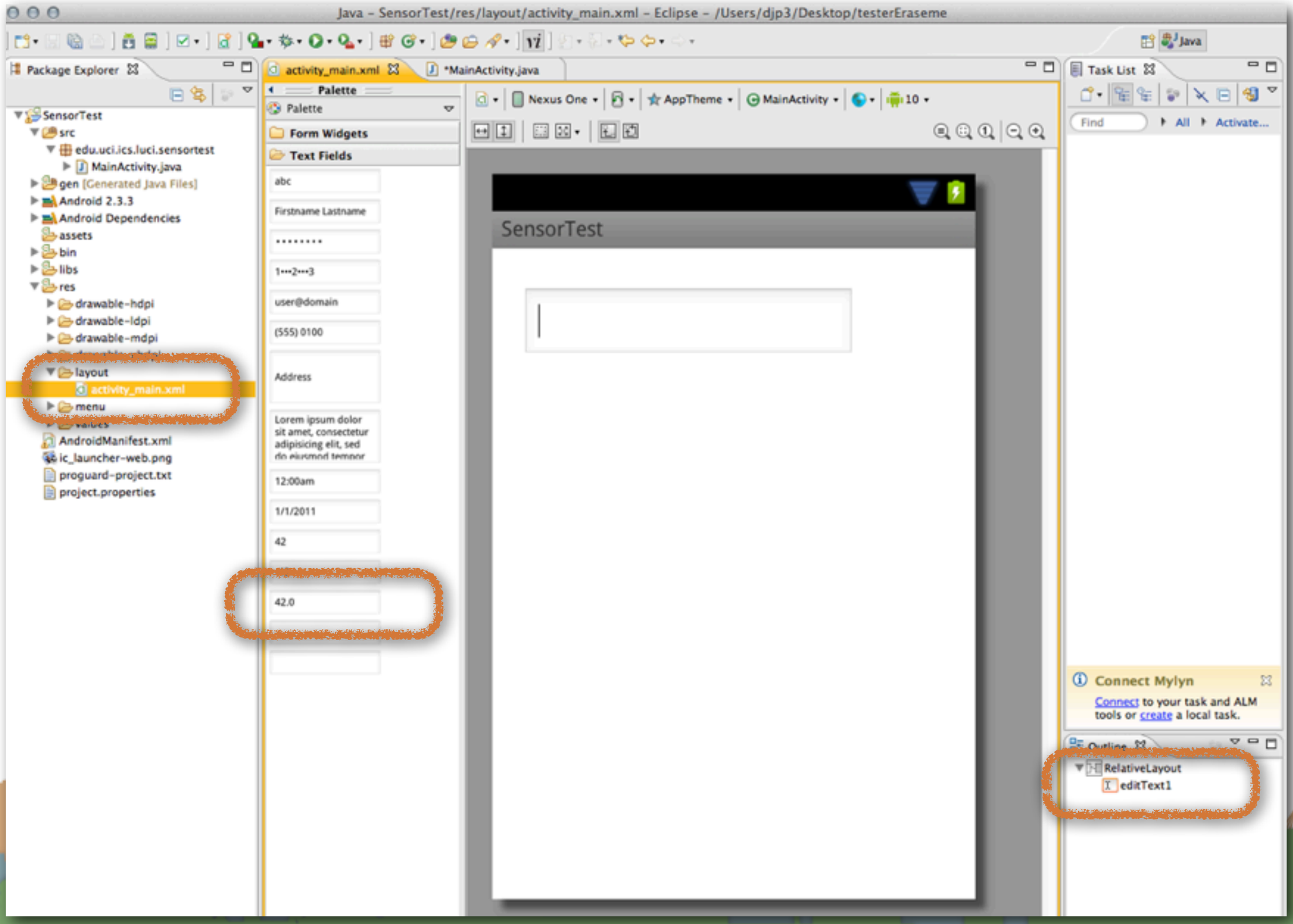
How to handle the assignment

- The Main Problem
  - Information from the phone's sensors are going to arrive much much faster than the phone can redraw the U/I
  - If you don't manage this, your application will crash while it backs up waiting for you U/I to draw
  - Let's do it the wrong but easy to understand way first

http://developer.android.com/guide/developing/device.html

How to handle the assignment

- Step 1: Create a place in the U/I to show the sensor data



How to handle the assignment

- Step 1: Create a place in the U/I to show the sensor data
  - The U/I object is a static class named “R”
- Step 2: Access the Android Sensor Service
- Step 3: Create a SensorEventListener that will handle the asynchronous callbacks
- Step 4: Tell the phone you are ready to get sensor readings
- Step 5: Tell the phone you don’t want sensor readings any more

How to handle the assignment

activity_main.xml | MainActivity.java

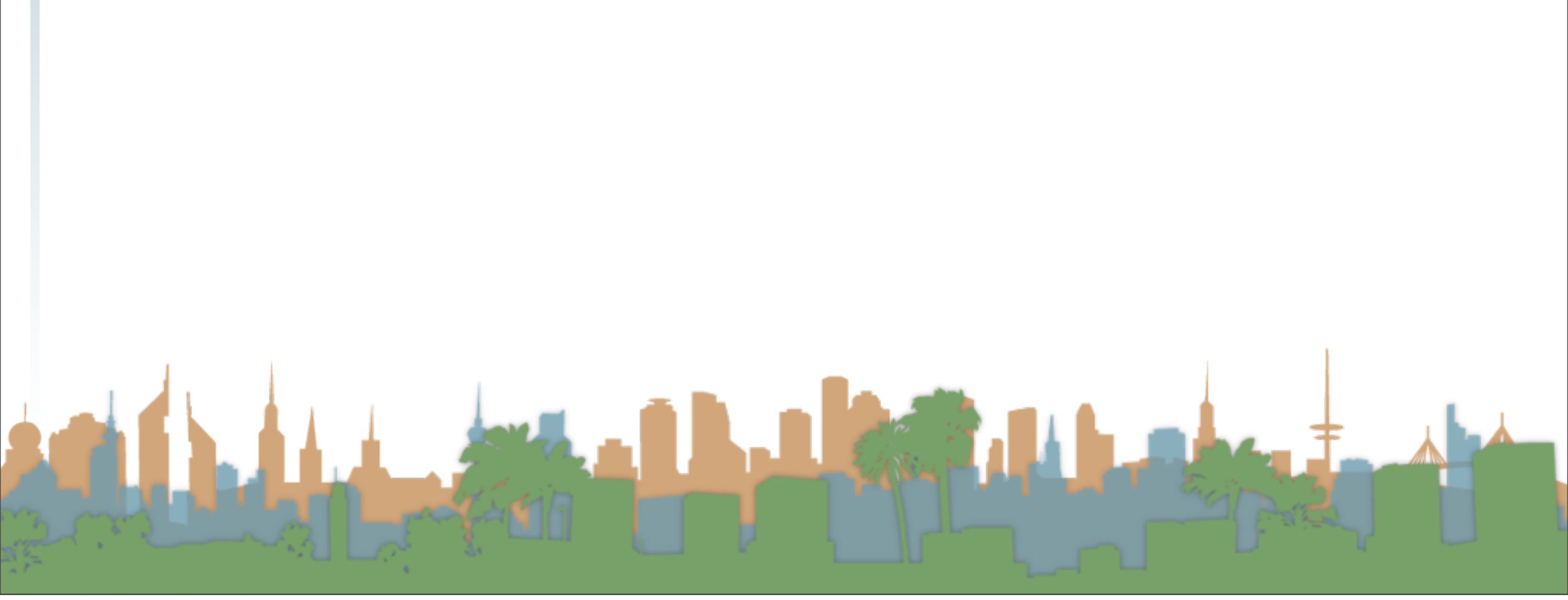
```
package edu.uci.ics.luci.sensortest;

import android.app.Activity;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.widget.TextView;

public class MainActivity extends Activity {

    private SensorManager mSensorManager;
    private TextView mTextViewLight;
    private SensorEventListener mEventListenerLight;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mTextViewLight = (TextView) findViewById(R.id.editText1);

        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

        mEventListenerLight = new SensorEventListener() {
            @Override
            public void onSensorChanged(SensorEvent event) {
                float[] values = event.values;
                mTextViewLight.setText("Light is " + values[0]);
            }

            @Override
            public void onAccuracyChanged(Sensor arg0, int arg1) {
            }
        };
    }
```

How t

```
public class MainActivity extends Activity {

    private SensorManager mSensorManager;
    private TextView mTextViewLight;
    private SensorEventListener mEventListenerLight;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mTextViewLight = (TextView) findViewById(R.id.editText1);

        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

        mEventListenerLight = new SensorEventListener() {
            @Override
            public void onSensorChanged(SensorEvent event) {
                float[] values = event.values;
                mTextViewLight.setText("Light is " + values[0]);
            }

            @Override
            public void onAccuracyChanged(Sensor arg0, int arg1) {
            }
        };
    }

    @Override
    public void onResume() {
        super.onResume();
        mSensorManager.registerListener(mEventListenerLight,
                mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT),
                SensorManager.SENSOR_DELAY_FASTEST);

    }

    @Override
    public void onStop() {
        mSensorManager.unregisterListener(mEventListenerLight);
        super.onStop();

    }
}
```

What it looks like when it's working

- Demo



Hints

```
public class MainActivity extends Activity{

    private SensorManager mSensorManager;
    private SensorEventListener mEventListenerLight;
    private TextView mTextViewLight;
    private float lastLightValue = 0.0f;
    private Lock lock = new ReentrantLock();


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        mTextViewLight = (TextView) findViewById(R.id.editText1);

        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

        mEventListenerLight = new SensorEventListener() {
            @Override
            public void onSensorChanged(SensorEvent event) {
                float[] values = event.values;
                lastLightValue = values[0];
                updateUI();
            }

            @Override
            public void onAccuracyChanged(Sensor arg0, int arg1) {
            }
        };

    }

    public void updateUI() {
        if(lock.tryLock()){
            mTextViewLight.setText("Light is " + lastLightValue);
            lock.unlock();
        }
    }
```

Hints

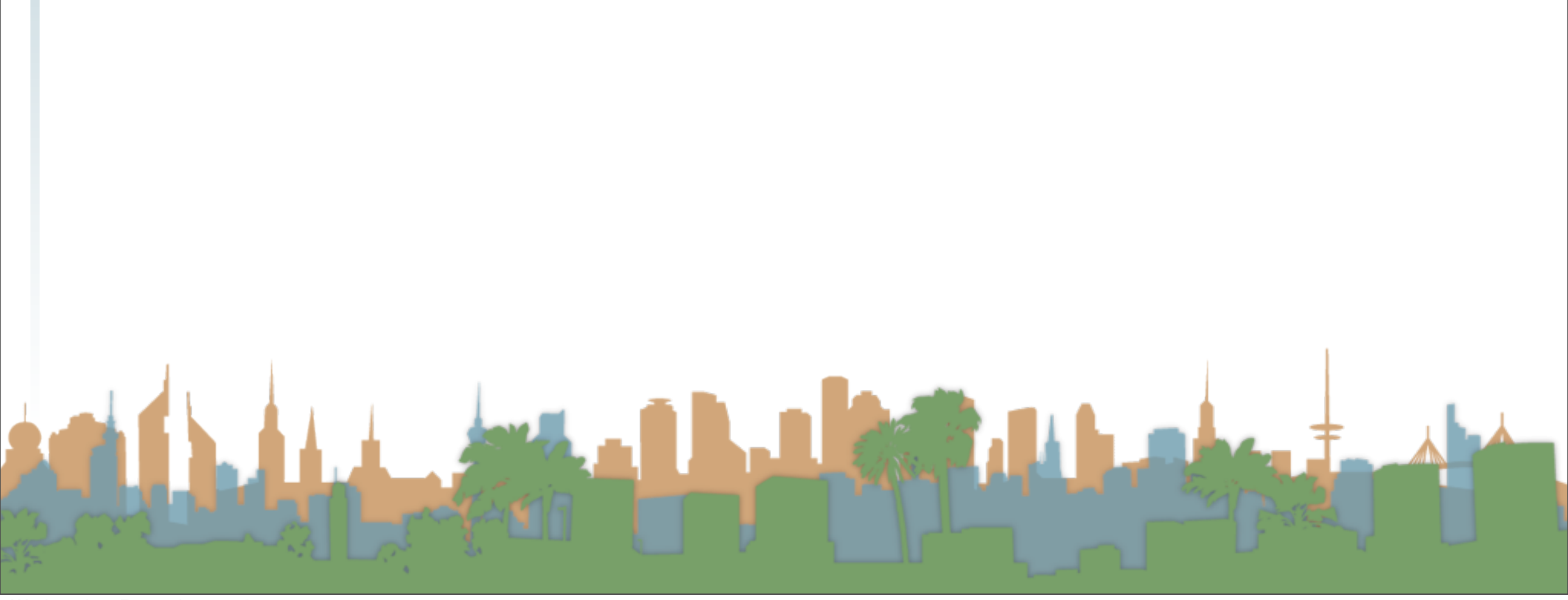
- Playing a sound
  - The key is the MediaPlayer call
  - Do not instantiate more than one MediaPlayer object

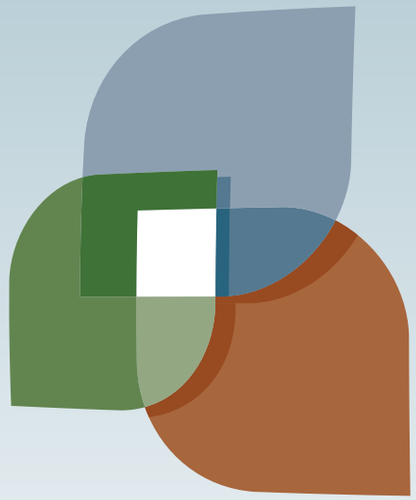
```
static MediaPlayer mp = new MediaPlayer();
public void playSound(String path) {
    if (mp.isPlaying()) {
        return;
    }
    mp.reset();
    try {
        mp.setDataSource(path);
        mp.prepare();
    } catch (Exception ex) {
        Log.d("main thread ex", ex.getStackTrace()[0].toString() + " path: " + path);
    }
    mp.start();
}
```

  - http://developer.android.com/guide/topics/media/index.html

Hints

- Playing a sound
  - You will need to get the audio media onto the phone





LUCI

Wednesday, December 5, 12