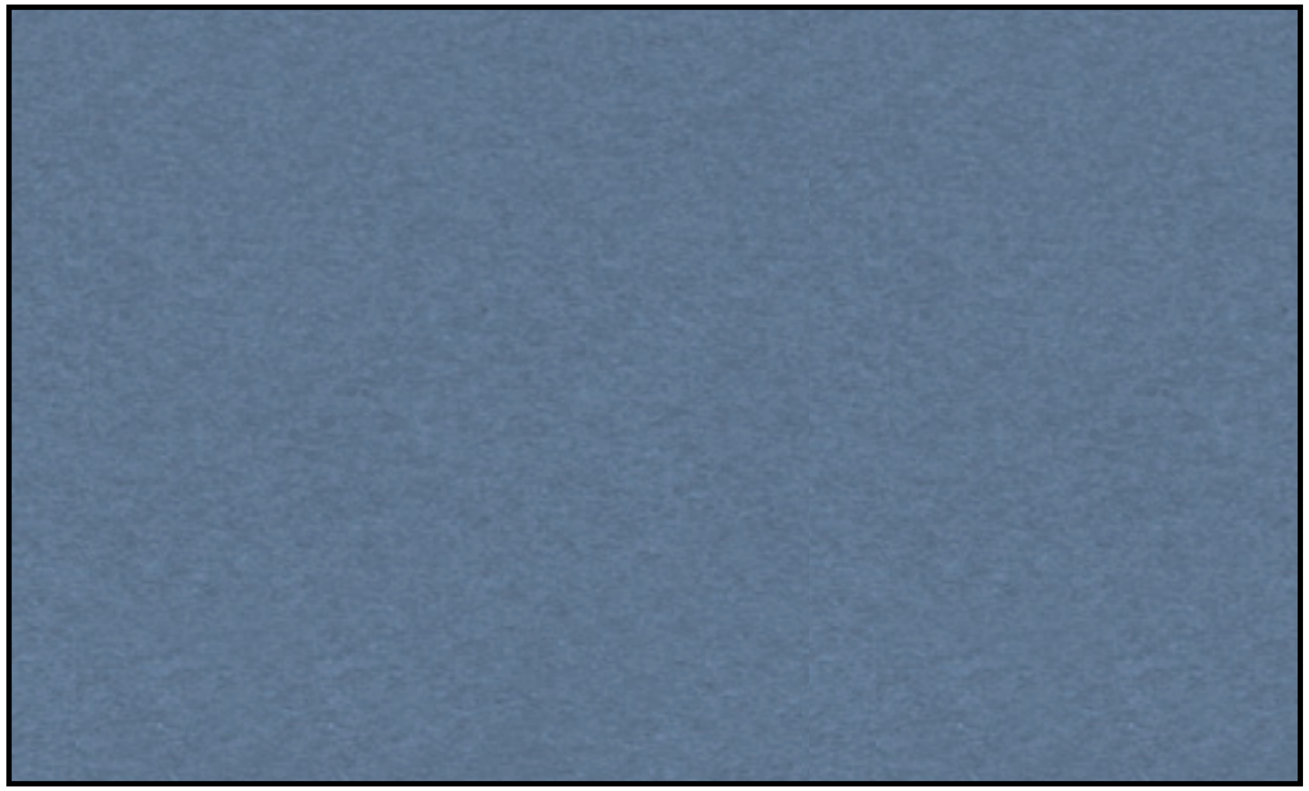
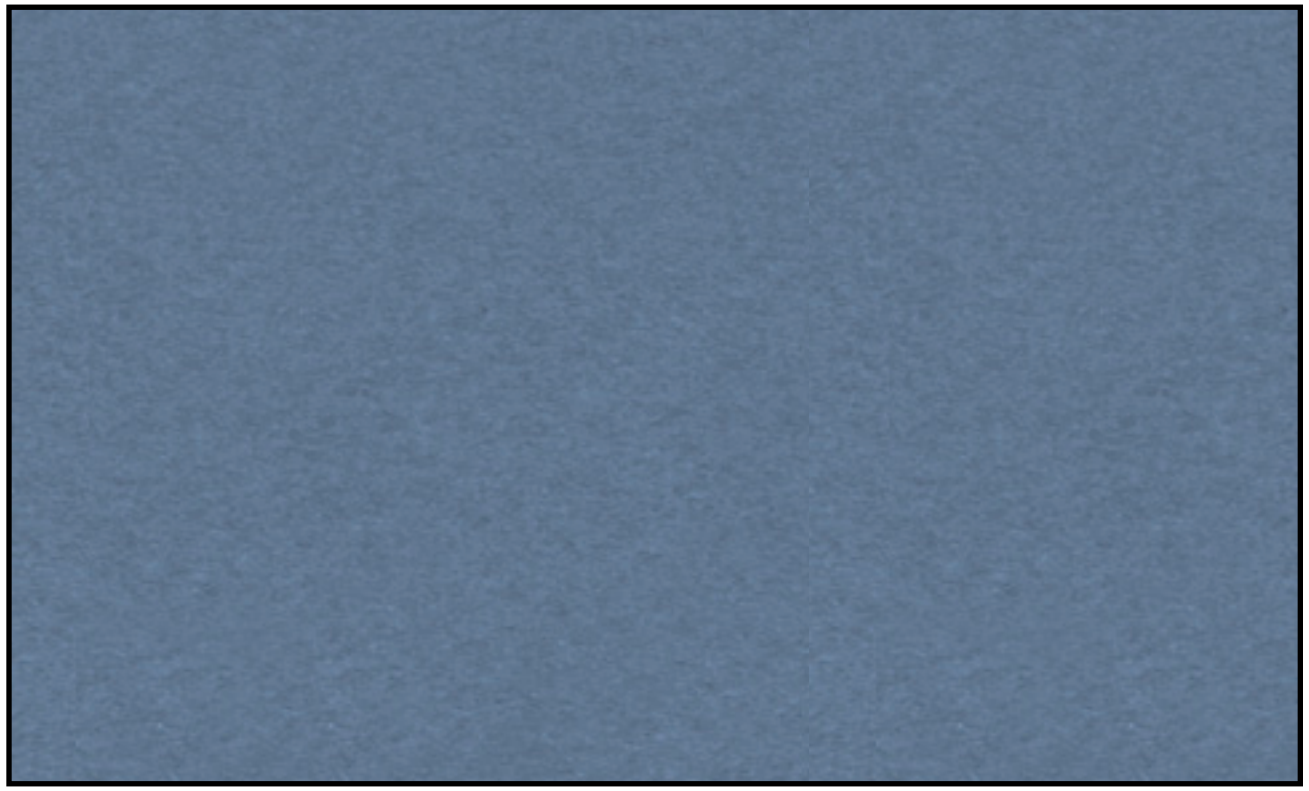
User Interaction: Intro to Android

Asst. Professor Donald J. Patterson
INF 133 Fall 2010

Multi-Touch Assignment Winner

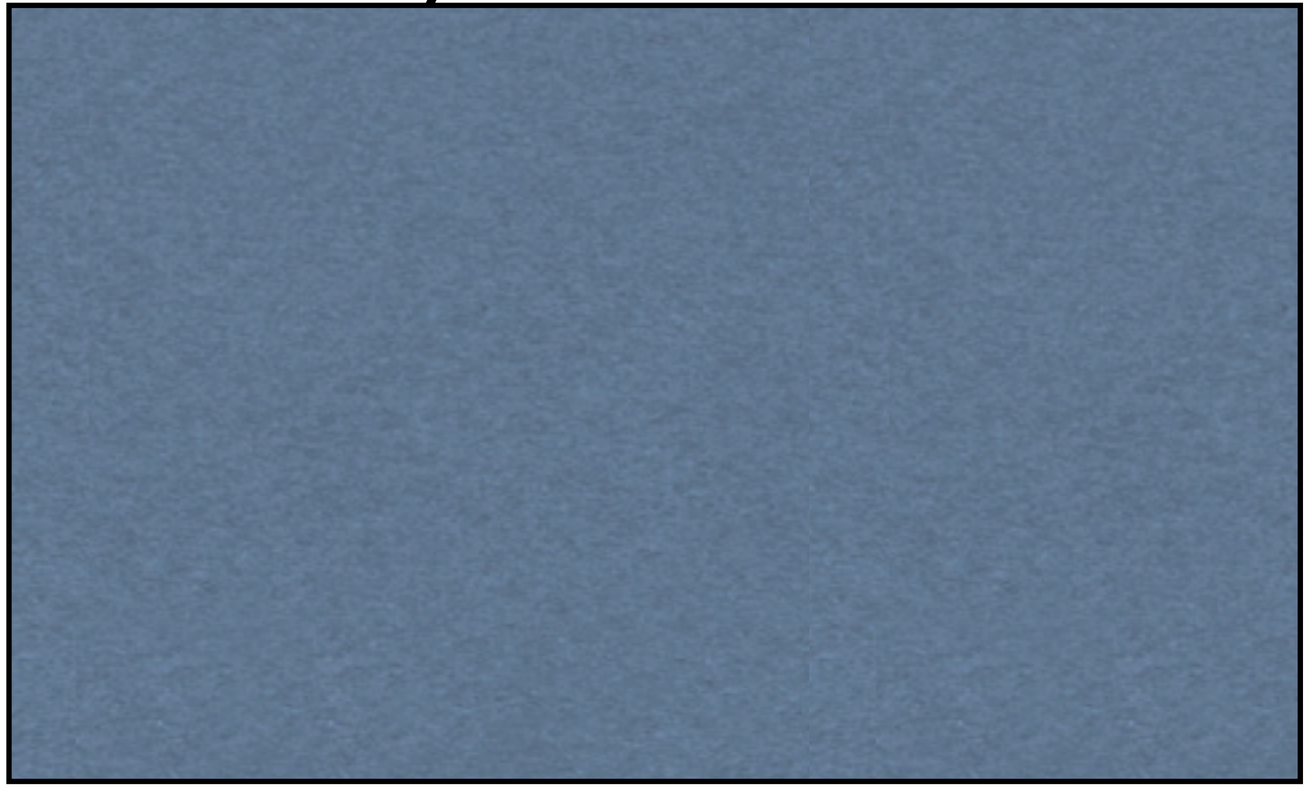


Multi-Touch Assignment Winner



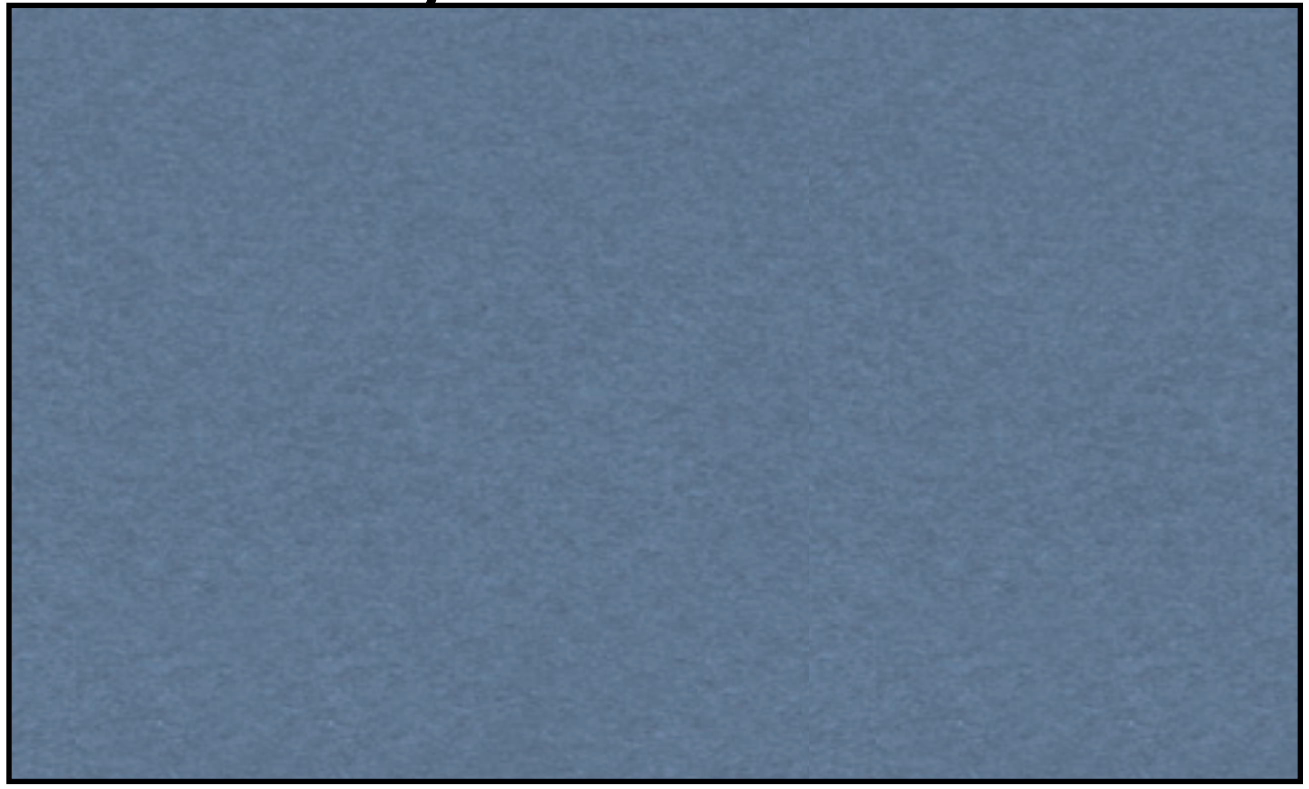
Multi-Touch Assignment Winner

- Volleyball

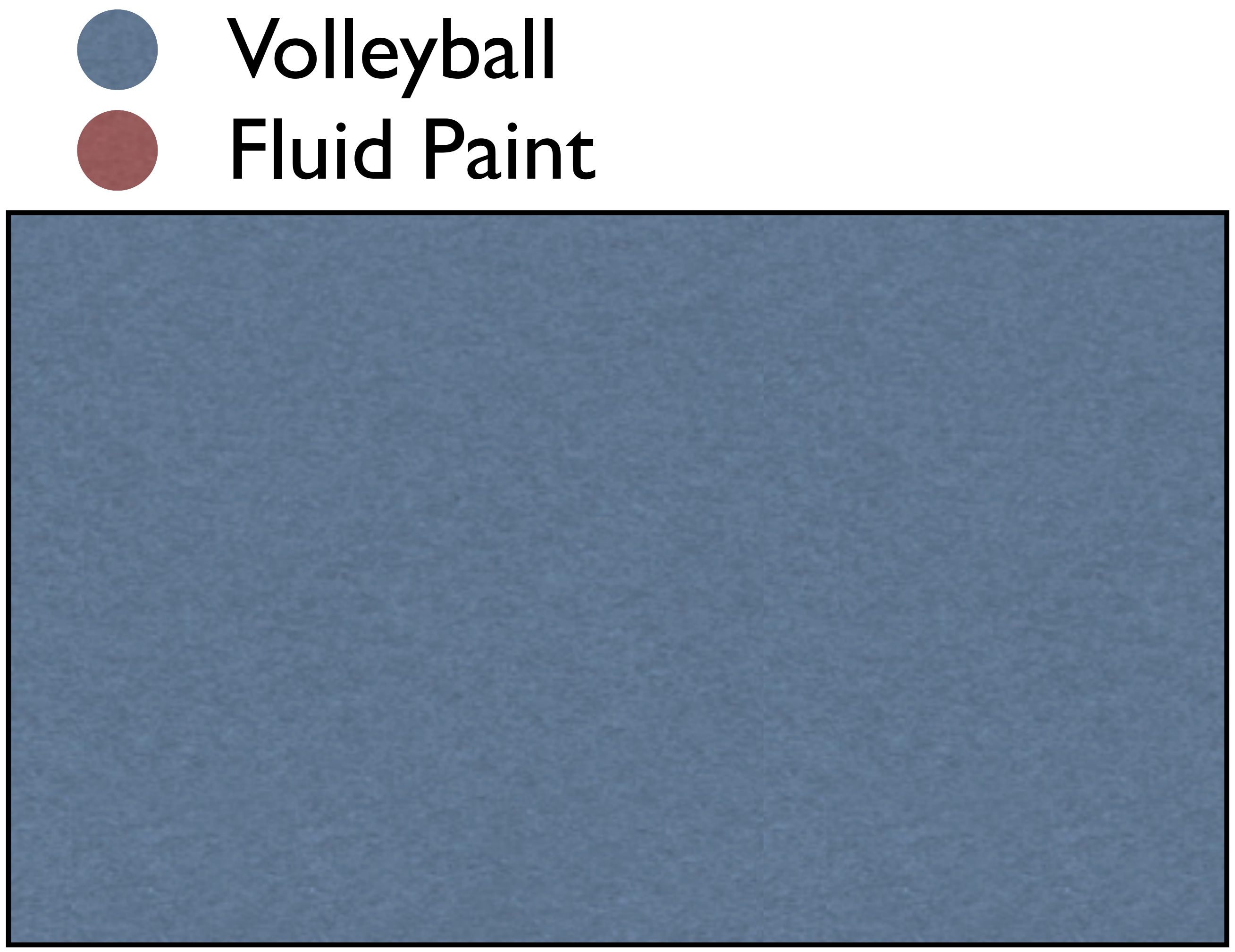


Multi-Touch Assignment Winner

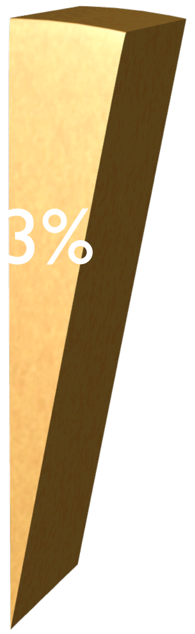
- Volleyball



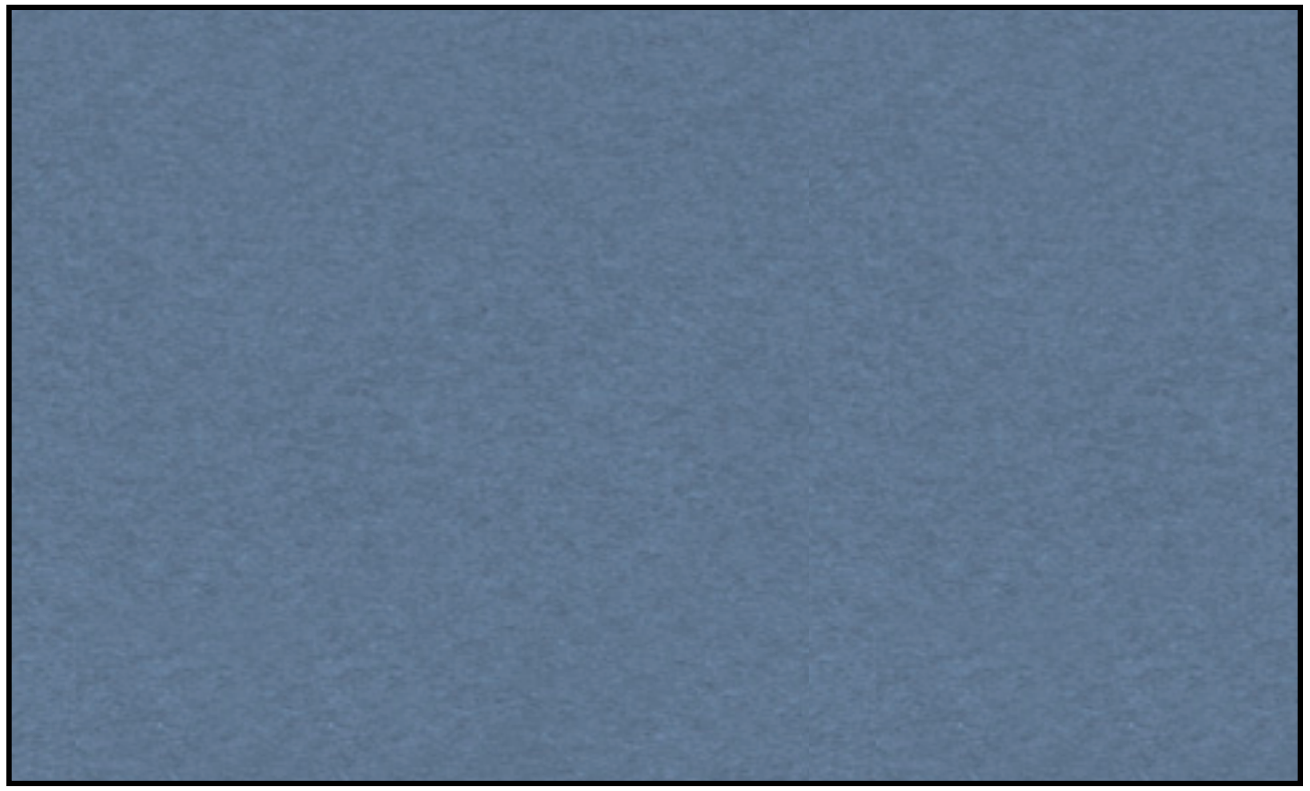
Multi-Touch Assignment Winner



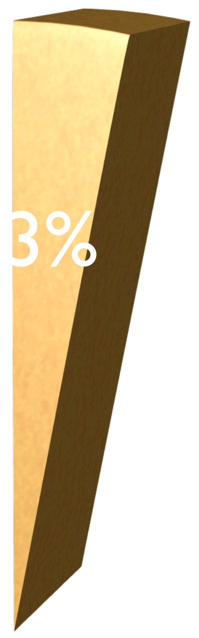
Multi-Touch Assignment Winner



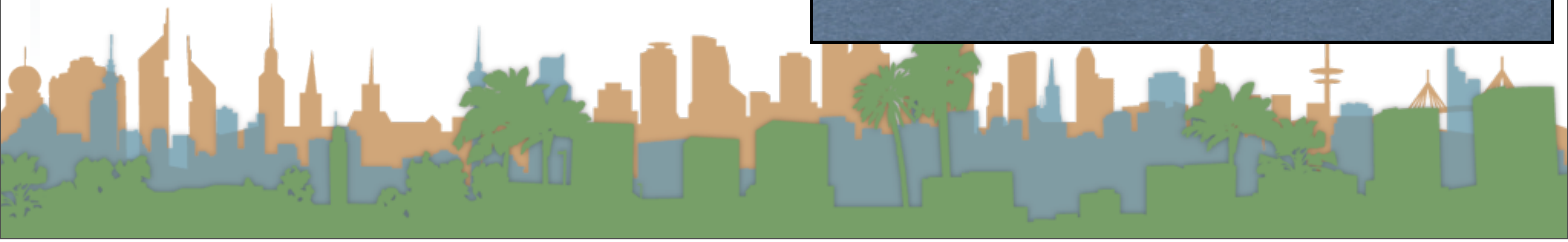
● Volleyball
● Fluid Paint



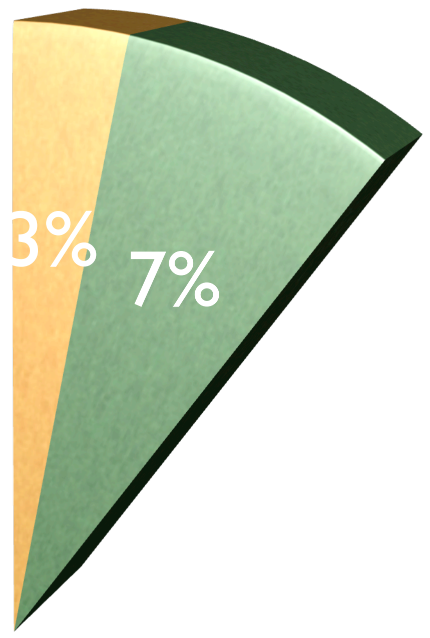
Multi-Touch Assignment Winner



● Volleyball
● Fluid Paint
● Ghetto Basketball

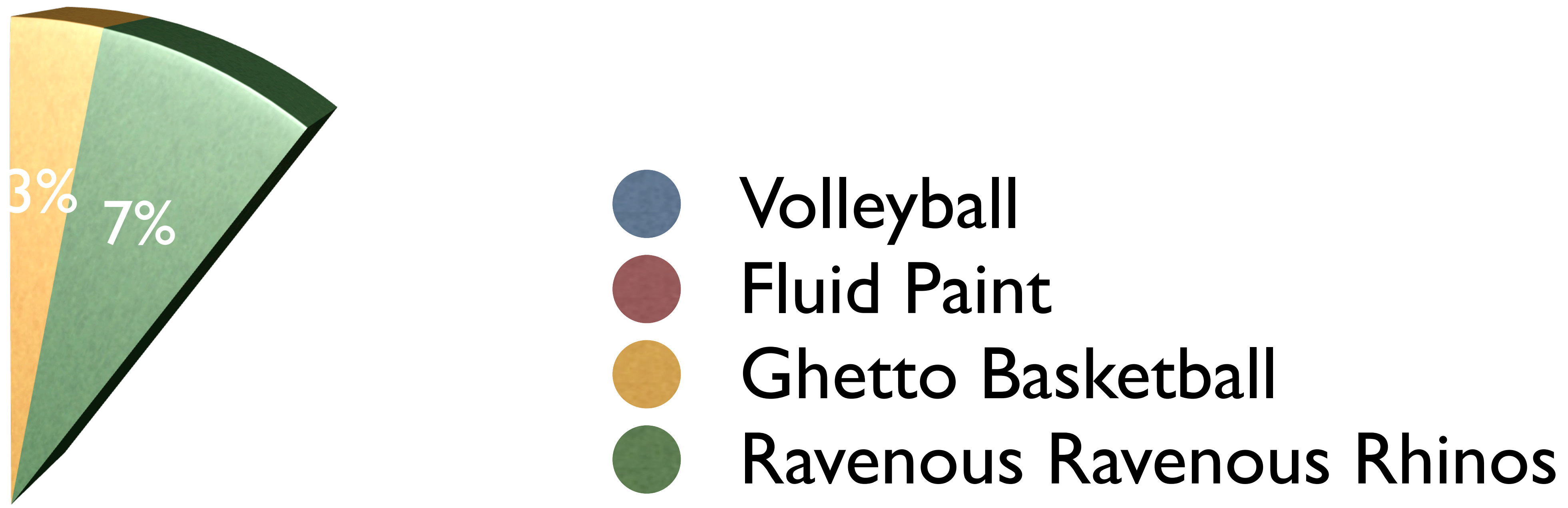


Multi-Touch Assignment Winner

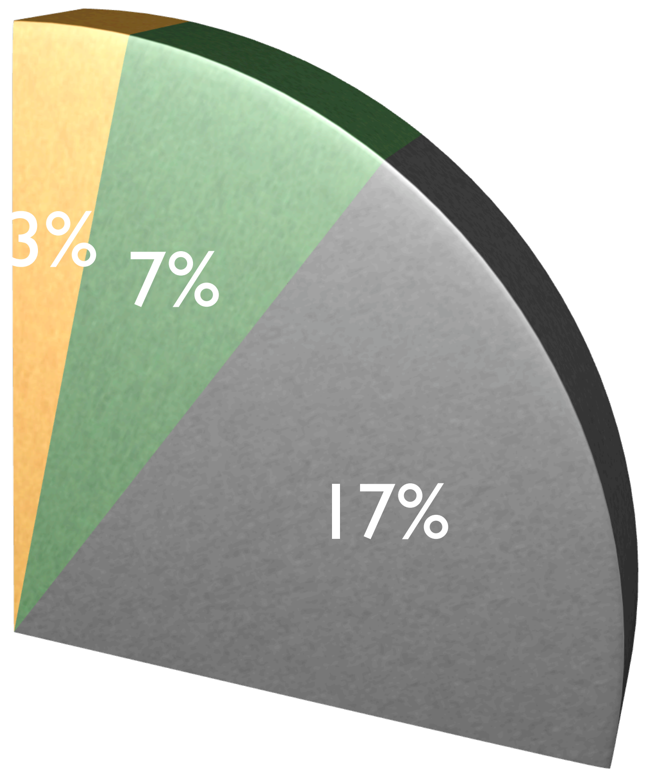


● Volleyball
● Fluid Paint
● Ghetto Basketball

Multi-Touch Assignment Winner

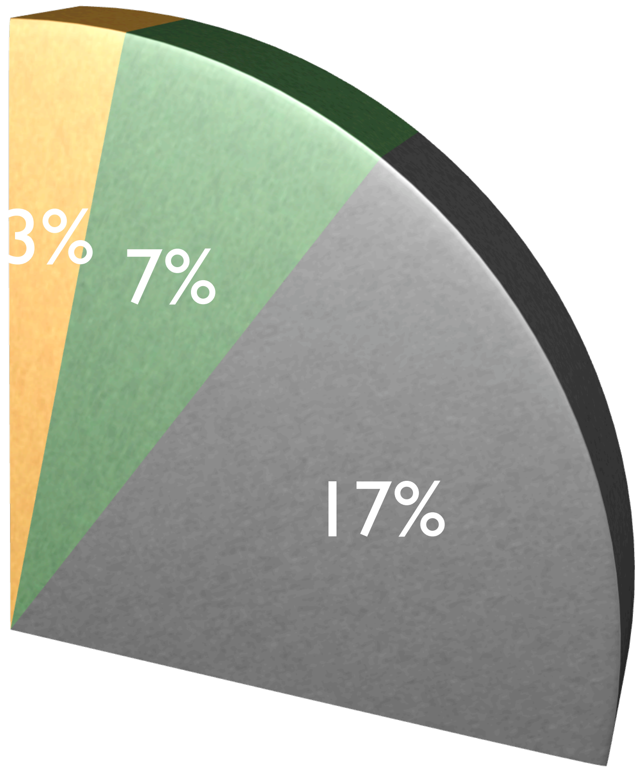


Multi-Touch Assignment Winner



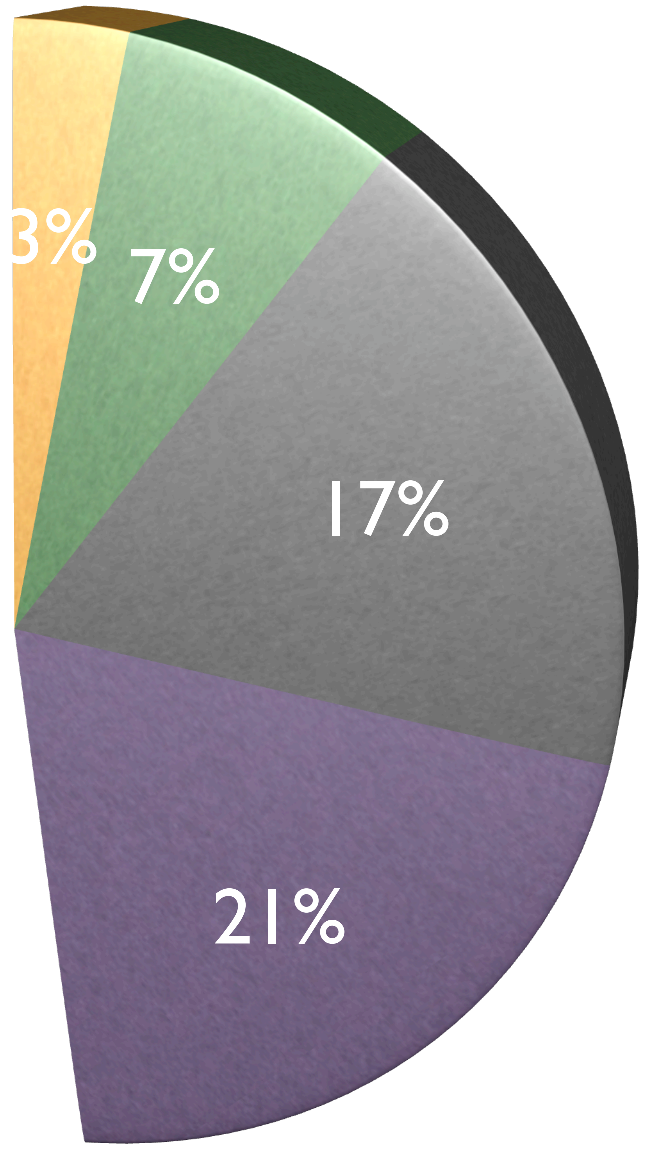
- ● Volleyball
- ● Fluid Paint
- ● Ghetto Basketball
- ● Ravenous Ravenous Rhinos

Multi-Touch Assignment Winner



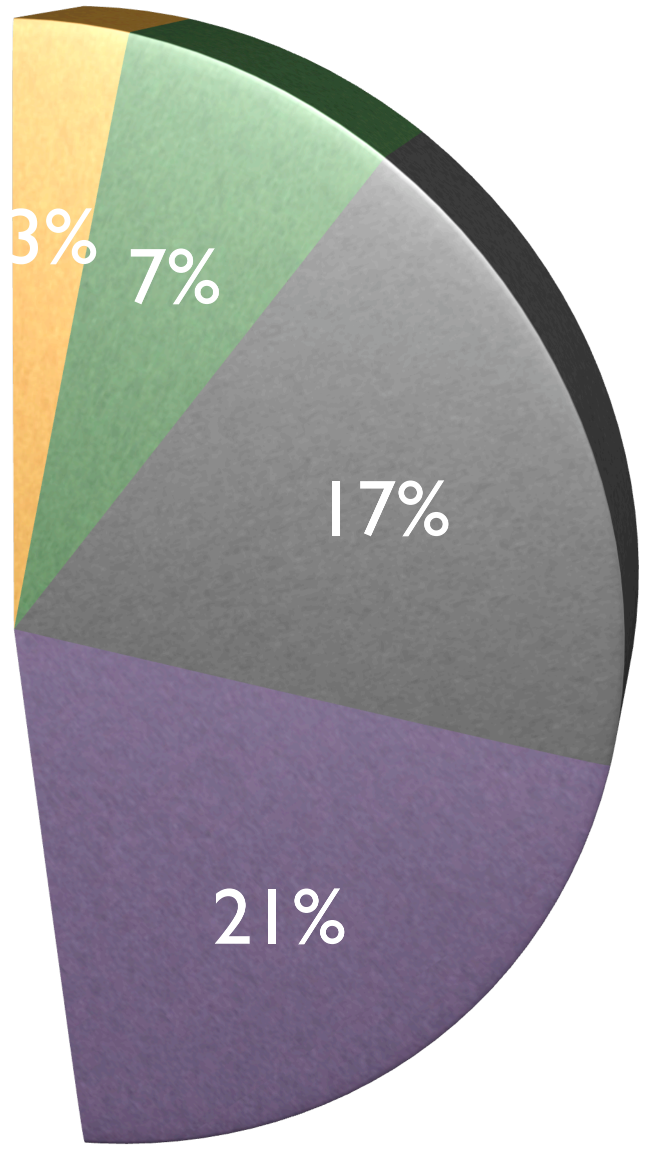
● Volleyball
● Fluid Paint
● Ghetto Basketball
● Ravenous Ravenous Rhinos
● Good Video

Multi-Touch Assignment Winner



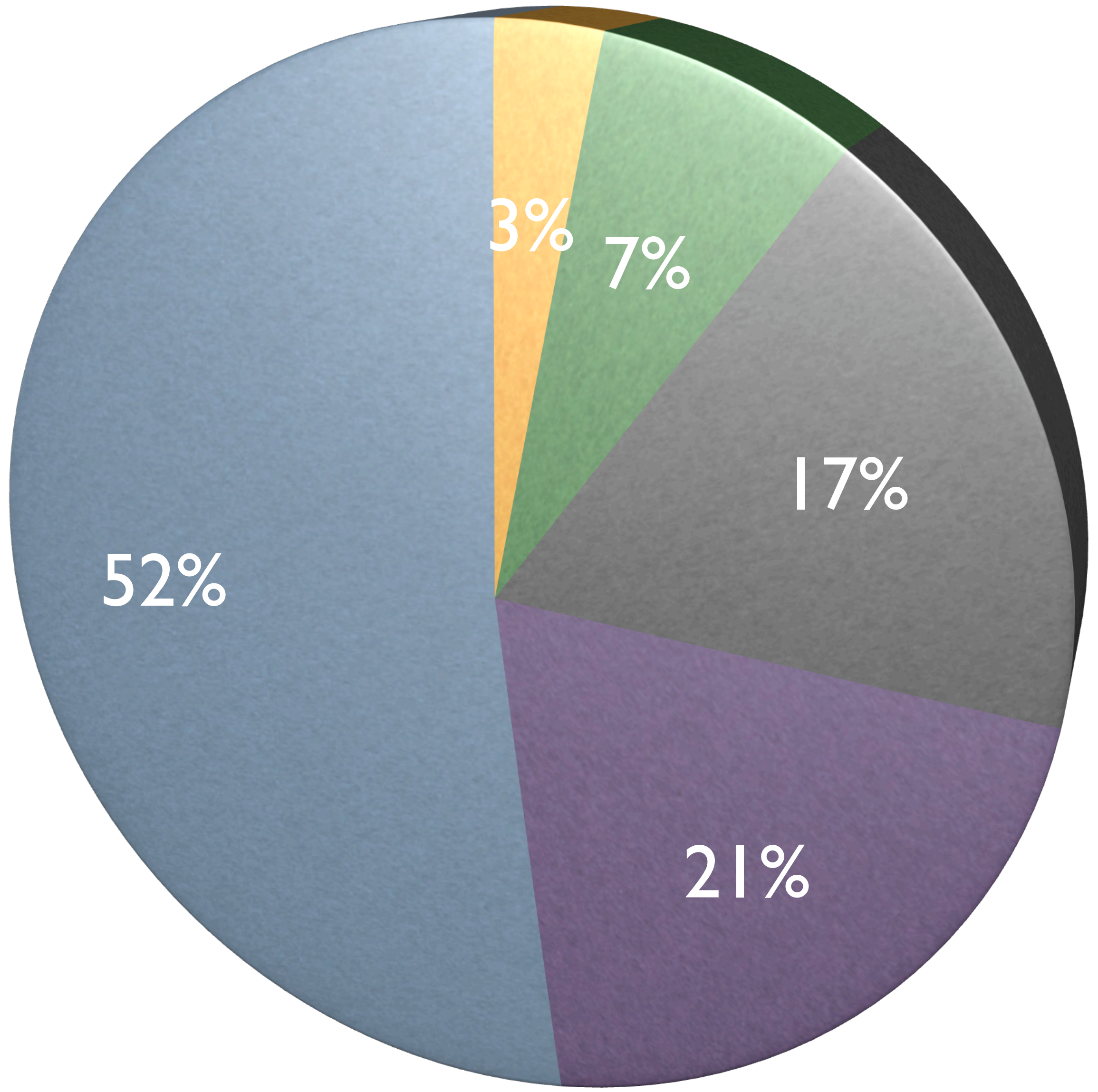
● Volleyball
● Fluid Paint
● Ghetto Basketball
● Ravenous Ravenous Rhinos
● Good Video

Multi-Touch Assignment Winner



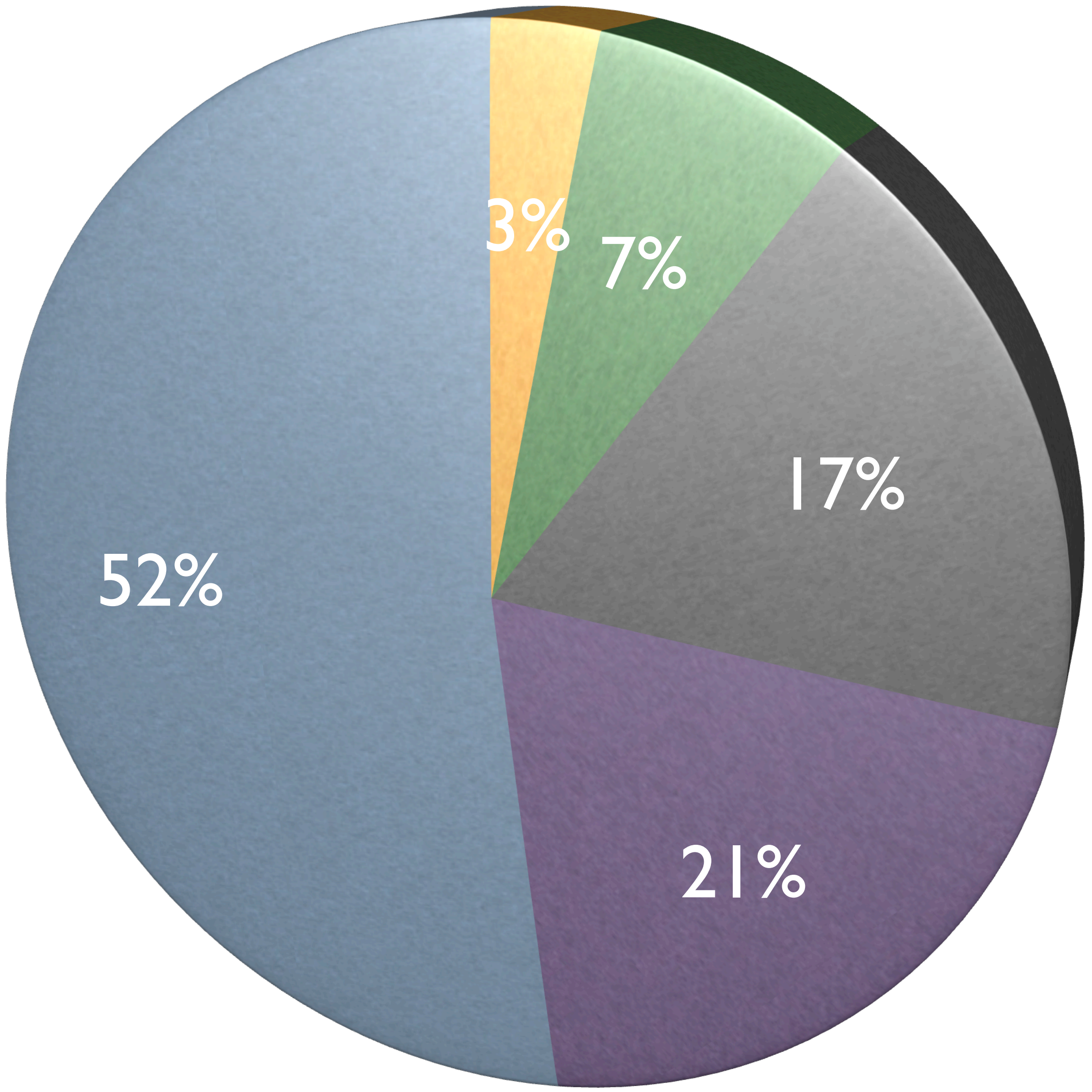
- ● Volleyball
- ● Fluid Paint
- ● Ghetto Basketball
- ● Ravenous Ravenous Rhinos
- ● Good Video
- ● Firing Range

Multi-Touch Assignment Winner



● Volleyball
● Fluid Paint
● Ghetto Basketball
● Ravenous Ravenous Rhinos
● Good Video
● Firing Range

Multi-Touch Assignment Winner



● Volleyball
● Fluid Paint
● Ghetto Basketball
● Ravenous Ravenous Rhinos
● Good Video
● Firing Range
● Space Hockey

Space Hockey

Assignment 5

NOKIA
N800
Web - Main Page - Wikimedi...
http://commons.wikimedia.org/wiki/Mai
NOKIA

Developing with Sensors on Android

- Requirements
- SDK/AVD
- Eclipse Plug-in
- Hello World

http://developer.android.com/guide/index.html

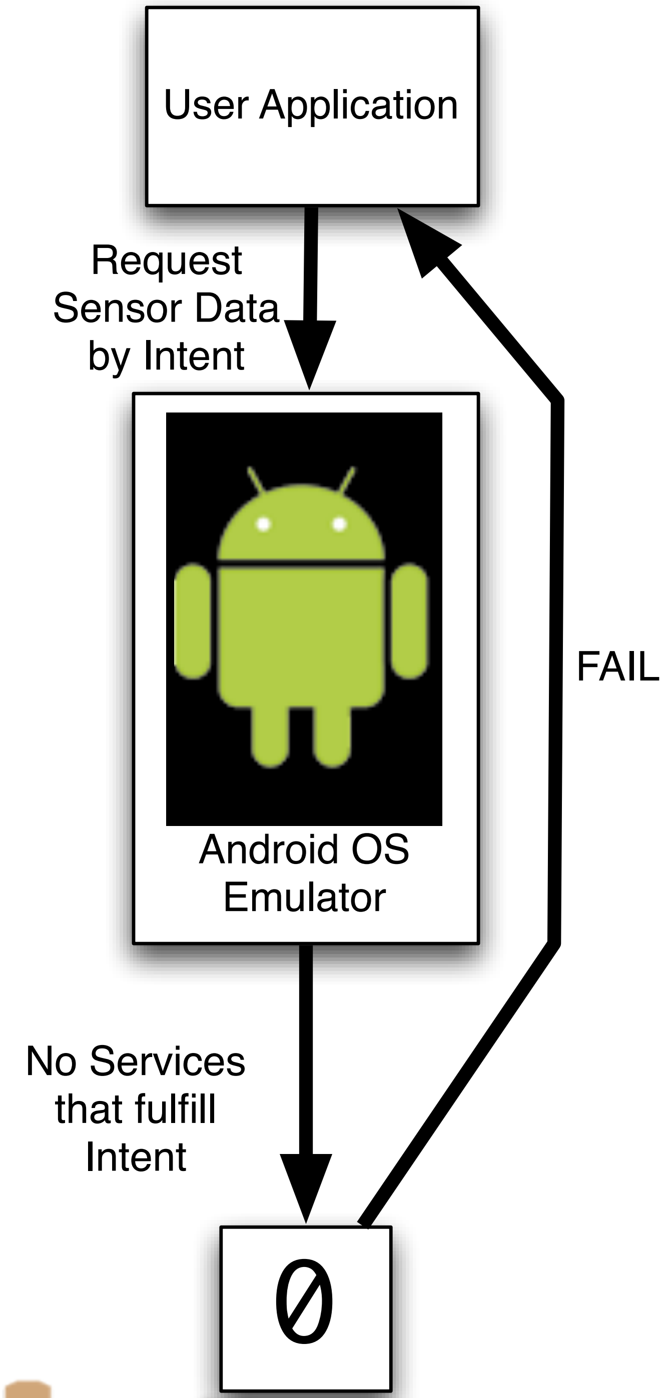
Developing with Sensors on Android

- Requirements
- SDK/AVD
- Eclipse Plug-in
- Hello World

http://developer.android.com/guide/index.html

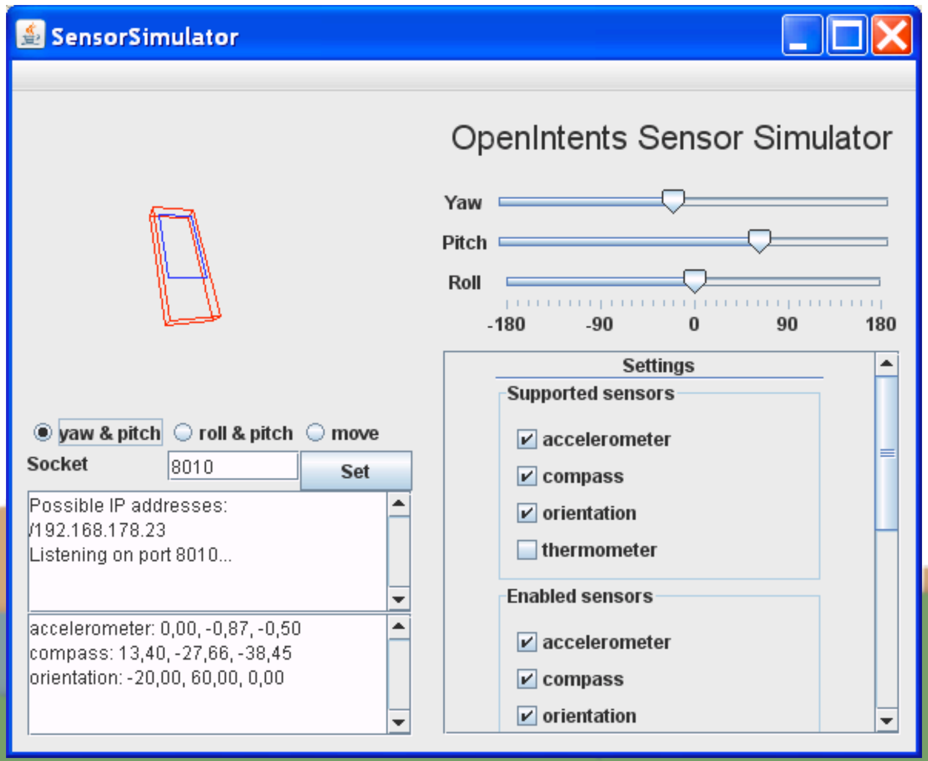
Intro to Android:

- On the emulator


User Application
Request Sensor Data by Intent
Android OS Emulator
No Services that fulfill Intent
FAIL
∅

Intro to Android:

- Simulating Sensors on the Emulator would require:



SensorSimulator

OpenIntents Sensor Simulator

Yaw

Pitch

Roll

-180 -90 0 90 180

(● yaw & pitch) ○ roll & pitch ○ move

Socket: 8010 Set

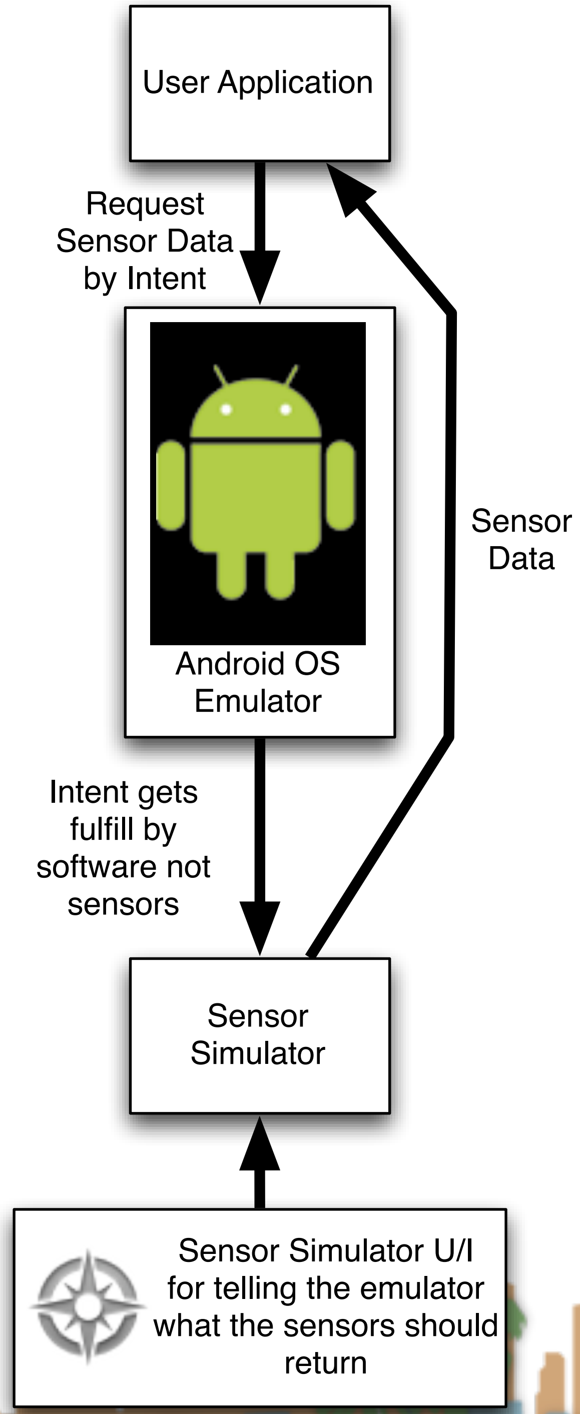
```
Possible IP addresses:
/192.168.178.23
Listening on port 8010...
```

```
accelerometer: 0,00, -0,87, -0,50
compass: 13,40, -27,66, -38,45
orientation: -20,00, 60,00, 0,00
```

Settings

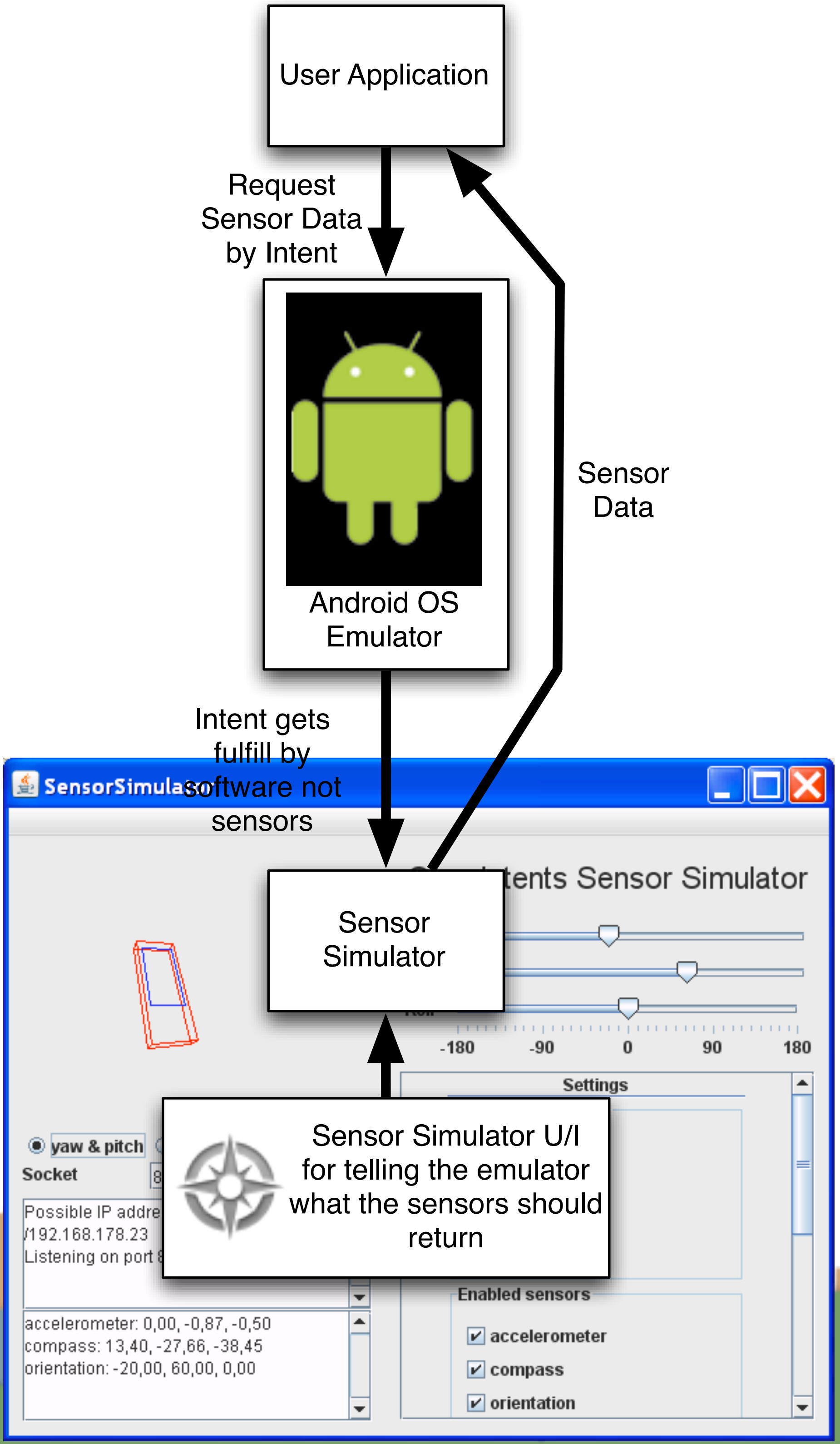
Supported sensors
- ☑ accelerometer
- ☑ compass
- ☑ orientation
- ☐ thermometer

Enabled sensors
- ☑ accelerometer
- ☑ compass
- ☑ orientation


User Application
Request Sensor Data by Intent
Android OS Emulator
Sensor Data
Intent gets fulfill by software not sensors
Sensor Simulator
Sensor Simulator U/I for telling the emulator what the sensors should return

Intro to Android:

- Simulating Sensors on the Emulator would require:

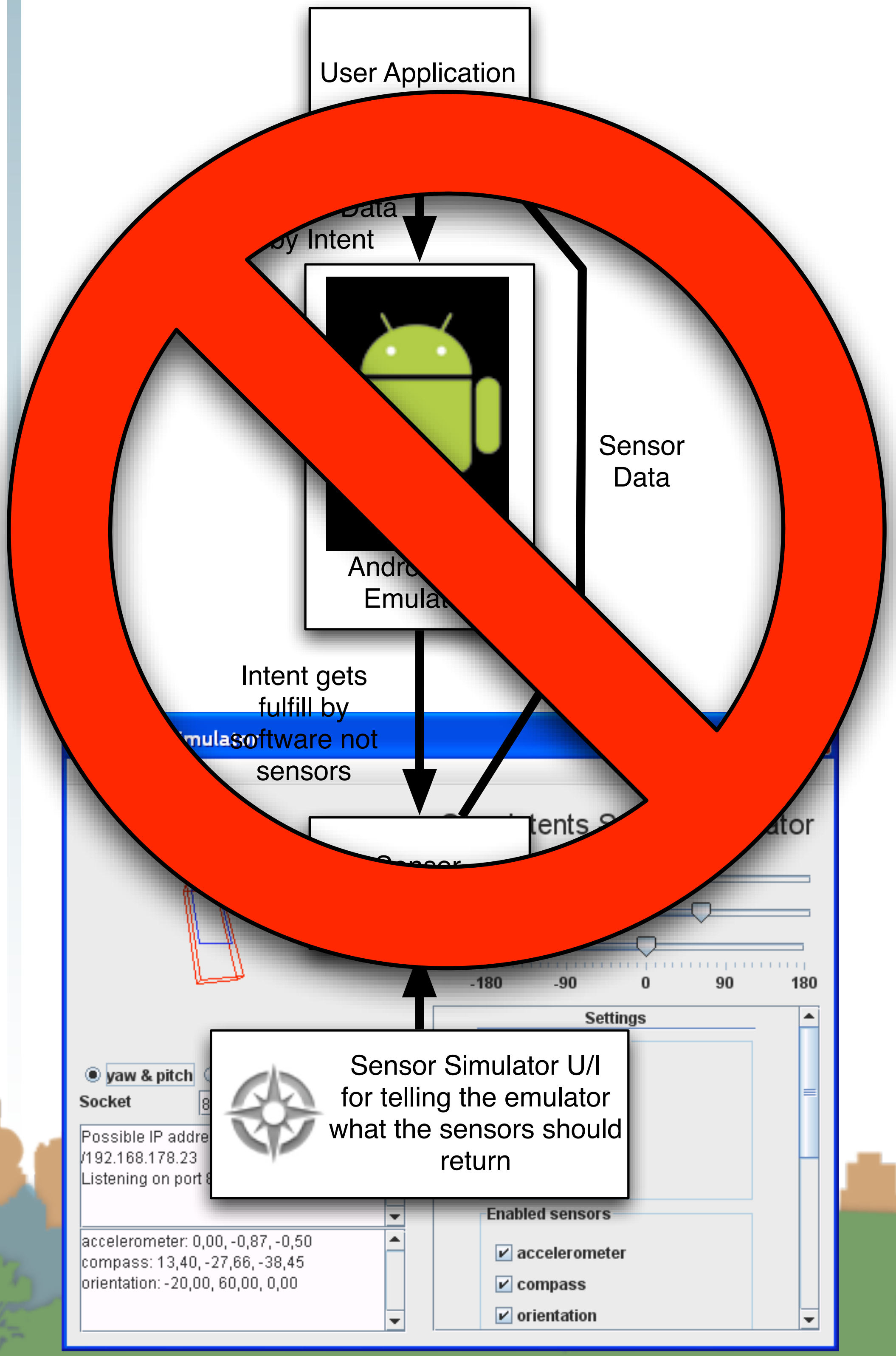


Possible IP addre
/192.168.178.23
Listening on port

accelerometer: 0,00, -0,87, -0,50
compass: 13,40, -27,66, -38,45
orientation: -20,00, 60,00, 0,00

Intro to Android:

- Simulating Sensors on the Emulator would require:

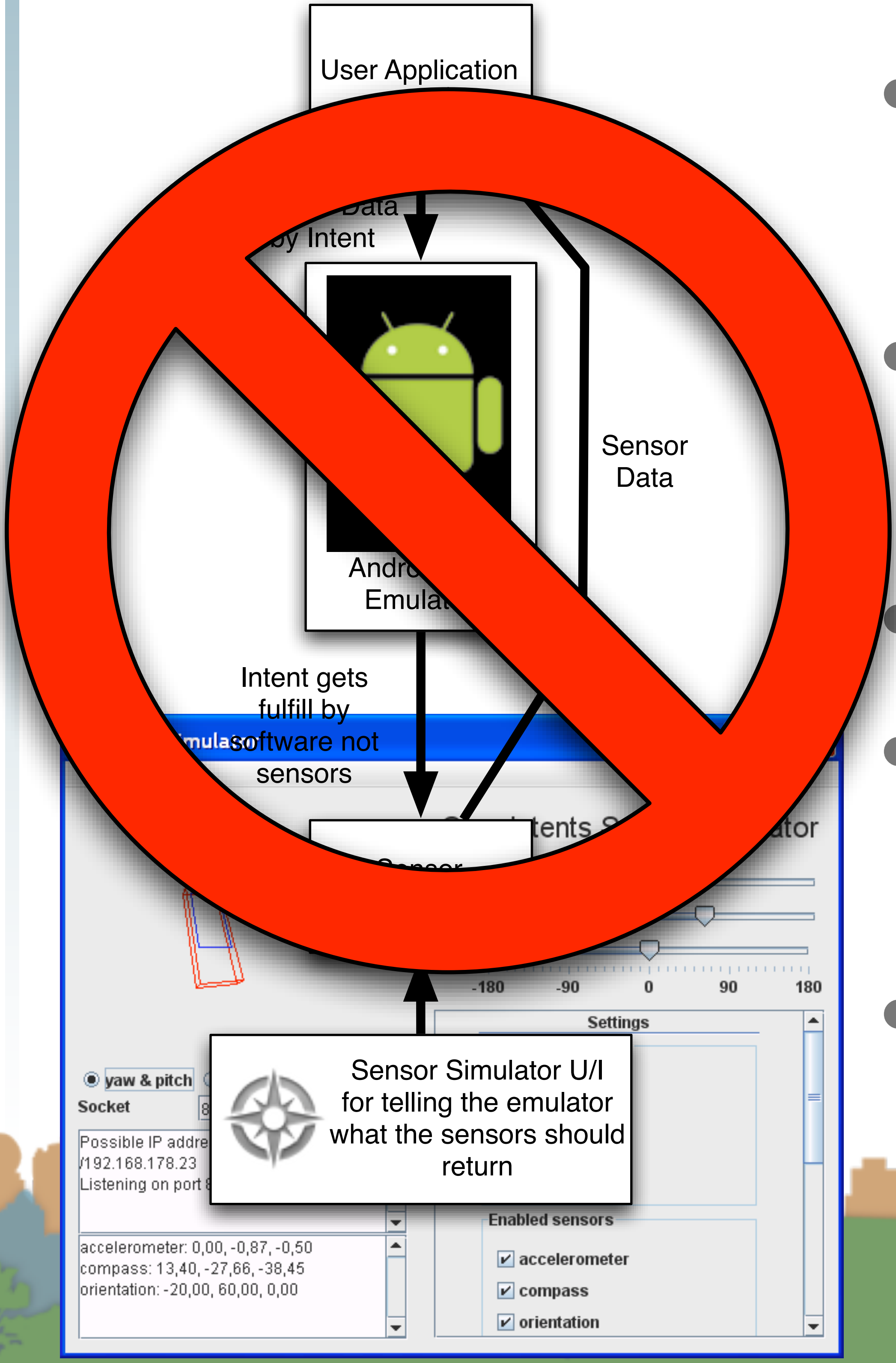


Possible IP addre
/192.168.178.23
Listening on port 8

accelerometer: 0,00, -0,87, -0,50
compass: 13,40, -27,66, -38,45
orientation: -20,00, 60,00, 0,00

Intro to Android:

- Simulating Sensors on the Emulator would require:



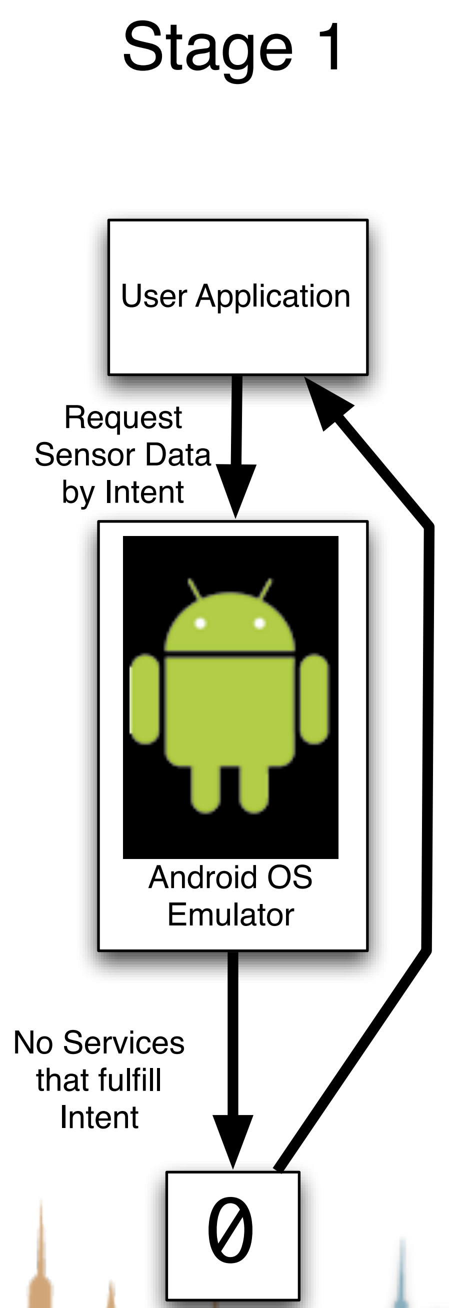
Possible IP addre
/192.168.178.23
Listening on port 8

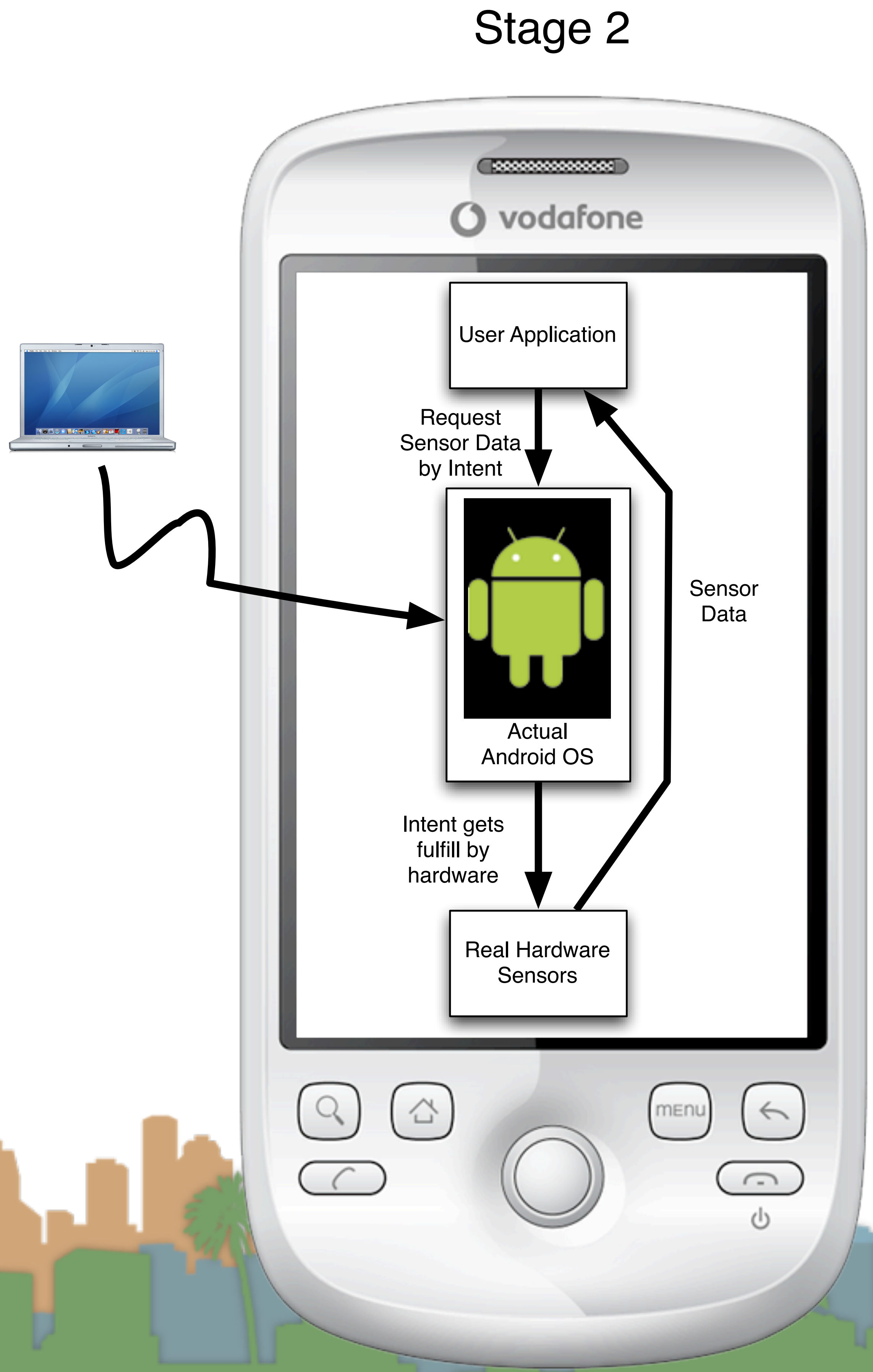
accelerometer: 0,00, -0,87, -0,50
compass: 13,40, -27,66, -38,45
orientation: -20,00, 60,00, 0,00

- This used to exist but has fallen out of currency with Android SDK
- Known package at OpenIntents only works with pre 2.0 SDK
- No known work around
- Instead we must develop on live devices for sensors
- Ok for accelerometers, not okay for GPS

Intro to Android:

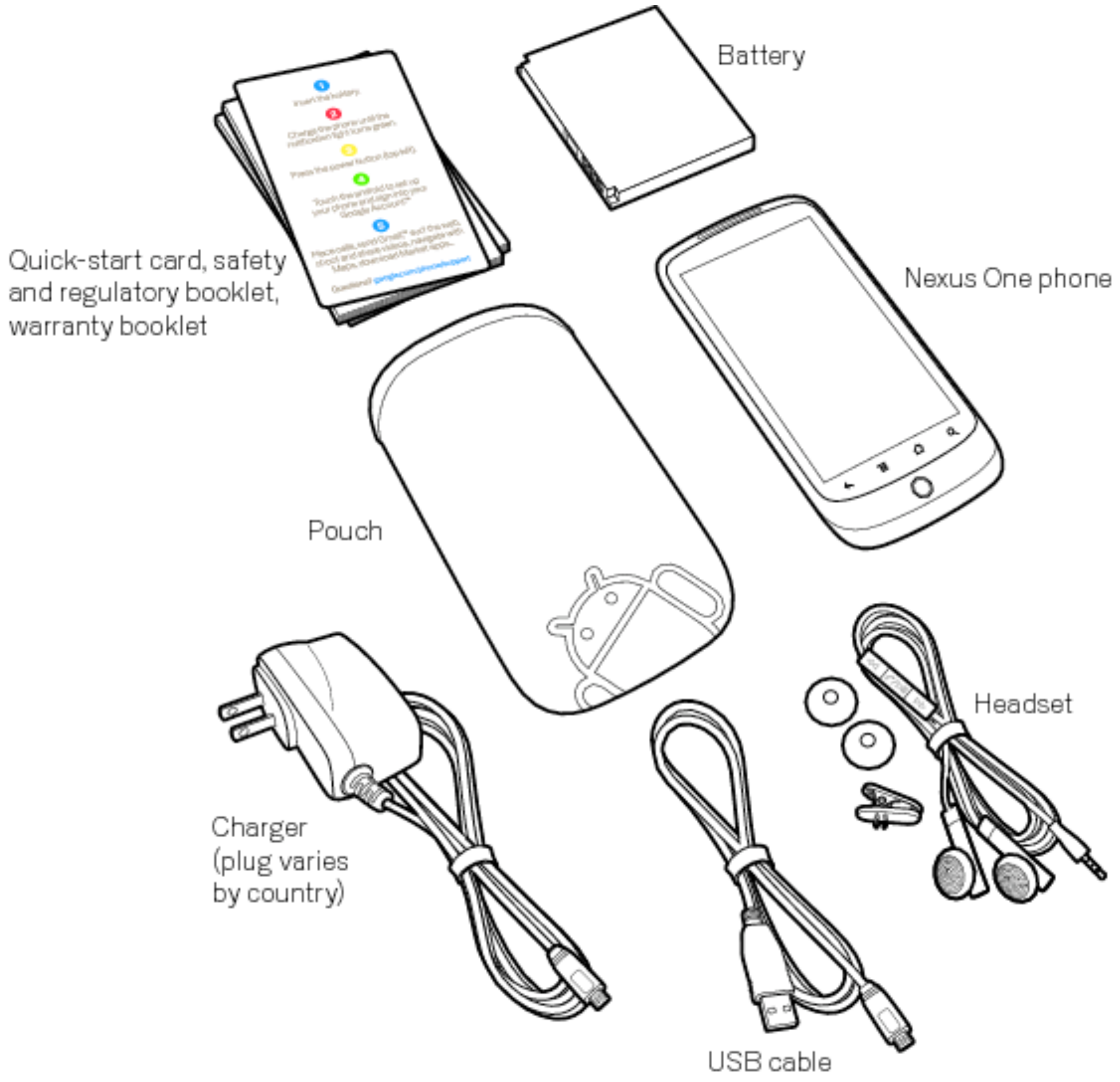
- Your approach





Intro to Android:

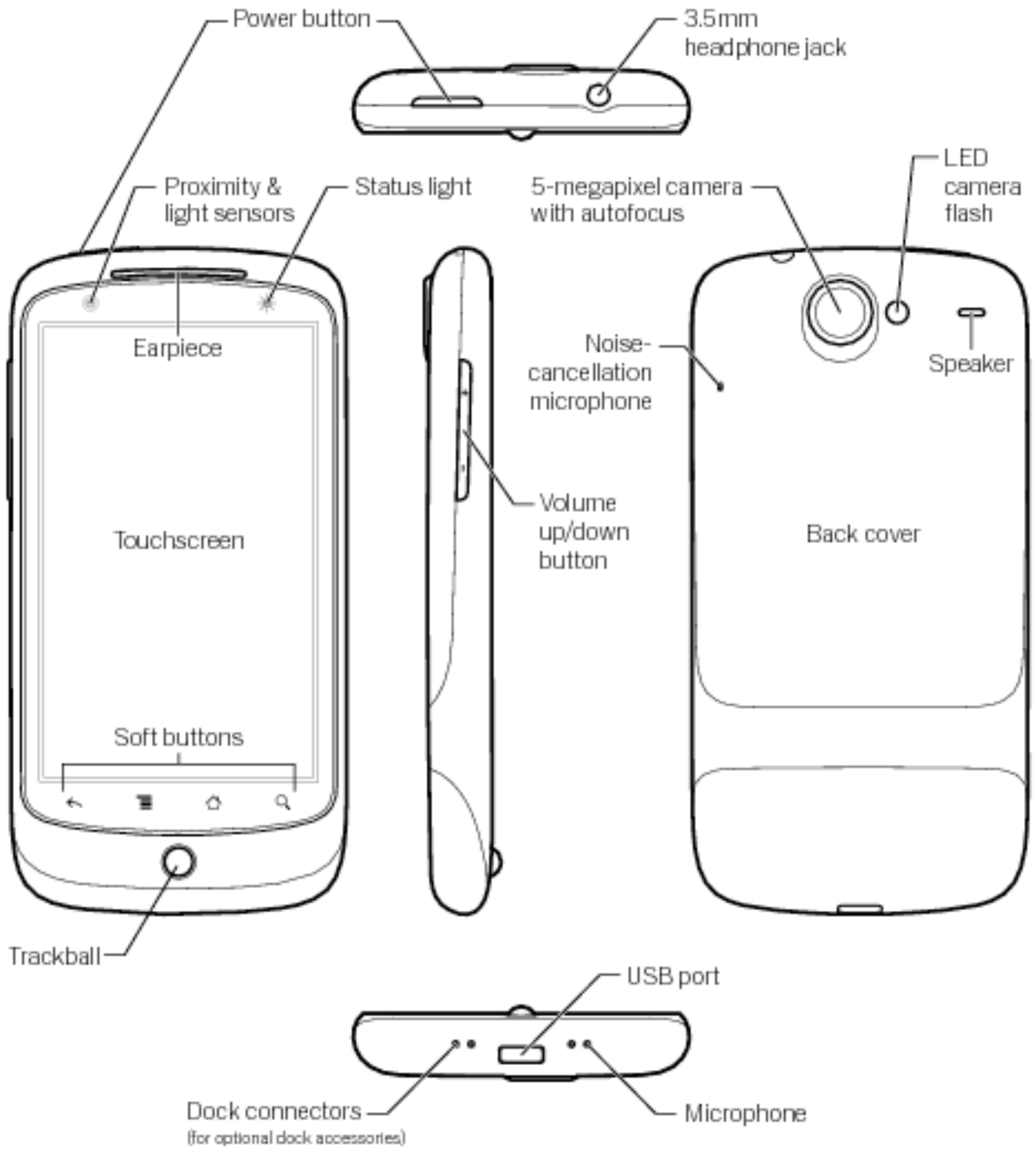
- Unpack the phone


Battery
Quick-start card, safety and regulatory booklet, warranty booklet
Nexus One phone
Pouch
Headset
Charger (plug varies by country)
USB cable

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

Intro to Android:

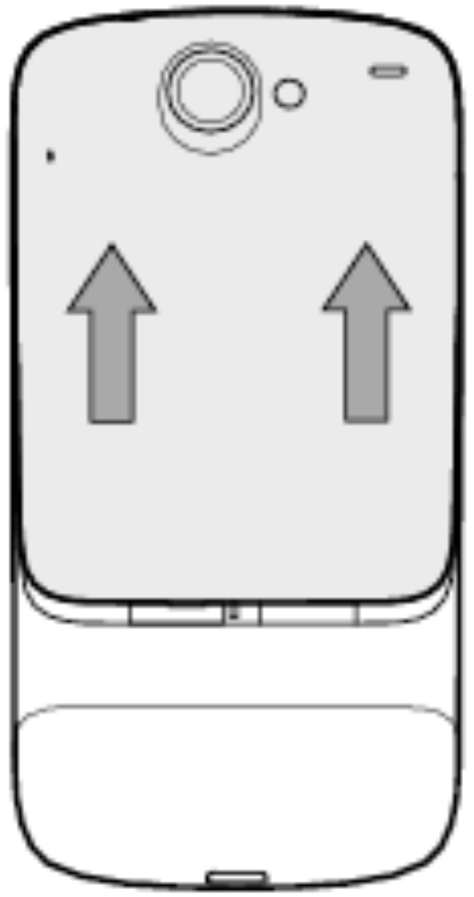
- Take a look at the sensors

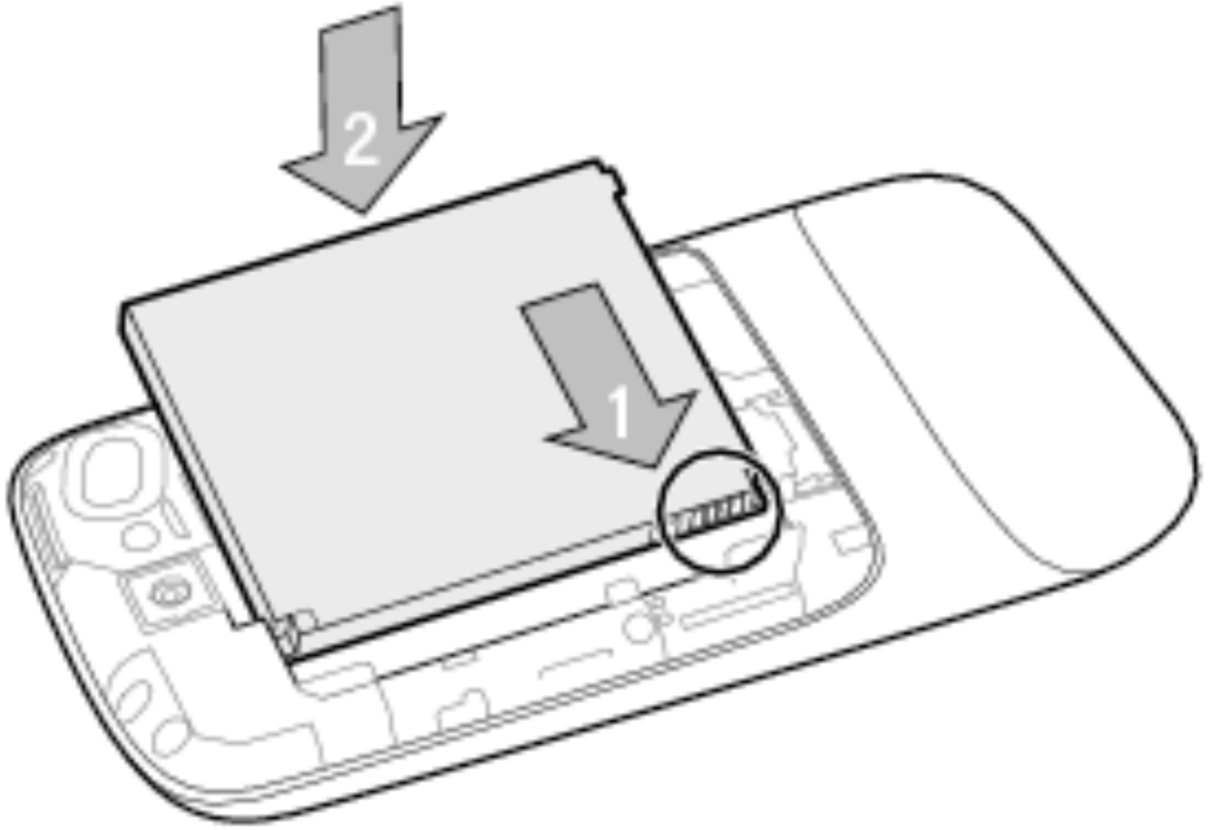


http://www.google.com/support/android/bin/topic.py?hl=en&topic=28930

Intro to Android:

- Install the battery - Do Not Damage My Phones!





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

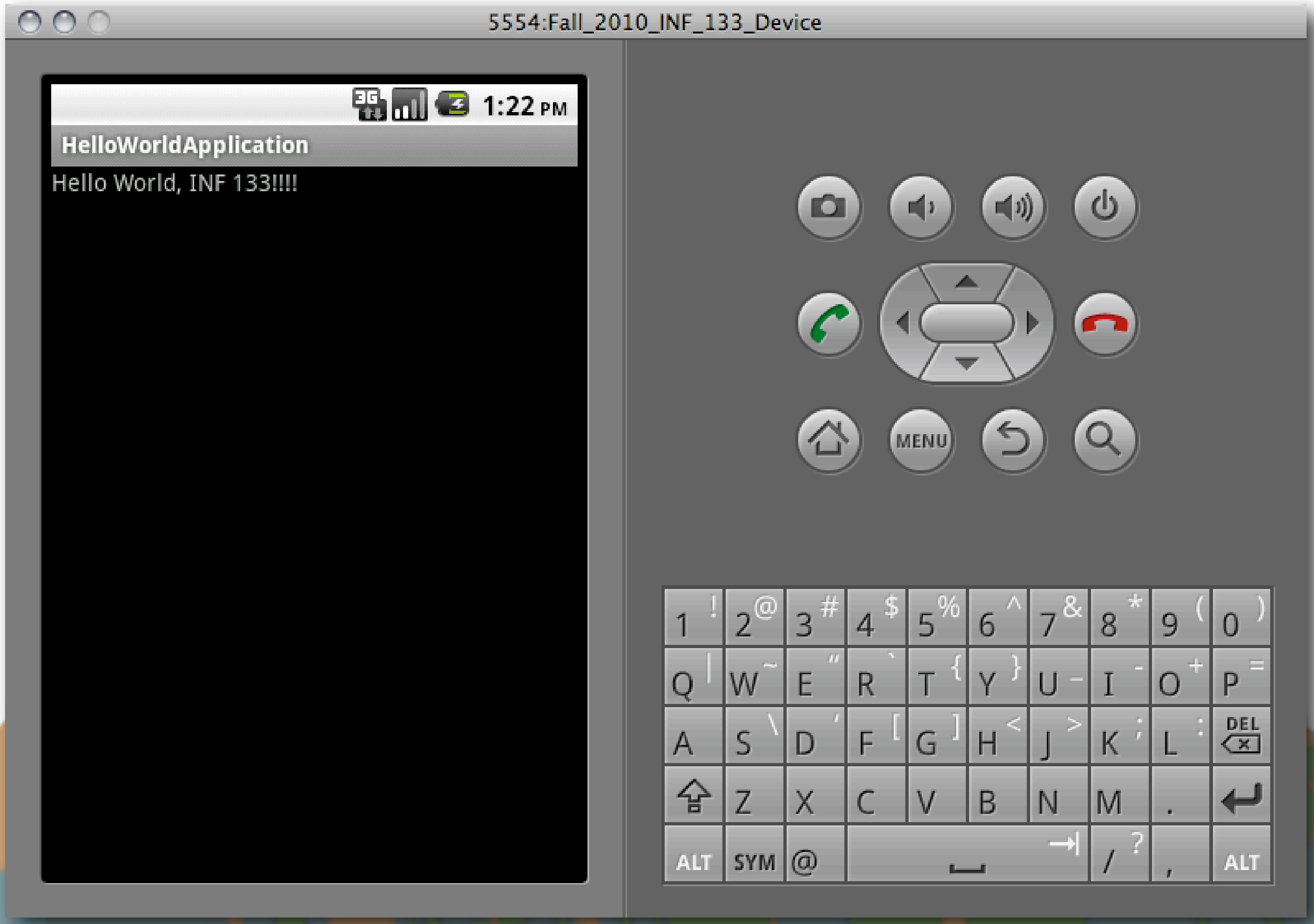
Intro to Android:

- Charge the phone to 100%
  - USB to computer
  - USB to wall plug
- While charging, go through on-phone tutorial
- Do not sync to your Google
- Enable Location reporting
- Set Date and Time
- Register your device with OIT (or send me the MAC address)
  - Home -> Menu -> Settings -> About Phone -> Status

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

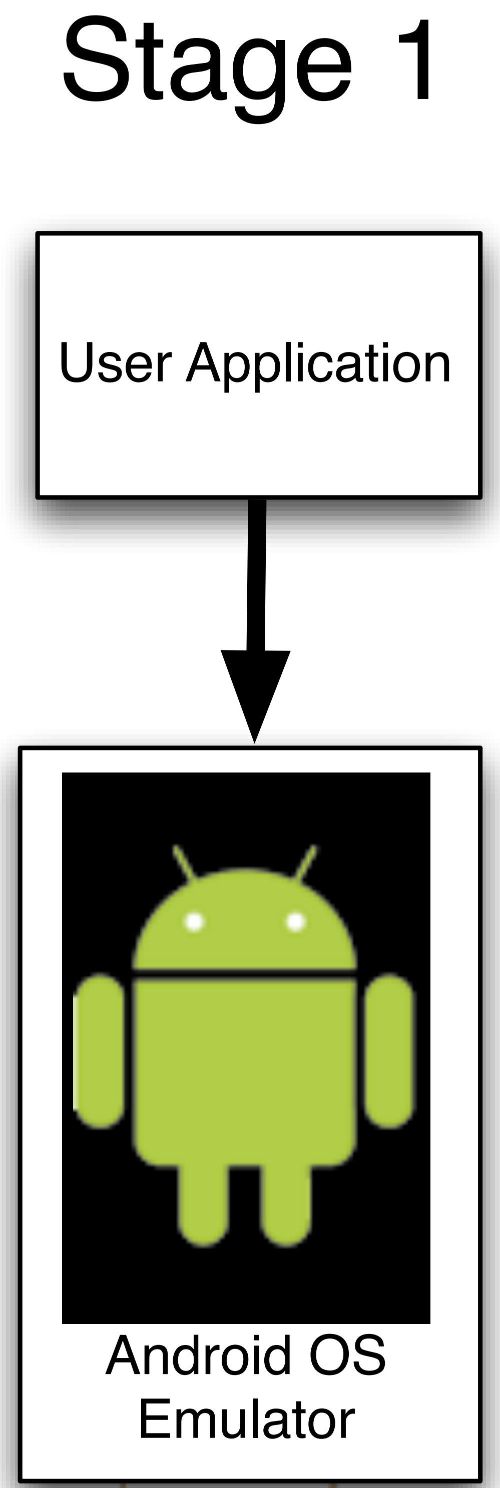
Intro to Android:

- Stage 1
  - Get a Hello World program running in Eclipse
  - Execute it on an emulated phone



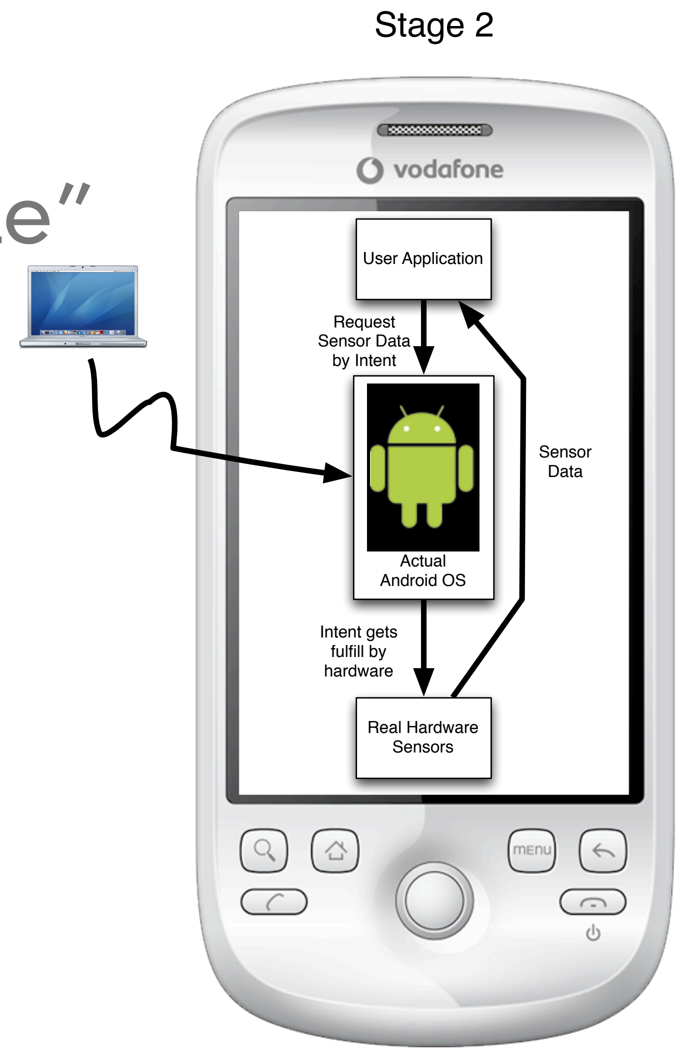
HelloWorldApplication

Hello World, INF 133!!!!



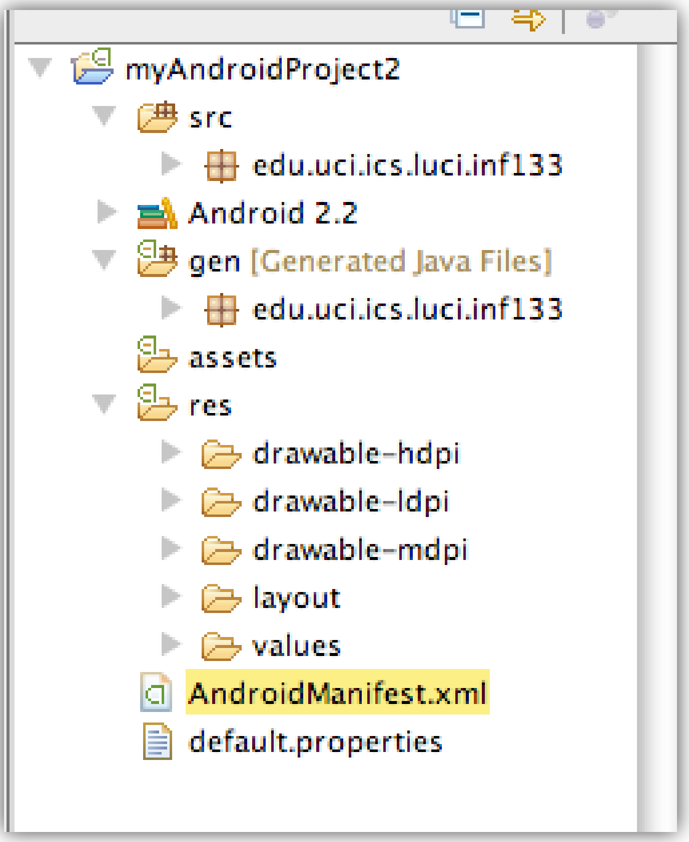
Intro to Android:

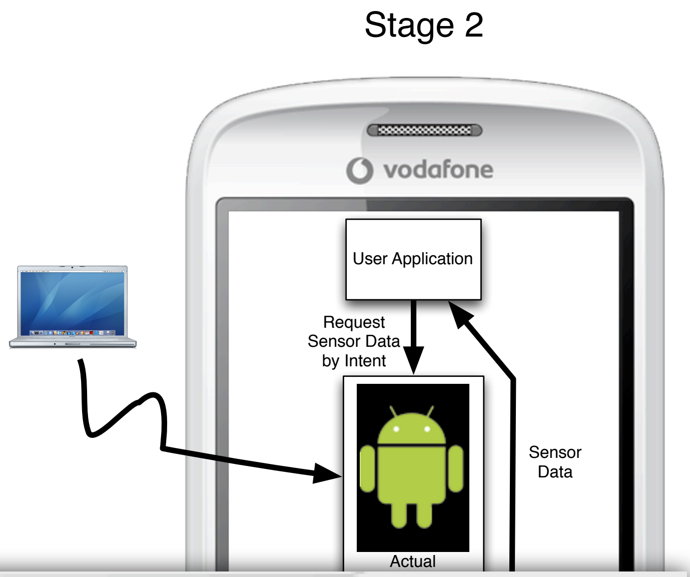
- Stage 2
  - Get a Hello World program running in Eclipse
  - Execute it on a real phone
    - Identify your application as “debuggable”

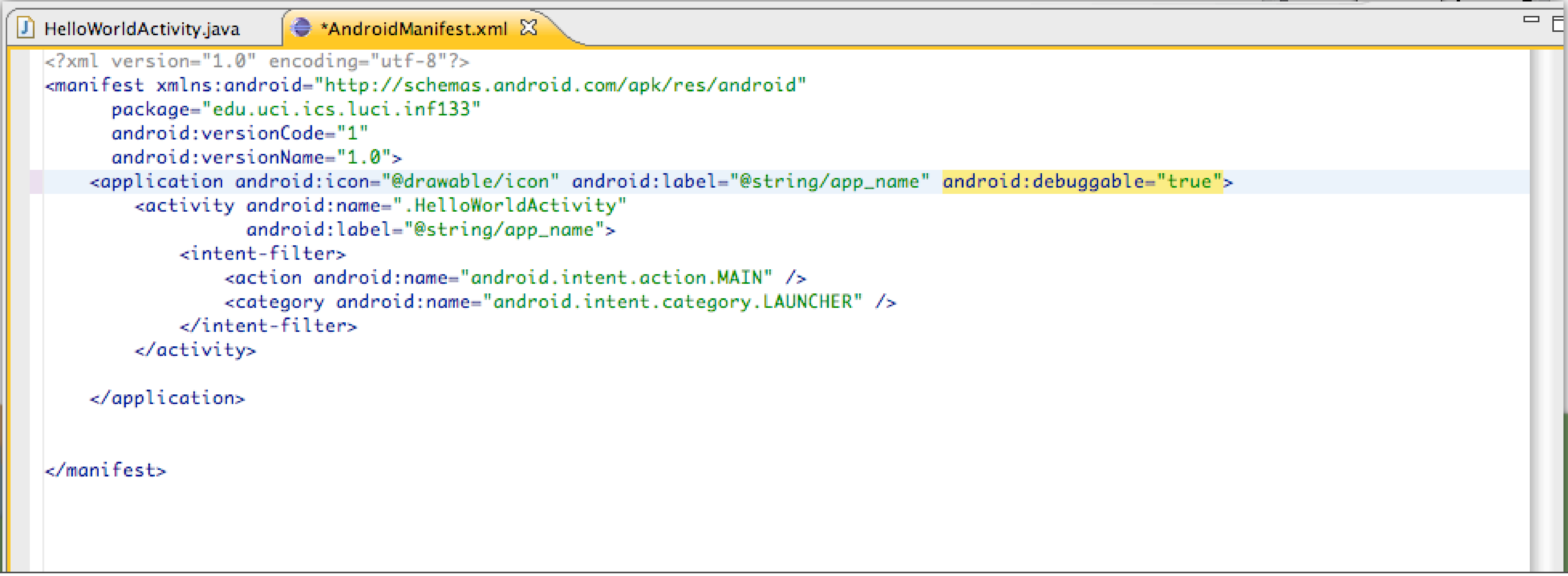


Intro to Android:

- Stage 2



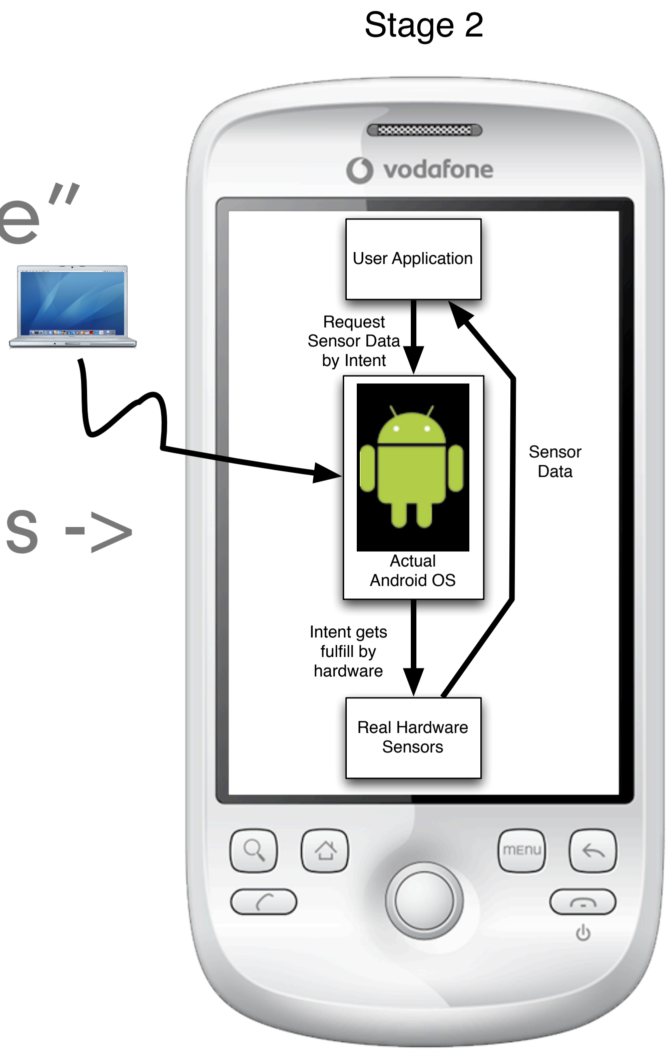




```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="edu.uci.ics.luci.inf133"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name" android:debuggable="true">
        <activity android:name=".HelloWorldActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>


</manifest>
```

Intro to Android:

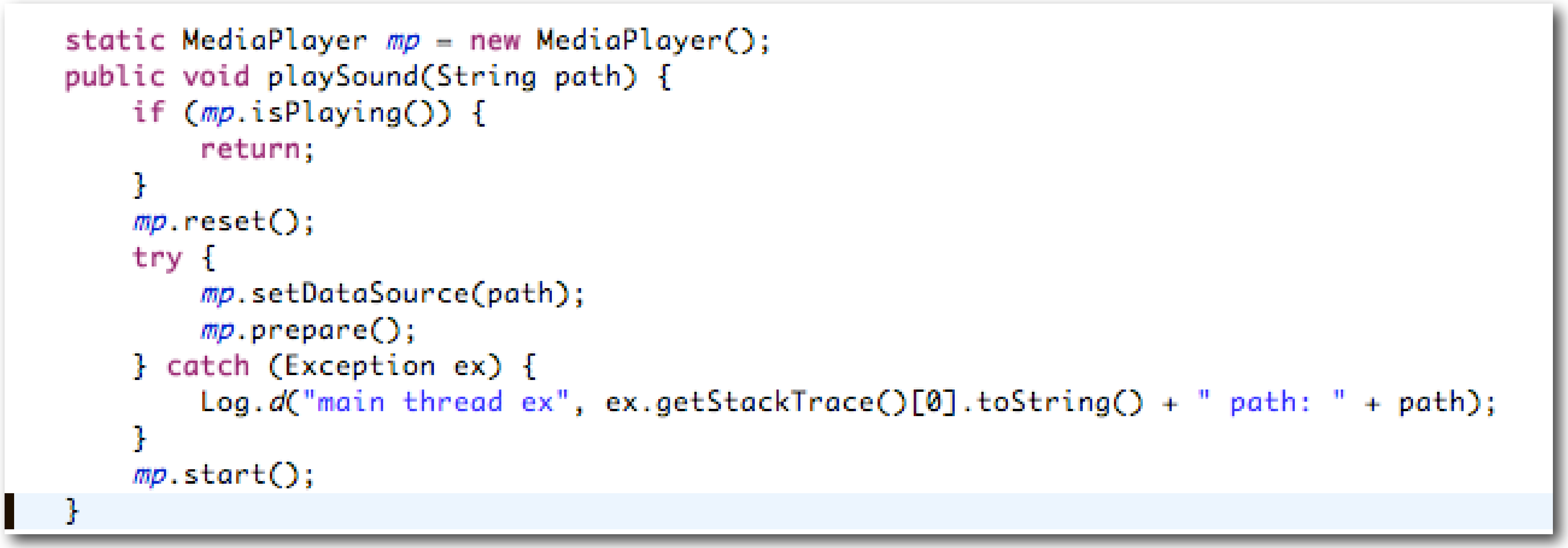
- Stage 2
  - Get a Hello World program running in Eclipse
  - Execute it on a real phone
    - Identify your application as “debuggable”
  - Turn on USB Debugging on the phone
    - Home -> Menu -> Settings -> Applications -> Development -> USB Debugging
  - Windows only: Download/install a driver
    - http://developer.android.com/sdk/win-usb.html
  - Run from Eclipse



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

Intro to Android:

- 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();
}
```

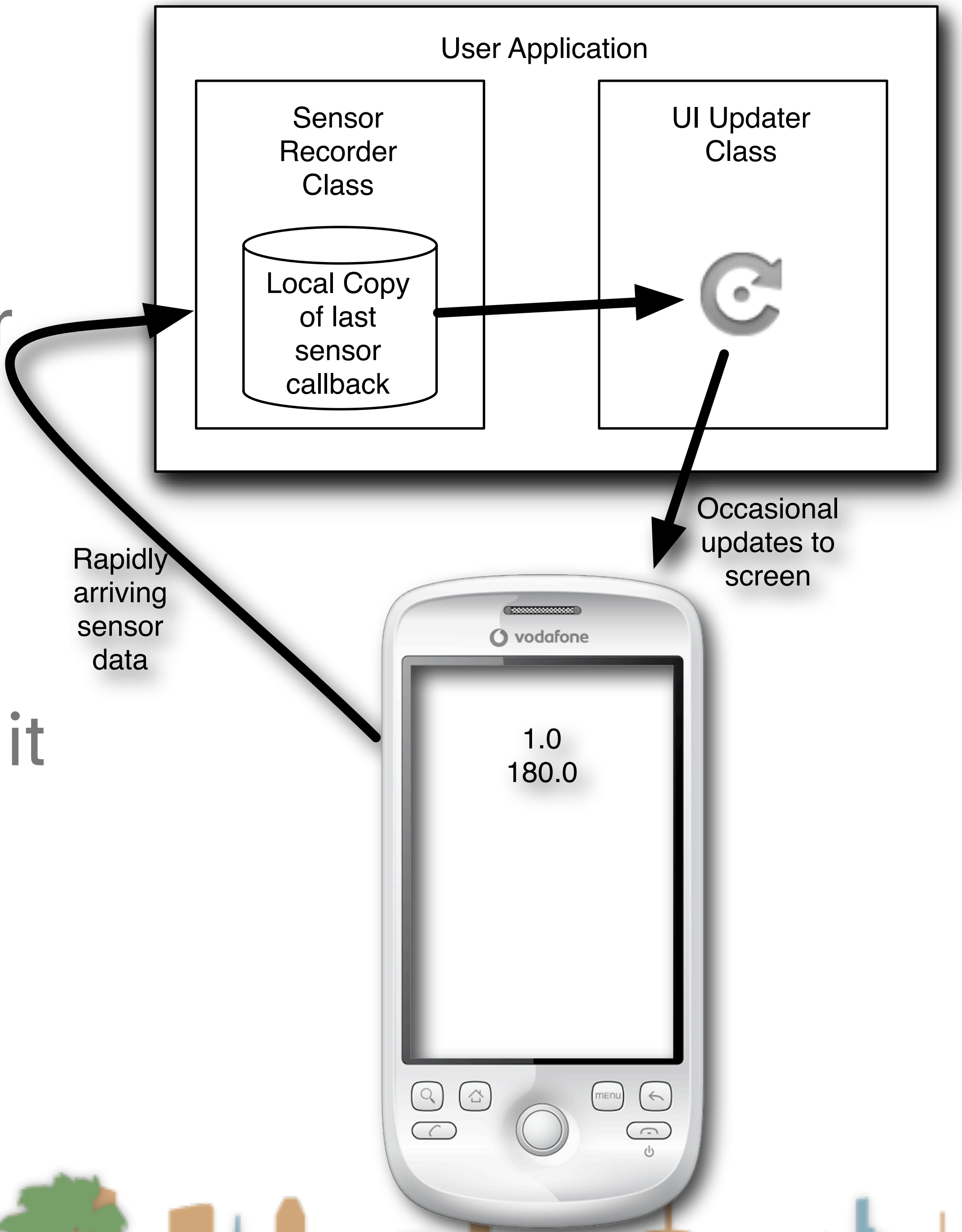
  - http://developer.android.com/guide/topics/media/index.html

Intro to Android:

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

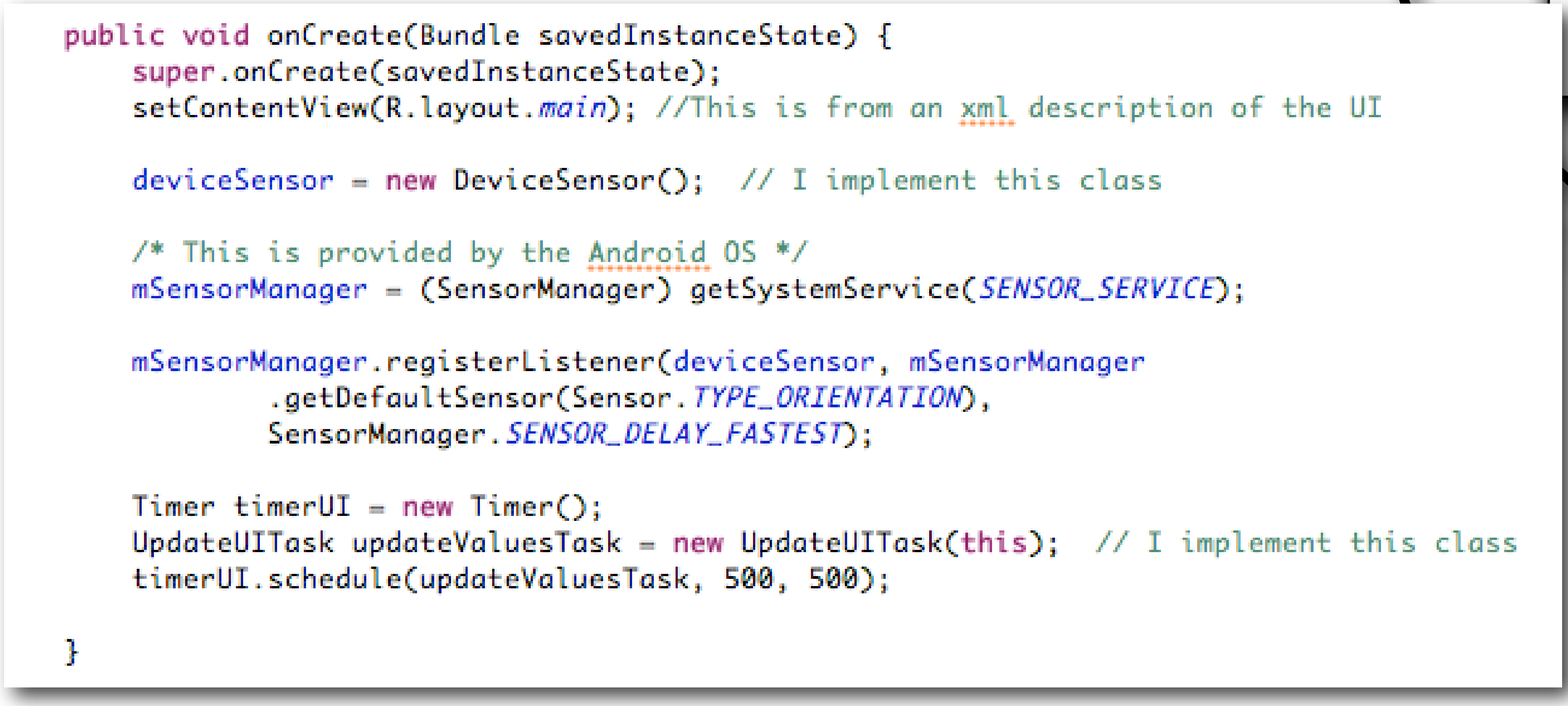
Intro to Android:

- One possible architecture for getting sensor readings
- Steps
  - Create a U/I for the data
  - Instantiate your Sensor Recorder
  - Register for sensor callbacks
  - Instantiate your UI Updater
    - Have a timer occasionally run it

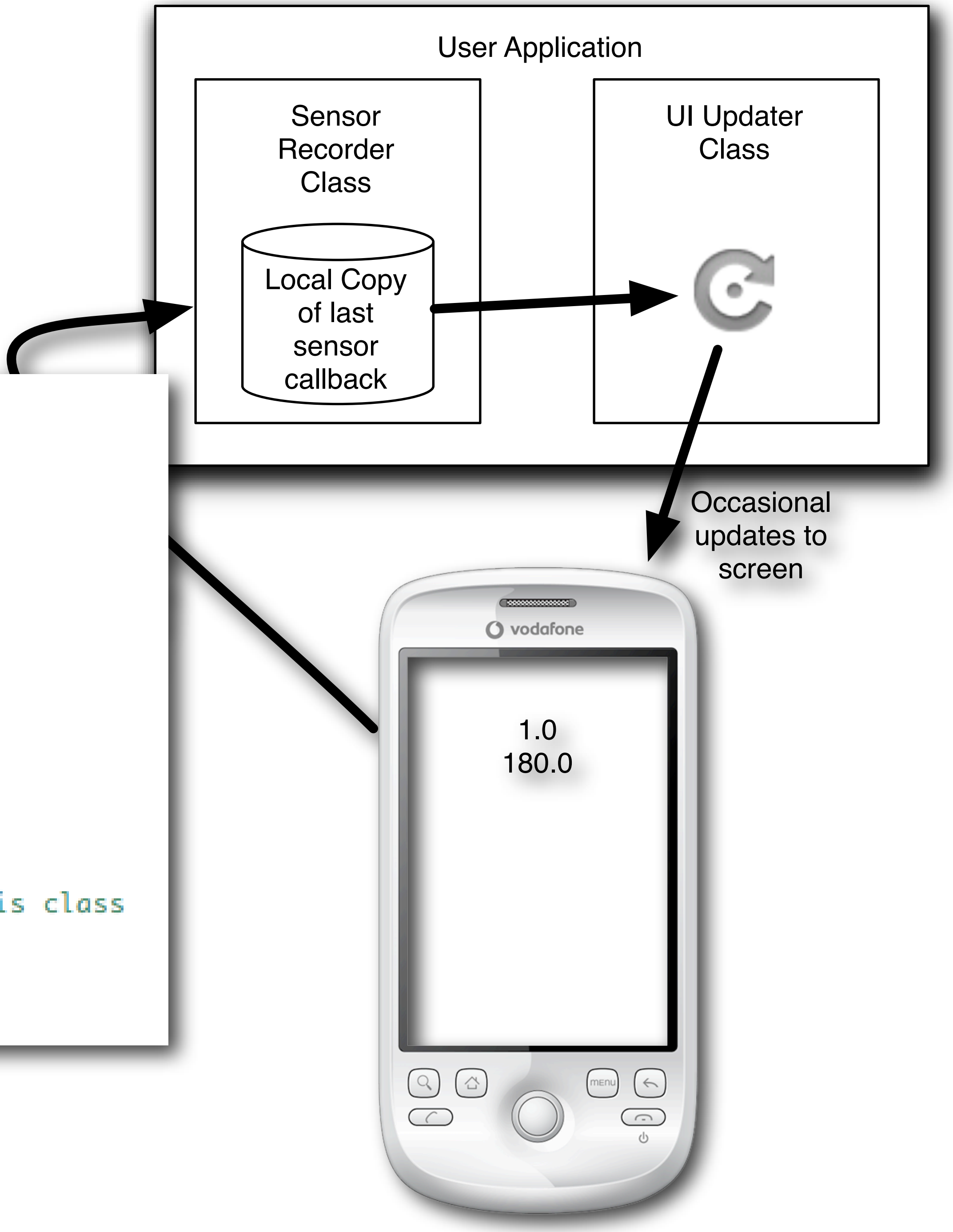

User Application
Sensor Recorder Class
Local Copy of last sensor callback
UI Updater Class
Occasional updates to screen
Rapidly arriving sensor data
vodafone
1.0
180.0

Intro to Android:

- Hints



```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main); //This is from an xml description of the UI

    deviceSensor = new DeviceSensor();  // I implement this class

    /* This is provided by the Android OS */
    mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);

    mSensorManager.registerListener(deviceSensor, mSensorManager
            .getDefaultSensor(Sensor.TYPE_ORIENTATION),
            SensorManager.SENSOR_DELAY_FASTEST);

    Timer timerUI = new Timer();
    UpdateUITask updateValuesTask = new UpdateUITask(this);  // I implement this class
    timerUI.schedule(updateValuesTask, 500, 500);

}
```



LUCI