

# Mobile and Sensor Systems

## Lecture 8: Practical: Mobile Phone Programming

Neal Lathia, Cecilia Mascolo, and Kiran Rachuri

# Android Programming

- Android Java
- Basic components
  - Activities
  - Services
  - Sensors
  - Alarms
  - Threads
  - ...
- More details at: <http://developer.android.com/index.html>



# EmotionSense Sensor Library

- Android sensor programming
  - Each sensor API has different methods
- EmotionSense Sensor Library (ES2 Library)
  - Provides an easy way to access sensor data
  - Supports a large number of sensors
  - One-off and continuous sensing models
  - Includes energy efficient sensing modules



# Requirements

- If you have an Android Phone, please bring it to the lecture
- Android Phone + USB to micro-USB cable
- Eclipse  
<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/junosr2>
- Android SDK <http://developer.android.com/sdk/index.html>
- ADT plugin for Eclipse  
<http://developer.android.com/tools/sdk/eclipse-adt.html>
- ES2 Demo Project [http://www.cl.cam.ac.uk/~kkr27/es2\\_demo\\_project.zip](http://www.cl.cam.ac.uk/~kkr27/es2_demo_project.zip)  
[Import this to your eclipse]

## Documentation:

- <http://developer.android.com/training/basics/firstapp/index.html>
- <https://docs.google.com/document/d/1TqThJULb-4e6TGb1gdkAaPCfyuXStjJpbnt7a0OZ9OE>
- <http://emotionsense.org/>

# Example - 1

```
// Get the instance of the sensor
manager
ESSensorManager sm =
ESSensorManager.getSensorManager
(context);
// Capture data from the microphone
sensor
MicrophoneData md = (MicrophoneData)
esSensorManager.getDataFromSensor
(SensorUtils.SENSOR_TYPE_MICROPHONE);
```

For Accelerometer, use:

```
SensorUtils.SENSOR_TYPE_ACCELEROMETER
```

# Example - 2

```
// Capture data continuously from the microphone
sensor

int subscriptionId = esSensorManager.
subscribeToSensorData
(SensorUtils.SENSOR_TYPE_MICROPHONE, listener);

// Implement the listener
class Listener implements SensorDataListener
{
    public void onDataSensed(SensorData sensorData)
    {
        .....
    }
}
```

# Programming exercise 1

Detect whether the user is moving or stationary

- Implement an activity and add a button to the UI
- On clicking the button, the program should detect whether the user is moving or not

# Programming exercise 1 (contd..)

## Detect whether the user is moving or stationary

- Classification technique:
  - Capture data from the accelerometer sensor for 4 seconds  
(Use `sensorManager.setSensorConfig()` to set `SENSE_WINDOW_LENGTH_MILLIS` to 4 seconds)
  - Calculate the magnitude of acceleration for each of the vectors;  
 $m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$
  - Calculate the Std. Deviation of the magnitudes
  - If (Std. Deviation > threshold) then the user is moving



# Programming exercise 2

## Detect the presence of environmental noise

- Capture data for about 3 seconds from the Microphone sensor
- Calculate the average of the amplitude values
- If (average > threshold) then status = 'noise'

# Programming exercise 3

Detect the presence of environmental noise on detecting the 'isNear' event on the proximity sensor

- Subscribe to the proximity sensor using `ESSensorManager`
- In the `onDataSensed()` method of the listener, check if `isNear()` is true and then invoke the noise detection procedure.

# Questions

Contact [neal.lathia@cl.cam.ac.uk](mailto:neal.lathia@cl.cam.ac.uk) or  
[kiran.rachuri@cl.cam.ac.uk](mailto:kiran.rachuri@cl.cam.ac.uk)