COMP 90018 Mobile Computing Systems Programming

Tutorial on Android Development

Chu Luo, Eman Bin Khunayn

{chu.luo, eman.bin}@unimelb.edu.au

Welcome!

Outcomes of this tutorial:

1. Know threads in Android
2. Learn to use sensors

UI/Main Thread

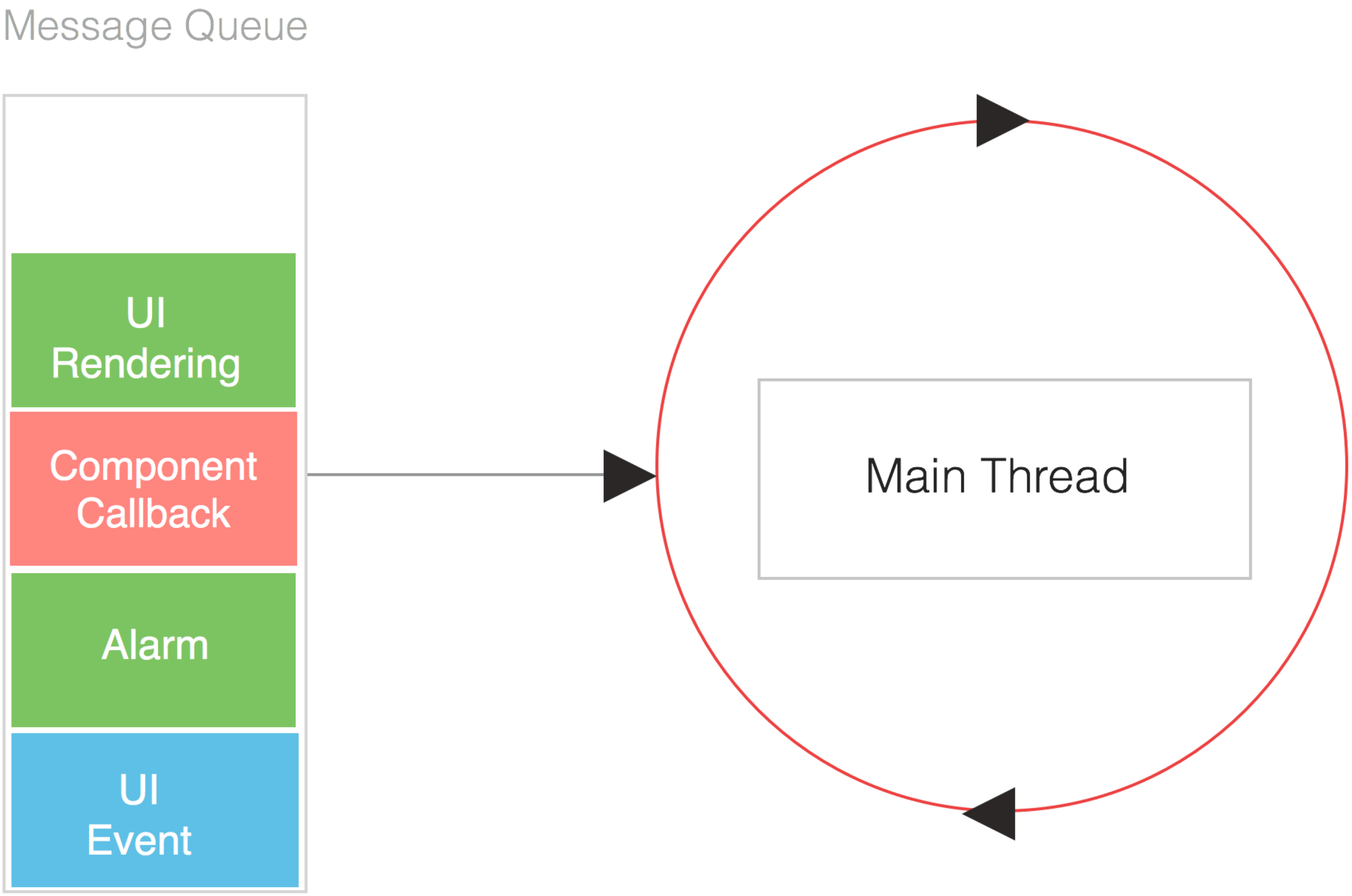
OS creates a thread of execution to launch an app (e.g., activity), called "main." (A.K.A: UI thread)

https://developer.android.com/guide/components/processes-and-threads.html

UI/Main Thread

Even, a Service is running on the main thread.

UI/Main Thread

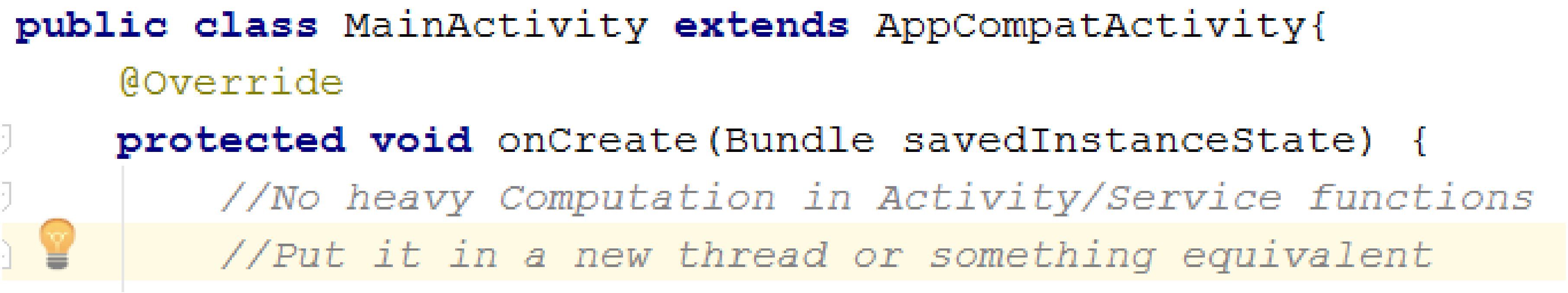


http://hvasconcelos.github.io/images/main_thread_queue.png

Two important principles:

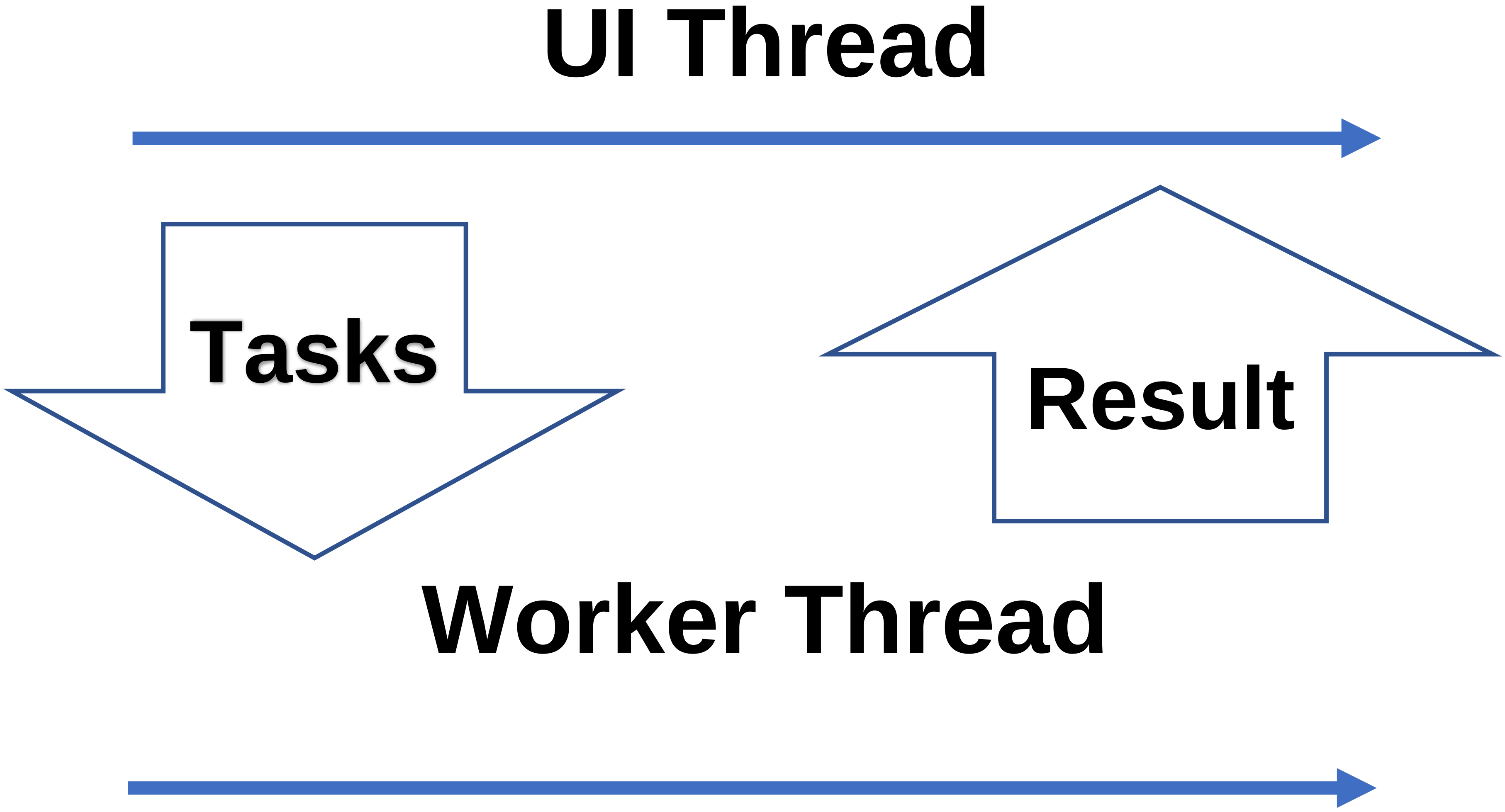
1. Do not block (overburden) the UI thread

2. Do not access UI from outside the UI thread



```java
public class MainActivity extends AppCompatActivity{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        //No heavy Computation in Activity/Service functions
        //Put it in a new thread or something equivalent
```

A better design:



UI Thread

Tasks

Result

Worker Thread

Multithreading: Many ways

1. Thread and Runnable
2. HandlerThread
3. AsyncTask
4. IntentService
5. ThreadPoolExecutor

https://developer.android.com/training/multiple-threads/index.html

Thread and Runnable: 2 ways

1. class PrimeThread extends Thread
2. class PrimeRun implements Runnable

Thread and Runnable

1. .start() to start a thread
2. . Interrupt() to quit
3. Mostly, a thread has a while loop and a temporarily cease execution (sleep time)

Be careful with the thread context

1. If an activity/service creates a thread, the thread will continue to run when the activity/service dies or stops.
2. If you don’t stop the thread now, it shouldn’t use objects of the dead activity/service (Should eventually stop it. Otherwise, horrible)

Thread and Runnable

Demo

Look at the Logcat console

AsyncTask and IntentService

For offloading tasks from UI thread to a worker thread (they will create this).

You don't need to manipulate a new thread for them.

ThreadPoolExecutor

ThreadPoolExecutor can manage a number of concurrent tasks and a number of threads.

https://developer.android.com/reference/java/util/concurrent/ThreadPoolExecutor.html

HandlerThread

For more advanced multithreading with Handlers:

(1) to schedule messages and runnables to be executed;

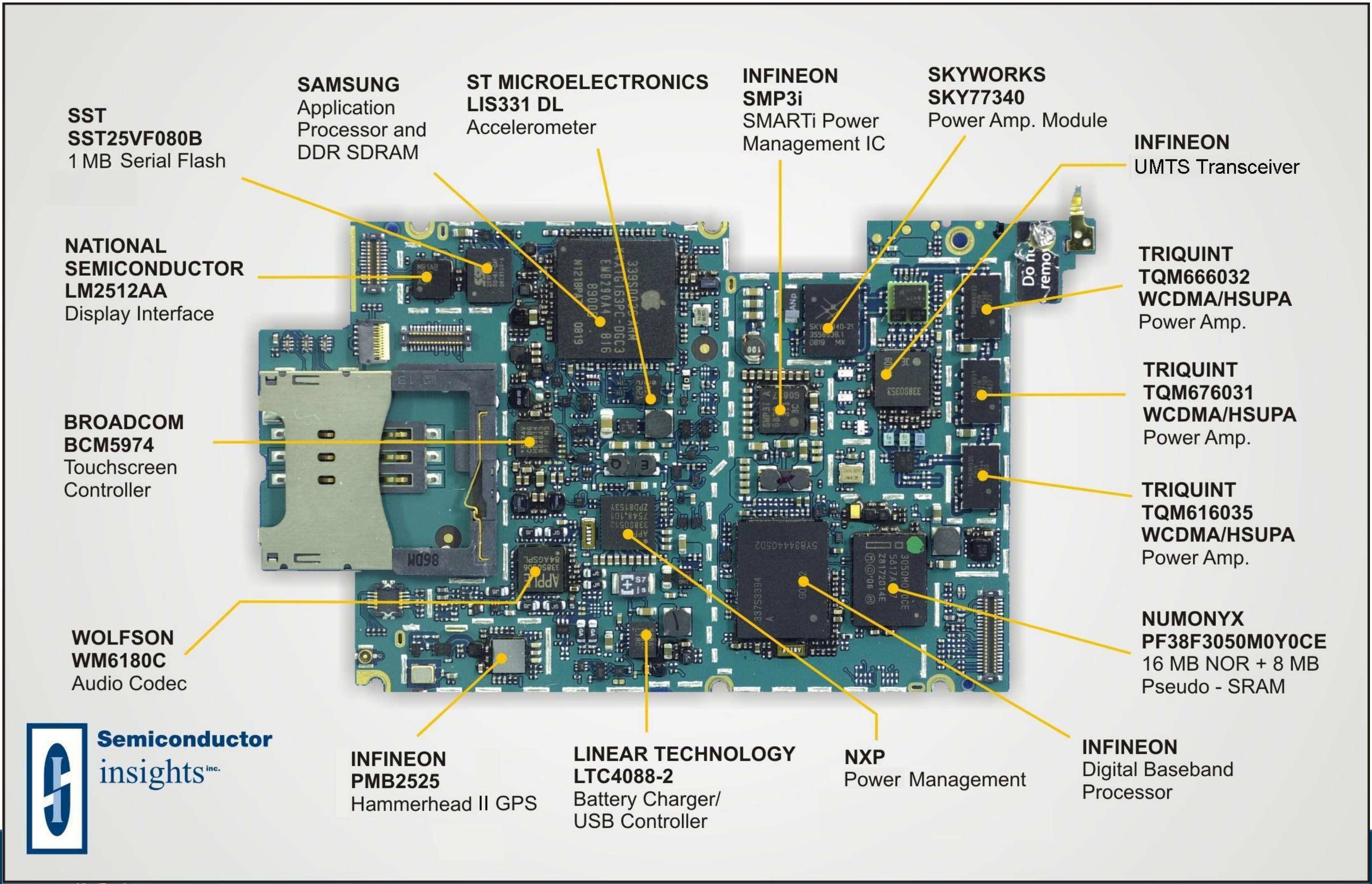
(2) to enqueue an action to be performed on this new thread.

HandlerThread

A typical usage scenario is sensor data collection.

So let’s turn to sensors now.

Phone Sensors



SST
SST25VF080B
1 MB Serial Flash

SAMSUNG
Application Processor and DDR SDRAM

ST MICROELECTRONICS
LIS331 DL
Accelerometer

INFINEON
SMP3i
SMARTi Power Management IC

SKYWORKS
SKY77340
Power Amp. Module

INFINEON
UMTS Transceiver

NATIONAL SEMICONDUCTOR
LM2512AA
Display Interface

BROADCOM
BCM5974
Touchscreen Controller

WOLFSON
WM6180C
Audio Codec

TRIQUINT
TQM666032
WCDMA/HSUPA
Power Amp.

TRIQUINT
TQM676031
WCDMA/HSUPA
Power Amp.

TRIQUINT
TQM616035
WCDMA/HSUPA
Power Amp.

NUMONYX
PF38F3050M0Y0CE
16 MB NOR + 8 MB Pseudo - SRAM

INFINEON
PMB2525
Hammerhead II GPS

LINEAR TECHNOLOGY
LTC4088-2
Battery Charger/
USB Controller

NXP
Power Management

INFINEON
Digital Baseband Processor

Semiconductor insights inc.

Android Sensors

- **Motion sensors**

acceleration forces and rotational forces: accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.

- **Environmental sensors**

environmental parameters: ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers.

- **Position sensors**

physical position of a device: orientation sensors and magnetometers.

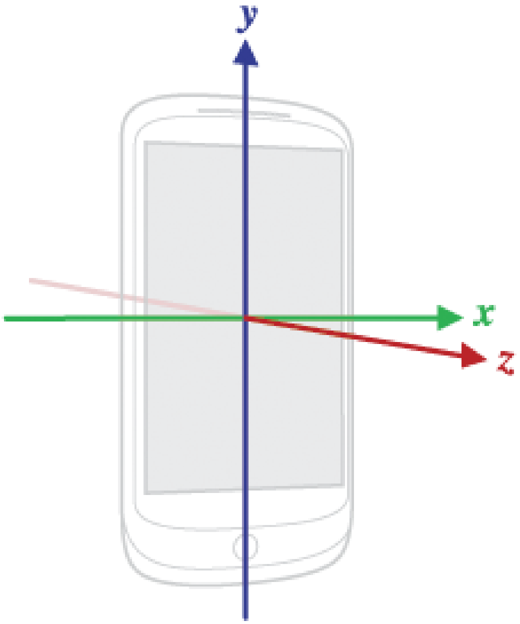
Android Location Service

- Android puts all location sensors in a service API. (So it is easy to use)

- Link: https://developer.android.com/guide/topics/location/index.html

List of sensors for this lab

- Accelerometer (movement, rate of change of velocity) (as a normal sensor)
- Location (as a special API)



Motion sensors

An **acceleration** sensor measures the acceleration applied to the device, WITHOUT the force of gravity.

Accelerometer

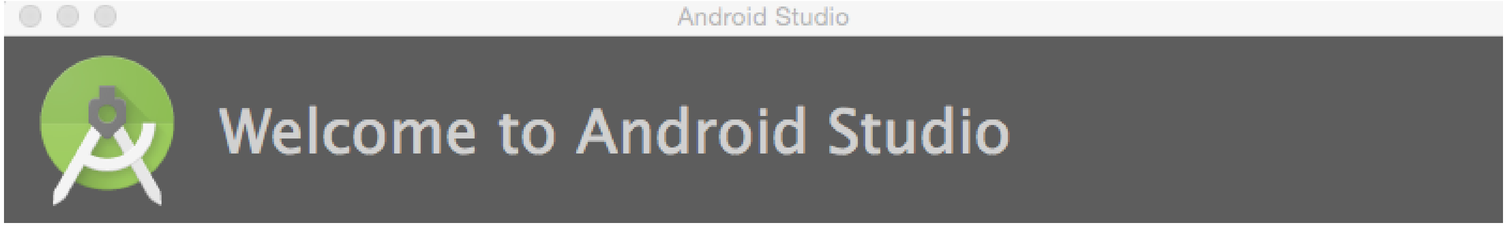
The phone is in the air, dropping, with gravity. The accelerometer reading is (0, 0, 0) m/s^2, if we don't consider measurement error. (although physical acceleration is (0,0,G))

The phone is on the table, with the force given by the table and gravity. The accelerometer reading is (0, 0, G) m/s^2, if no measurement error. (G≈9.8 on our planet)

STARBUCKS
STARBUCKS COFFEE
Careful, the beverage you're about to enjoy is extremely hot.
COFFEE

Example

- Create an Android app to collect such data


Android Studio

Welcome to Android Studio

Reference

- https://developer.android.com/reference/android/hardware/SensorManager.html
- https://developer.android.com/reference/android/hardware/SensorEventListener.html

Collect sensor data

1. Implement the interface SensorEventListener in an Activity or a Service (mostly using Service).
2. Initialise SensorManager
3. Implement onAccuracyChanged() and onSensorChanged() function

Using a HandlerThread

Sensor data collection may happen very often. So It is better to have a new thread to do this.

Demo using a HanderThread

Do computation in another thread

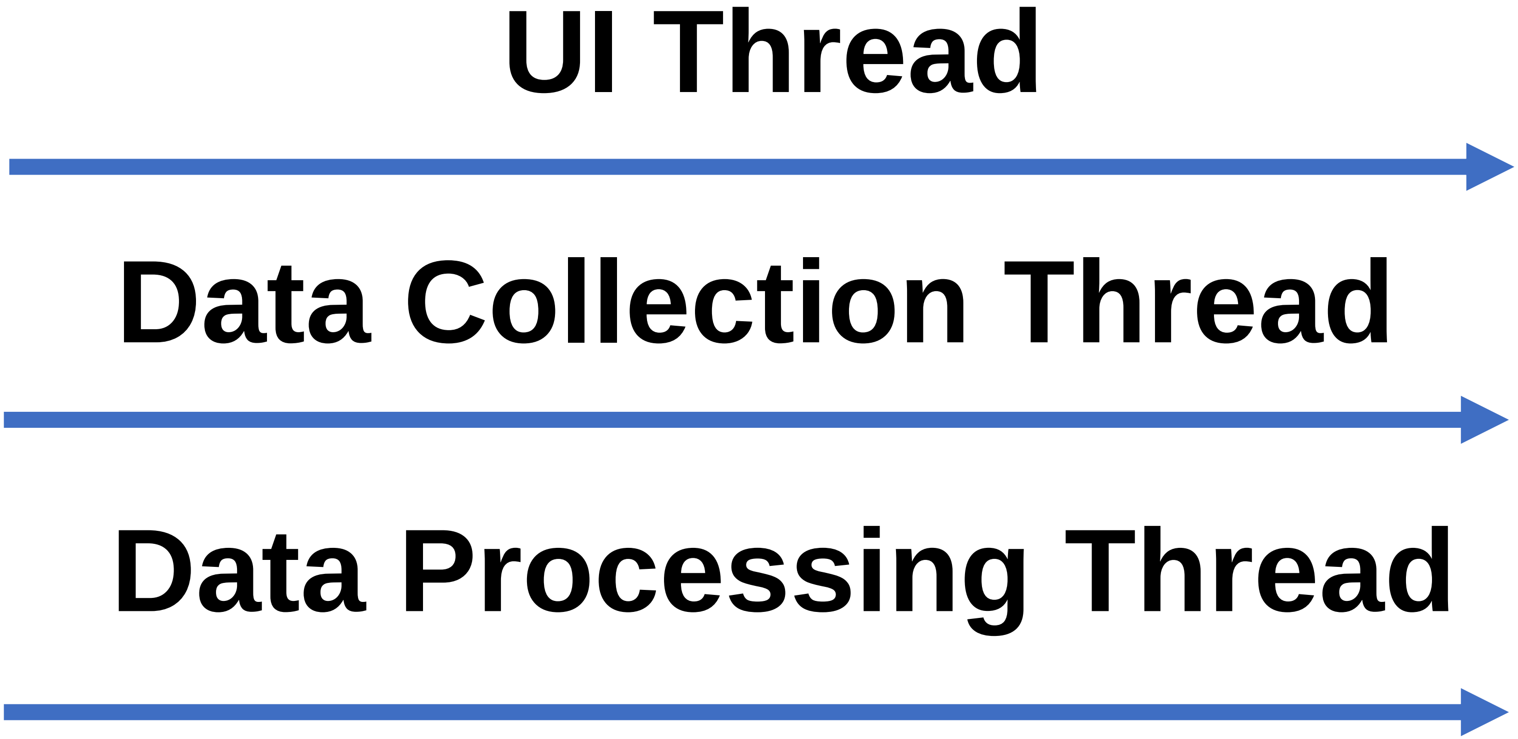
(Your Kalman Filter goes here!)

The data collection HandlerThread only responds to new sensor data.

Computation needs a more stable and manageable thread.

Demo doing computation on a new thread.

Overview of continuous sensing



UI Thread

Data Collection Thread

Data Processing Thread

Location Sensing

This is different from other sensors.

1. You need a LocationManager
2. A Service does the low-level sensing : Context.LOCATION_SERVICE

Permission Check For Location

1. Location Sensing requires Permission from users.

2. API 23 or above needs runtime Permission Check

Demo For Location Sensing

Exercise:

1. Create a Service collecting sensor data.

2. Put data collection in a HandlerThread

3. Do computation in another thread

More learning directions:

1. Ty to process raw data.

2. Design Content Providers to store/read data.

3. To use Azure (e.g., SQL database).

See you next week

COMP 90018

Tutorial on Android Development

Chu Luo, Eman Bin Khunayn

{chu.luo, eman.bin}@unimelb.edu.au