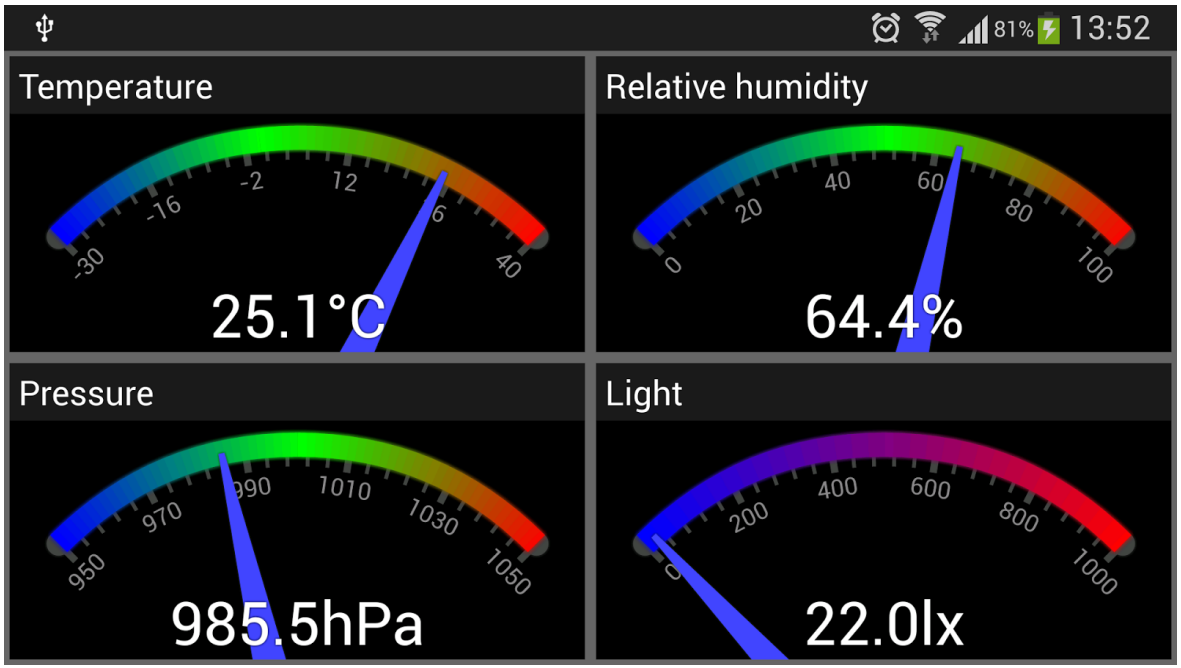
La gestion des capteurs sous Android

Frédéric Guinand

IUT du Havre



Sensors $\leftrightarrow$ capteurs

- il existe trois principales catégories de capteurs sur les machines Android :
  - les capteurs de mouvement (motion)
  - les capteurs d'environnement
  - les capteurs de localisation

- Les capteurs peuvent être accédés au travers du Sensor Framework qui fournit des classes et des interfaces pour :
  - vérifier la disponibilité de certains capteurs sur votre machine cible
  - enregistrer des *listener* aux *Events* produits par les capteurs
  - récupérer les mesures brutes des capteurs et définir la période d'acquisition de ces données

Sensor Framework

- la classe `SensorManager` est utilisée pour la création d'un *Sensor Service*
- la classe `Sensor` pour créer de nouvelles classes pour des capteurs spécifiques
- la classe `SensorEvent`, une instance de cette classe fournit des informations sur les événements produits par les capteurs
- l'interface `SensorEventListener` est utilisée pour l'enregistrer un Objet comme *listener* des *Sensor Events*.
  - `onSensorChanged(SensorEvent se)`
  - `onAccuracyChanged(Sensor s, int accuracy)`

Procédure générale de mise en oeuvre

1. récupérer une référence à SensorManager
2. récupérer une référence au Sensor (le capteur cible)
3. vérifier que ce type de capteur existe sur la machine (simple : la référence du point précédent est non nulle)
4. enregistrer un listener au capteur cible
5. implémenter les méthodes d'interface `onSensorChanged()` et `onAccuracyChanged()`

Lister les capteurs

- nous devons récupérer une référence au Sensor Manager
- nous invoquons la méthode `getSensorList()` avec en paramètre tous les types de capteurs

```
SensorManager sm = (SensorManager)this.getSystemService(SENSOR_SERVICE);
List<Sensor> availableSensors = sm.getSensorList(Sensor.TYPE_ALL);
```

- pour chaque capteur de la liste nous pouvons invoquer différentes méthodes :
  - `getName()`
  - `toString()` (liste l'ensemble des informations sur la capteur)
  - `getMinDelay()` retourne une valeur non nulle si la donnée est produite de manière périodique ou nulle si la valeur est modifiée seulement lorsque la valeur change

Liste de tous les capteurs

K330 3-axis Accelerometer
10000
++++++++++++++
YAS532 Magnetic Sensor
10000
++++++++++++++
K330 Gyroscope sensor
10000
++++++++++++++
Barometer Sensor
66700
++++++++++++++
MAX88920 Proximity Sensor
0

Exemples

Détection de la lumière

- sur certains matériels vous avez un capteur de lumière, il peut apparaître sous le nom RGB Sensor
- c'est un capteur environnemental
- pour l'utiliser vous devez :
  1. obtenir une référence au SensorManager : `sm`
  2. obtenir une référence au capteur :
     ```
     Sensor lightSensor =
     sm.getDefaultSensor(Sensor.TYPE_LIGHT)
     ```
  3. vérifier la disponibilité (ou la présence) du capteur :
     ```
     lightSensor != null
     ```
  4. enregistrer un listener au capteur cible (minDelay vaut 0)
     ```
     sm.registerListener(anObject,lightSensor,
          SensorManager.SENSOR_DELAY_NORMAL) ;
     ```
  5. implémenter les méthodes de l'interface
- obtenir les données depuis le SensorEvent se :

```
float lightInLux = se.values[0] ;
```

détecteur de proximité

- sur certains matériels vous avez un détecteur de proximité, il apparaît le plus souvent sous le nom : proximity sensor
- c'est un capteur de position
- pour l'utiliser vous devez :
  1. obtenir une référence au SensorManager : `sm`
  2. obtenir une référence au capteur :
     ```
     Sensor proximSensor =
     sm.getDefaultSensor(Sensor.TYPE_PROXIMITY)
     ```
  3. vérifier la disponibilité (ou la présence) du capteur :
     ```
     proximSensor != null
     ```
  4. enregistrer un listener au capteur cible (minDelay vaut 0)
     ```
     sm.registerListener(anObject,proximSensor,
          SensorManager.SENSOR_DELAY_NORMAL) ;
     ```
  5. implémenter les méthodes de l'interface
- obtenir les données depuis le SensorEvent se :

```
float proxInCentimeters = se.values[0] ;
```

Accéléromètre

- L'accéléromètre est un capteur qui détecte les changements de vitesse selon 3 axes. Plusieurs capteurs ont la charge de la mesure de l'accelération : l'accéléromètre linéaire et l'accéléromètre 3-axes qui apparaît souvent sous le nom de 3-axis Accelerometer
- c'est un capteur de mouvement (motion sensor)
- pour l'utiliser vous devez :
  1. obtenir une référence au SensorManager : `sm`
  2. obtenir une référence au capteur :
     ```
     Sensor accelSensor =
     sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER)
     ```
  3. vérifier la disponibilité (ou la présence) du capteur :
     `accelSensor != null`

Accéléromètre

- pour l'utiliser vous devez :
  3. enregistrer un listener pour le capteur cible (minDelay vaut 10000 (en microsecondes)
     comme la valeur de `minDelay` est non nulle vous devez choisir la période d'acquisition des données parmi :
     - `SENSOR_DELAY_UI`
     - `SENSOR_DELAY_NORMAL`
     - `SENSOR_DELAY_GAME`
     - `SENSOR_DELAY_FASTEST`

     ```
     sm.registerListener(anObject,accelSensor,
         SensorManager.SENSOR_DELAY_NORMAL) ;
     ```
  4. implémenter les méthodes de l'interface
- obtenir les données depuis SensorEvent se :

```
float xacc = se.values[0] ;
float yacc = se.values[1] ;
float zacc = se.values[2] ;
```

Bonnes pratiques de programmation

Register/unregister

Bonnes pratiques

- enregistrer (register) les `listeners` dans la méthode `onResume()`
- et relâcher (unregister) les `listeners` dans la méthode `onPause()`

Longues tâches

- si votre application a besoin d'exécuter de longues opérations à partir de données d'un capteur :
  - vous ne devez pas exécuter ces tâches dans le corps de la méthode `onSensorChanged()`
  - vous devez déléguer leur exécution dans une autre méthode à laquelle vous attacher un Thread dédié pour cette tâche

Références

- http ://developer.android.com/guide/topics/sensors/sensors_overview.html
- http ://developer.android.com/reference/android/hardware/SensorEvent.html
- http ://developer.android.com/guide/topics/sensors/sensors_environment.html
- http ://developer.android.com/guide/topics/sensors/sensors_position.html