Les capteurs sous android

Plan

I. Définitions

I.1. Transducteur

Un transducteur est un dispositif convertissant un signal physique en un autre ; par exemple un signal lumineux en signal nerveux (vision animale) ou signal électrique (photorécepteur). [wikipedia.org]

I.2. Instrument de mesure

En physique et en sciences de l'ingénieur, mesurer consiste à comparer une grandeur physique qui caractérise un objet (ou un événement) avec celle de même nature choisie comme unité de mesure. La valeur numérique de la grandeur mesurée est le nombre qui fixe la relation entre la grandeur mesurée et l'unité de mesure choisie. Le dispositif qui permet de réaliser la mesure est un instrument de mesure (ou appareil de mesure). [wikipedia.org]

I.3. Capteur

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. On fait souvent (à tort) la confusion entre capteur et transducteur : le capteur est au minimum constitué d'un transducteur.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même, disposant d'un affichage ou d'un système de stockage des données. Le capteur, lui, en est dépourvu.

Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation.

En sciences, l'instrumentation est une technique de mise en œuvre d'instruments de mesure, d'actionneurs, des capteurs, de régulateurs, en vue de créer un système d'acquisition de données ou de commande. [wikipedia.org]

II. Liste des capteurs sous android

Les grandeurs physiques le plus souvent mesurées actuellement par un dispositif mobile fonctionnant sous la plate-forme Android™ sont les suivantes :

- l'accélération **(Accelerometer)**;
- le champ magnétique **(Magnetic_Field)**;
- la température **(Temperature)**;
- l'intensité lumineuse **(Light)** ;
- la pression atmosphérique **(Pressure)**;
- la position géographique de l'appareil **(GPS** (n'est pas un type android.hardware.Sensor) [software.intel.com]**)**;
- l'orientation de l'appareil dans l'espace **(Orientation)**;
- la distance de l'appareil avec un élément de son environnement **(Proximity)** ;
- la mesure de la rotation en termes de vitesse autour de chaque axe **(Gyroscope)**;

La liste n'est pas exhaustive et pourra évoluer avec l'apparition de nouveaux composants. [techniques-ingenieur.fr]

Le tableau qui suit présente les informations nécessaires sur les différents capteurs android [android2ee.com].

Nom du capteur	**Sémantique**	**Unité**	***Dimension du vecteur***	**valeurs**
Accelerometer	Mesure de l'accélération (gravité incluse)	m/s^2	3	[0] axe x [1] axe y [2] axe z
Gyroscope	Mesure la rotation en termes de vitesse autour de chaque axe	Radian/seconde	3	[0] vitesse angulaire autour de x [1] vitesse angulaire autour de y [2] vitesse angulaire autour de z
Light	Mesure de la luminosité	Lux	1	[0] valeur
Magnetic_Field	Mesure du champ magnétique	µTesla	3	[0] axe x [1] axe y [2] axe z
Orientation	Mesure l'angle entre le nord magnétique	degrés	3	[0] Azimut entre l'axe y et le nord [1] Rotation autour de l'axe x (-180,180) [2] Rotation autour de l'axe y (-90,90)
Pressure	Mesure la pression	KPascal	1	[0]valeur
Proximity	Mesure la distance entre l'appareil et un objet cible	mètre	1	[0]valeur
Temperature	Mesure la température	Celsius	1	[0]valeur

L'azimut (parfois orthographié azimuth1) est l'angle dans le plan horizontal entre la direction d'un objet et une direction de référence. Le terme est issu de l'espagnol acimut, lui-même issu de l'arabe السمت (as-simt), qui signifie direction.

Cette référence peut être le Nord géographique ou magnétique. L'azimut est mesuré depuis le nord en degrés de 000° à 359° dans le sens rétrograde (sens des aiguilles d'une montre) : ainsi l'Est est au 90°, le Sud au 180° et l'Ouest au 270°. [wikipedia.org]

III. Programmation des capteurs sous android

III.1. Déclaration du capteur dans AndroidManifest

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android2ee.android.tuto.sensor.light" android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-feature android:name="android.hardware.sensor.accelerometer" android:required="true" />
    <uses-feature android:name="android.hardware.sensor.barometer"   android:required="true" />
    <uses-feature android:name="android.hardware.sensor.compass"   android:required="true" />
    <uses-feature android:name="android.hardware.sensor.gyroscope"  android:required="true" />
    <uses-feature android:name="android.hardware.sensor.light"   android:required="true" />
    <uses-feature android:name="android.hardware.sensor.proximity"  android:required="true" />
 </manifest>
```

III.2. Lister les capteurs du terminal android

```
SensorManager mg;
// Instancier le gestionnaire des capteurs, le SensorManager
mg = (SensorManager) getSystemService(SENSOR_SERVICE);

private void listerCapteurs() {
  // Trouver tous les capteurs de l'appareil :
  List<Sensor> sensors = mg.getSensorList(Sensor.TYPE_ALL);
  // La chaîne descriptive de chaque capteur
  StringBuffer sensorDesc = new StringBuffer();
  //pour chaque capteur trouvé, construire sa chaîne descriptive
  for (Sensor sensor : sensors) {
    sensorDesc.append("New sensor detected : \r\n");
    sensorDesc.append("\tName: " + sensor.getName() + "\r\n");
    sensorDesc.append("\tType: " + getType(sensor.getType()) + "\r\n");
    sensorDesc.append("Version: " + sensor.getVersion() + "\r\n");
    sensorDesc.append("Resolution (in the sensor unit): " + sensor.getResolution() + "\r\n");
    sensorDesc.append("Power in mA used by this sensor while in use" + sensor.getPower() +"\r\n");
    sensorDesc.append("Vendor: " + sensor.getVendor() + "\r\n");
    sensorDesc.append("Maximum range of the sensor in the sensor's unit."
+sensor.getMaximumRange() + "\r\n");
    sensorDesc.append("Minimum delay allowed between two events in microsecond"+ " or zero if this
sensor only returns a value when the data it's measuring changes"   + sensor.getMinDelay() +
"\r\n");
  }
  //Faire quelque chose de cette liste
  Toast.makeText(this, sensorDesc.toString(), Toast.LENGTH_LONG).show();
  // Pour trouver un capteur spécifique :
  Sensor gyroscopeDefault = mg.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
  // Pour trouver tous les capteurs d'un type fixé :
  List<Sensor> gyroscopes = mg.getSensorList(Sensor.TYPE_GYROSCOPE);
}
private String getType(int type) {
  String strType;
  switch (type) {
    case Sensor.TYPE_ACCELEROMETER: strType = "TYPE_ACCELEROMETER";break;
    case Sensor.TYPE_GRAVITY:strType = "TYPE_GRAVITY";break;
    case Sensor.TYPE_GYROSCOPE:  strType = "TYPE_GYROSCOPE";  break;
    case Sensor.TYPE_LIGHT:strType = "TYPE_LIGHT";break;
    case Sensor.TYPE_LINEAR_ACCELERATION:strType = "TYPE_LINEAR_ACCELERATION"; break;
    case Sensor.TYPE_MAGNETIC_FIELD:strType = "TYPE_MAGNETIC_FIELD";break;
    case Sensor.TYPE_ORIENTATION:strType = "TYPE_ORIENTATION";break;
    case Sensor.TYPE_PRESSURE:strType = "TYPE_PRESSURE";break;
    case Sensor.TYPE_PROXIMITY:     strType = "TYPE_PROXIMITY";   break;
    case Sensor.TYPE_ROTATION_VECTOR:    strType = "TYPE_ROTATION_VECTOR";break;
    case Sensor.TYPE_TEMPERATURE:strType = "TYPE_TEMPERATURE";break;
    default: strType = "TYPE_UNKNOW";break;
  }
  return strType;
}
```

III.3. Utilisation d'un capteur pour écouter le changement de valeurs

Quand on souhaite travailler avec les capteurs, la première chose à faire est de savoir écouter leur changement d'état (de valeurs). Pour cela, c'est assez simple, il suffit de les instancier puis de s'enregistrer en tant qu'écouteur dans la méthode onResume et de se désenregistrer dans la méthode onPause. Pour les écouter il suffit d'implémenter l'interface SensorEventListener (souvent au niveau de votre activité). *[android2ee.com]*

Au lieu d'un long discours, quelques lignes de code éclaireront cette explication. Prenons l'exemple de l'accéléromètre :

```
public class SensorAccelerationTutoActivity extends Activity implements SensorEventListener {
  SensorManager mg;
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Instancier le gestionnaire des capteurs, le SensorManager
```

```
        mg = (SensorManager) getSystemService(SENSOR_SERVICE);
        // Instancier l'accéléromètre
        accelerometer = mg.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
      }
      @Override
      protected void onPause() {
        // unregister the sensor (désenregistrer le capteur)
        mg.unregisterListener(this, accelerometer);
        super.onPause();
      }
      @Override
      protected void onResume() {
        /* Ce qu'en dit Google dans le cas de l'accéléromètre :
        * «  Ce n'est pas nécessaire d'avoir les évènements des capteurs à un rythme trop rapide.
        * En utilisant un rythme moins rapide (SENSOR_DELAY_UI), nous obtenons un filtre
        * automatique de bas-niveau qui "extrait" la gravité  de l'accélération.
        * Un autre bénéfice étant que l'on utilise moins d'énergie et de CPU. »
        */
        mg.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_UI);
        super.onResume();
      }
      @Override
      public void onAccuracyChanged(Sensor sensor, int accuracy) { }
      @Override
      public void onSensorChanged(SensorEvent event) {
        // Récupérer les valeurs du capteur
        float x, y, z;
        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
        x = event.values[0];
        y = event.values[1];
        z = event.values[2];
      }
```

}Cet exemple est celui de l'accéléromètre, mais il s'applique à tous les capteurs.

III.4. Le SensorEvent

```
public class SensorAccelerationTutoActivity extends Activity implements SensorEventListener {
  SensorManager mg;
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Instancier le gestionnaire des capteurs, le SensorManager
    mg = (SensorManager) getSystemService(SENSOR_SERVICE);
    // Instancier l'accéléromètre
    accelerometer = mg.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
  }
  @Override
  protected void onPause() {
    // unregister the sensor (désenregistrer le capteur)
    mg.unregisterListener(this, accelerometer);
    super.onPause();
  }

  @Override
  protected void onResume() {
    /* Ce qu'en dit Google dans le cas de l'accéléromètre :
    * «  Ce n'est pas nécessaire d'avoir les évènements des capteurs à un rythme trop rapide.
    * En utilisant un rythme moins rapide (SENSOR_DELAY_UI), nous obtenons un filtre
    * automatique de bas-niveau qui "extrait" la gravité  de l'accélération.
    * Un autre bénéfice étant que l'on utilise moins d'énergie et de CPU. »
    */
    mg.registerListener(this, accelerometer, SensorManager.SENSOR_DELAY_UI);
    super.onResume();
  }
  @Override
  public void onAccuracyChanged(Sensor sensor, int accuracy) {  }
  @Override
            public void onSensorChanged(SensorEvent event) {
```

```
    // Récupérer les valeurs du capteur
    float x, y, z;
    if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
    x = event.values[0];
    y = event.values[1];
    z = event.values[2];
  }
}
```

III.5. Le capteur de la lumière

Ce capteur est extrêmement facile d'utilisation, mais sa sensibilité dépend du fabricant. Certains Smartphones ne renvoient que des puissances de 10 (10, 100, 1000, et ainsi de suite). Ce capteur permet de savoir quelle est l'intensité lumineuse détectée par votre téléphone (l'unité est le Lux). *[android2ee.com]*

Ainsi pour écouter les changements de valeur de ce capteur, il vous faut :

- le déclarer dans votre fichier AndroidManifest le capteur de lumière ;
- faire étendre votre activité (ou la classe qui écoute le capteur) de SensorListener ;
- déclarer et instancier un objet Sensor de type light ;
- s'enregistrer/se désenregistrer en tant qu'écouteur de ce capteur ;
- faire quelque chose lors d'un changement de valeur.

```
public class SensorLightTutoActivity extends Activity implements SensorEventListener {
  float l;
  Sensor light;
  @Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    // Instancier le SensorManager
    mg = (SensorManager) getSystemService(SENSOR_SERVICE);
    // Instancier le capteur de lumière
    light = mg.getDefaultSensor(Sensor.TYPE_LIGHT);                …
  }
  @Override
  protected void onPause() {
    // désenregistrer notre écoute du capteur
    mg.unregisterListener(this, light);
    super.onPause();
  }
  @Override
  protected void onResume() {
    /* * enregistrer notre écoute du capteur*/
    mg.registerListener(this, light, SensorManager.SENSOR_DELAY_GAME);
    super.onResume();
  }
  @Override
  public void onAccuracyChanged(Sensor sensor, int accuracy) { }
  @Override
  public void onSensorChanged(SensorEvent event) {
    // Mettre à jour uniquement dans le cas de notre capteur
    if (event.sensor.getType() == Sensor.TYPE_LIGHT) {
      // La valeur de la lumière
      l = event.values[0];
    }
  }
}
```

III.6. Le capteur de proximité

Ce capteur est extrêmement facile d'utilisation, mais sa sensibilité dépend du fabricant. Certains Smartphones ne renvoient que soit 0, soit 5 mètres. Pour comprendre pourquoi il faudrait le nommer « capteur qui détecte la présence du corps humain au niveau de l'écouteur de l'appareil ». S'il détecte une présence il renvoie 0 sinon il renvoie 5. *[android2ee.com]*

III.7. L'accéléromètre

À partir de ce capteur Android en dérive deux de plus, le Linear Acceleration et la Gravity.

L'accéléromètre fournit le vecteur de force (ou d'accélération, c'est la même chose) tridimensionnel (x,y,z).

À partir de ce vecteur le système déduit la composante gravitationnelle (toujours un vecteur tridimensionnel). Celle-ci est celle renvoyée par le capteur Gravity. Le capteur Linear Acceleration déduit du champ de force cette composante fournissant un vecteur (tridimensionnel) épuré de la gravité. Cela explique pourquoi les capteurs Gravity et Linear Acceleration n'ont pas de numéro de série et sont fournis par Google. *[android2ee.com]*

III.8. Le capteur électromagnétique

Le capteur électromagnétique permet de connaître les valeurs du vecteur électromagnétique qui s'applique à votre téléphone. Le référentiel est le même que celui utilisé par le capteur d'accélération. Les valeurs sont en micro-tesla.

Au-delà du vecteur en lui-même, sa norme permet de savoir ce que l'on appelle la valeur électromagnétique. Si vous placez votre appareil à proximité d'un aimant (par exemple des baffles) vous verrez les valeurs augmenter. *[android2ee.com]*

III.9. Le capteur d'orientation

Comme son nom l'indique ce capteur donne l'orientation (le nord). Ce capteur est la boussole de votre appareil. Il y a trois notions à comprendre avec ce capteur :

- l'azimut : donne l'angle avec le Nord magnétique ;
- le pitch : donne l'angle autour de l'axe des x. En français cela se dit le tangage, mais on garde la nomenclature anglaise pour être en cohérence avec le SDK ;
- le roll : donne l'angle autour de l'axe des y. En français cela se dit le roulis, aussi garde la nomenclature anglaise pour être en cohérence avec la SDK.

L'azimut varie entre 0 et 360°, il représente l'angle avec le nord dans le sens des aiguilles d'une montre.

Le pitch varie entre -180 et 180, il représente l'inclinaison haut-bas de l'appareil selon l'axe Y (parallèle au sol, perpendiculaire au sol). On obtient les valeurs suivantes :

- la valeur 0, l'appareil est parallèle au sol, face vers le ciel ;
- la valeur +/- 180 est l'appareil parallèle au sol, face vers le sol ;
- la valeur 90 est l'appareil perpendiculaire au sol face, tête vers le bas ;
- la valeur -90 est l'appareil perpendiculaire au sol face, tête vers le haut.

Enfin le roll varie entre -90 et 90, il représente l'inclinaison droite-gauche de l'appareil selon l'axe des X (si l'appareil penche à gauche ou à droite). On obtient les valeurs suivantes (quand sa face est vers le haut) :

- la valeur 0, l'appareil ne penche pas ;
- la valeur 90 est l'appareil penche à gauche ;
- la valeur -90 est l'appareil penche à droite.

Il y a deux façons distinctes pour obtenir cette orientation, soit en écoutant directement le capteur d'orientation (ce qui n'a pas l'air d'être la bonne pratique), soit en utilisant le champ magnétique et le champ de force. *[android2ee.com]*

III.10. Le capteur vecteur d'orientation

Ce capteur représente le vecteur de rotation de l'appareil, de la même manière que l'orientation. Ce n'est pas un capteur physique mais plus un capteur interpolé (comme Gravity).

Références :

[wikipedia.org] http://fr.wikipedia.org/wiki/Capteur
[techniques-ingenieur.fr] http://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/systemes-embarques-42588210/programmation-de-capteurs-sur-dispositifs-mobiles-h1595/
[android2ee.com] http://www.android2ee.com/tutoriaux/les-capteurs.html
[software.intel.com] https://software.intel.com/fr-fr/articles/developing-sensor-applications-on-intel-atom-processor-based-android-phones-and-tablets