P82 Mobile Application Development - Android

Sensors

Device Dependent Sensor

♦ 3 categories:

♦ Motion sensors:

/ Using the forces of acceleration and rotation on the three axes, it is possible to determine in which direction the device is moving by using the following motion sensors: the accelerometer, the gravity sensors, the gyroscopes and the vectors of rotation vectors.

♦ Position sensors:

/ determine the position of the device: the orientation sensors and the magnetometer

♦ Environmental sensors:

/ there are three sensors (barometer, photometer and thermometer) that measure atmospheric pressure, illumination and ambient temperature.

http://openclassrooms.com/courses/creez-des-applications-pour-a
ndroid / les-sensors

Using Sensors

```
<uses-feature

    android:name="android.hardware.sensor.accelerometer"
    android:required="true" />
```

- Uses the SensorManager object

```
SensorManager sensorManager =
    (SensorManager)getSystemService(Context.SENSOR_SERVI CE);
```

- Can Get the list of Sensors Available

```
List<Sensor> sList = sensorManager.getSensorList(Sensor.TYPE_ALL);
```

Using Sensors

- Verify the presence of a Sensor with getDefaultSensor(int Type) method

- Sensor accelerometer =
  sensorManager.getDefaultSensor(Sensor.*TYPE_ACCELERO METER*);

```
if(accelerometer != null) {
    //There is at least one accelerometer
}else {
    // There are none
}
```

- Can get the consumption

- accelerometre.getPower()

SensorEventListener Interface

```
@Override
public void onSensorChanged(SensorEvent event) { }
```

```
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {}
```

Adding the listener

- To save processing

  / adding the listener

```
@Override
  protected void onResume() {
      super.onResume();
      mSensorManager.registerListener(this, accelerometre,
  SensorManager.SENSOR_DELAY_NORMAL);
  }
```

Alternately you could use : SensorManager.*SENSOR_DELAY_GAME*

  / Removing a Listener

```
@Override
  protected void onPause() {
          super.onPause(); mSensorManager.unregisterListener(this);
  }
```

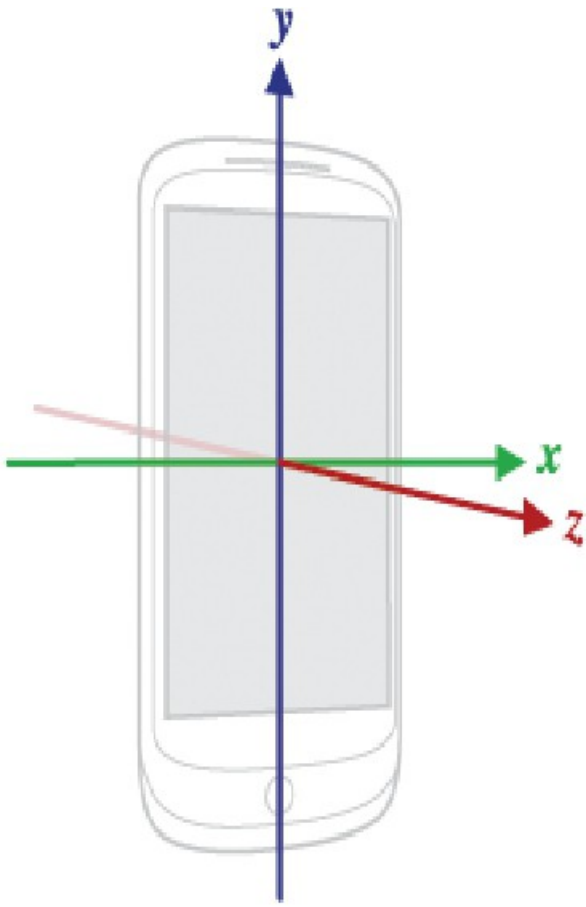
Redefining the SensorChange Method

- Most sensors will return a flow chart with three values
  - @Override

```
public void onSensorChanged(SensorEvent event) {
        float x = event.values[0];
        float y = event.values[1];
        float z = event.values[2];
}
```

Acceleration

The x, y and z values represent the acceleration force in m/s^2, including Earth's gravity..



To know the acceleration without gravity used

Sensor.TYPE_LINEAR_ACCELERATION

http://developer.android.com/guide/to
pics / sensors / sensors_overview.html