SCIENCES SORBONNE UNIVERSITÉ

Sensors

Renault@lrde.epita.fr

Description

Sensors

- are device dependent
- produce raw data

Many sensors may be available on devices

- Based on user's movements
  - **Accelerometer, Gyro, rotation sensors, ...**
- Based on the environment
  - **Temperature, barometer, ...**
- Based on positions
  - **Magnetometers, GPS, ...**

Sensor Framework

SensorManager	Helps to calibrate and detect constants
Sensor	An instance of a given sensor
SensorEvent	Information stored after an event on the sensor
SensorEventListener	Listener on the sensor

Detecting Device Sensors

List available sensors

```
SensorManager mSensorManager =
    (SensorManager) getSystemService(Context.SENSOR_SERVICE);
List<Sensor> deviceSensors =
    mSensorManager.getSensorList(Sensor.TYPE_ALL);
for (Sensor s: deviceSensors)
    Toast.makeText(getApplicationContext(), s.toString(),
                   Toast.LENGTH_SHORT).show();
```

Types of Sensors

- Sensor.TYPE_GYROSCOPE
- Sensor.TYPE_LINEAR_ACCELERATION
- Sensor.TYPE_GRAVITY
- Sensor.TYPE_ALL

Using Light Sensor (1/2)


```
public class MainActivity extends ActionBarActivity
                          implements SensorEventListener {
 private SensorManager mSensorManager;
 private Sensor mLight;

 @Override
 public final void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mSensorManager =
      (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    mLight = mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
}

@Override
public final void onAccuracyChanged(Sensor sensor,
                                    int accuracy)  {
       // Do something here if sensor accuracy changes.
}
```

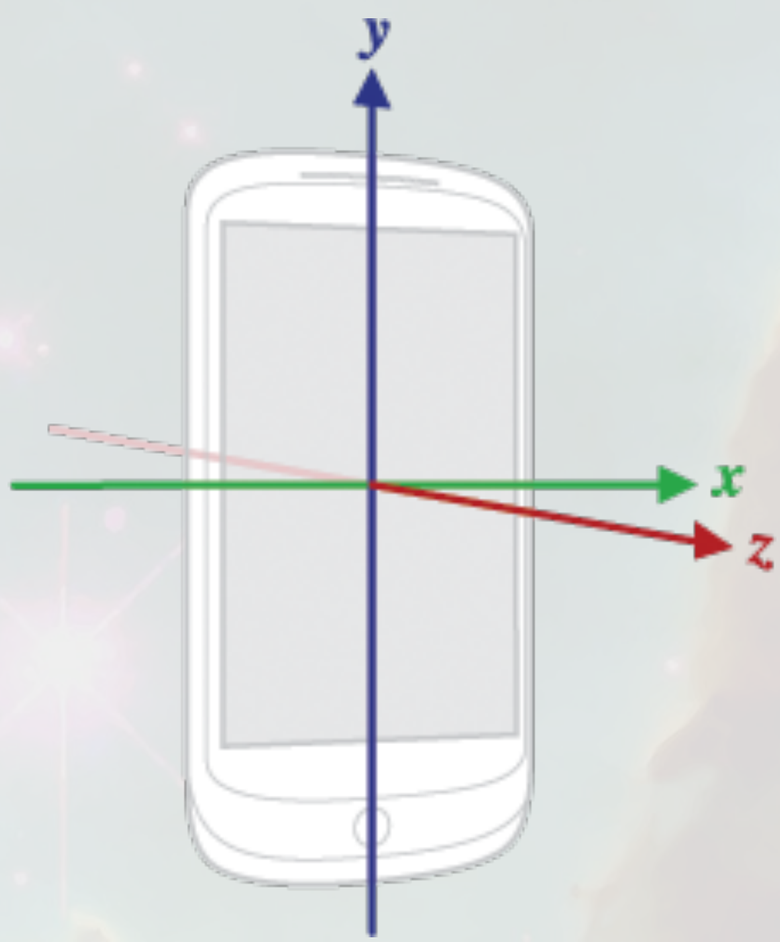
Using Light Sensor (2/2)


```java
@Override
public final void onSensorChanged(SensorEvent event) {
  // The light sensor returns a single value.
  // Many sensors return 3 values, one for each axis.
  float lux = event.values[0];
  TextView v = (TextView) findViewById(R.id.display);
  v.setText("LUX : " + lux );
}
@Override
protected void onResume() {
  super.onResume();
  mSensorManager.registerListener(this, mLight,
                                  SensorManager.SENSOR_DELAY_NORMAL);
}
@Override
protected void onPause() {
  super.onPause();
  mSensorManager.unregisterListener(this);
}
```

Motion Sensor

Works similarly

- Register to SensorEventListener in onResume
- Unregister in onStop
- Register for accelerometer event (for instance)
- onSensorChange provides informations
  - **x in event.values[0]**
  - **y in event.values[1]**
  - **z in event.values[2]**



Runtime Detection of Sensors

- Sensors can be plugged and unplugged anytime
  - Sensors may be broken!

- Detect sensors at runtime

```
private SensorManager mSensorManager;
   ...
mSensorManager =
    (SensorManager) getSystemService(Context.SENSOR_SERVICE);
if (mSensorManager.getDefaultSensor(Sensor.TYPE_PRESSURE)
     != null){
       // Success! There's a pressure sensor.
}
else {
   // Failure! No pressure sensor.
}
```

Summary

Do not forget to declare in AndroidManifest.xml

```
<uses-feature android:name=
                  "android.hardware.sensor.accelerometer"
              android:required="true"
/>
```

Multiple sensors are available on devices BUT

- Choose low refreshing rate to be friendly with battery

All sensors use the same interface

- one can register to all sensors