

Android alkalmazásfejlesztés

Szenzorok

GPS és helymeghatározás

OE-NIK

2013. március 11.

Sicz-Mesziár János

sicz-mesziar.janos@
nik.uni-obuda.hu



Szenzorok



Szenzorok Androidon

◎ Mozgásérzékelők

Tipikus felhasználás

Accelerometer	Hardware	Mozgás detektálás(shake, tilt, ...)
Gravity	Hardware/Software	Mozgás detektálás (shake, tilt, ...)
Gyroscope	Hardware	Forgás detektálása (spin, turn, ...)
Linear acceleration	Hardware/Software	Gyorsulás adott tengely mentén
Rotation vector	Hardware/Software	Mozgás és forgás detektálás

◎ Pozíciós szenzorok

Magnetic field	Hardware	Íránytű
Orientation	Software	Eszköz helyzetének meghatározása
Proximity	Hardware	Telefon helyzete a hívás alatt

◎ Környezeti szenzorok

Ambient temperature	Hardware	Környezeti hőmérséklet mérése
Light	Hardware	Háttérvilágítás szabályzása
Pressure	Hardware	Légnyomás változás figyelése
Relative humidity	Hardware	Abszolút, relatív páratartalom
Temperature	Hardware	Belső, eszköz hőmérséklet

Szenzorok támogatottsága

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
<u>TYPE_ACCELEROMETER</u>	Yes	Yes	Yes	Yes
<u>TYPE_AMBIENT_TEMPERATURE</u>	Yes	n/a	n/a	n/a
<u>TYPE_GRAVITY</u>	Yes	Yes	n/a	n/a
<u>TYPE_GYROSCOPE</u>	Yes	Yes	n/a ¹	n/a ¹
<u>TYPE_LIGHT</u>	Yes	Yes	Yes	Yes
<u>TYPE_LINEAR_ACCELERATION</u>	Yes	Yes	n/a	n/a
<u>TYPE_MAGNETIC_FIELD</u>	Yes	Yes	Yes	Yes
<u>TYPE_ORIENTATION</u>	Yes ²	Yes ²	Yes ²	Yes
<u>TYPE_PRESSURE</u>	Yes	Yes	n/a ¹	n/a ¹
<u>TYPE_PROXIMITY</u>	Yes	Yes	Yes	Yes
<u>TYPE_RELATIVE_HUMIDITY</u>	Yes	n/a	n/a	n/a
<u>TYPE_ROTATION_VECTOR</u>	Yes	Yes	n/a	n/a
<u>TYPE_TEMPERATURE</u>	Yes ²	Yes	Yes	Yes

¹ Megjelenése Android 1.5, de Android 2.3 óta érhető el

² Szenzor elérhető, de elavult (deprecated)

Tudni érdeemes

☉ Szenzor támogatás gyártó és Android verzió függő

- http://developer.android.com/guide/topics/sensors/sensors_overview.html

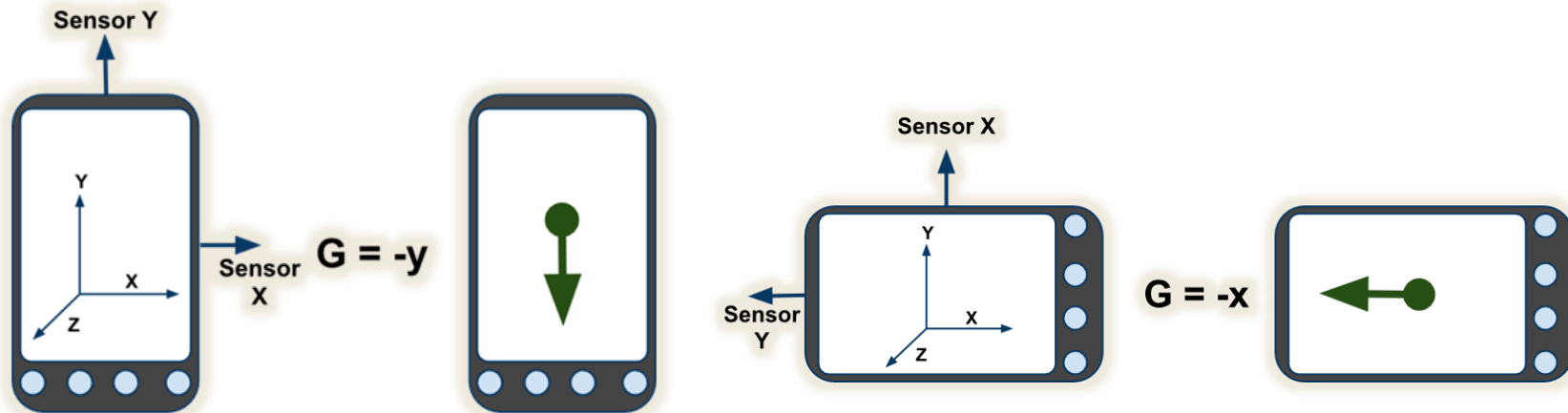
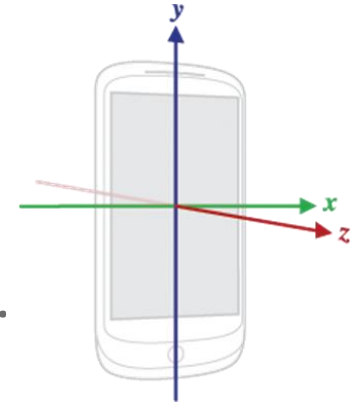
☉ Koordináta rendszer

- 3 tengelyű koordináta rendszer. (X, Y, Z)
- Portrai mód az alapértelmezett.

Landscape: a koordinátarendszer nem fordul el.

(hasonlóan az OpenGL koordináta rendszerhez)

De forgatható: [SensorManager.remapCoordinateSystem\(\)](#)



Tudni érdeemes (2)

⦿ Az emulátor nem támogatja a szenzorok emulálását!

- De van alternatíva → OpenIntents SensorSimulator
<http://code.google.com/p/openintents/wiki/SensorSimulator>

⦿ Mintavételezés és áramfelvétel HTC Dream esetén

- SENSOR_DELAY_NORMAL 10mA (*orientáció detektálás*)
- SENSOR_DELAY_UI 15mA (*~1adat/sec*)
- SENSOR_DELAY_GAME 80mA
- SENSOR_DELAY_FASTEST 90mA

⦿ Az adat egy float[] tömbben érkezik

- values[0]
- values[1]
- values[2]

Kamera, mikrofon, touchscreen is szenzor, csak másképpen kezeljük. 😊

Ajánlott videó a szenzorokkal kapcsolatban:

<http://www.youtube.com/watch?v=C7JQ7Rpwn2k>

Szenzorok elérése gyakorlatban

⊙ Jogosultság kérése ebben az esetben nem szükséges.

⊙ SensorManager példányosítása:

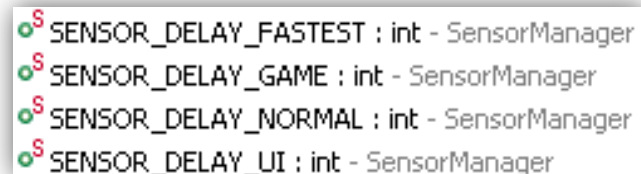
```
SensorManager manager =  
    (SensorManager) getSystemService (SENSOR_SERVICE) ;
```

⊙ SensorEventListener implementálása:

```
SensorEventListener listener = new  
    SensorEventListener () {  
    public void onSensorChanged (SensorEvent event) {}  
    public void onAccuracyChanged (Sensor s, int a) {}  
};
```

⊙ Feliratkozás a szenzor adatokra

```
manager.registerListener (  
    listener,  
    manager.getDefaultSensor (Sensor.TYPE_PROXIMITY) ,  
    SensorManager.SENSOR_DELAY_FASTEST  
);
```



- SENSOR_DELAY_FASTEST : int - SensorManager
- SENSOR_DELAY_GAME : int - SensorManager
- SENSOR_DELAY_NORMAL : int - SensorManager
- SENSOR_DELAY_UI : int - SensorManager

GPS és helymeghatározás



- GPS
- Network Location
- Proximity Alert
- Fájlformátumok

Helymeghatározás Android alatt

„An Android phone always knows where it is.”

Ed Burnette – Hello, Android

Adatok forrása:

◎ GPS :

- legpontosabb, de
- „csak” kültéren használható,
- nagyobb fogyasztás,
- lassú információszerzés (mint ahogy felhasználó szeretné)

◎ Hálózati információk alapján (Wi-Fi, Cell-ID) :

- kevésbé pontos,
- kültéri és beltéri használat,
- gyors információszerzés,
- kevesebb fogyasztás

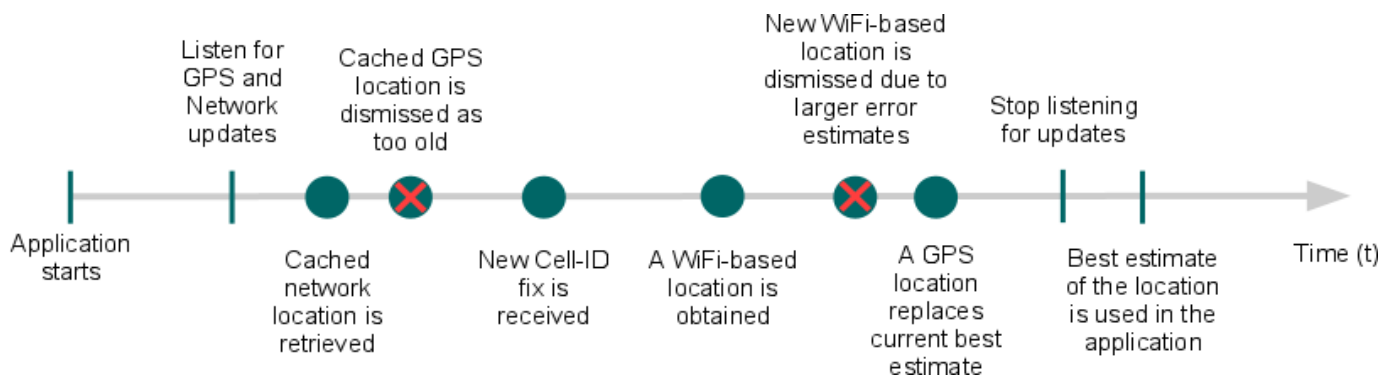
Nehézségek

◎ Felhasználó helymeghatározásában rejlő nehézségek:

- Felhasználó mozgásban (gyakori mérés kell)
- Változó pontosság:
Lehet, hogy a 10mp-el korábbi adat pontosabb, mint az új adatból származó becslés
- Van-e GPS a készülékben?
 - Be van-e kapcsolva?
 - Elérhető-e adat?
 - Elég pontos-e?

Google I/O 2009:
Fogyasztás?

◎ Pozíció meghatározásának ideje



Helymeghatározás a gyakorlatban

1. Jogosultság kérése: **AndroidManifest.xml / Permissions**

- `android.permission.ACCESS_FINE_LOCATION`
- `android.permission.ACCESS_COARSE_LOCATION`
- `android.permission.ACCESS_MOCK_LOCATION`

Pontos adatok

Közelítő adatok

Hamis adatok -
emulátorhoz

2. LocationManager elérése

```
LocationManager locationManager =  
    (LocationManager) getSystemService(LOCATION_SERVICE);
```

3. LocationListener definiálása

```
LocationListener myListener = new LocationListener() {  
    public void onStatusChanged(...) {}  
    public void onProviderEnabled(...) {}  
    public void onProviderDisabled(...) {}  
    public void onLocationChanged(...) {}  
};
```

GPS status változik

Szolgáltatás elérhető

Szolgáltatás nem elérhető

Pozíció változik

4. A myListener regisztrálása

```
locationManager.requestLocationUpdates(  
    locationManager.GPS_PROVIDER, 0, 0, myListener);
```

További érdekes adatok

- ⊙ A felhasználó szeretne gyorsan pozíciót kapni, ezért lekérhető az utoljára ismert helyzete:

```
locationManager.getLastKnownLocation();
```

- ⊙ Két pont közötti távolság:

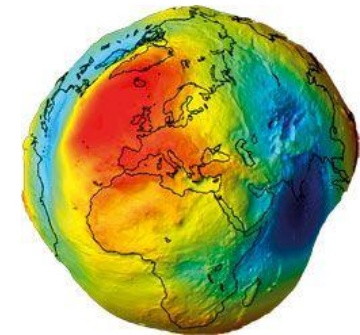
```
Location.distanceTo(Location dest);
```

WGS84 ellipszoid
támogatás

- ⊙ Szatellit információk

GpsStatus.Listener megvalósítása:

```
public void onGpsStatusChanged(int event) {  
    switch(event) {  
        case GpsStatus.GPS_EVENT_FIRST_FIX: break;  
        case GpsStatus.GPS_EVENT_STARTED: break;  
        case GpsStatus.GPS_EVENT_STOPPED: break;  
        case GpsStatus.GPS_EVENT_SATELLITE_STATUS:  
            GpsStatus status = locationManager.getGpsStatus(null);  
            for(GpsSatellite sat:status.getSatellites()) { /**/ }  
            break;  
        }  
    }  
}  
locationManager.addGpsStatusListener(new GpsStatus.Listener() {...});
```



NMEA nyersadatok

◎ ASCII alapú adatközlés, szabványos mondatok formájában:

- \$ jellel kezdődnek
- Ezt követi a küldő fél + mondat típusa
- Adatok vesszővel vannak elszeparálva
- * jelzi az utolsó értéket, utána checksum (XOR)

```
$GPGLL,4916.45,N,12311.12,W,225444,A,*1D
```

◎ Gyakorlatban:

```
locationManager.addNmeaListener(  
    new NmeaListener() {  
        public void onNmeaReceived(long t, String nmea) {  
            Log.d("NIK", nmea);  
        }  
    }  
);
```

GPS adatok feldolgozásához ajánlott: 😊
[GPS alapú rendszerek – Léczfalvy Ádám](#)

Proximity Alert

- ⊙ Közelségi riasztás, != proximity sensor
- ⊙ Jelzés, ha az adott pozícióhoz megadott rádiuszon belülrre / kívülrre kerülünk.
- ⊙ Jelzés → Intent kibocsátása
 - Extra data (KEY_PROXIMITY_ENTERING)
 - Boolean típusú:
 - **True:** belép a területre
 - **False:** kilép a területről
- ⊙ Energiatakarékos!:
 - Kellően nagy távolság esetén csak a hálózat adataira épít. Kis távolság esetén automatikusan GPS-re vált.
 - Lekapcsolt képernyő esetén, csak 4 percenként ellenőriz.

Fájlformátumok

- ◎ GPX - GPS eXchange file
 - XML alapú adattárolás
 - Egyéni értékekkel is bővíthető
- ◎ KML - Keyhole Markup Language file
 - XML alapú formátum
 - Pontok, vonalak, képek, sokszögek és megjelenítési modellek tárolására és modellezésére találták ki
 - KMZ a KML tömörített formátuma
- ◎ CSV – Comma-separated values
- ◎ TCX - Training Center XML
 - Garmin szabványa, hasonló, mint a GPX
 - Tárol aktív sportolói életben szokásos jellemzőket is, mint: szívverés, kalória, kerékpárnál fordulat/perc, ...

