

# ANDROID ALKALMAZÁSFEJLESZTÉS

Szenzorok,  
GPS helymeghatározás



[sicz-mesziar.janos@nik.uni-obuda.hu](mailto:sicz-mesziar.janos@nik.uni-obuda.hu)

Sicz-Mesziár János

2017. április 2.

# 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

# Tudni érdeemes

## Szenzor támogatás gyártó és Android verzió függő

- [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html)

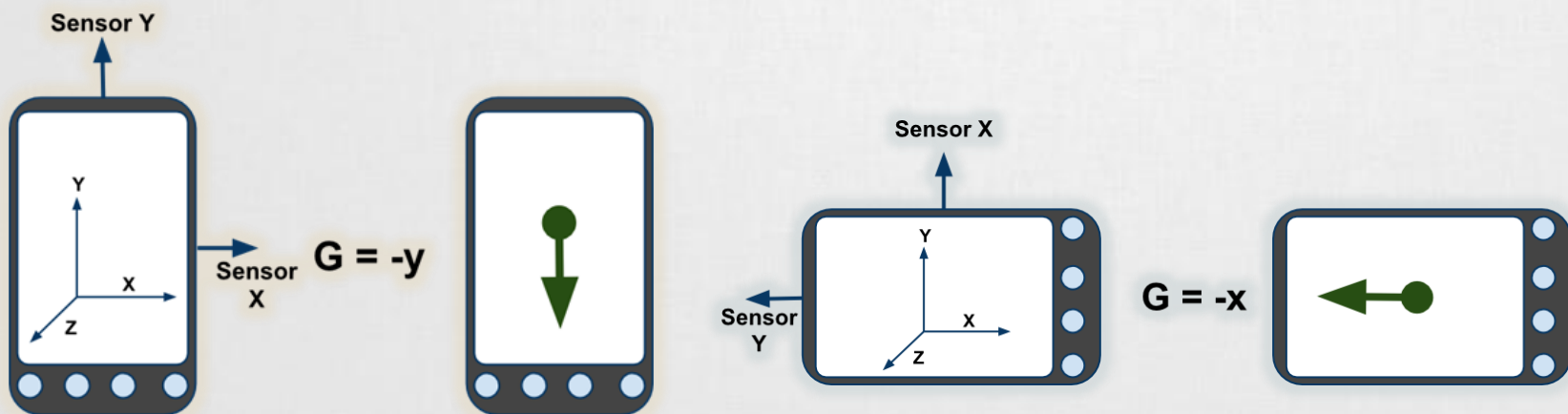
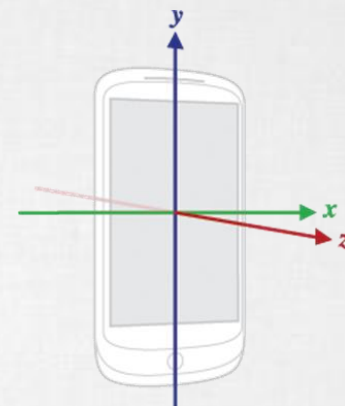
## Koordináta rendszer

- 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 Sensor Simulator~~  
<http://code.google.com/p/openintents/wiki/SensorSimulator>

### Mintavételezés és áramfelvétel HTC Dream esetén

- SENSOR\_DELAY\_NORMAL
- SENSOR\_DELAY\_UI
- SENSOR\_DELAY\_GAME
- SENSOR\_DELAY\_FASTEST

### 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 a 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  
);
```

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



GPS



# 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

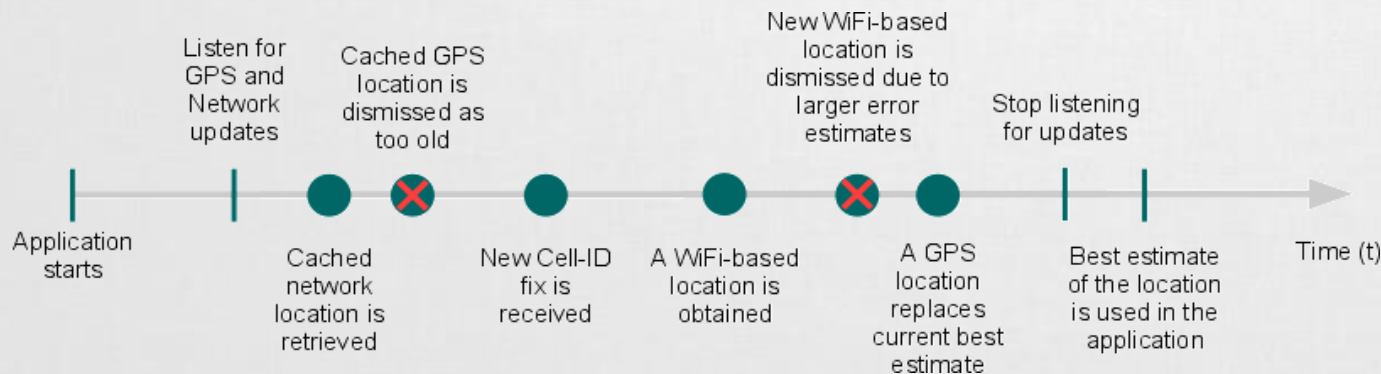
# Nehézségek

## 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` Pontos adatok
- `android.permission.ACCESS_COARSE_LOCATION` Közelítő adatok
- `android.permission.ACCESS_MOCK_LOCATION` 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);
```

# GPS adatok szimulálása

Location

Cellular

Battery

Phone

Directional pad

Fingerprint

Settings

Help

GPS data point

Decimal

Sexagesimal

Latitude  
37.422

Longitude  
-122.084

Altitude (meters)  
0.0

SEND

GPS data playback

Delay (sec)	Latitude	Longitude	Elevation	Name	Description
0	47.53357	19.03385	145		
1	47.5335	19.03389	151		
1	47.53355	19.03387	154		
3	47.53359	19.03378	160		
7	47.53361	19.03371	165		
4	47.53365	19.03366	166		
4	47.53368	19.03359	168		

Speed 1X

LOAD GPX/KML

# További érdekes adatok

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

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

**Két pont közötti távolság:**

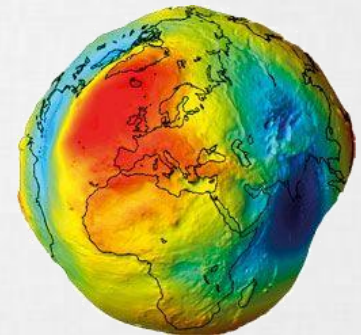
```
Location.distanceTo(Location dest);
```

**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);  
        }  
    }  
);
```

# Proximity Alert

Közelségi riasztás, != **proximity sensor**

Jelzés, ha az adott pozícióhoz megadott rádiuszon belülrre / kívülre 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.

# Fájl formátumok

## **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 fordulát/perc, ...

# Location APIs

A Google Play Services része:

<https://developer.android.com/google/play-services/location.html>

- **Fused location provider**

Továbbfejlesztett helymeghatározás, egyszerűbb API, azonnali hozzáférés az utolsó pozícióhoz,

- **Activity Recognition**

fizikai tevékenységek detektálása, mint például áll, gyalogol, fut, vezet, kerékpár, ... + egyéni mozgás minták alkalmazása

- **Geofencing APIs**

kijelölt területre való belépés és kilépés detektálása

