ANDROID ALKALMAZÁSFEJLESZTÉS

Szenzorok,
GPS helymeghatározás

Sicz-Mesziár János
sicz-mesziar.janos@nik.uni-obuda.hu

Mezei József
mezei.jozsef@nik.uni-obuda.hu

2018. október 21.

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	Iránytű
Orientation	Software	Eszköz helyzetének meghatározása
Proximity	Hardware	Telefon helyzete a hívás alatt

Környezeti szenzorok

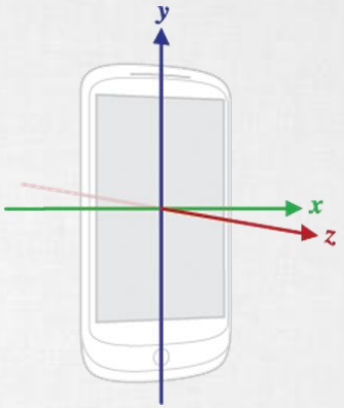
Ambient temperature	Hardware	Környezeti hőmérséklet mérése
Light	Hardware	Háttérvilágítás szabályzása
Pressure	Hardware	Légnyomás változás figyelése
Relative humidity	Hardware	Abszolút, relatív páratartalom
Temperature	Hardware	Belső, eszköz hőmérséklet

Tudni érdemes

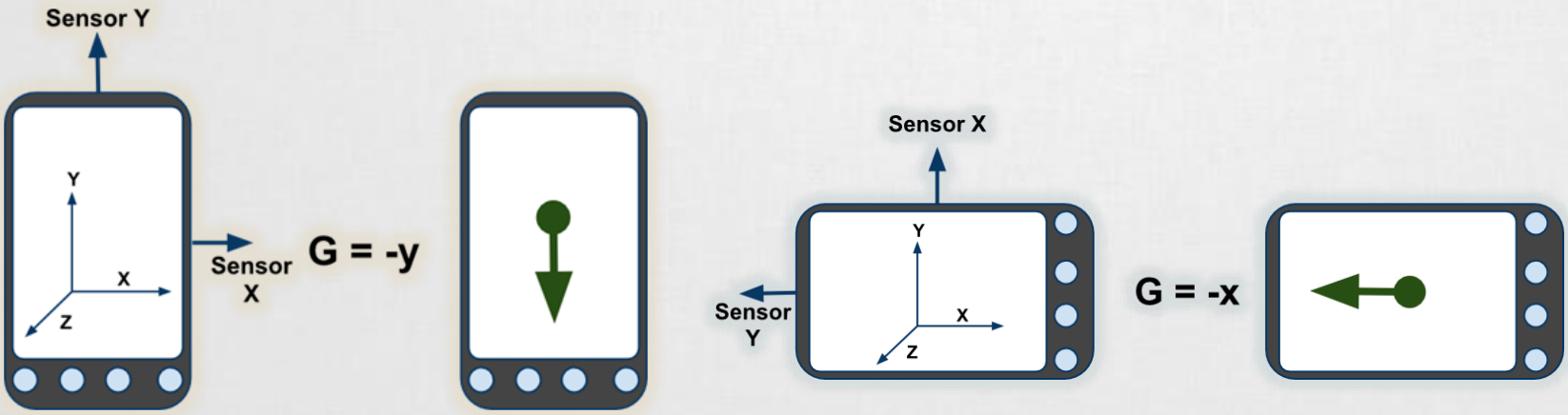
Szenzor támogatás gyártó és Android verzió függő

- 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 érdemes (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
- SENSOR_DELAY_UI
- SENSOR_DELAY_GAME
- SENSOR_DELAY_FASTEST

Az adat egy float[] tömben 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:

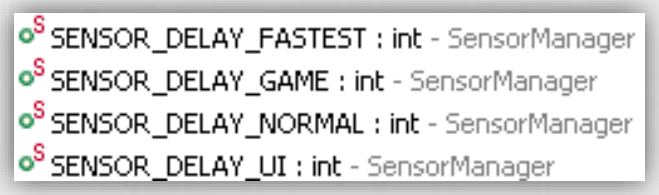
```
val manager = getSystemService(Context.SENSOR_SERVICE) as SensorManager
```

SensorEventListener implementálása:

```
val listener = object : SensorEventListener {
     override fun onAccuracyChanged(sensor: Sensor?, accuracy: Int) {}
     override fun onSensorChanged(event: SensorEvent?) {}
   }
```

Feliratkozás a szenzor adatokra

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



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

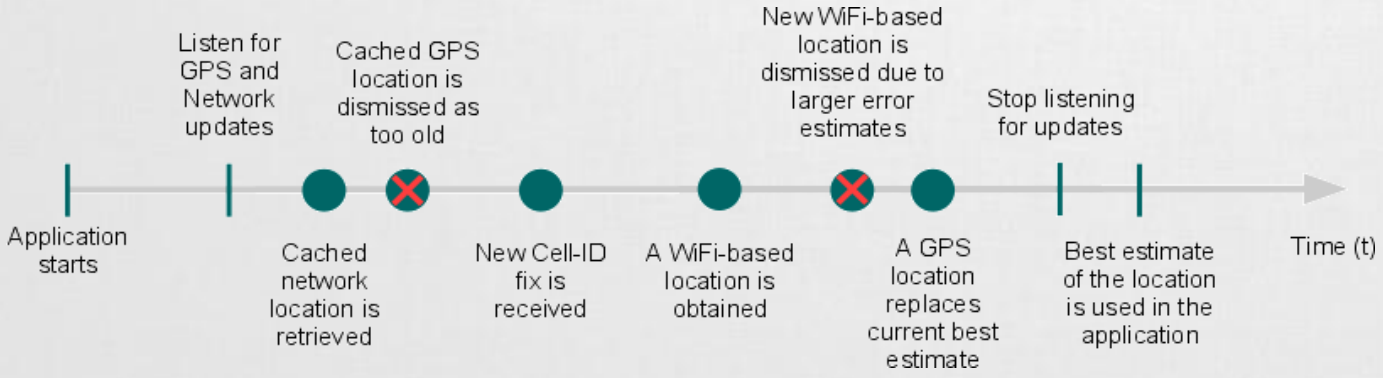
Nehézségek

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
- android.permission.ACCESS_COARSE_LOCATION
- android.permission.ACCESS_MOCK_LOCATION

Pontos adatok

Közelítő adatok

Hamis adatok - emulátorhoz

2. LocationManager elérése

```
val manager = getSystemService(Context.LOCATION_SERVICE) as
    LocationManager
```

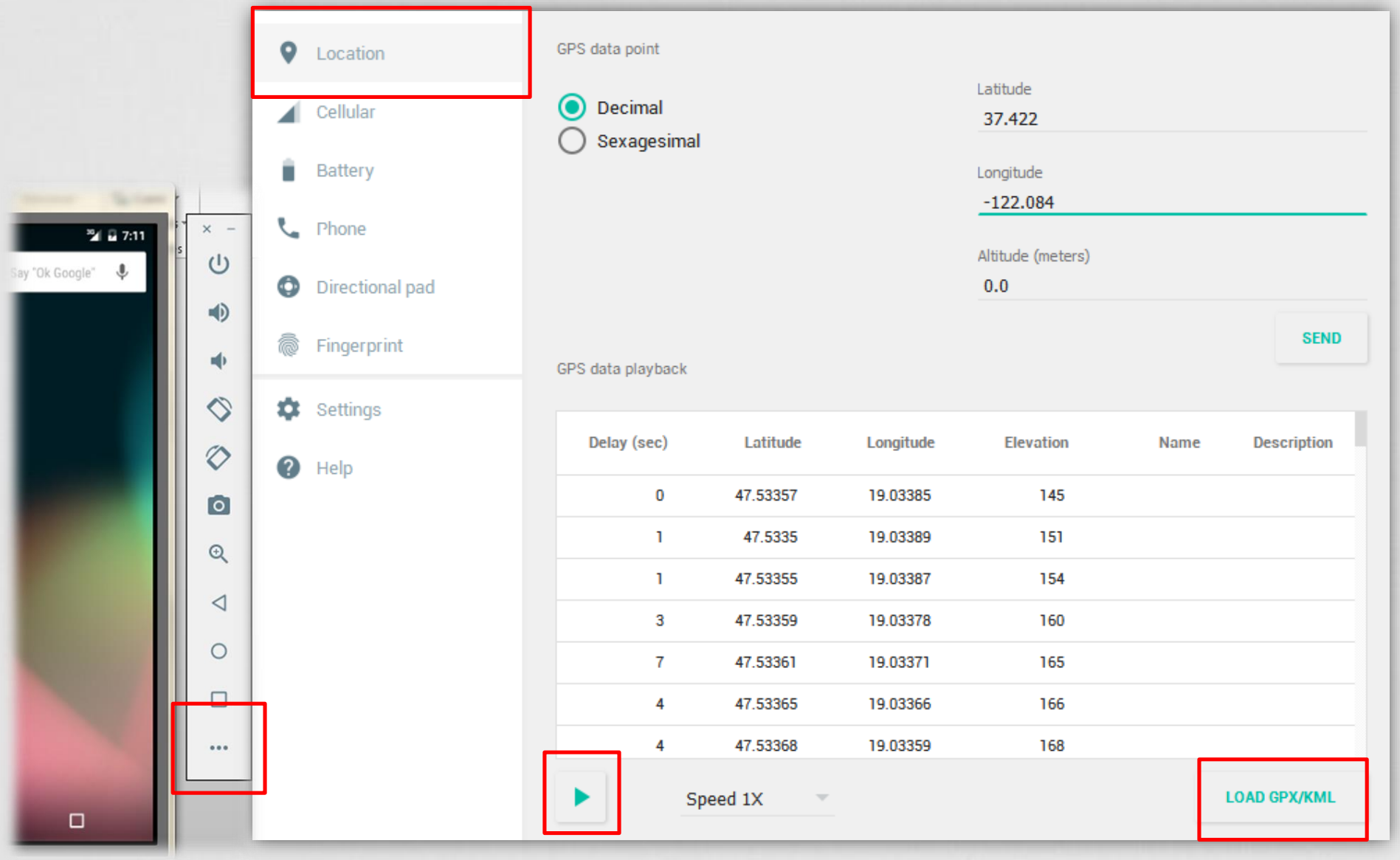
3. LocationListener definiálása

```
val listener = object : LocationListener {
   override fun onLocationChanged(location: Location?) {}
   override fun onStatusChanged(
        provider: String?, status: Int, extras: Bundle?) {}
   override fun onProviderEnabled(provider: String?) {}
   override fun onProviderDisabled(provider: String?) {}
   }
```

4. A myListener regisztrálása

```
manager.requestLocationUpdates(
   LocationManager.GPS_PROVIDER, 0, 0, myListener);
```

GPS adatok szimulálása



További érdekes adatok

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

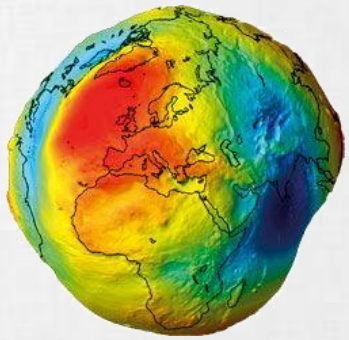
```
locationManager.getLastKnowLocation();
```

Két pont közötti távolság:

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

WGS84 ellipszoid támogatás

Szatellit információk



```
val statusListener = object: GpsStatus.Listener {
     override fun onGpsStatusChanged(event: Int) {
          when (event) {
               GpsStatus.GPS_EVENT_FIRST_FIX -> {}
               GpsStatus.GPS_EVENT_STARTED -> {}
               GpsStatus.GPS_EVENT_STOPPED -> {}
               GpsStatus.GPS_EVENT_SATELLITE_STATUS -> {
                    val status = locManager.getGpsStatus(null)
                    for (sat in status.getSatellites()) { /**/ }
               }
          }
     }
}

locManager.addGpsStatusListener(statusListener);
```

RAW nyersadatok

NMEA: ASCII alapú adatközlés, szabványos mondatok formájában:

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

Gyakorlatban:

```
locationManager.addNmeaListener(
            NmeaListener { t, nmea ->
                    Log.d("NIK", nmea)
              }
        )
```

RAW GNSS mérések:

- 7.0 óta néhány eszköz támogatja a közvetlen hozzáférést
- https://developer.android.com/guide/topics/sensors/gnss.html

Proximity Alert

Közelségi riasztás, != proximity sensor

Jelzés, ha az adott pozícióhoz megadott rádiuszon belülre / 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 fordulat/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 Recognitaion**
  fizikai tevékenységek detektálása, mint például áll, gyalogol, fut, vezet, kerékpár, … + egyéni mozgás minták alkalmazása
- **Geofencig APIs**
  kijelölt területre való belépés és kilépés detektálása