

# Android alkalmazásfejlesztés

Helymeghatározás

- GPS
- Network location

**Sicz-Mesziár János**

sicz.mj@gmail.com

2011. március 24.

OE-NIK



# 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) :

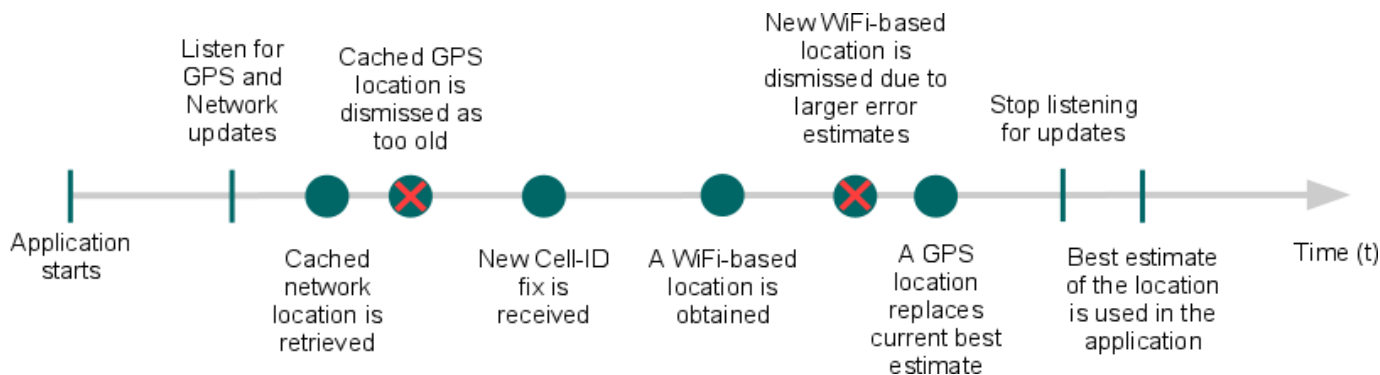
- kevésbé pontos,
- kültéri és beltéri használat,
- gyors információszerzés,
- kevesebb fogyasztás

# 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?

## ◎ 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(...) {}  
};
```

## 4. A myListener regisztrálása

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

# További érdekes adatok

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

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

- ⊙ Két pont közötti távolság:

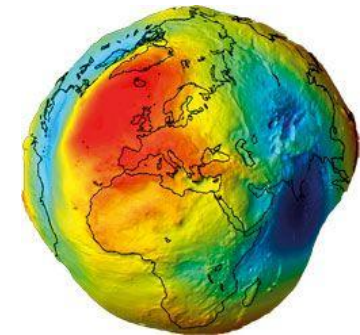
```
Location.distanceTo(Location dest);
```

**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](#)

# 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

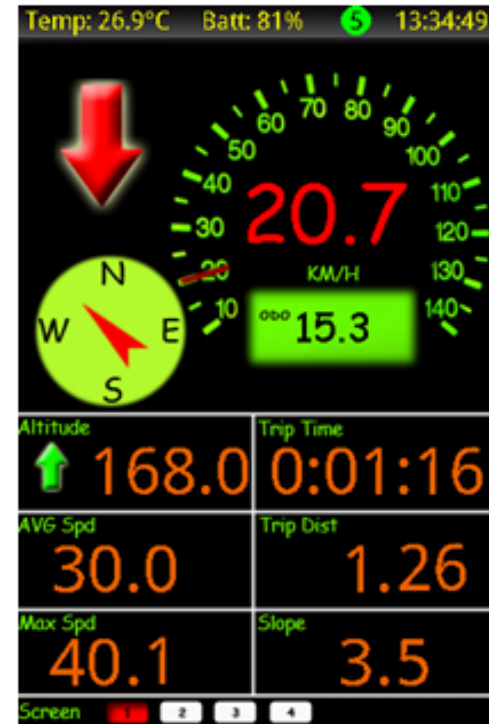
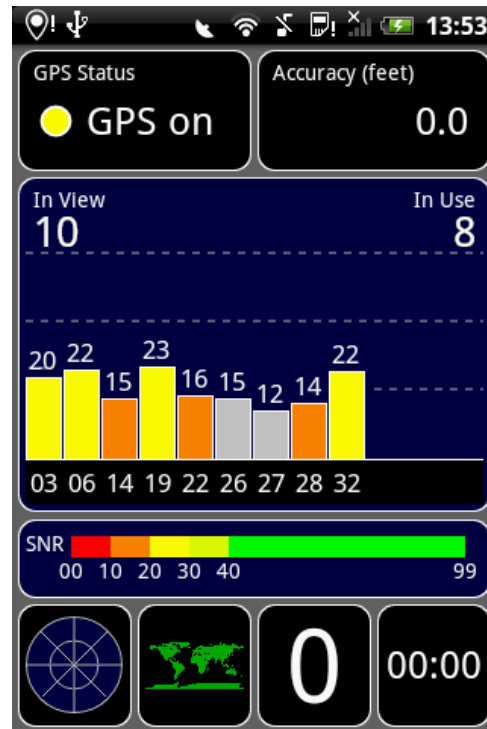
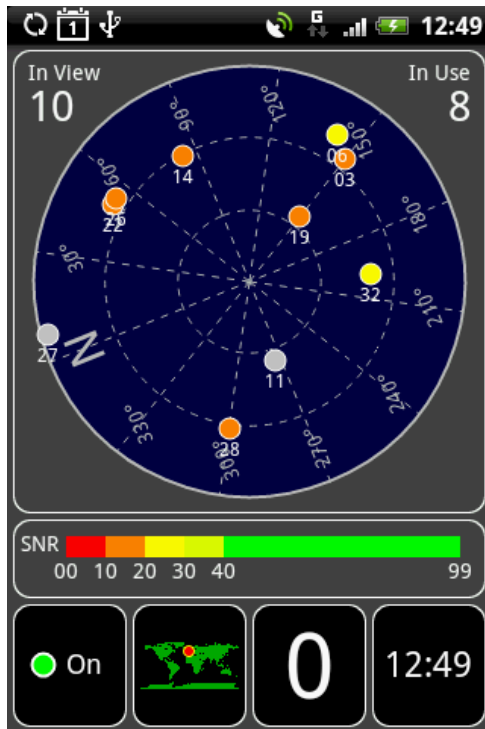
## ◎ CSV – Comma-separated values

## ◎ TCX - Training Center XML

- Garmin szabványa, hasonló, mint a GPX
- Tárol aktív sportolói életben szokásos jellemzőket is, mint: szívverés, kalória, kerékpárnál fordulat/perc, ...

# Jó kis programok kedvcsinálónak

- Legnépszerűbb GPS teszt alkalmazás az Android Marketen



- Berta Norbert szakdoga munkája: GPS-es útvonalelemző kerékpárhoz