

Android alkalmazásfejlesztés

Szenzorok megismerése
Érzékelők használata

Sicz-Mesziár János

sicz.mj@gmail.com

2011. április 14.

OE-NIK



Mi a szenzor?

- ◎ Olyan elem, amely a környezet jellemzőit méri és arról információval szolgál, valamilyen fizikai, kémiai, biológia változásra érzékeny eszköz.

◎ Típusai

■ Passzív

Csakis a környezet fizikai jellemzőinek megváltozására épít.

Pl.: hő-, fény-, nyomásszenzor

■ Aktív

A méréshez valamilyen segédjelet bocsájt ki. (fény, rádióhullám, ...)

Pl.: közelségérzékelő, kapacitív érintőképernyő

Fontosabb paraméterek:

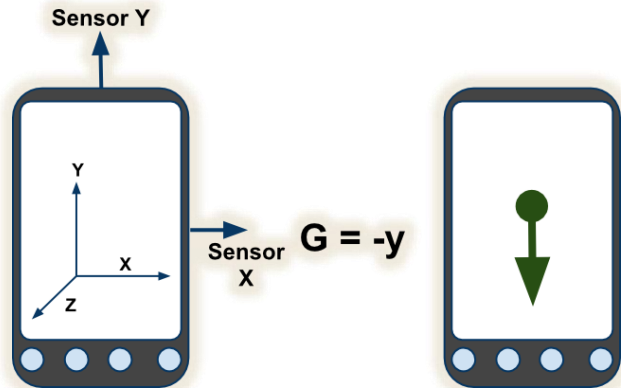
- Érzékenység
- Méréstartomány
- Jel/zaj viszony
- Adaptációs képesség
- Karakterisztika



Accelerometer

[Szenzor teszt videó](#) 😊

- ◎ Gyorsulásmérő, tipikusan egy BMA150-es szenzor
- ◎ Android készülékekben 3-tengelyű gyorsulásmérő



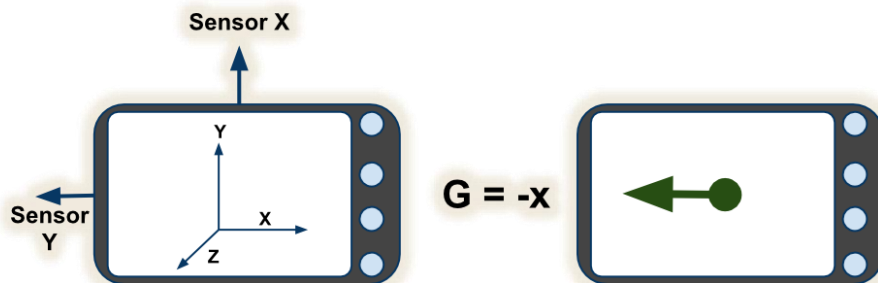
Mértékegysége: m/s^2

$G = -9.81 m/s^2$

- ◎ [SensorManager.remapCoordinateSystem\(\)](#)

Segítségével a koordináták transzformálhatóak.

Pl.: telefon megdöntése miatt (landscape)

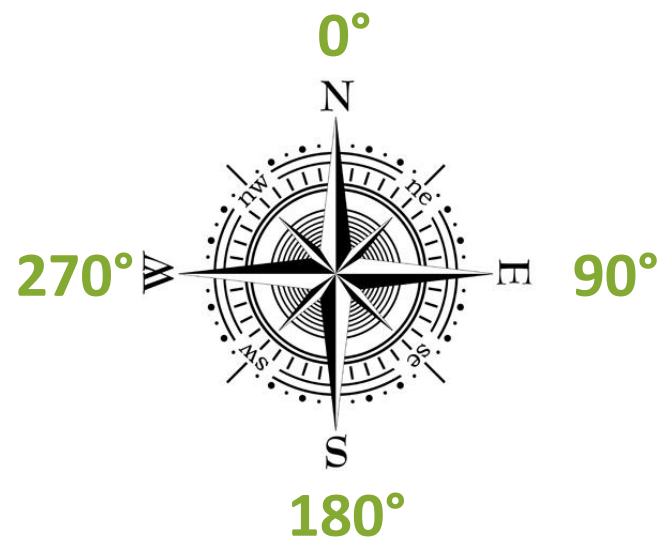
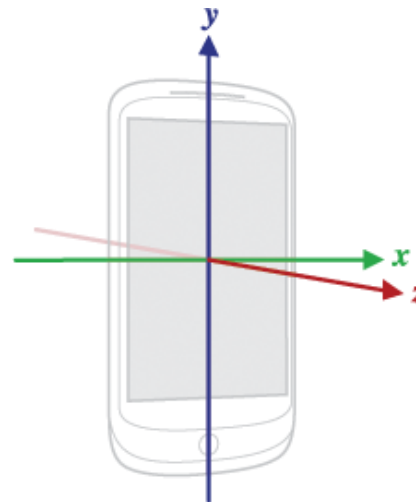


Orientation sensor

- ◎ Irány szenzor
- ◎ Mértékegység fokban értendő
- ◎ Digitális iránytű
- ◎ X, Y és Z tengely:

- values[0]: Azimuth (0 - 359):
 - 0 = Észak
 - 90 = Kelet
 - 180 = Dél
 - 270 = Nyugat
- values[1]: Pitch (-180 – +180)
- values[2]: Roll (-90 – +90)

- ◎ Az értékek ugyanúgy remap...() segítségével transzformálhatóak



Magneticfield

- ⦿ X, Y és Z tengelyen mért mágneses mező
- ⦿ Jellemzően AK8973-as szenzorral
- ⦿ Mértékegység: micro-Tesla (uT)
- ⦿ Fémekre, mágneses és elektromágneses zajokra érzékeny

Proximity

- ⦿ Közelség érzékelő, aktív szenzor – segédfényt használ
- ⦿ Működése infraleddel:



[Demonstráló videó](#)

- ⦿ Bináris adat vagy mért távolság centiméterben
- ⦿ `Sensor.getMaximumRange\(\)` –el lekérdezhető a maximális értékhatár

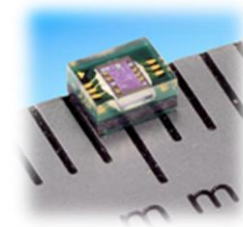
Még néhány szenzor

◎ Temperature

- Hőmérséklet érzékelése
- Nem mindegyik készülékben, de 1 hőszenzor az akkumulátornál mindig szokott lenni 😊

◎ Light

- Fényszenzor, **SI lux** mértékegységben értendő
- Összes Android okostelefonnál elérhető



◎ Pressure

- Légnyomásmérő
- `SensorManager.getAltitude()` : légnyomás alapján magasság

◎ Gyroscope

- Pörgettyű
- Mértékegység: radian / sec

Szenzorok elérése gyakorlatban

◎ Jogosultság kérése ebben az esetben nem szükséges.

◎ SensorManager példányosítása:

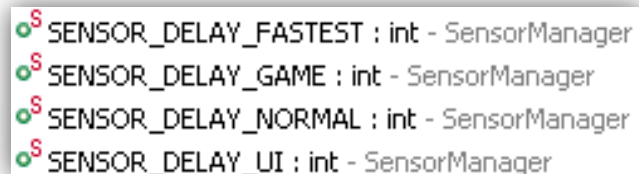
```
SensorManager manager =  
    (SensorManager) getSystemService (SENSOR_SERVICE) ;
```

◎ 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

Emulátorban

◎ Szenzor szimulátor letöltése [OpenIntents oldaláról](#)

◎ bin/SensorSimulatorSettings-x.x.x.apk telepítése emulátoron:

```
adb install SensorSimulatorSettings-x.x.x.apk
```

◎ Standard Java alkalmazás futtatása PC-n:

```
bin/sensorsimulator-x.x.x.jar
```

◎ Telepített APK alkalmazásban IP cím megadása



◎ Külső JAR fájl hozzáadása az Android projekthez

◎ Eredeti kód cseréje alábbiak szerint, egyébként ugyanaz:

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



```
mSensorManager = SensorManagerSimulator.getSystemService (this,  
    SENSOR_SERVICE) ;
```

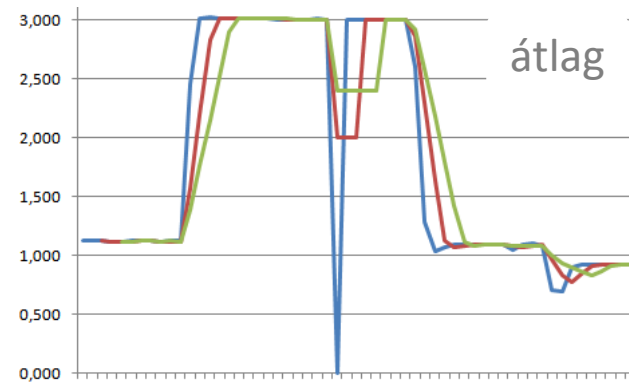
```
mSensorManager.connectSimulator () ;
```

Szűrők alkalmazása

Cél: hibák, pontatlanságok kiszűrése

◎ Átlagolás:

- Páratlan számú mérési adat átlagolása
- 1-1 jelentős érték (pl.: mérés-hiba) befolyásolja
- Ezen javít a legkisebb és legnagyobb értékek eldobása

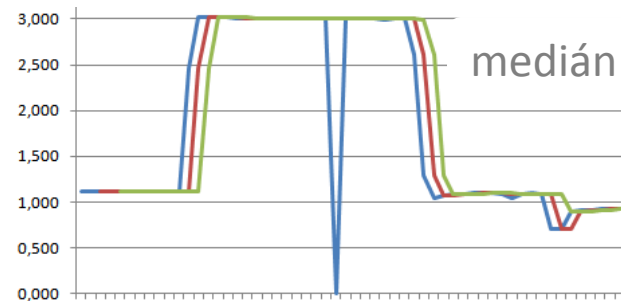


◎ Küszöb

- Meghatározott érték alatt, felett nem veszem figyelembe a kapott adatokat

◎ Medián

- Zajszűrésre egy jó módszer
- Rendelkezésre álló adatokat sorba rendezem és veszem a középső elemet



◎ Csúszóablakos megoldás (alkalmazkodó)

- Fix nagyságú ablak adataival dolgozom, pl.: utolsó 5 adat

Választható feladatok! 😊



E-Dobókocka | Interaktív irányítású kígyós játék

Szenzorok kezelése

E-Dobókocka

- ⦿ Készíts egy interaktív dobókockát, mely a telefonkészülék rázására „dob” egy számot 1 és 6 között!
- ⦿ A szám egy véletlen generátor által adott szám legyen! Csak 1 és 6 közötti szám szükséges.
- ⦿ A rázás alatt folyamatosan új szám generálódjon, mind addig, amíg a rázás véget nem ért!
- ⦿ Használj szűrőt, hogy kicsit „rázósabb” körülmények között is használható legyen!
 - Pl.: vonaton utazva 😊
- ⦿ Szám kiírása helyett legyen valami látványos grafikai megjelenése.
 - Pl.: számlapképek cseréje, véletlen elforgatása, ...

Kígyós játék interaktív irányítással

- ⦿ A klasszikus Snake kígyós játék megvalósítása, melynek irányítása nem 4 irányú gombokkal történik, hanem a telefon megdöntésével.
- ⦿ Egy egyszerű vonal halad előre a zárt térben, mely irányt változtat a készülék megdöntésével megegyező irányba.
- ⦿ A játék véget ér, ha:
 - a kígyó a saját testébe harap
 - a pálya szélére ért