

Android alkalmazásfejlesztés

Az Android rendszer és a fejlesztői
környezet bemutatása

OE-NIK

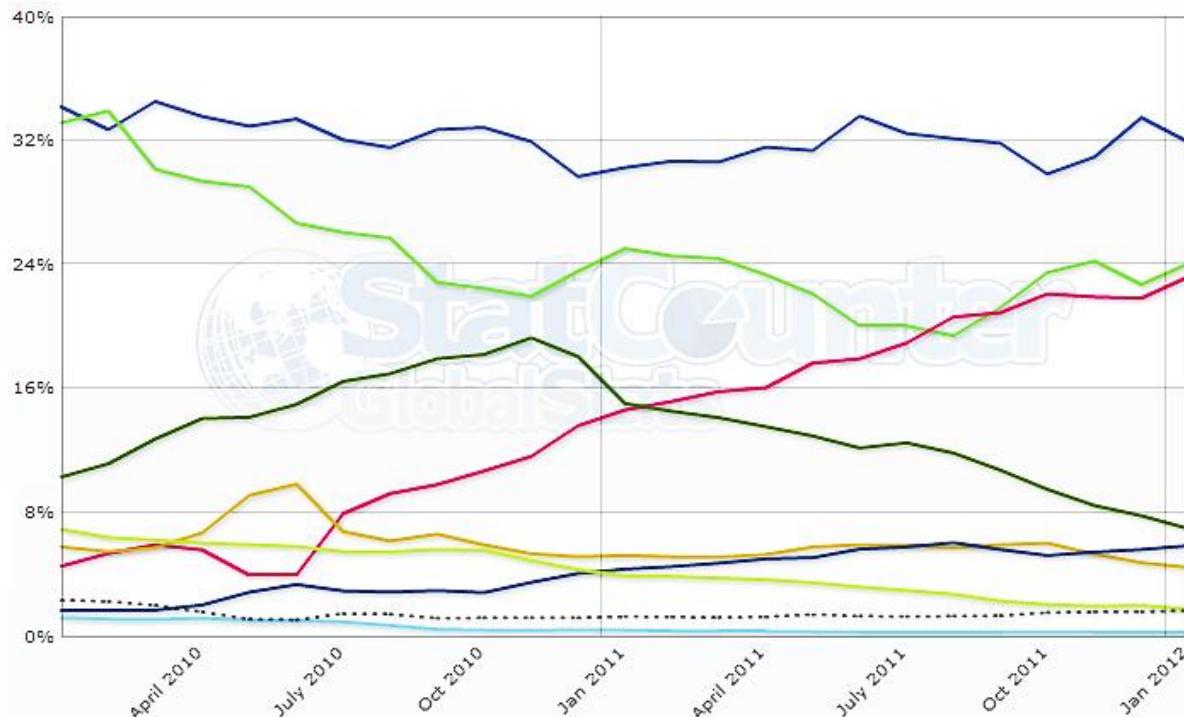
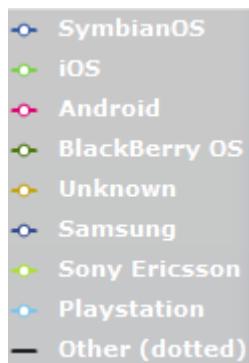
2012. február 13.

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



Az Android története

- Eredetileg az Android Inc. fejlesztése volt
- Google Inc. 2005-ben felvásárolta ☺
- Mára az Android mögött több cég is (pl.: gyártók) megtalálható → Open Handset Alliance (OHA)
- Első elérhető készülék: T-mobile G1 (HTC Dream), 2008 októberében jelent meg, Android 1.0 rendszer (Base)
- Azóta:



Forrás:

http://gs.statcounter.com/#mobile_os-ww-monthly-201001-201201

Android verziók

Api level

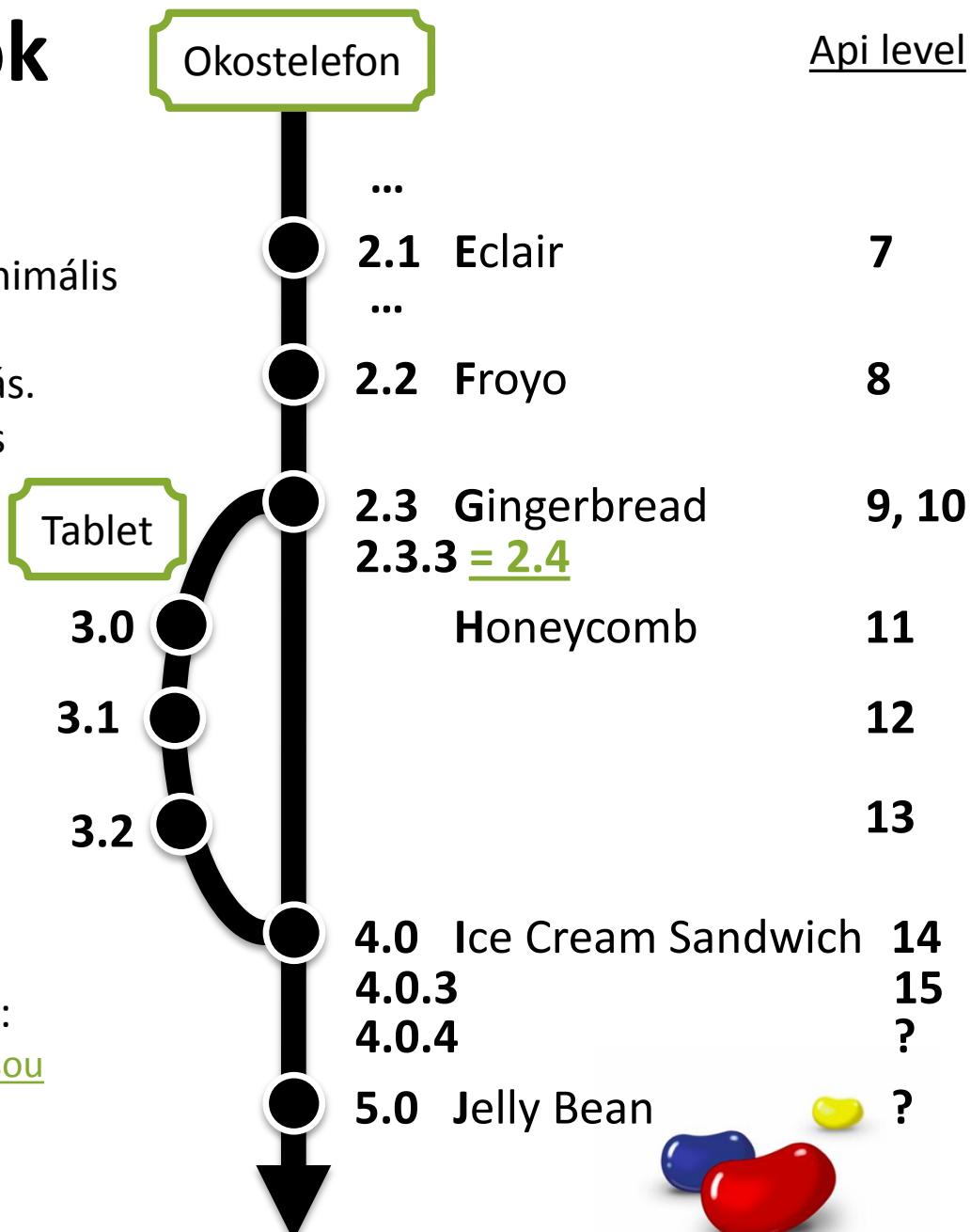
- Folyamatosan növekvő szám
- Fejlesztéskor használjuk a minimális Android verzió megjelölésére
- API level alapján eset választás.

Pl.: adott funkció csak bizonyos verzió óta érhető el.

- Megkönyíti a referenciában való tájékozódást (filter):
<http://developer.android.com/reference/packages.html>

Milyen verzióra érdemes fejleszteni?

Két hetente frissülő statisztika:
<http://developer.android.com/resources/dashboard/platform-versions.html>



Amit jó tudni a platformról

- Linux kernelre épülő operációs rendszer
http://developer.android.com/guide/basics/what-is-android.html#os_architecture

Linux 2.6.* kernelre épül, de az Android 4.0-tól már az új 3.0.* kernelre.
- Dalvik VM virtuális gépen menedzselt Java kód futtatása
(!= Sun JVM, és J2ME-vel sem kompatibilis)
- Open Source
- Párhuzamosan futhat több alkalmazás is (nincs megkötés)
- Android Market: egy instant alkalmazáspiac
- Ingyenes és fizetős szoftverek (30%-70%) ill. alkalmazáson belüli vásárlás lehetősége (2.3 óta)
- Offline alkalmazástelepítés (unknow source) megengedett
- A mobilalkalmazás egyetlen fájlból áll (*.apk)
- Fejlesztői környezet is ugyanúgy fejlődik, mint maga a rendszer ([ADT](#))

Programozás

- **SDK - Java (Dalvik VM)**
- **NDK - C/C++ (Natív kód, Java kódból elérhető - JNI)**
- **ADK – Accessory Development Kit (3.1 óta)**
- További kísérletek
 - Script nyelvekkel (SL4A segítségével)
Pl.: Python, Perl, JavaScript, ...
<http://code.google.com/p/android-scripting/>
 - C# (MonoDroid kezdeményezés)
<http://monodroid.net/>

+1 érdekesség:

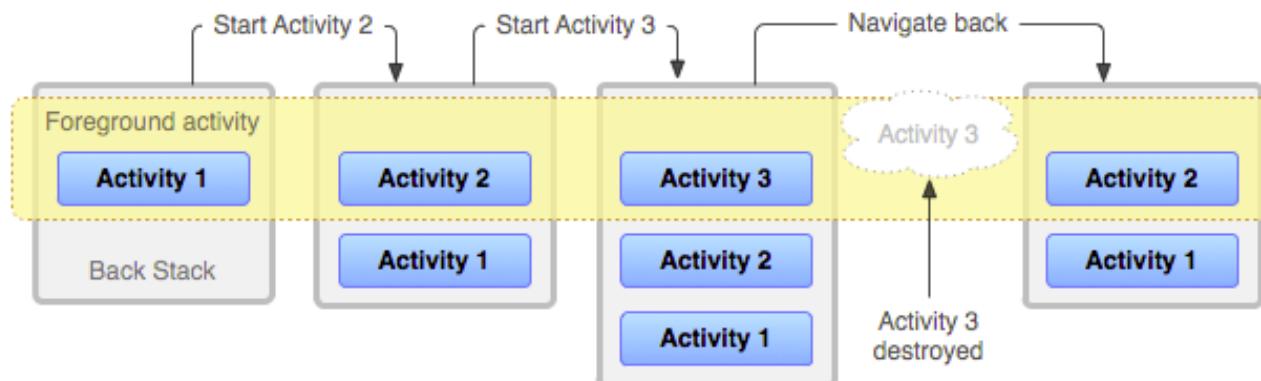
Android app Windows-on

Fejlesztés

- Hivatalosan támogatott IDE: Eclipse
- Az SDK rendelkezik emulátorral (kicsit lassú, időnként problémás – szenzorok?)
- Futtathatunk közvetlen telefonon (gyors, kényelmes – de kell készülék)
- Bibliánk: <http://developer.android.com/>

Android alkalmazásokról

- Az alkalmazások egyenrangúak
- minden programot és képernyőt egy stack-be rögzít
- Vissza (Back) gomb hatására az előző elemre vált át
(Gyakorlatilag, mint böngészőben az előzmények)
- Az alkalmazások (hivatalosan) nem bezárhatóak, ezt maga a rendszer végzi, ha szükség van rá. (Pl.: kevés memória)
- Nem ablakok, hanem teljes képernyőt kitöltő Activity-k
- Egy alkalmazás több Activity-ből állhat
- minden Activity-nek saját életciklusa van!

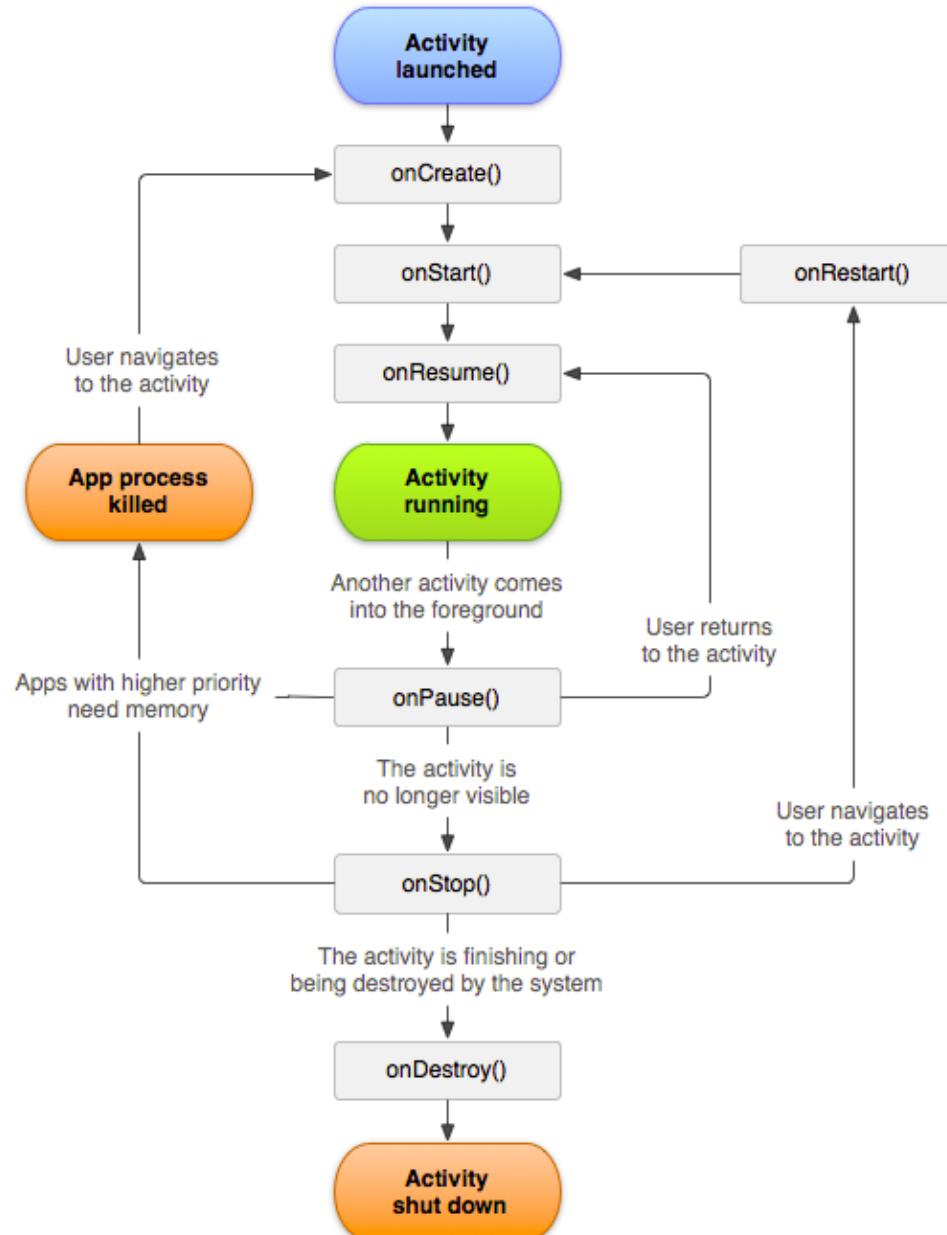


Activity életciklus modell

Célja:

- Váratlan események kezelése.
Pl.: bejövő hívás, akku merülése
- Erőforrások hatékony kezelése

A modellen ábrázolt metódusokat a rendszer automatikusan meghívja, amikor az szükséges.



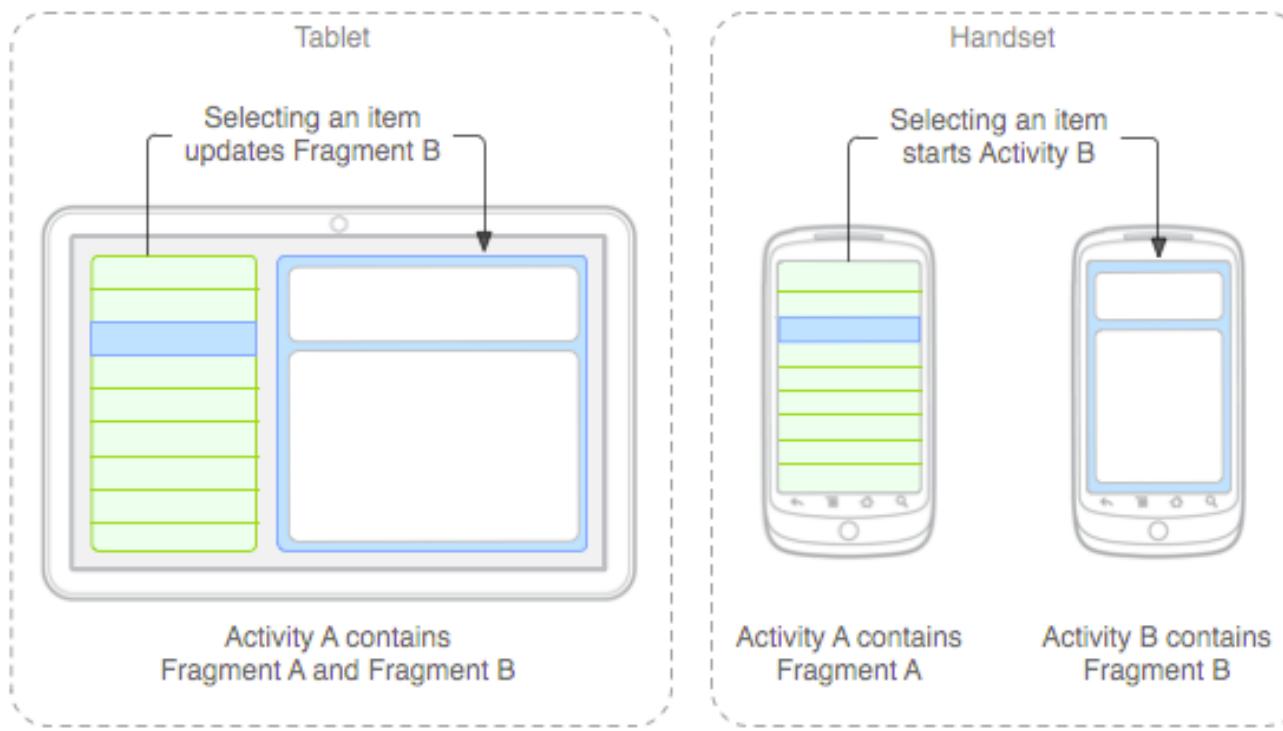
Forrás:

<http://developer.android.com/guide/topics/fundamentals.html#actlife>

Ugyanez másképp:

<http://www.google.hu/images?q=android+activity+lifecycle>

Fragments API



- Tervezési filozófia nagyobb képernyők kezelésére
- Gyakorlati szempontból megadhatjuk mely felületek kerülhetnek egymás mellé.
- Bővebben:
<http://developer.android.com/guide/topics/fundamentals/fragments.html>

Fejlesztői környezet



Fejlesztői környezet telepítése

Mi kell hozzá?

- JDK telepítse
- Eclipse IDE
- Android SDK
- ADT plugin (ADT = Android Development Tools)

Hogyan?

● Windows:

[http://szyrt.hu/downloads/Docs/Android_fejleszto_kornyezet_kialkitasa
Windowson.pdf](http://szyrt.hu/downloads/Docs/Android_fejleszto_kornyezet_kialkitasa_Windowson.pdf)

● Ubuntu 10.04 és felette:

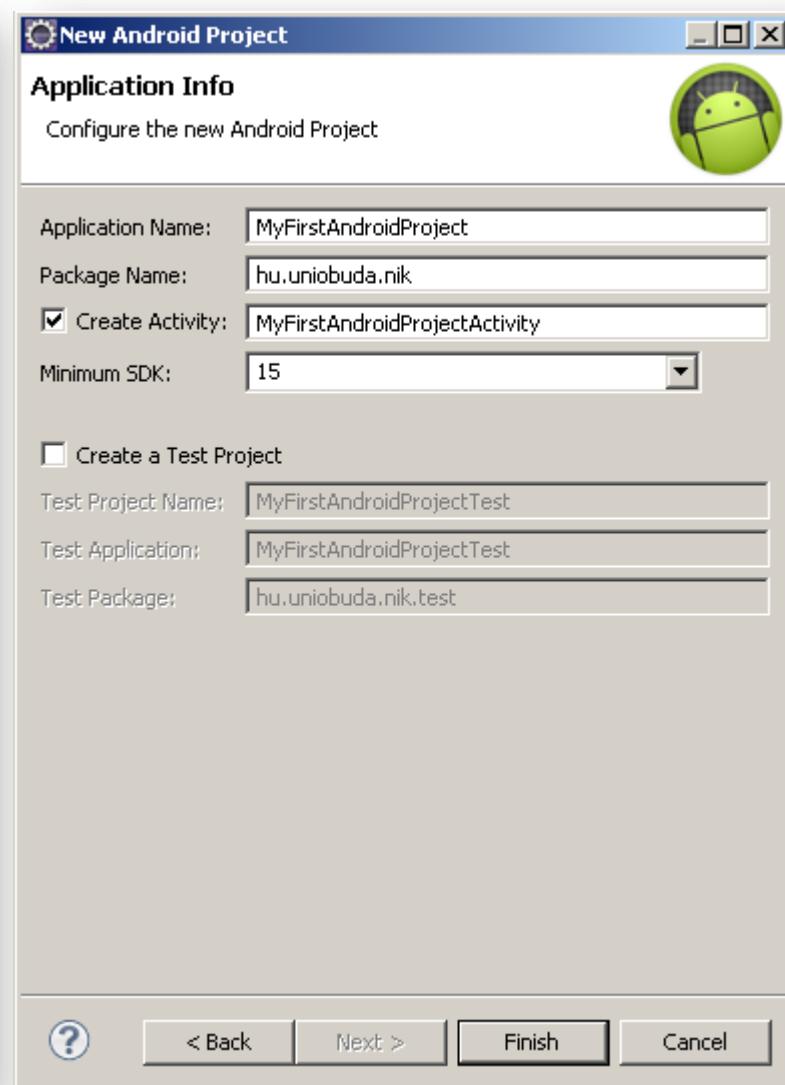
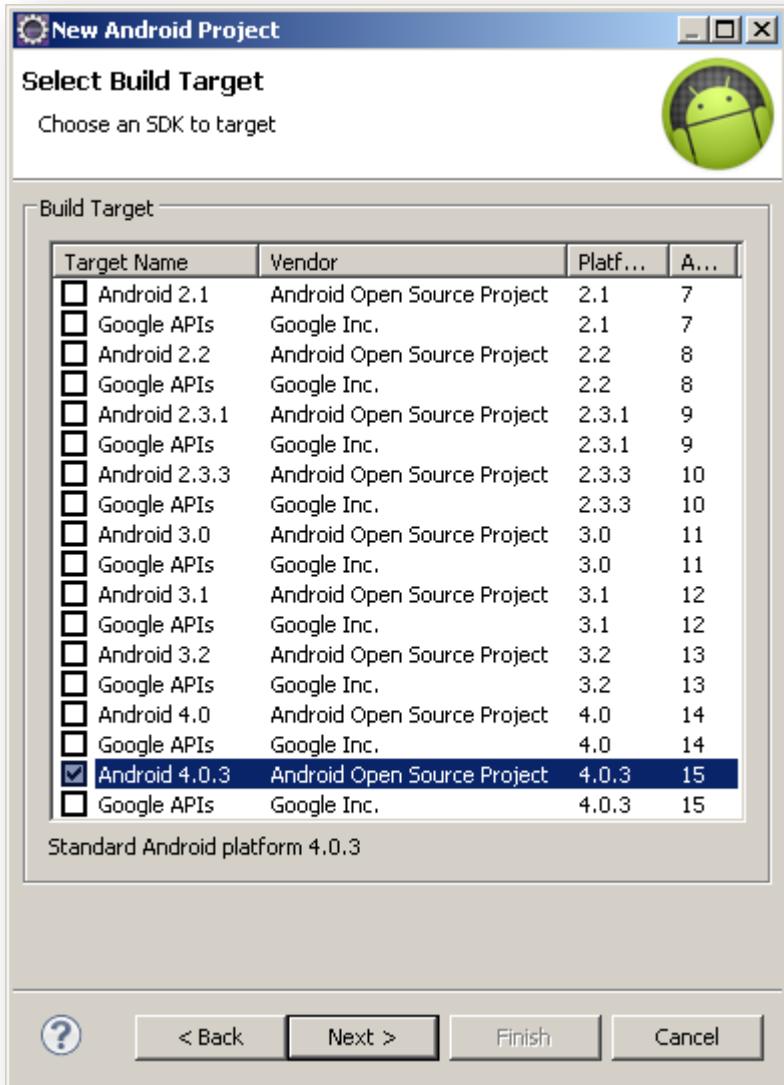
[http://szyrt.hu/downloads/Docs/Android_fejleszto_kornyezet_kialkitasa
Ubuntu 10.04_en.pdf](http://szyrt.hu/downloads/Docs/Android_fejleszto_kornyezet_kialkitasa_Ubuntu_10.04_en.pdf)

● Eredeti leírás:

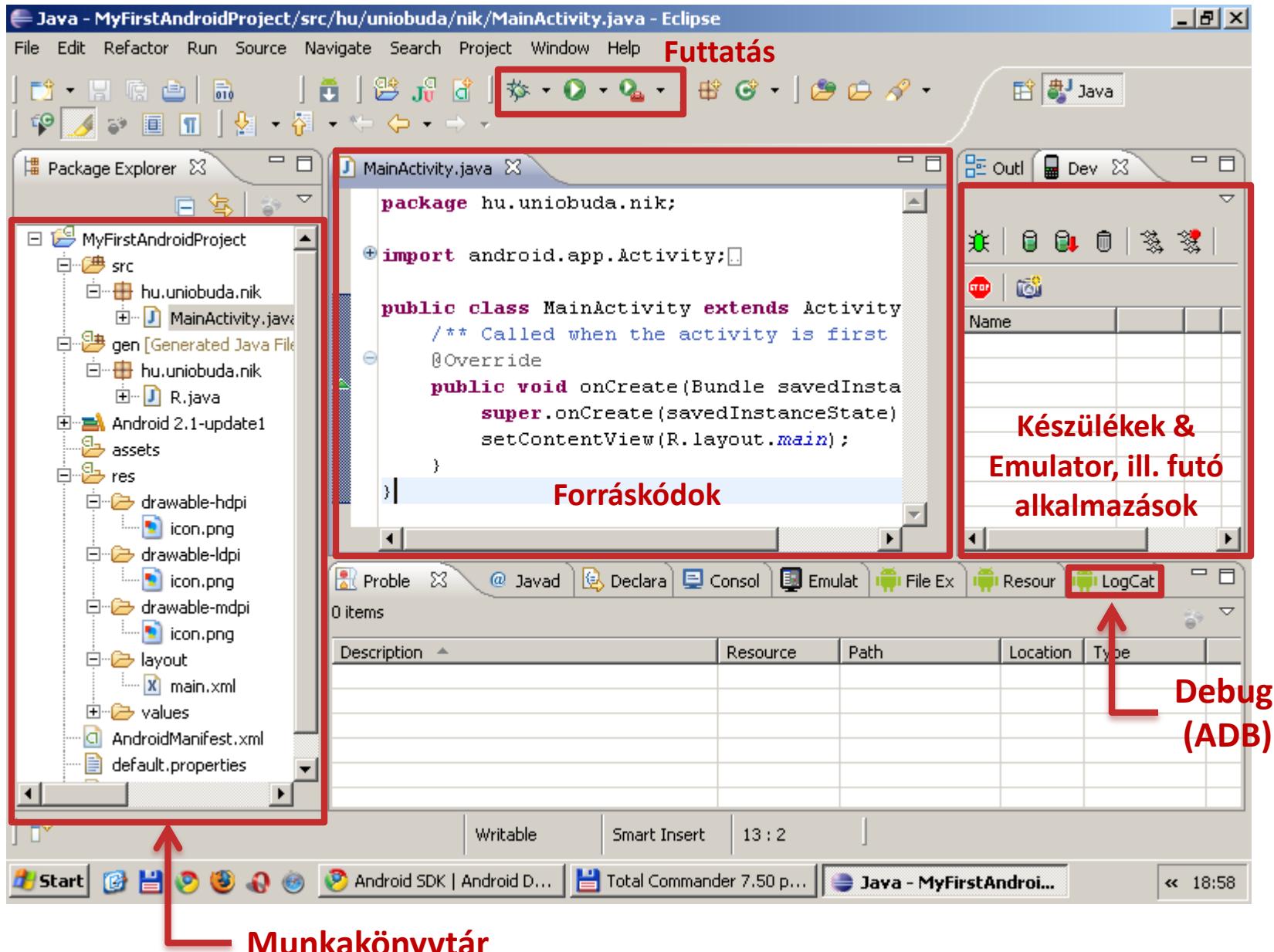
<http://developer.android.com/sdk/installing.html>

Új Android projekt

- File > New > Other... > Android > Android project



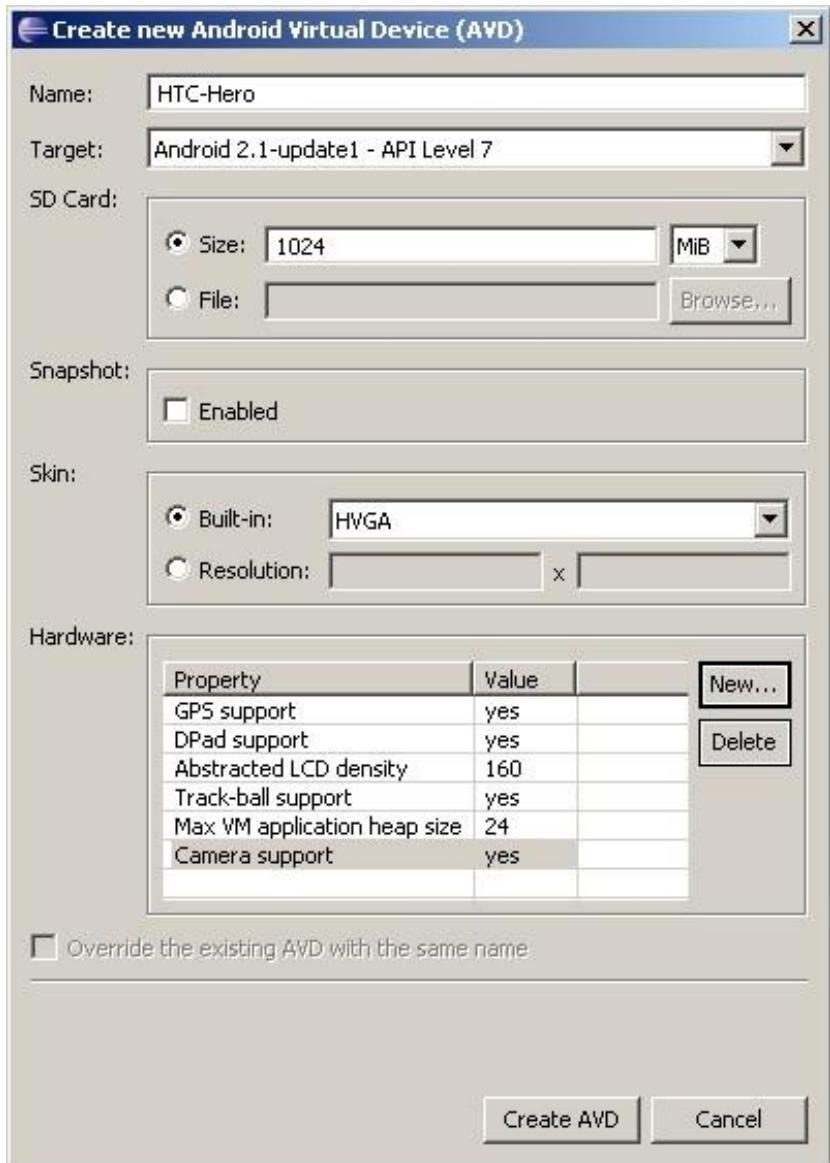
Ismerkedés a felülettel



A munkakönyvtár szerkezete

src/	Az Activity Java fájljai és minden egyéb Java fájl.
gen/	ADT által generált R.java fájl helye. Segítségével érhetjük el Java kódban az res/ könyvtárban található tartalmakat.
assets/	Alapértelmezetten üres könyvtár. RAW (nyers adat) tartalmakat tárolhatunk itt.
res/	Különböző források, melyeket beépítenénk az alkalmazásba.
res/drawable/	Képek és rajzok. Pl.: PNG, JPG, GIF, Ikonok, Drawable-t leíró XML-ek (selector, shape, ...)
res/layout/	Activity felületét, layout-okat leíró XML fájlok.
res/menu/	Menü felépítését leíró XML-ek. Pl.: Options menü
res/values/	Különböző értékek, szövegek, számok, színek, tömbök tárolása XML formában. Pl.: nyelvi fájlok
res/xml/	További XML-ek, pl.: Beállítás felületének leírása
AndroidManifest.xml	Fontos információk, használati jogok, min. SDK verzió, Activity-k, Service-ok, Broadcast-ek nyílvántartása, stb...

Futtatás: Emulátoron



Új Android virtuális eszköz hozzáadása

Window > Android SDK and AVD Manager > New

Beállíthatjuk

- Hardveres jellemzőket
- Android verziót

Emulátorról tudni érdemes:

- Lassúcska szegény
- New/Edit után, első indításkor mindig lassabb, mint egyébként
- Emulátor is egy virtuális gép
- Nem minden támogat: szenzorok, bluetooth, ...
- Emulátor irányítása billentyűzettel

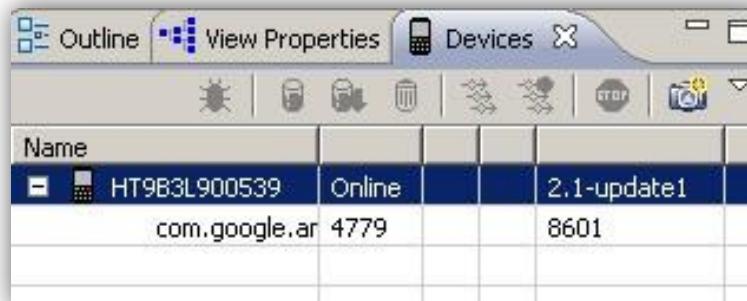
Futtatás: Emulátoron (2)



Futtatás: Közvetlenül telefonon

1. "debuggable," jelző beállítása AndroidManifest.xml-ben!
2. Telefonon USB debugging engedélyezése!
Settings > Applications > Development
3. USB driver letöltése az ADB-nek (Android Debug Bridge):
 - Window > Android SDK and ... > Available packages > Third party ...
 - Vagy OEM Driverek:
<http://developer.android.com/sdk/oem-usb.html>
4. USB kábellel összekötök (**Kábelt nem otthon hagyni!**)
Az ADB telepítésénél meg kell adni a driver helyét.
5. Szükség esetén az Eclipse és/vagy rendszer újraindítása

- Ha sikerült, akkor hasonló:



- Eredeti leírás :
<http://developer.android.com/guide/developing/device.html>

Debug - DDMS

• Dalvik Debug Monitor Server

Eclipse-ben: Window > Open Perspective > Other... > DDMS

Függetlenül: [SDK path] / tools / ddms.bat

Lépései:

1. Breakpoint elhelyezése, mint bármely ismert fejlesztői környezetben.
2. Devices fül alatt a kiválasztott készüléken az adott alkalmazás „debug” módba kapcsolása.



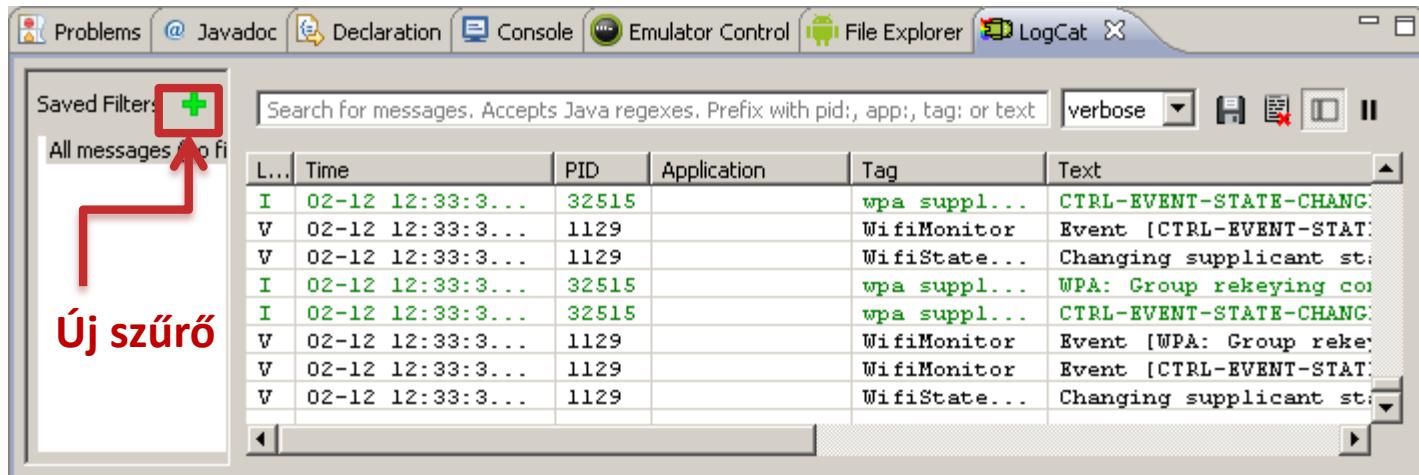
Breakpoint elhelyezve

```
// Irántlós szám vagy max ujj tölmegatás
ujjak_szama = (byte) Math.min(ujjak.length, e)
```

- A debug módban futtatott alkalmazás lassú.
- De utólag is átválthatunk!

Debug(2) - LogCat

- Készülék vagy emulátor kiválasztása a Devices fülön
- LogCat fül alatt listázódik minden ami a rendszerben történik



Ajánlott filter használata:

- Log.d("címke", "Ez történik:" + this.toString());
- Új szűrő:



