

ANDROID ALKALMAZÁSFEJLESZTÉS

Android rendszer és a fejlesztői környezet
megismerése



sicz-mesziar.janos@nik.uni-obuda.hu

Sicz-Mesziár János

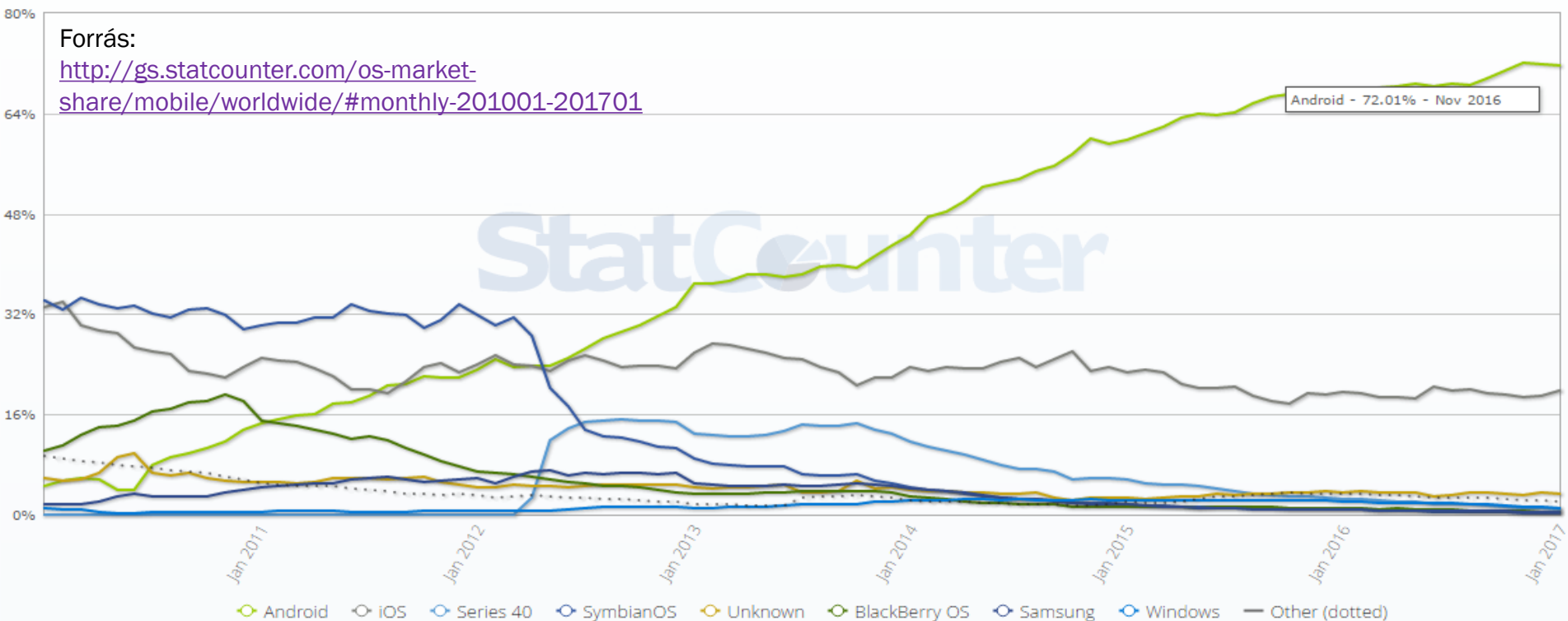
2017. február 15.

Android kezdetekben

- Kezdetben Android Inc. fejlesztése volt, Google 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)

Forrás:

<http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201001-201701>



Android 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 menedzselte Java kód futtatása (!= Sun JVM), 4.4-től ART megjelenik, 5.0-tól ART alapértelmezett

- Open Source

- Párhuzamosan futtat több alkalmazás is (nincs megkötés)

- Google Play Store: egy instant alkalmazás piac

- *Ingyenes és fizetős szoftverek (30%-70%) ill. alkalmazáson belüli vásárlás lehetősége (2.3 óta), előfizetési lehetőségek*

- Offline alkalmazástelepítés (*unknown source*) megengedett

- A mobilalkalmazás egyetlen fájlból áll (*.apk)

- Fejlesztői környezet is ugyanúgy fejlődik, mint maga a rendszer

<http://developer.android.com/tools/studio/index.html>

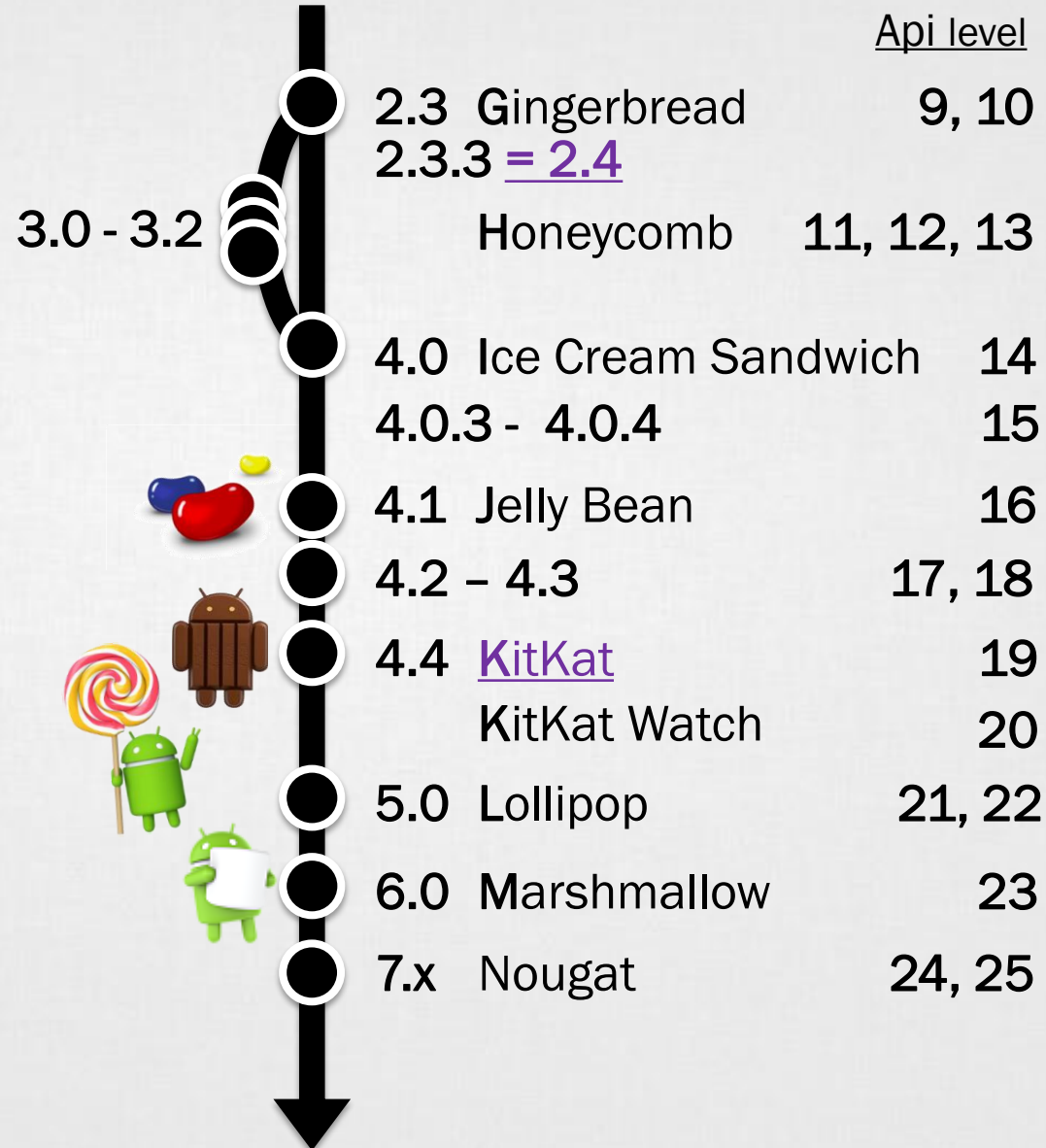
Android verziókról

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

Milyen verzióra érdemes fejleszteni?

<http://developer.android.com/about/dashboards/index.html>



Android alkalmazásfejlesztés

SDK*Dalvik VM, ART***NDK***C/C++, JNI***ADK****Wear****TV****Auto**2014
óta**Eclipse Kepler**

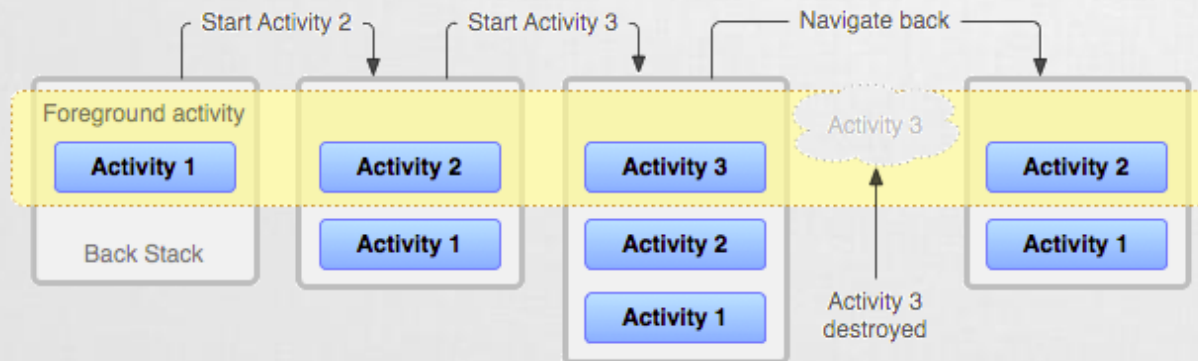
<http://developer.android.com/sdk/index.html>

**Android Studio**

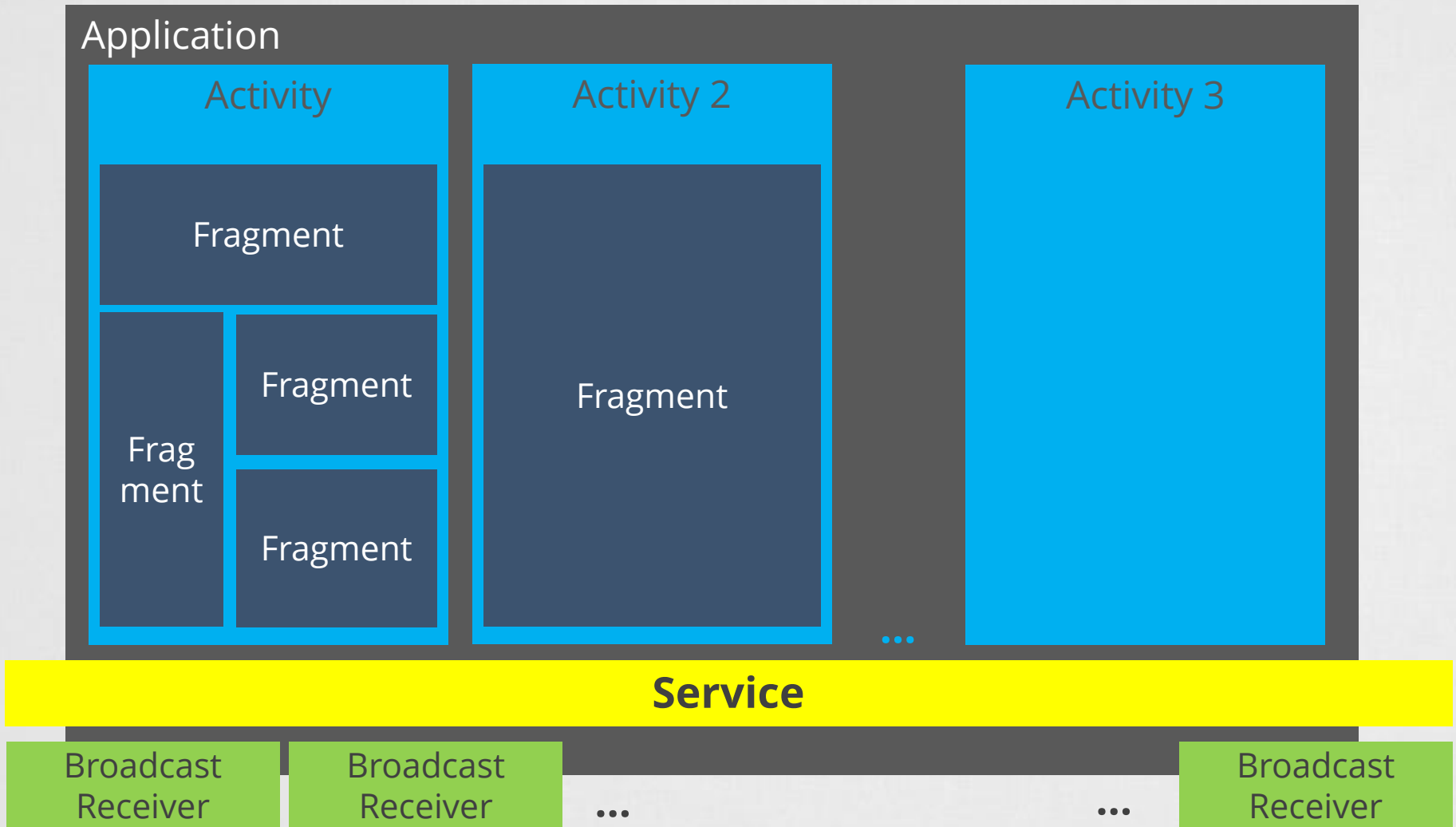
<http://developer.android.com/sdk/installing/studio.html>

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!



Alkalmazások felépítése

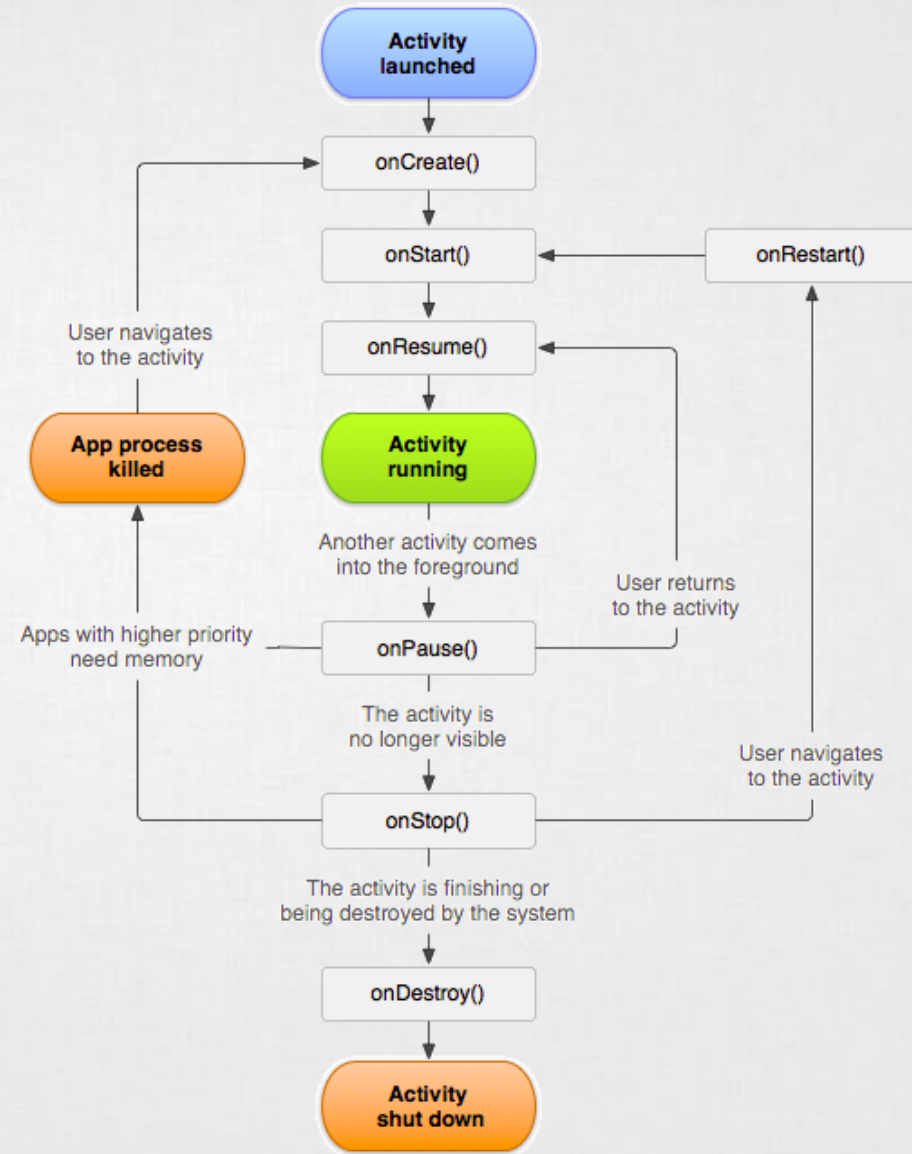


Activity

Célja:

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

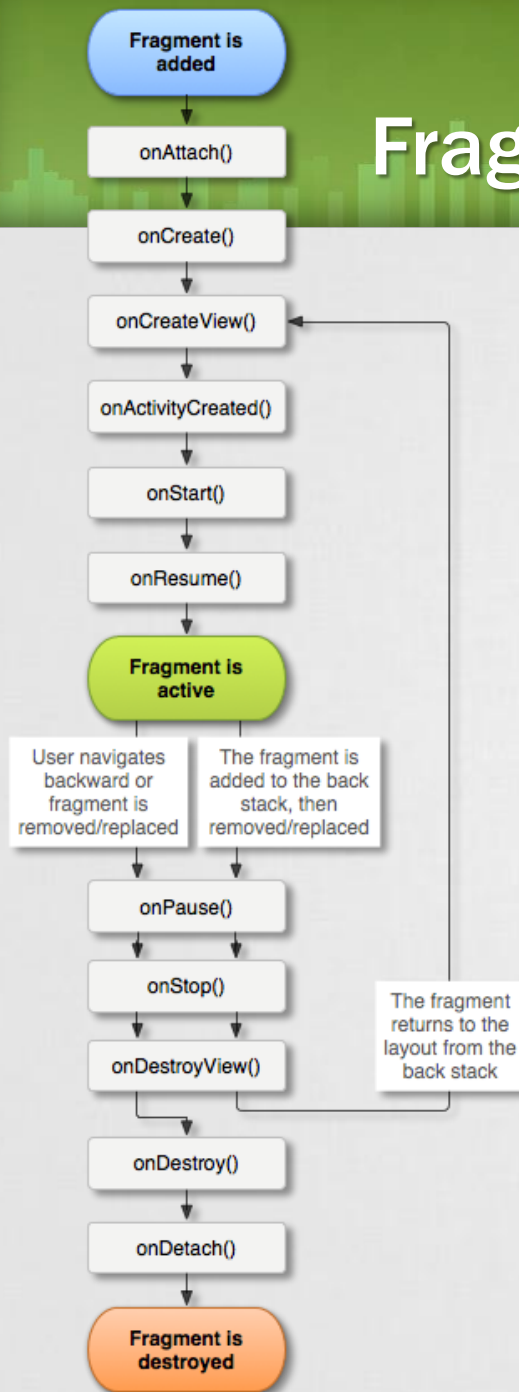
A modellen ábrázolt metódusokat a rendszer automatikusan meghívja, amikor az szükséges.



Forrás:

<http://developer.android.com/reference/android/app/Activity.html>

Fragment



- Tervezési filozófia nagyobb képernyők kezelésére
- Gyakorlati szempontból megadhatjuk mely felületek kerülhetnek egymás mellé.
- Lényegében Activity-n belüli építő elem.
- Életciklus-modell szinkronban az Activity-vel

- Bővebben:

<http://developer.android.com/guide/components/fragments.html>

<http://developer.android.com/reference/android/app/Fragment.html>



Fejlesztői környezet megismerése



Fejlesztő környezet telepítése

Mi kell hozzá?

- JDK telepítése
- Android Studio telepítése
<http://developer.android.com/sdk/index.html>
- Android SDK

Forrás:

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



Fejlesztő környezet bemutatása

The screenshot displays the Android Studio interface with several key components highlighted by red boxes and labels:

- Futtatás (Run):** The Run button (a green play icon) in the top toolbar is highlighted.
- Projekt könyvtár (Project View):** The left sidebar shows the project structure, including folders like `src`, `main`, `java`, `hu`, `index`, `elsoalkalmazas`, and `MainActivity`.
- Forráskódok (Source Code):** The main editor window shows the `MainActivity.java` file with the following code:

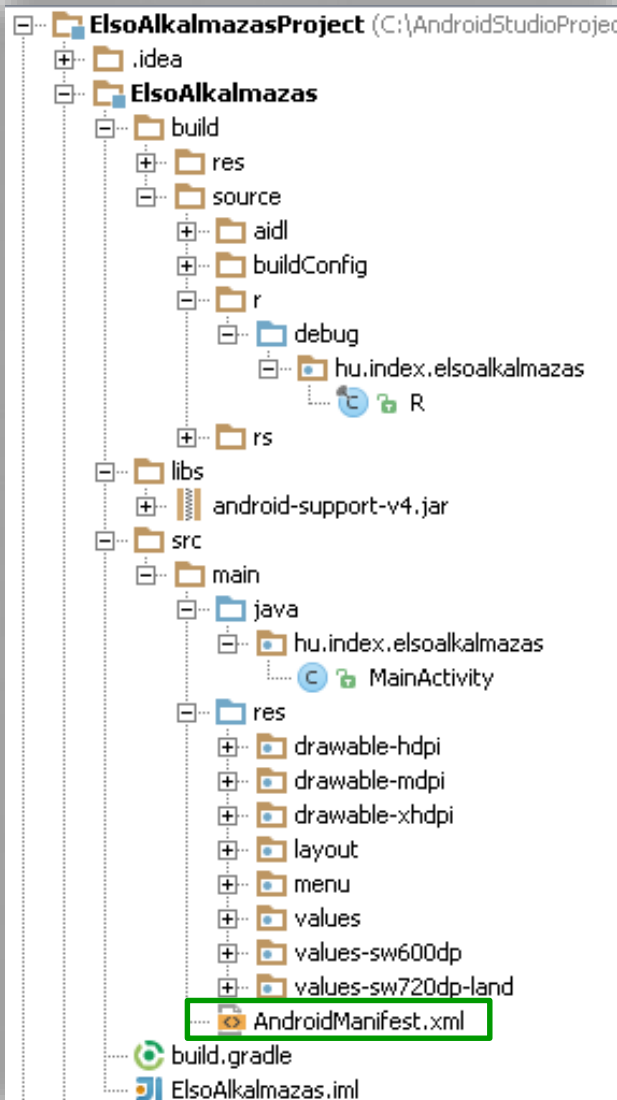

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```
- Eszközök (Tools):** The bottom toolbar shows the device selection dropdown menu, currently set to "LGE LG-P880 Android".
- LogCat - Debug:** The LogCat window at the bottom shows the following log output:


```
06-05 00:53:14.200 293-360/? D/StateMachine: he
06-05 00:53:14.200 293-360/? D/StateMachine: he
06-05 00:53:14.200 293-360/? D/StateMachine: pr
06-05 00:53:14.200 293-360/? D/WifiStateMachine
06-05 00:53:14.220 293-360/? D/StateMachine: he
```

At the bottom of the IDE, a status bar indicates: "Compilation completed successfully in 41 sec (a minute ago)".

Projekt felépítése



AndroidManifest.xml

Fontos információk, használati jogok, min. SDK verzió, Activity-k, Service-ok, Broadcast-ek nyilvántartása, stb...

R.java

Segítségével érhetjük el Java kódban az res/ könyvtárban található tartalmakat.

Libraries

Különböző könyvtárak, library-k helye.
Pl.: *.jar fájlok

Java source

Logikát leíró, Java fájlok, forráskódok helye. Pl.: Activity fájlok

Resources

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

Gradle

A Gradle egy nyílt forráskódú automatizált fordítást támogató eszköz, melynek célja, hogy mint egy karmester összefogja a fordítással járó folyamatokat.



- Gondoskodik a függőségek beszerzéséről (*Dependency Management*)
- Taskokat hajt végre, mely bővíthető
- Plugin-ek írhatóak hozzá (*Groovy, Java*)
- Tesztek futtatásának támogatása
- Deploy támogatása
- Parancssorból is elindítható, pl Android esetén:
`gradlew assembleDebug`
- Akár modulonkénti *.gradle fájlok, mint build script fájlok
- Bővebben:
<http://developer.android.com/tools/building/configuring-gradle.html>

Futtatás: Emulátor

Új Android virtuális eszköz hozzáadása

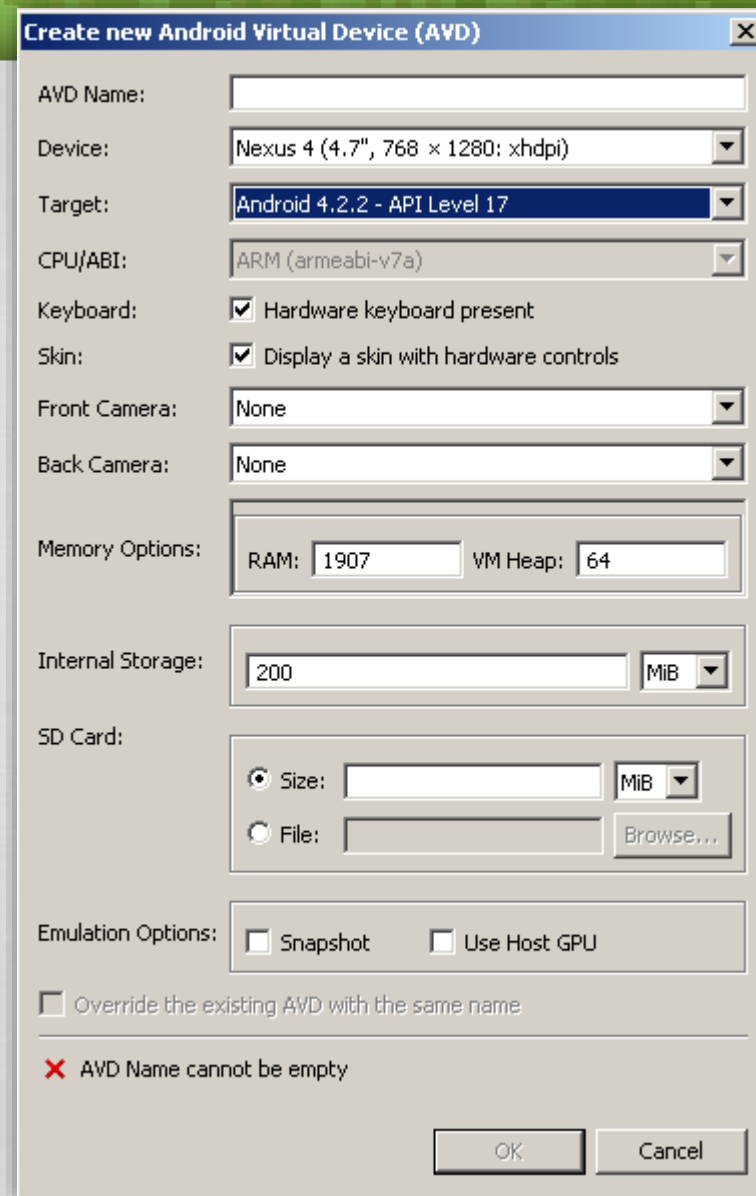
Tools > Android > AVD Manager > New

Beállíthatjuk

- Hardveres jellemzőket
- Android verziót
- Emulációs paraméterek (GPU render)

Emulátorról tudni érdemes:

- Lassúcska szegény
- New/Edit után, első indításkor mindig lassabb, mint egyébként
- Emulátor is egy virtuális gép
- Nem mindent támogat:
szenzorok, bluetooth, ...
- [Emulátor irányítása billentyűzettel](#)



Futtatás: Emulator (2)



Futtatás: Genymotion

- Emulátor alternatíva
- Open Source project
- Egy x86-os Androidot futtató VirtualBox virtuális gép
- <http://www.genymotion.com/>

Miben jobb?

- Gyorsabb, nincs ARM emulálás
- OpenGL és multiscreen támogatás
- Szenzorok szimulálása
- Könnyebb hamis pozíció emulálás

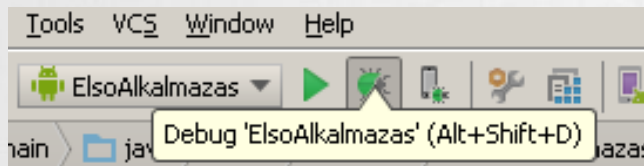


Debug - DDMS

Dalvik Debug Monitor Server

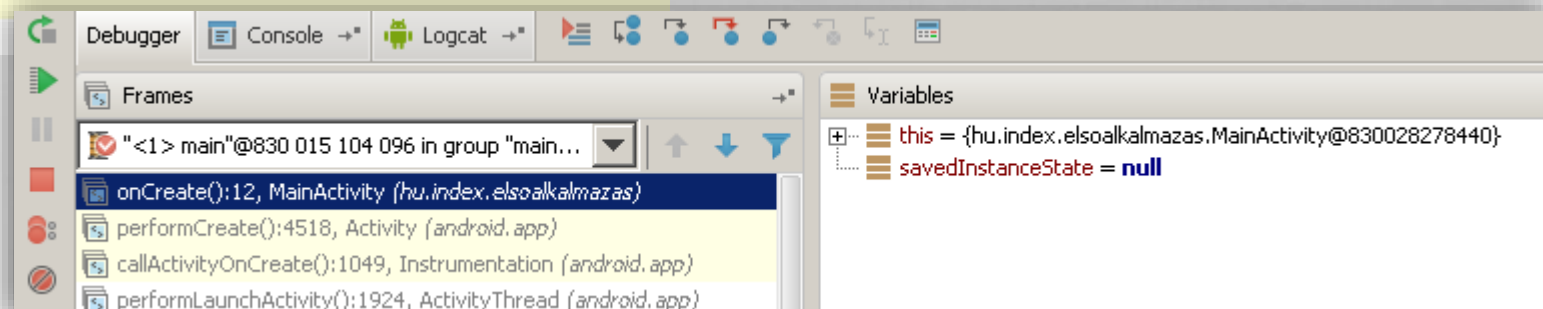
- Függetlenül: [SDK path] / tools / ddms.bat
- IntelliJ: Tools > Android > Monitor (DDMS)

vagy:



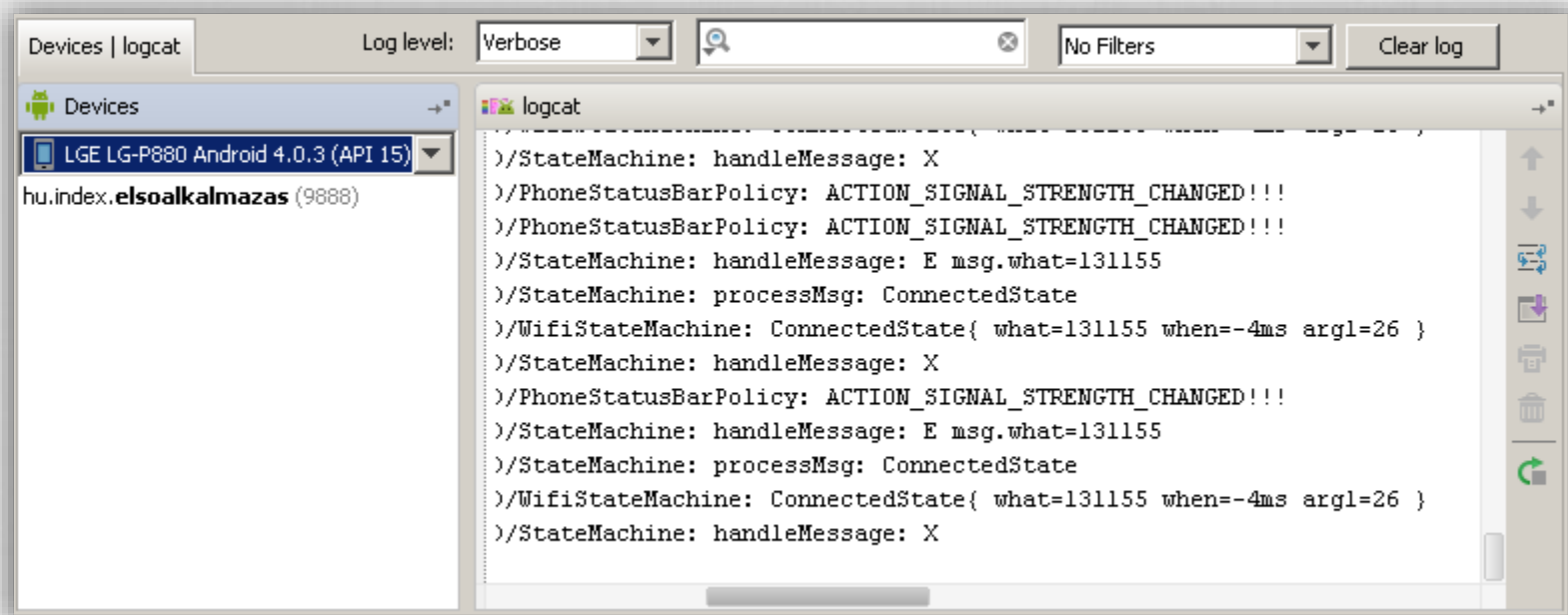
Eredmény:

```
7 public class MainActivity extends Activity {  
8  
9     @Override  
10    protected void onCreate(Bundle savedInstanceState) {  
11        super.onCreate(savedInstanceState);  
12        setContentView(R.layout.activity_main);  
13    }  
14
```



Debug(2) - LogCat

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



Ajánlott filter használata:

⦿ `Log.d("cimke", "Ez történik:" + this.toString());`