

# ANDROID ALKALMAZÁSFEJLESZTÉS

Android rendszer és a fejlesztői környezet  
megismerése



Sicz-Mesziár János

[sicz-mesziar.janos@nik.uni-obuda.hu](mailto:sicz-mesziar.janos@nik.uni-obuda.hu)

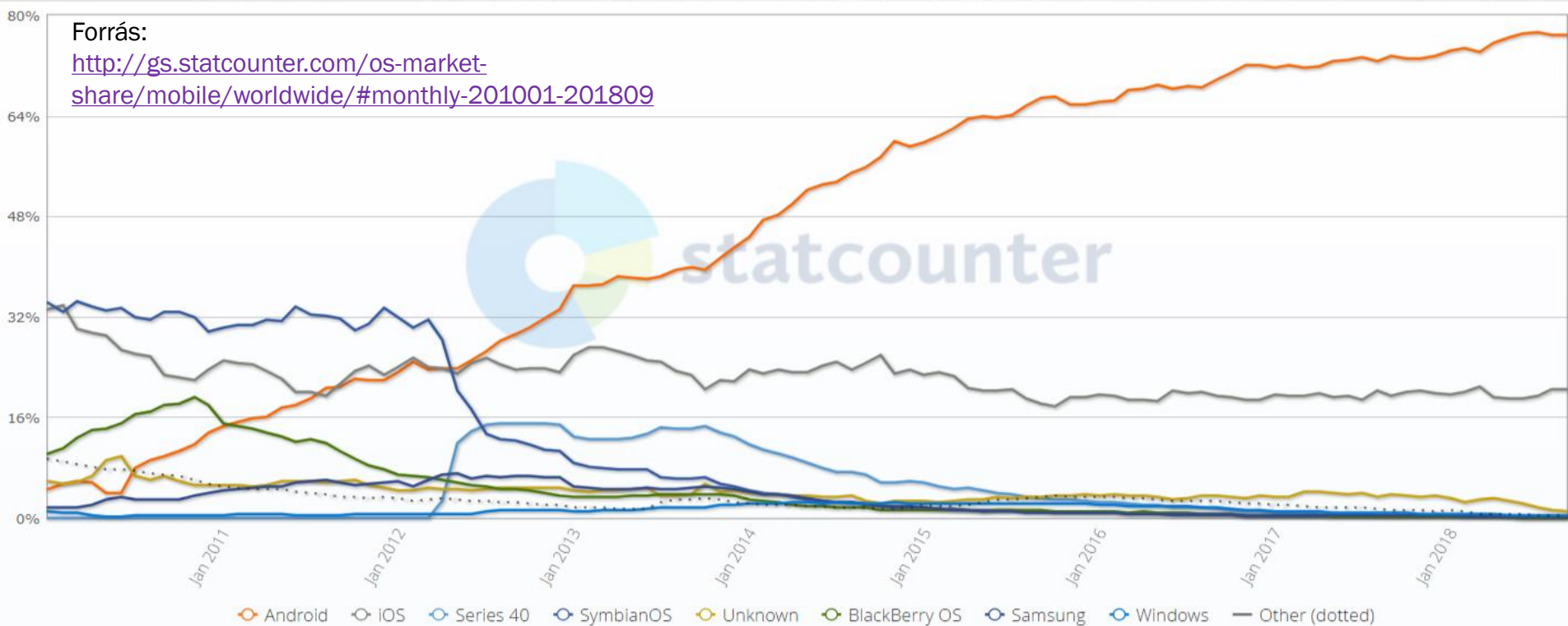
Mezei József

[mezei.jozsef@nik.uni-obuda.hu](mailto:mezei.jozsef@nik.uni-obuda.hu)

2018. szeptember 8.

# 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)



# Android platformról

- Linux kernelre épülő operációs rendszer  
[http://developer.android.com/guide/basics/what-is-android.html#os\\_architecture](http://developer.android.com/guide/basics/what-is-android.html#os_architecture)
- Dalvik VM 1.0 - 4.4,  
ART 4.4-től megjelenik,  
ART 5.0 alapértelmezett
- Open Source: [source.android.com](http://source.android.com)
- Párhuzamosan futhat több alkalmazás is (nincs megkötés)
- Google Play Store: egy instant alkalmazás piac
  - *Ingyenes* alkalmazások
  - *Fizetős* szoftverek (30-70%),
  - *Alkalmazáson belüli* vásárlás lehetősége (30-70%),
  - *Előfizetési* lehetőségek (30-70%, de 12 hónap után 15%-85%)
- Offline alkalmazástelepítés (*unknown source*)
- Mobilalkalmazás egyetlen fájlból áll (*\*.apk*)  
*App Bundle és On-Demand modules*
- Fejlesztői környezet is ugyanúgy fejlődik, mint maga a rendszer  
<https://developer.android.com/studio/intro/>

# 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):  
<https://developer.android.com/reference/packages>

## Milyen verzióra érdemes fejleszteni?

<https://developer.android.com/about/dashboards/>



# Android alkalmazásfejlesztés

**SDK**

Java, Kotlin

**NDK**

C/C++, Assembly

**ADK****Wear****TV****Auto****Things****Eclipse Kepler**

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

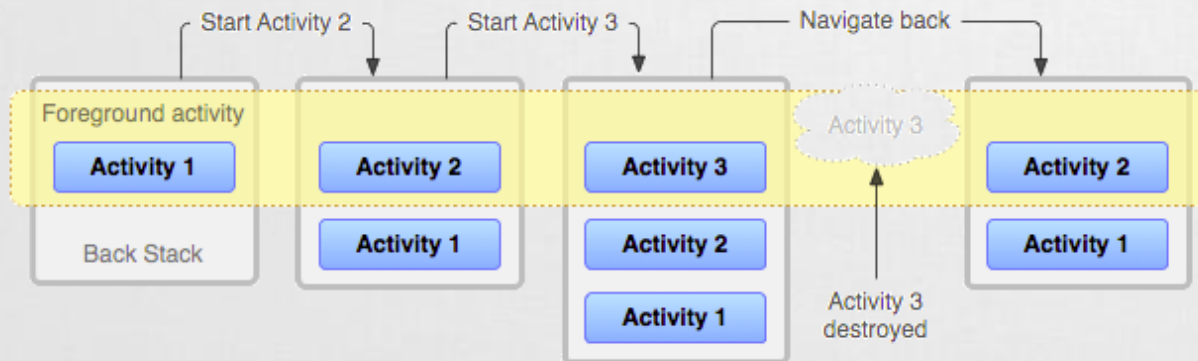
**Android Studio**

<https://developer.android.com/studio/>

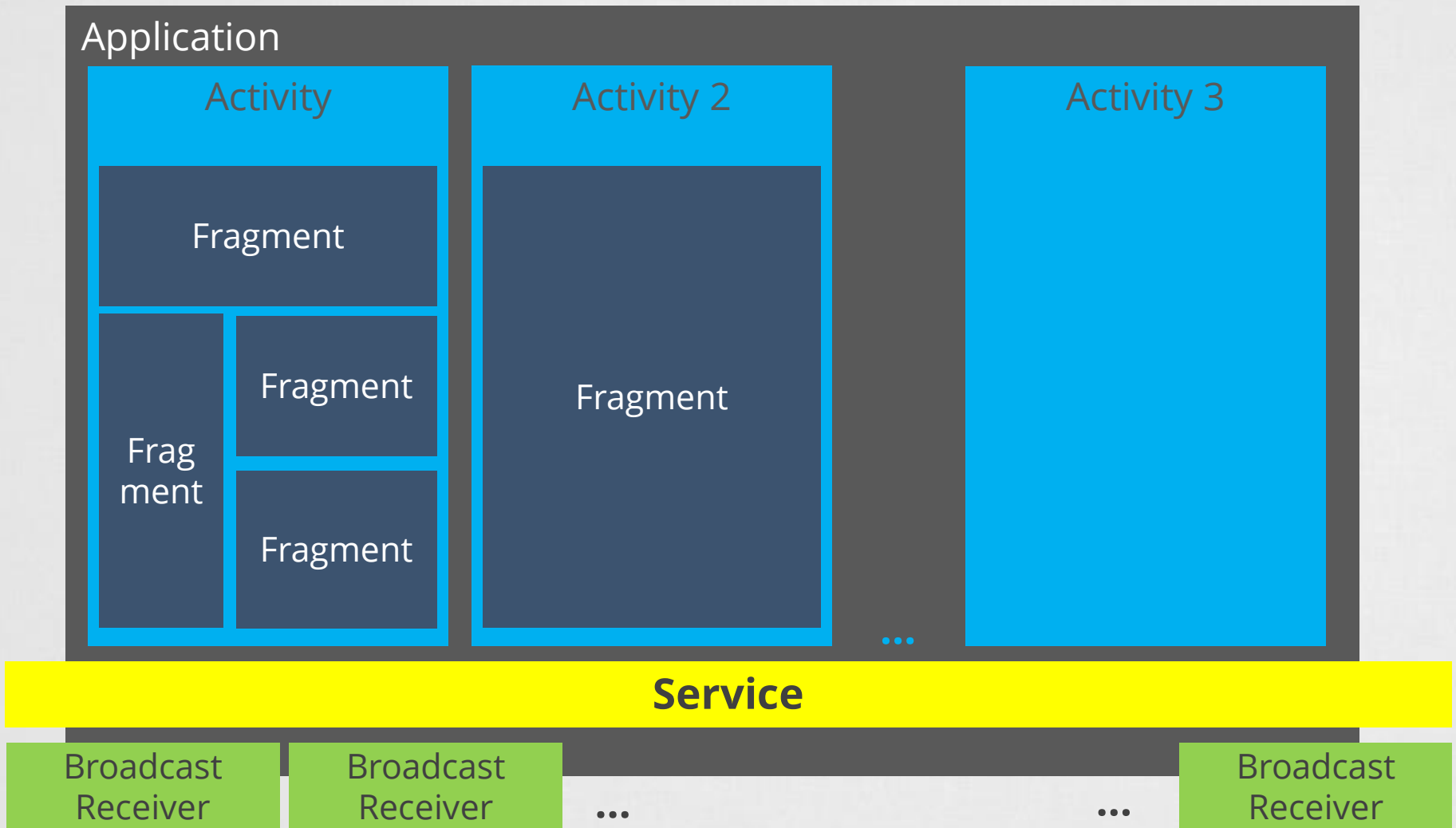


# Android alkalmazásokról

- 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

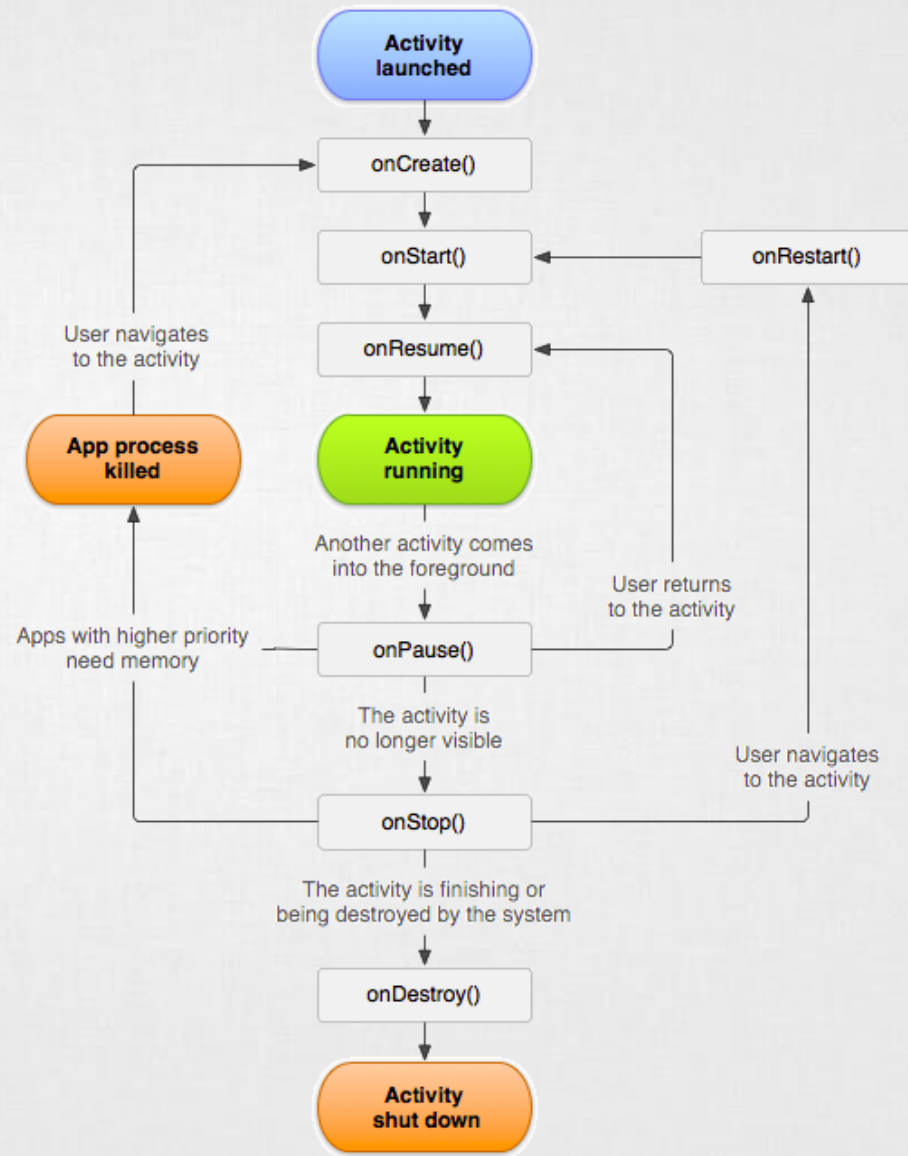


# Activity

## Célja:

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

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

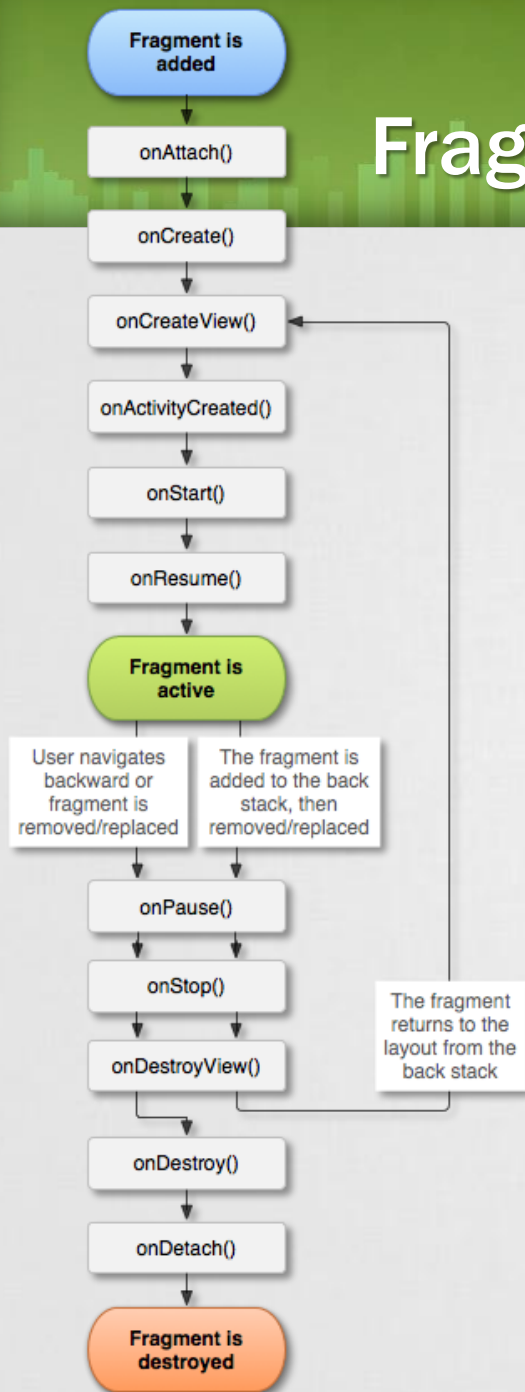


Forrás:

<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?

- Android Studio telepítése  
<https://developer.android.com/studio/>
- Android SDK

Forrás:

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



# Fejlesztő környezet bemutatása

**Futtatás**

**Forráskódok**

**Projekt könyvtár**

**Eszközök**

**LogCat - Debug**

```

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

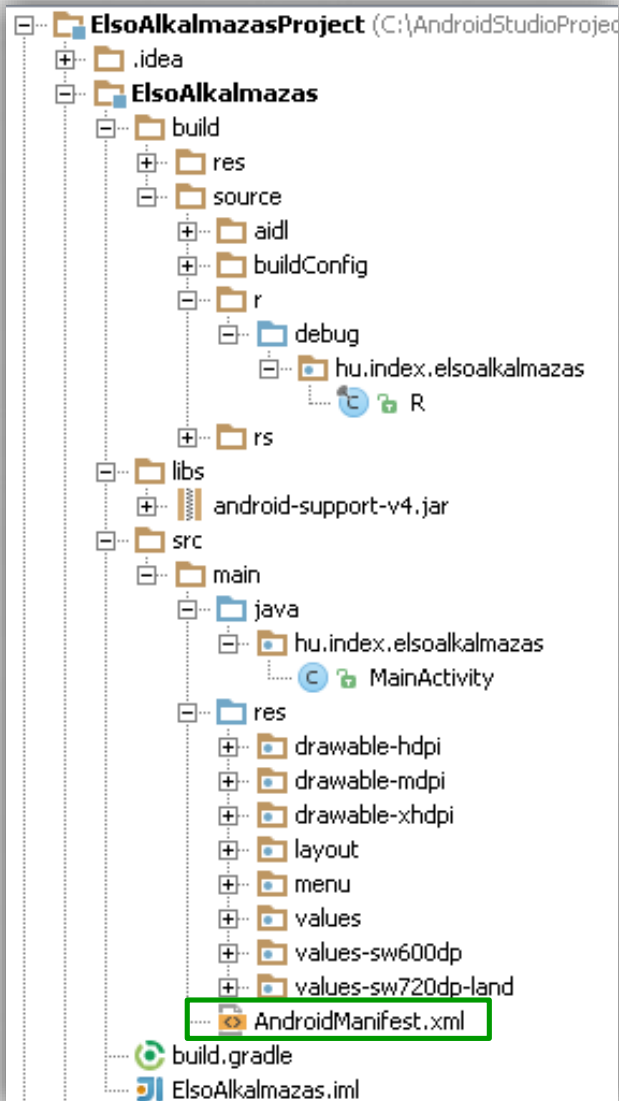
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

```

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

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: Emulator

## Új Android virtuális eszköz hozzáadása

Tools > Android > AVD Manager > New

## Beállíthatjuk

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

## Emulátorról tudni érdemes:

- Emulátor is egy virtuális gép
- x86 image használata ajánlott Intel CPU-s gépeken
- Nem mindent támogat: nfc, bluetooth, ...

Virtual Device Configuration

Select Hardware  
Android Studio

Choose a device definition

Category	Name	Play Store	Size	Resolution	Density
TV	Pixel XL		5,5"	1440x...	560dpi
Wear	Pixel		5,0"	1080x...	xxhdpi
Phone	Nexus S		4,0"	480x8...	hdpi
Tablet	Nexus One		3,7"	480x8...	hdpi
	Nexus 6P		5,7"	1440x...	560dpi
	Nexus 6		5,96"	1440x...	560dpi
	Nexus 5X		5,2"	1080x...	420dpi
	Nexus 5		4,95"	1080x...	xxhdpi
	Nexus 4		4,7"	768x1...	xhdpi
	Galaxy Nexus		4,65"	720x1...	xhdpi
	5.4" FWVGA		5,4"	480x8...	mdpi

**Nexus 5X**

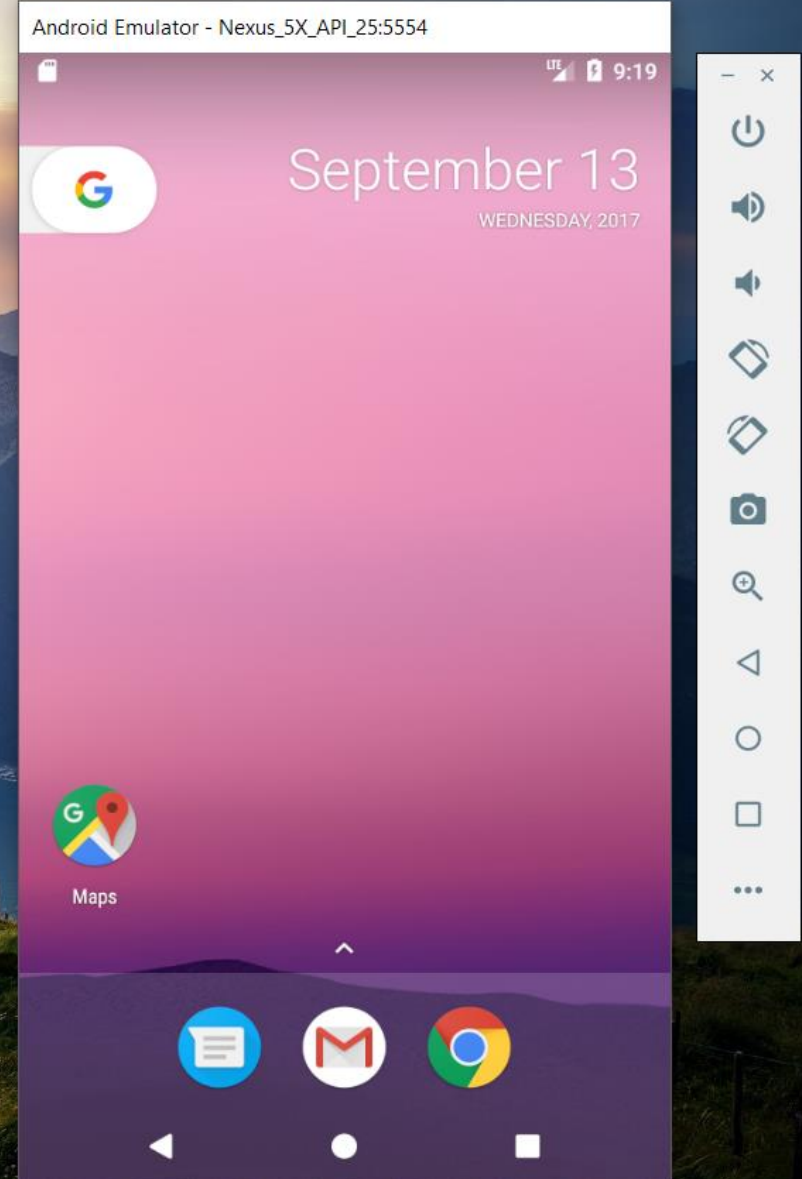
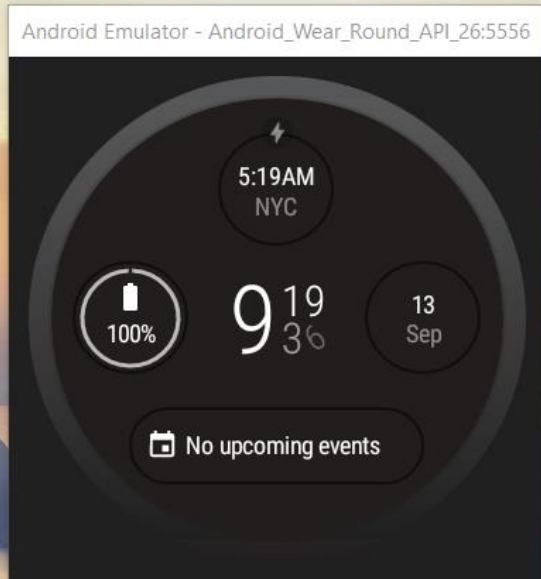
1080px

5.2"

1

New Hardware Profile Import Hardware Profiles

# 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

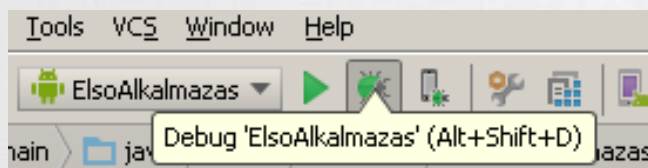


# Debug - DDMS

## Dalvik Debug Monitor Server

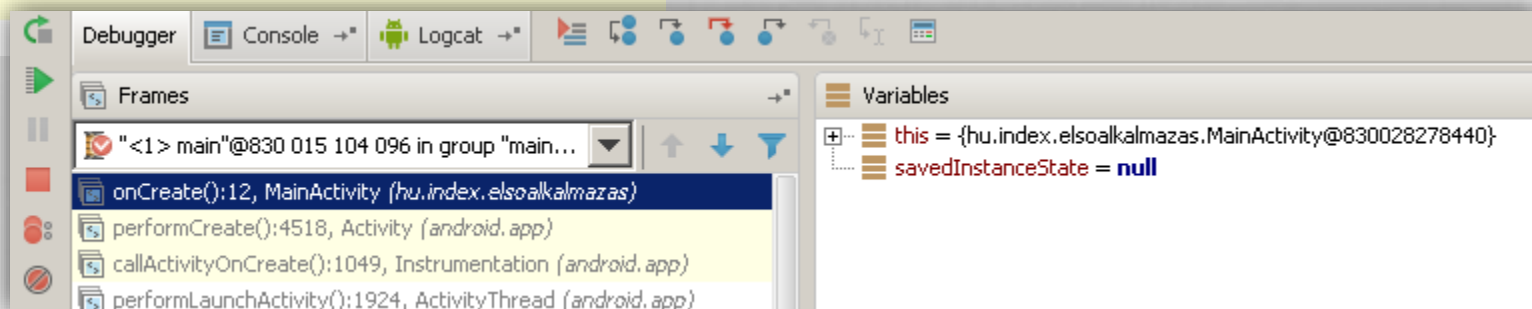
- Függetlenül: [SDK path] / tools / ddms.bat
- IntelliJ: Tools > Android > Monitor (DDMS)

vagy:



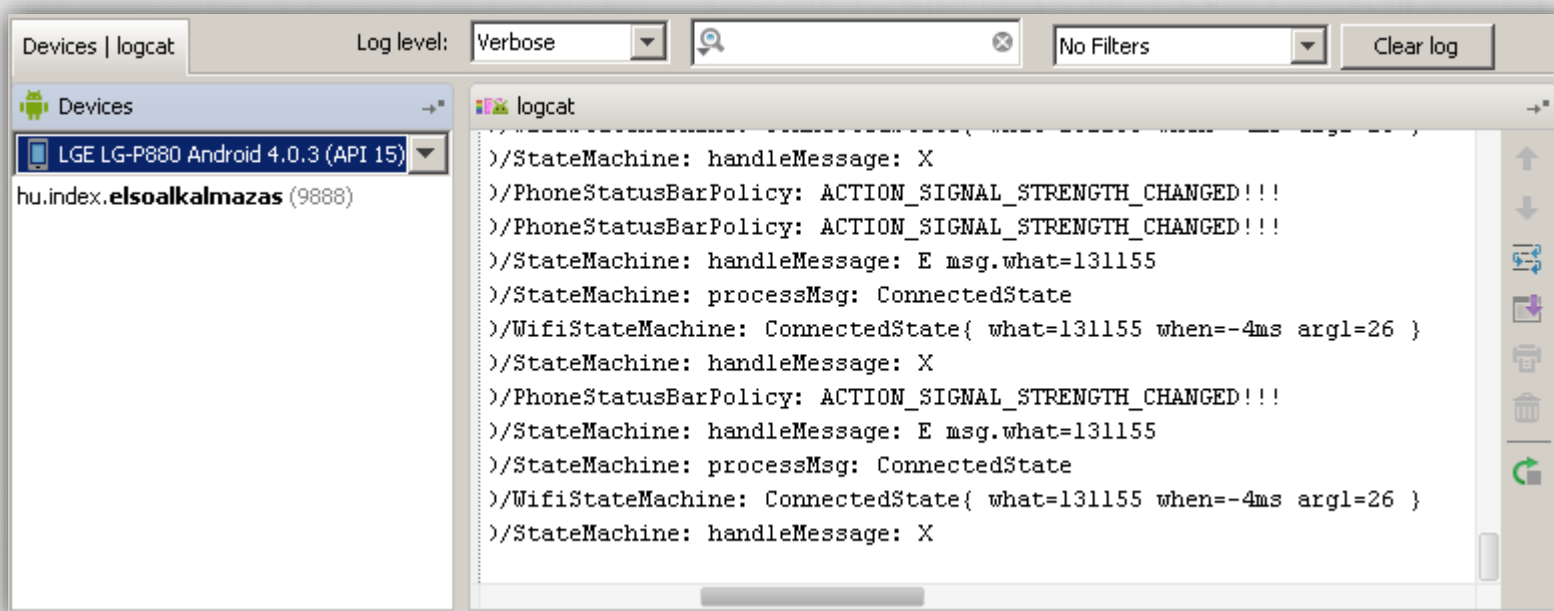
## Eredmény:

```
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());`