

1. Mi az Android?

Az Android a Google Linux alapokra helyezett, ám Java nyelven programozható beágyazott platformja. Az Android érdekességét éppen ez a két jellemző adja: az alapszint működését és az eszközök kezelését egy módosított Linux biztosítja, amelyen egy DalvikVM virtuális gép fut, és az operációs rendszer szolgáltatásait már Java nyelven írt programok használják.

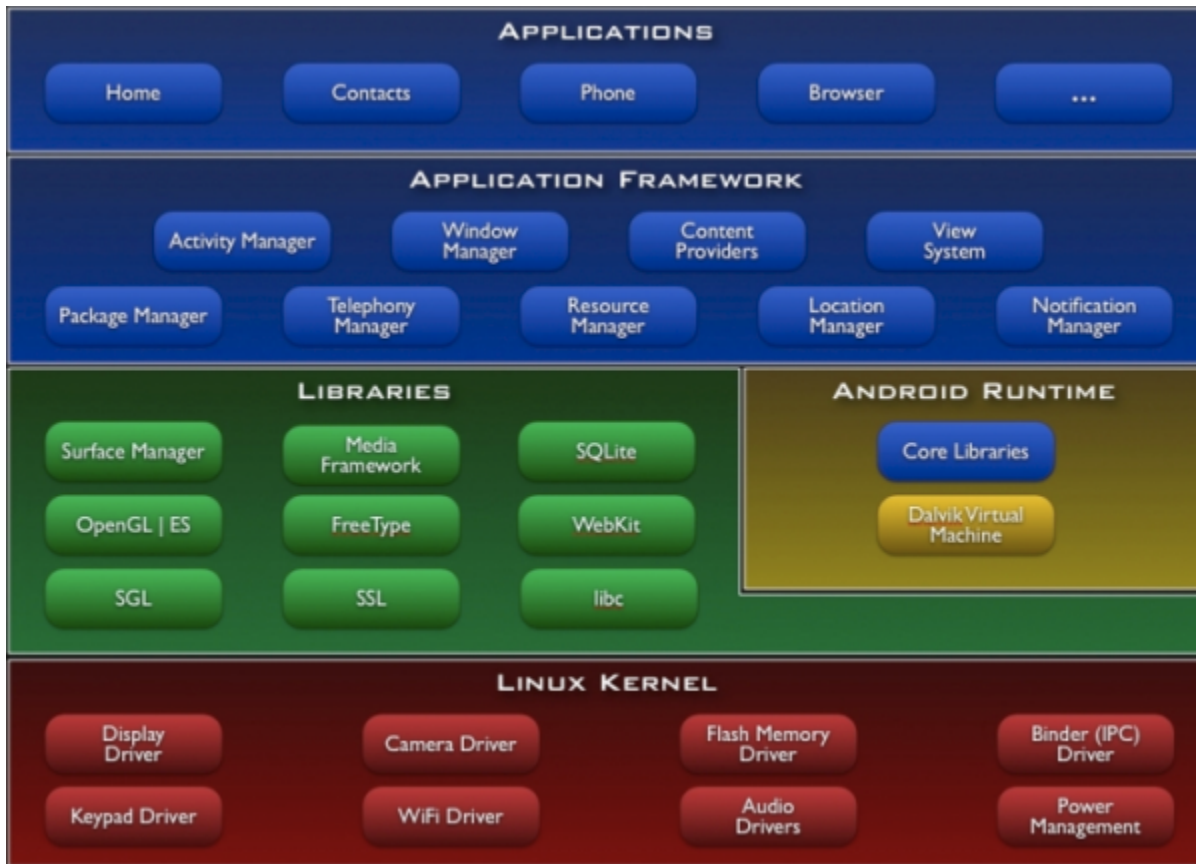
1.1. A platformról

Az Android platform abból a célból született, hogy egységes nyílt forrású alapja legyen a mobil eszközöknek (és itt elsősorban a *smartphone* kategóriát kell érteni, mintsem egyszerű mobiltelefonokat). Az elképzelés alapja egy Linux alapú operációs rendszer volt, amelyet úgy alakítanak át, hogy képes legyen problémák nélkül kezelni a mobil eszközök integrált hardvereit (érintképernyő, WiFi, HSDPA, Bluetooth, stb.). Az első lépéseknél nem volt szó Java nyelvről, azonban a Google 2005 júliusában megvásárolta az Android nevű céget, és új irányt adott a fejlesztésnek: a Linux kernel fölé egy virtuális gép került, amely a felhasználói felület kezeléséért és az alkalmazások futtatásáért felel.

Természetesen ez nem ment egyik napról a másikra, és a Google az első évben igen csendesen dolgozott, 2007 elején kezdtek kiszivárogni olyan hírek, hogy a Google belép a mobil piacra. Az iparági pletykák végül igaznak bizonyultak, bár sok esetben túlzó állításokat és rémhíreket olvashattunk a híroldalakon. 2007. november ötödikén az Open Handset Alliance bejelentette az Android platformot. Az OHA tagjai között több tucat mobil technológiában érdekelt céget megtalálunk, akik érdekeltek egy nyílt forrású platform bevezetésében.

Napjainkra az Android platform iránt a mobiltelefon és a Tablet gyártók érthető izgalma leszámitva nagy érdeklődés mutatkozik a gépjárművek fedélzeti számítógépét és navigációját szállító cégek, illetve az ipari automatizálás irányából is, hiszen minden olyan helyen kényelmes az Android, ahol alapvetően kicsi a kijelző, limitáltak az erőforrások és az adatbevitel nem egerrel és/vagy billentyűzettel történik.

A platform egyszerűnek tűnik, és messzire nézve egyszerű is, vessünk egy pillantást az alábbi ábrára:



Mint láthatjuk, a platform alapját a vörös színnel jelölt Linux kernel adja, amely tartalmazza a hardver által kezelendő eszközök meghajtó programjait. Ezeket azon cégek készítik el, amelyek az Android platformot saját készülékükön használni kívánják, hiszen a gyártónál jobban más nem ismerheti a mobil eszközbe integrált perifériákat. Ez a kis méretű kernel adja a memória kezelést, a folyamatok ütemezését és az alacsony fogyasztást elsegítő teljesítmény-kezelést is.

A kernel szolgáltatásait használják a Linux rendszerekben meglévő különféle programkönyvtárak, mint a *libc*, az *SSL* vagy az *SQLite*; ezek C/C++ nyelven vannak megvalósítva, és a Linux kernelen futnak közvetlenül. Részben ezekre épül a Dalvik virtuális gép, amely egyáltalán nem kompatibilis a Sun virtuális gépével, teljesen más az utasítás készlete, és más bináris programot futtat. A Java programok nem egy-egy .class állományba kerülnek fordítás után, hanem egy nagyobb Dalvik Executable formátumba, amelynek kiterjesztése .dex, és általában kisebb, mint a forrásul szolgáló .class állományok mérete, mivel a több Java fájlban megtalálható konstansokat csak egyszer fordítja bele a Dalvik fordító. A virtuális gép más, mint a Java alatti megszokott virtuális gép, vagyis a Java csak mint nyelv jelenik meg!

A kék színnel jelölt részekben már csak Java forrást találunk, amelyet a virtuális gép futtat, s ez adja az Android lényegét: a látható és tapintható operációs rendszert, illetve a futó programokat. A virtuális gép akár teljesen elrejtje a Linux által használt fájlrendszert, és csak az Android Runtime által biztosított fájlrendszert láthatjuk.

1.2. A kiadott verziók

Az Android 1.0 (Apple pie :) platform 2008. október 21-én került kiadásra Apache licence alatt, amely – egy szk fanatikusból álló rétegen kívül - nem nyerte el igazán az átlag felhasználók tetszését. A platform stabilitása megfelel volt ugyan, inkább a használhatósága volt nehézkes, sok esetben pedig a kinézete nem volt megfelelő. A HTC által gyártott G1 is inkább csak koncepció-telefon volt – elsősorban a fejlesztők munkáját, illetve felkeltend a cégek érdeklődését, mint használható mobil eszköz. A Google azonban nem ült a babérjain, hanem komoly fejlesztésekbe kezdett...

Android 1.1 (Banana bread)

Ez a *jelentéktelen* mértékben elterjedt kiadás 2009. februárban jelent meg, felkerült a G1 telefonokra, sok apró hibát javított, amely az eltelt pár hónap alatt napvilágra került, de az igazán éget problémákra nem adott gyógyírt.

Android 1.5 (Cupcake)

Majd fél évvel az els verzió után jelent meg a 2.6.27 verziójú Linux kernelen alapuló Android platform, amely tucatnyi újítást tartalmazott:

- Szoftveres billentyzet automatikus kiegészítés funkcióval
- A2DP Bluetooth támogatás, illetve automatikus headset csatlakozás
- Új UI komponensek
- Animációk a képernyéváltások között
- Feljavított Copy-Paste funkció
- Videók és képek közvetlen feltöltése a YouTube és a Picasa portálokra

Android 1.6 (Donut)

A 2009 szeptemberében megjelent 1.6 verzió az elz kiadás ráncfelvarrásaként érkezett:

- Android Market javítások
- Feljavított galéria funkcionalitások (több kép kijelölése közös mvelethez)
- Hangfelismerésen alapuló funkciók
- A teljes platformban keresni képes keres
- A használt technológiák frissítése, WVGA felbontás támogatása, egyéb optimalizálások

Android 2.0/2.1 (Eclair)

A 2009 októberében kiadásra kerül 2.0 verzió a Google régebb óta dolgozott, s a változások is mélyebbre hatottak, ami a verziószámából is látszik:

- 2.6.29 Linux kernel
- Hardverre optimalizálás
- Változatos képernyőméretek és felbontások támogatása (NetBook és Tablet támogatás)
- Újraértelmezett grafikus felület
- HTML5 támogatás
- Multi touch támogatás
- Bluetooth 2.1 támogatás
- "É" háttér

A 2.0 kiadás után nem sokkal érkezett a 2.0.1 verzió, amely több apró - de bosszantó - hibát javított, illetve 2010 januárban a 2.1 verzió, amely további javításokat hozott az Android világába. Az *Eclair* legsikeresebb verziója a 2.1 lett, a 2.0 és a 2.0.1 minimálisan terjedt el, az OHA tagok eszközeire csak 2.1 került fel.

Android 2.2 (Froyo)

2010 májusában mutatta be a Google az I/O fejlesztői konferencián:

- Feljavított böngésző: Flash 10.1 és akár háromszor gyorsabb JavaScript
- JIT támogatás, amely a CPU igényes feladatokat 400-500 százalékkal gyorsíthatja
- Stream és push támogatás
- Adhoc WiFi megosztás
- Teljesítmény és felületi javítások



TODO: további Android verziók!

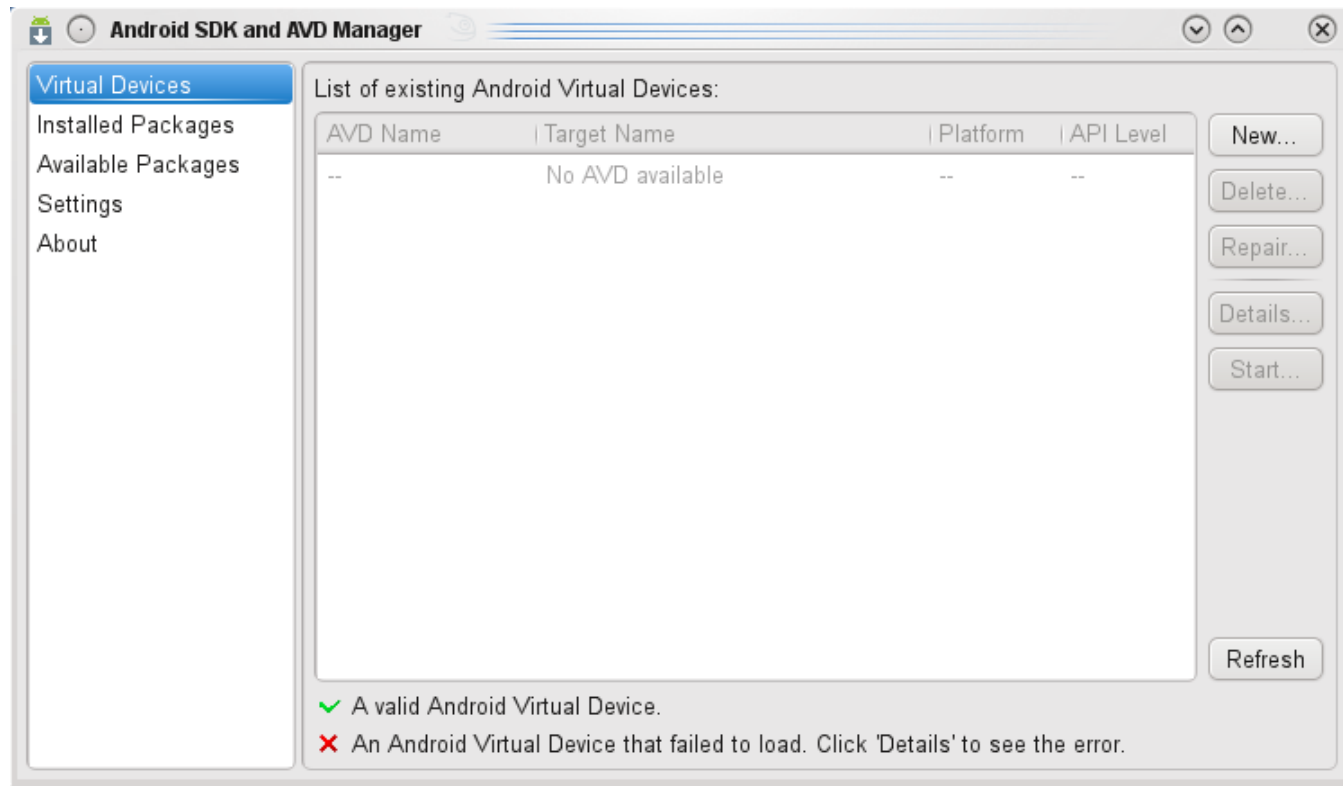
1.3. A fejlesztőkörnyezet

A platform részletesebb ismertetését illetve használatát több oldalon is megtalálhatjuk, a JavaForum portálon a **fejlesztés** a fbb téma, ezért az Android platform további boncolgatása helyett nézzünk meg, mi kell ahhoz, hogy alkalmazásokat tudjunk írni egy Android platformot futtató telefonra.

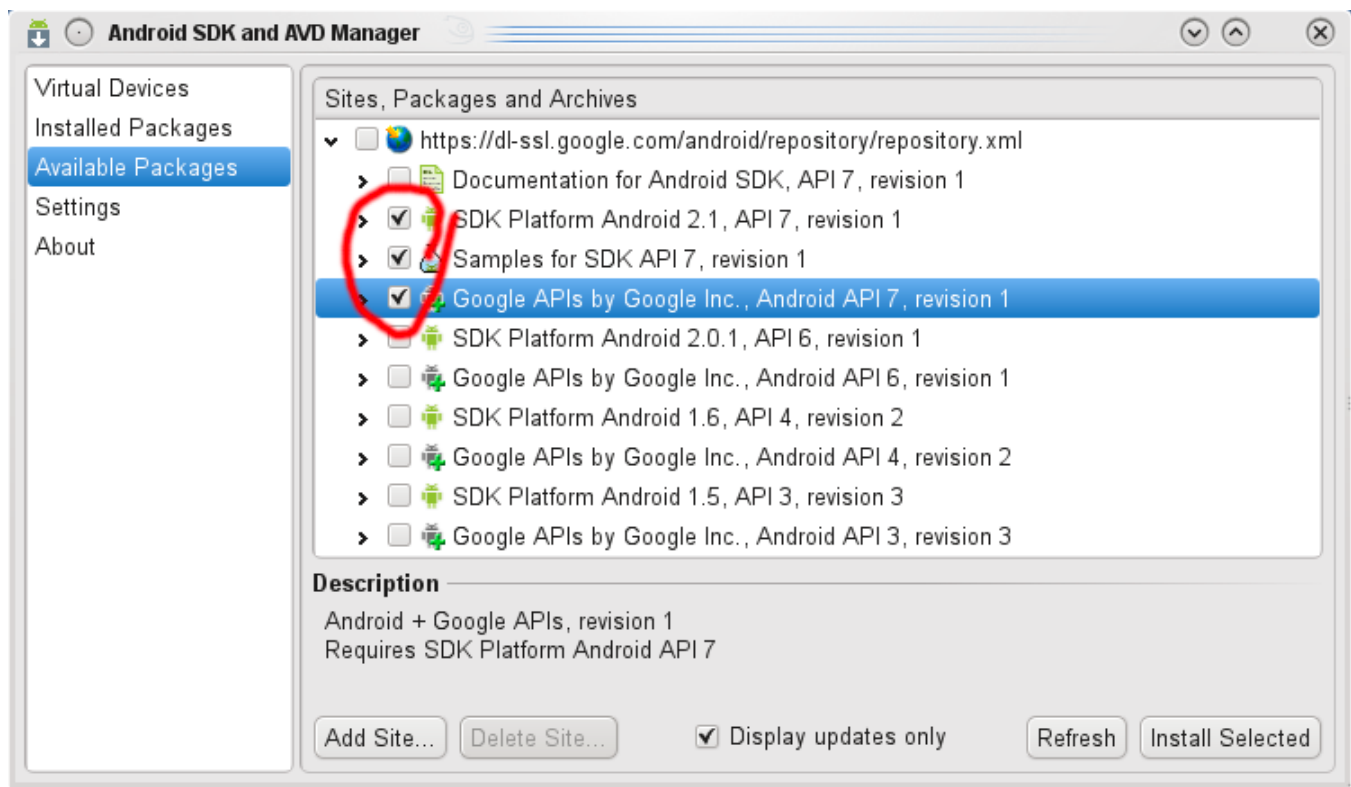
A Google els körben Eclipse alá készítette el a fejlesztőkörnyezetét, ám egy ideje létezik plugin NetBeans környezethez is, s mindkét esetben le kell töltenünk az Android SDK állományt a <http://developer.android.com/sdk/index.html> címről - mindenki a saját operációs rendszeréhez való, a cikk Linux alatt készült, tehát a Linux-os utat fogjuk végigjárni – Eclipse és NetBeans IDE használatával, illetve IDE független Maven megoldással is.

1.3.1 Közös lépések

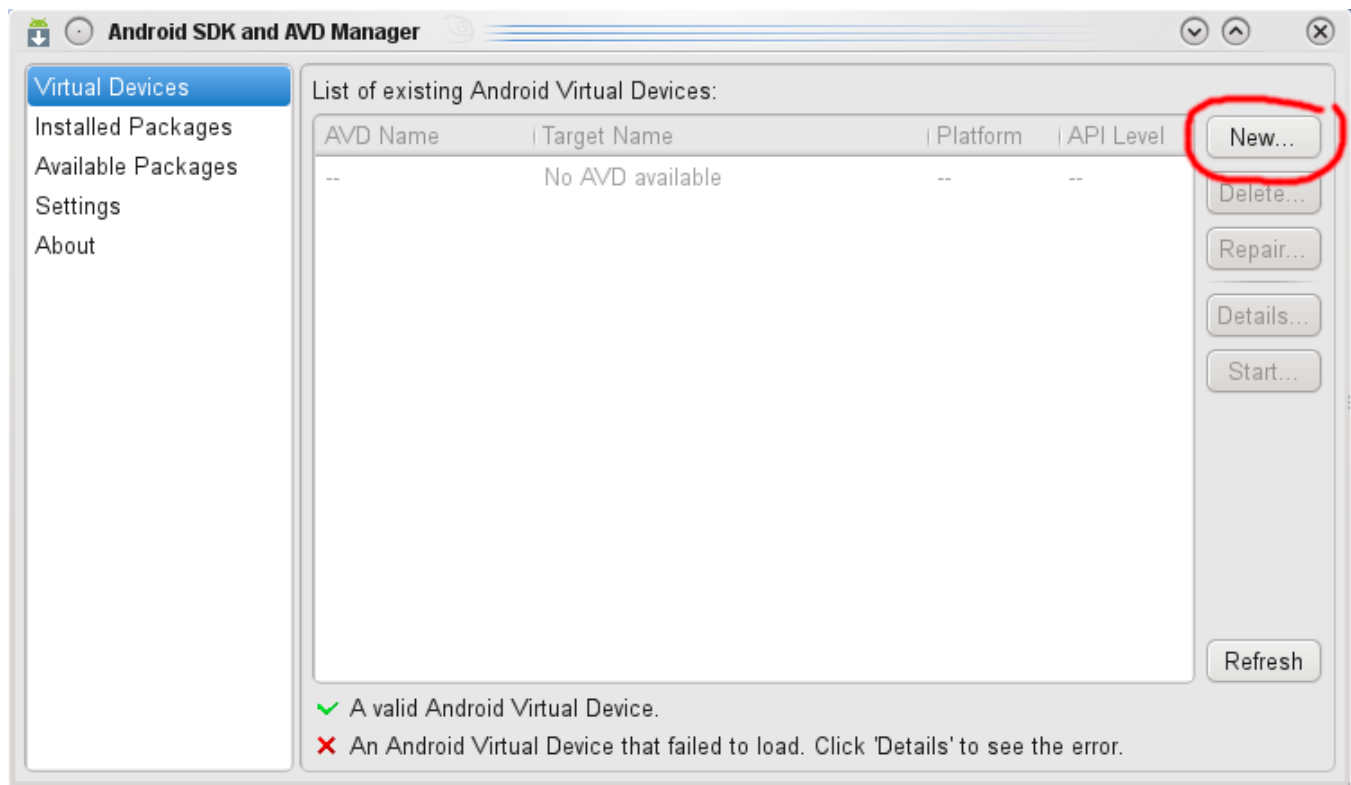
Az elz bekezdésben letöltött állományt tömörítsük ki a számunkra szimpatikus helyre, nálam ez a `/home/auth.gabor/android-sdk-linux_86/` könyvtár lett. Ennek a *tools* könyvtárban lévő *android* programot kell futtatnunk ahhoz, hogy a szükséges környezetet letöltsük – illetve frissítsük a meglévő, ha a Google kiad egy újabb platform verziót.



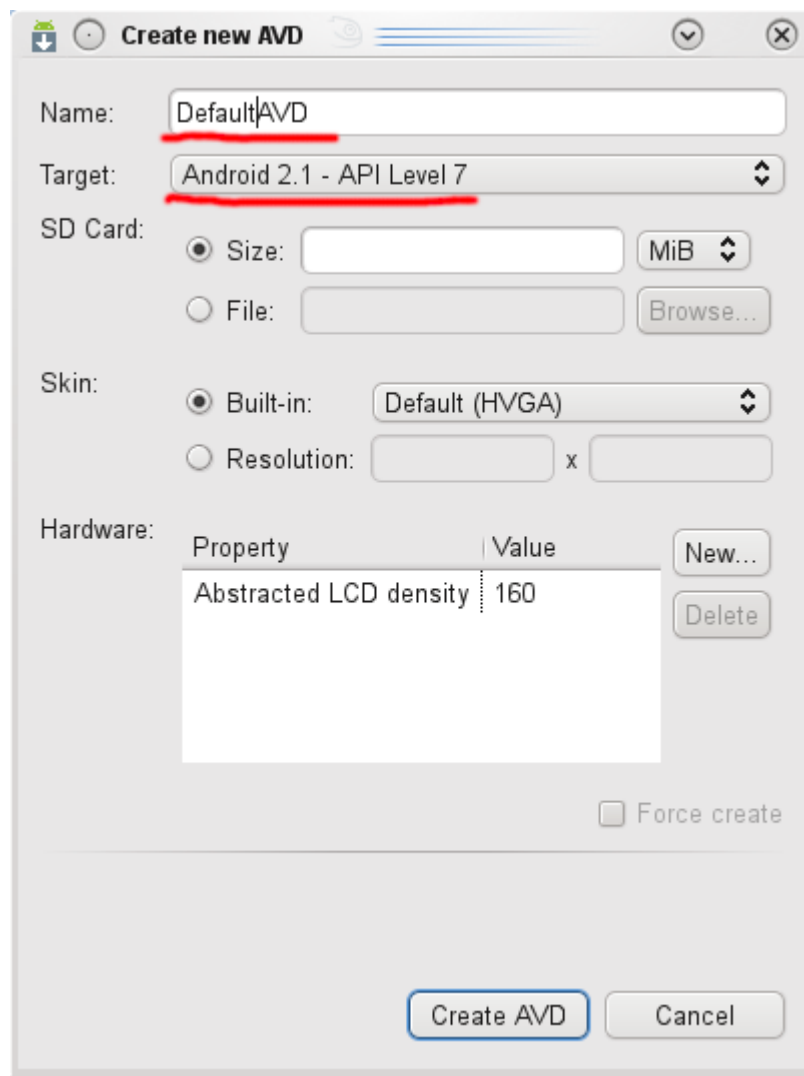
A programmal létre kell hoznunk virtuális eszközt, amelyen majd futtatni tudjuk a programunkat, ehhez először az *Available Packages* menüponton kell kattintanunk, majd a felbukkanó ablakban kiválasztani a kívánt platform verziót:



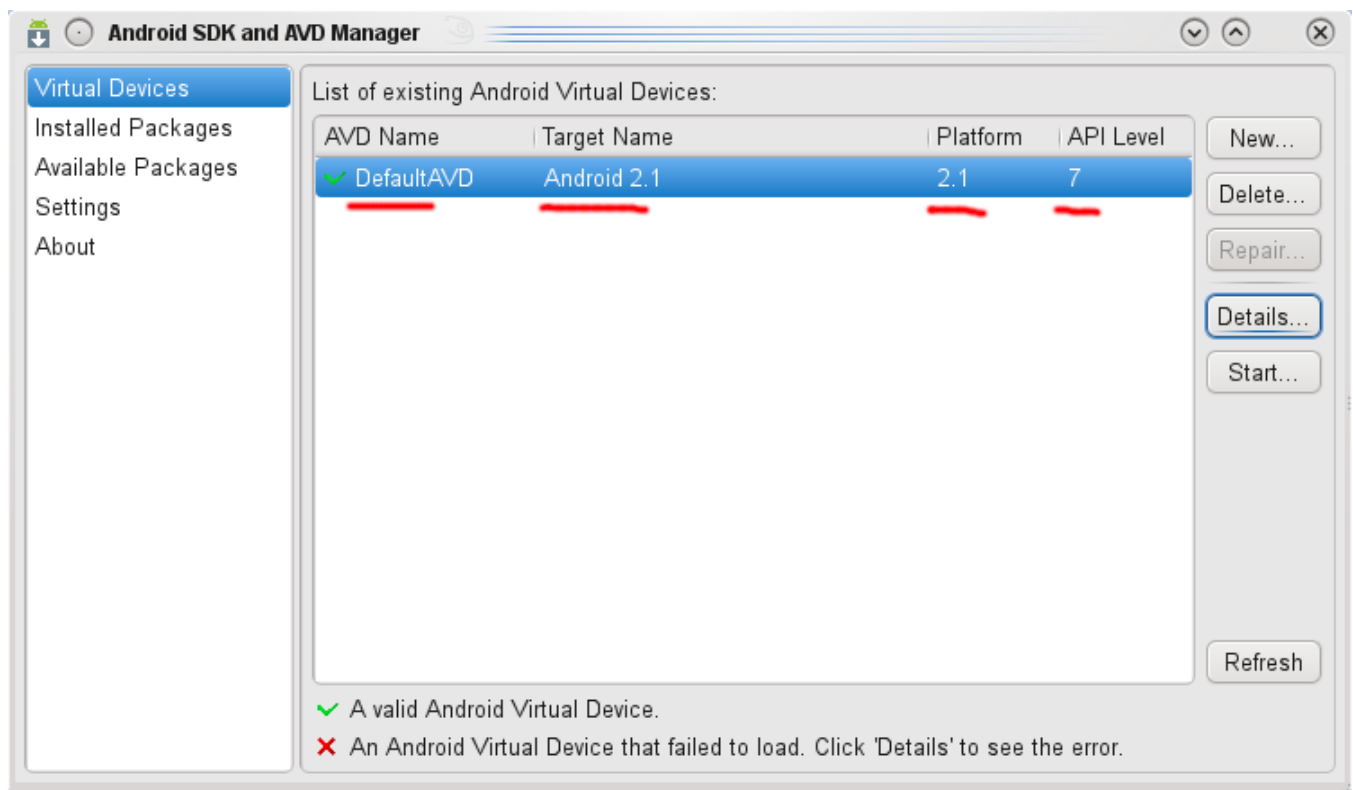
Az *Install Selected* gomb megnyomása - és a szokásos licenc elfogadása - után letöltődik és kicsomagolódik közel 250MB-nyi adat, amely szükséges lesz az emulátorhoz és a fejlesztéshez. A továbbiakban létre kell hoznunk legalább egy *Android Virtual Device* (AVD) konfigurációt, amelyet a *New* gomb megnyomásával tudunk elkezdni:



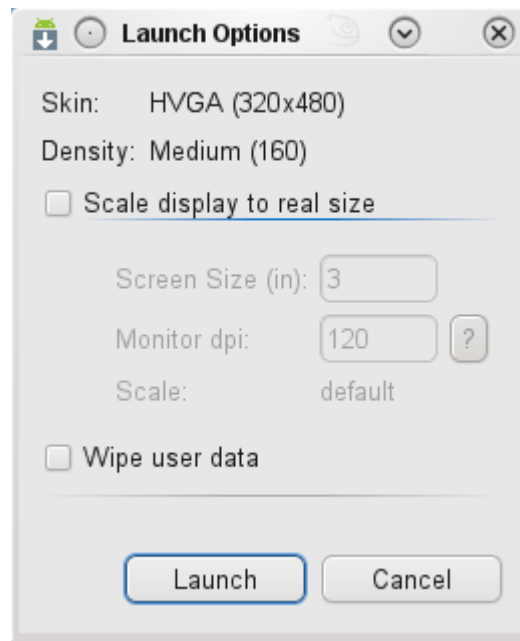
Ezek után ki kell töltenünk a felbukkanó ablakban az AVD nevét és a platform verziószámát:



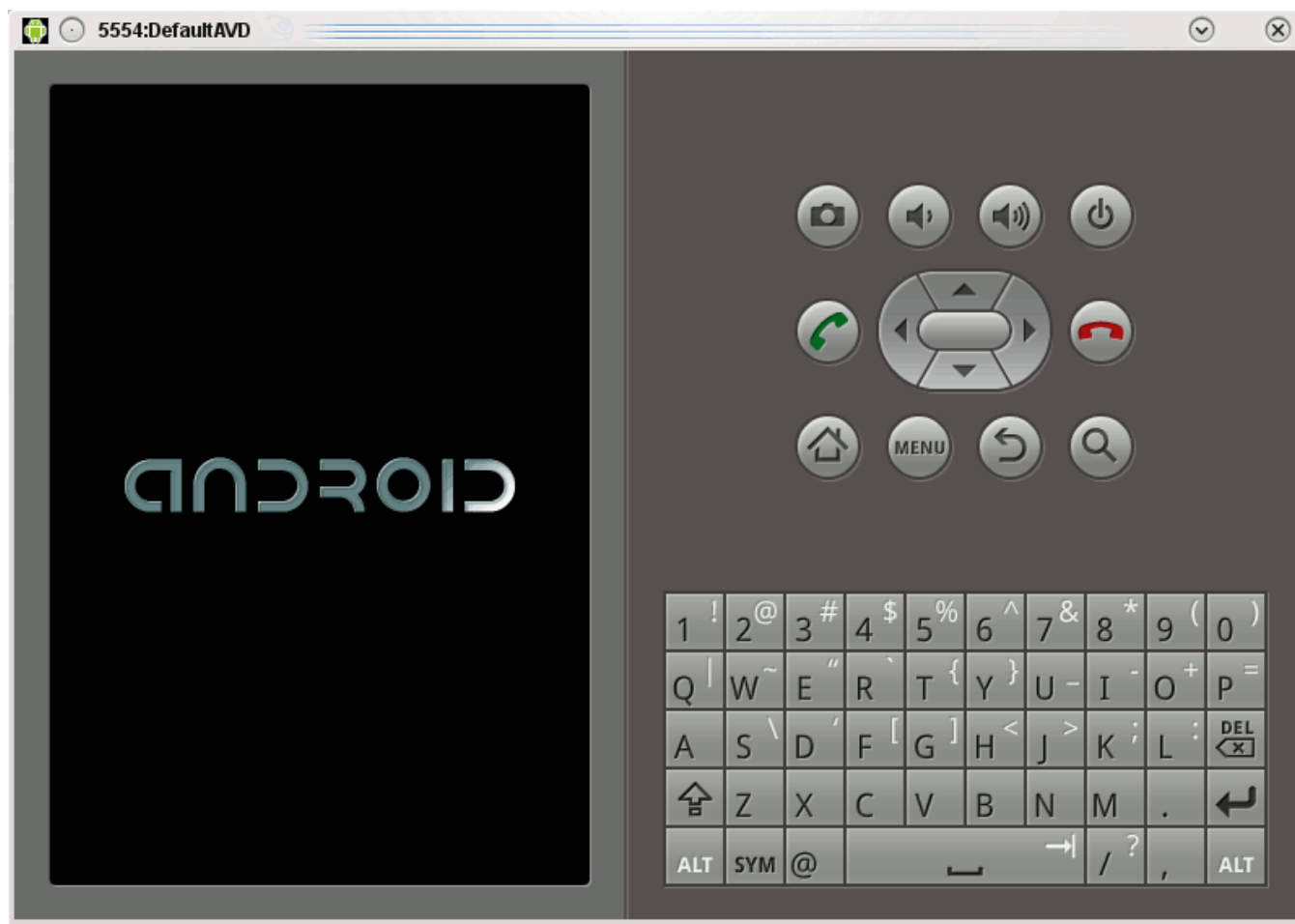
Természetesen egyéb jellemzőket is megadhatunk, több tucatnyi lehetőségünk van, amelyek az emulált hardver jellemzőit befolyásolják, ezeket most hagyjuk figyelmen kívül, hozzuk létre az virtuális eszközt a *Create AVD* gomb megnyomásával. Az *AVD Manager* listájában láthatjuk az újonnan létrehozott eszköz főbb jellemzőit:



A Start... gomb megnyomásával tudjuk a kiválasztott eszközt elindítani:



A Launch megnyomása után indul az emulátor:



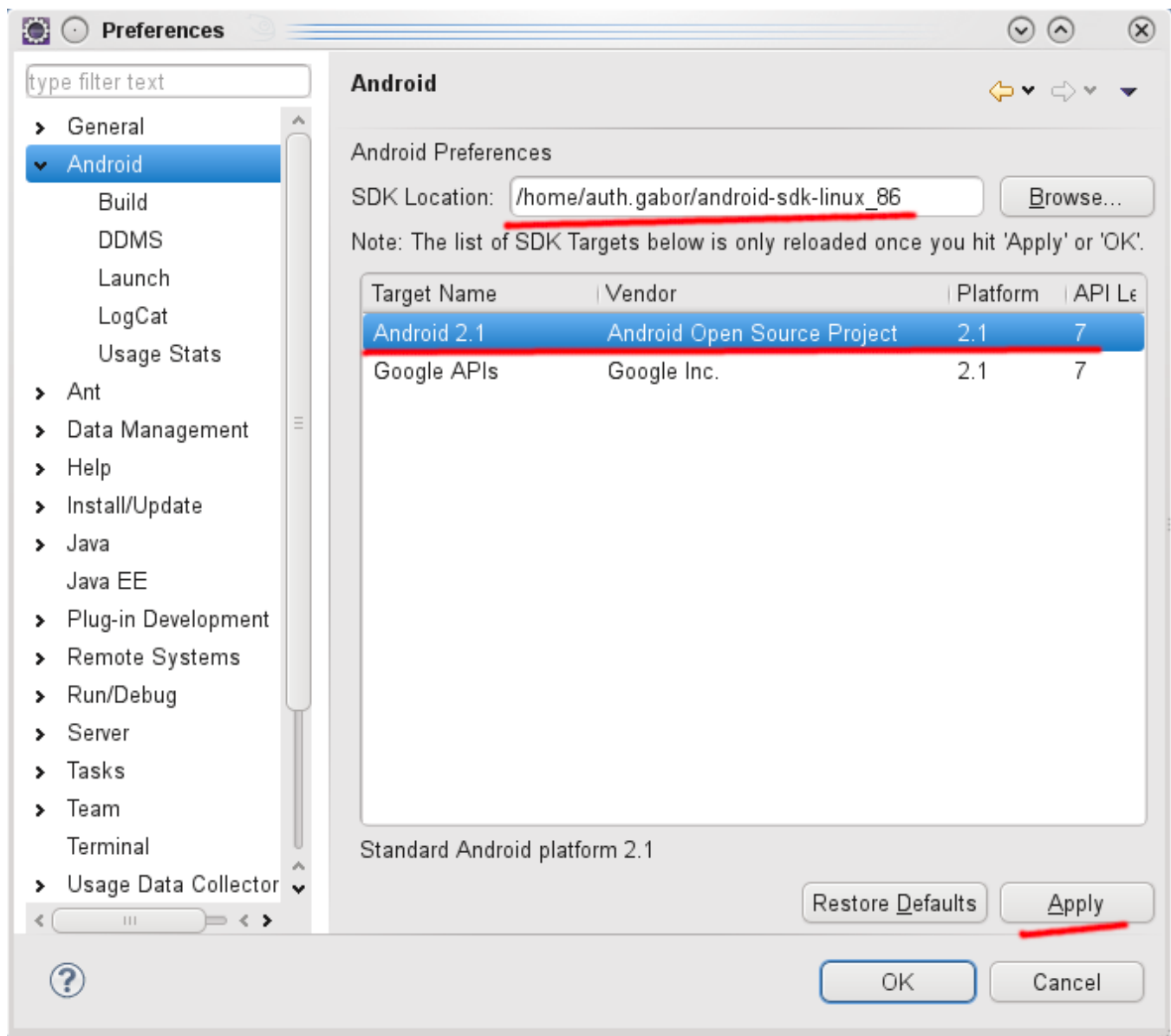
Amíg az els indítás megtörténik, igyunk meg egy jó ers kávé, ugyanis ezek után már csak össze kell barátkoztatni az SDK-t a kedvenc fejleszt-környezetünkkel... 😊

1.3.2. A fejleszt környezet beállítása

Az Android fejlesztéshez nem feltétlen kell fejlesztkörnyezet, ám sokat segít, jelen cikkben a két elterjedt ingyenes környezet beállításait mutatom meg.

1.3.2.1. Eclipse

Eclipse esetén meg kell szerezni a legújabb [ADT plugin](#)-t, majd telepíteni kell azt a megszokott módon, egyszeren kövessük a Google útmutatását. A telepítés után az Eclipse környezet újraindul, de mielőtt egy új projekt létrehozásába kezdenénk, állítsuk be az *Android SDK* helyét:



A *Browse...* gomb segítségével keressük meg a telepített Android SDK könyvtárát, majd nyomjuk meg az *Apply* gombot, ugyanis az Eclipse csak ekkor olvassa fel az ott lévő platformokat. Most már létrehozhatjuk az első Android projektet, ehhez az *Android* csoportból választjuk ki az ott árválkodó *Android Project* típust, majd lépünk a következő képernyőre:

New Android Project

Creates a new Android Project resource.

Project name:

Contents

- ☒ Create new project in workspace
- ☐ Create project from existing source
- ☒ Use default location

Location:

- ☐ Create project from existing sample

Samples:

Build Target

Target Name	Vendor	Platform	API Le
☒ Android 2.1	Android Open Source Project	2.1	7
☐ Google APIs	Google Inc.	2.1	7

Standard Android platform 2.1

Properties

Application name:

Package name:

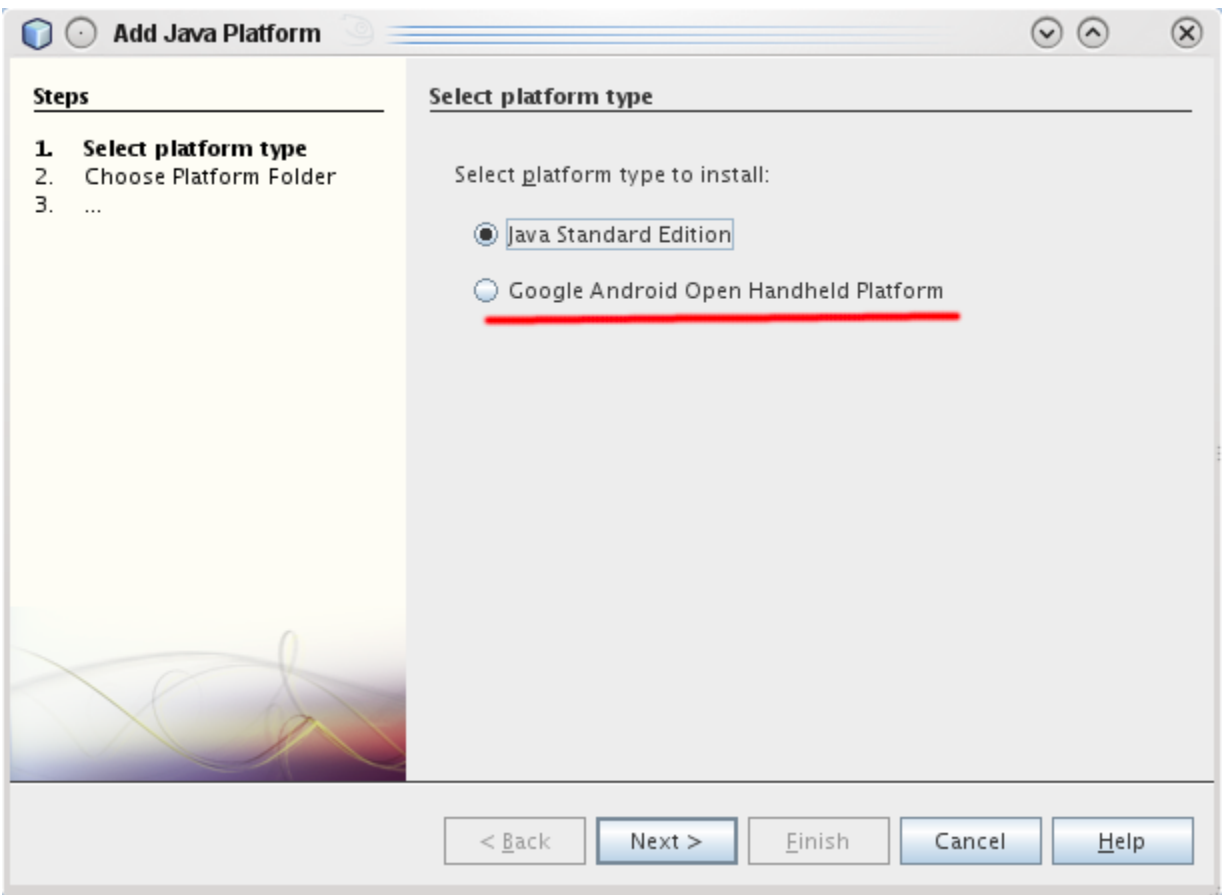
☒ Create Activity:

Min SDK Version:

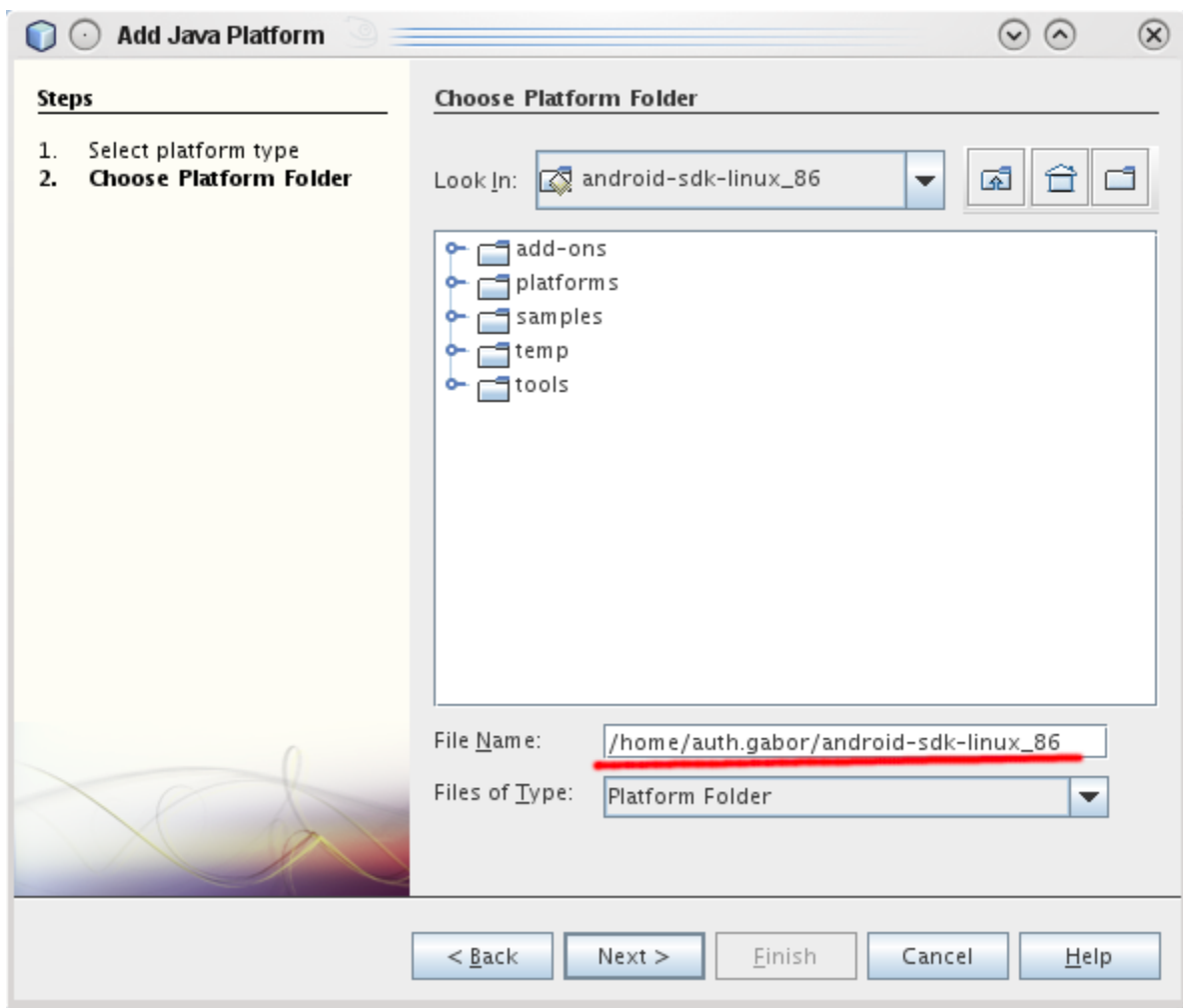
Töltsük ki értelmesen a szükséges mezket, majd kattintsunk a *Finish* gombon. Az Eclipse ekkor létrehozza a projektet, amelyre építve megírhatjuk az els alkalmazásunkat.

1.3.2.2. NetBeans

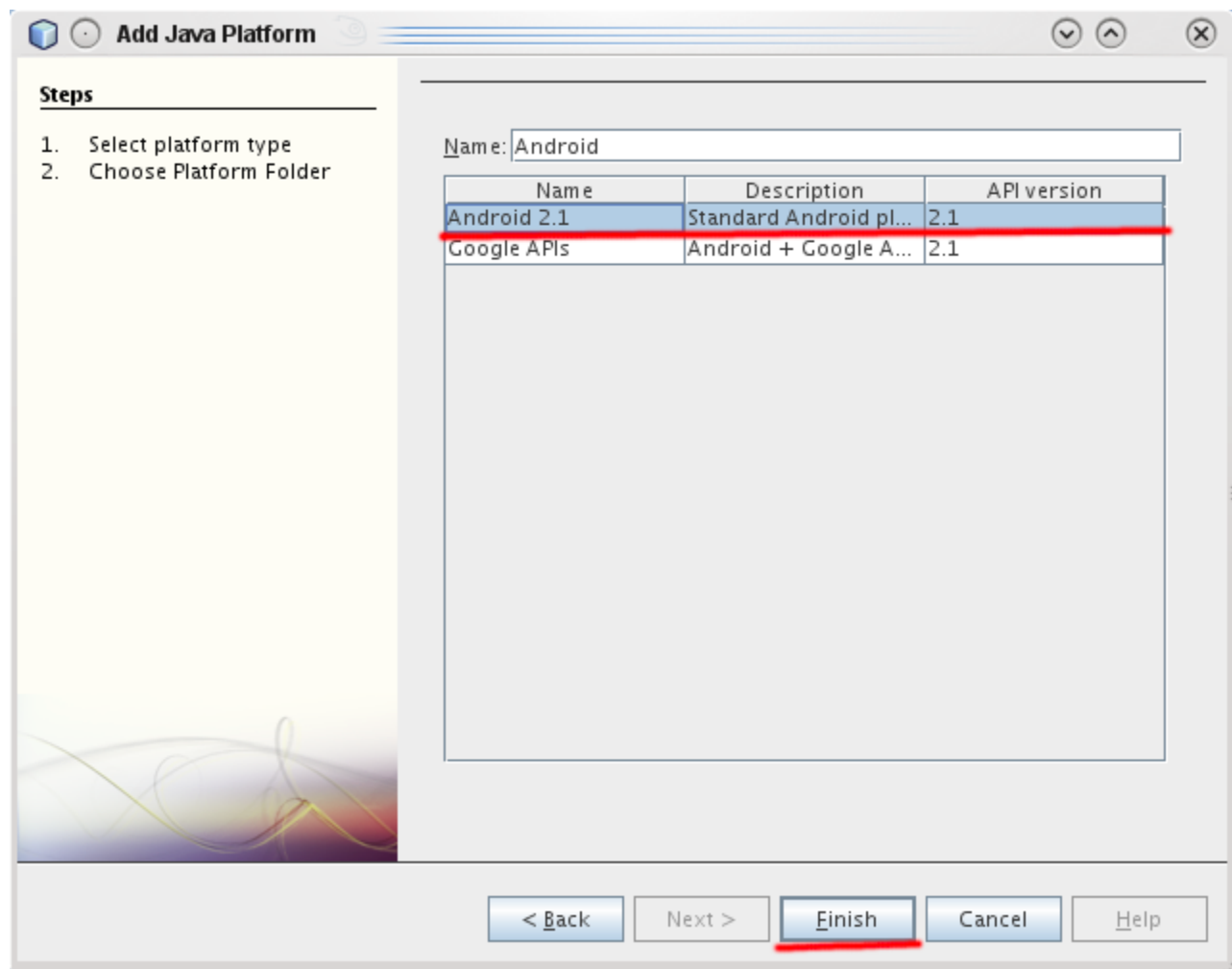
NetBeans esetén keressük meg a Kenai oldalon az [nbandroid](#) plugin-t, és adjuk hozzá az *Options* menü *Plugins* menüpontját használva (a *Settings* fülön az *Add* gomb lesz a barátunk), a NetBeans letölti és feltelepíti a plugin-t, amelyet használatba is vehetünk, ám be kell állítanunk a használandó Android SDK helyét. Ezt a *Tools* menü *Java Platforms* menüpontjában tehetjük meg:



Kattintsunk a *Google Android Open Handheld Platform* pontra, majd a *Next* gombra:



Az Android SDK helyének megadása után meg kell adnunk a platform verzióját is:



Ezzel az integráció be is fejeződött, a *File* menüben a *New Project...* menüpontban hozzunk létre egy új *Android Application* projektet:

S kész is vagyunk... 😊

1.3.2.3. Maven2

Grafikus fejlesztőkörnyezetekkel annyi a probléma, hogy nem mindig lehet automatizált tesztek és *build* szerverek használni, a nagyobb projektek és azok függőségeinek kezelése se egyszer, s a Maven pont erre jó. Ez a fejezet azon fejlesztőknek szól, akik ismerik a Maven2 elnyeit és hátrányait, s használják is.

A szükséges plugin a <http://code.google.com/p/maven-android-plugin/> címen megtalálható, a telepítése nem annyira egyszer, ugyanis nem található meg a népszerű repókban - és nem biztosít saját repót se, így letöltés után nekünk kell a saját helyi vagy a távoli repóba telepítenünk, amely könnyen elvégezhető a hozzáadott leírás alapján. Aki a gyors megoldások híve, megtalálja a szükséges állományokat a <http://nexus.javaforum.hu/nexus/content/repositories/thirdparty/> repóban, amit igyekszem frissen tartani (vigyázat, lehetnek benne elavult 3rd party fájlok!).

Egy tetszőleges helyen hozzuk létre az alábbi (unalomig ismert :) könyvtárszerkezetet:

- HelloJavaForum
 - src
 - main
 - java
 - manifest
 - resources

Az *src* könyvtár mellé hozzuk létre a *pom.xml* állományt, amely leírja a projektet:

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>

  <groupId>hu.javaforum.android</groupId>
```

```
<artifactId>HelloJavaForum</artifactId>
<version>1.0.0</version>
<packaging>apk</packaging>
<name>HelloJavaForum demo application</name>
<description>HelloJavaForum demo application</description>

<organization>
  <name>javaforum.hu</name>
  <url>javaforum.hu</url>
</organization>

<distributionManagement>
  <repository>
    <id>releases</id>
    <name>Releases</name>
    <url>http://nexus.javaforum.hu/nexus/content/repositories/releases</url>
  </repository>
  <snapshotRepository>
    <id>snapshots</id>
    <name>Snapshots</name>
    <url>http://nexus.javaforum.hu/nexus/content/repositories/snapshots</url>
  </snapshotRepository>
</distributionManagement>

<modules>
</modules>

<dependencies>
  <!-- The internal dependencies -->

  <!-- The external dependencies -->
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>[4.4,)</version>
    <scope>test</scope>
  </dependency>

  <!-- The container provided dependencies -->
  <dependency>
    <groupId>android</groupId>
    <artifactId>android</artifactId>
    <version>1.5_r3</version>
    <scope>provided</scope>
  </dependency>

  <!-- The 'transitive' external dependencies -->

  <!-- The 'transitive' container provided dependencies -->
</dependencies>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<repositories>
  <repository>
    <id>javaforum</id>
    <name>Javaforum Repository</name>
    <url>http://nexus.javaforum.hu/nexus/content/repositories/public</url>
    <releases>
      <enabled>true</enabled>
      <updatePolicy>always</updatePolicy>
      <checksumPolicy>warn</checksumPolicy>
    </releases>
    <snapshots>
      <enabled>false</enabled>
      <updatePolicy>never</updatePolicy>
      <checksumPolicy>fail</checksumPolicy>
    </snapshots>
  </repository>
</repositories>
```

```

</repositories>

<build>
  <plugins>
    <plugin>
      <groupId>com.jayway.maven.plugins.android.generation2</groupId>
      <artifactId>maven-android-plugin</artifactId>
      <configuration>
        <sdk>
          <path>${env.ANDROID_HOME}</path>
          <platform>3</platform>
        </sdk>
        <device>emulator</device>
        <deleteConflictingFiles>true</deleteConflictingFiles>
        <androidManifestFile>${project.basedir}/src/main/manifest/AndroidManifest.xml</androidManifestFile>
        <resourceDirectory>${project.basedir}/src/main/resources/</resourceDirectory>
      </configuration>
      <extensions>true</extensions>
    </plugin>

    <plugin>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.6</source>
        <target>1.6</target>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>

```

Láthatjuk, hogy hivatkozunk a *nexus.javaforum.hu* repóra (akár így is lehet hagyni - akár akár saját repóra cserélni), ebben találja meg a Maven2 azokat a függéseket, amelyeket fellepítettünk. Az *src/main/manifest* könyvtárban hozunk létre egy *AndroidManifest.xml* állományt:

AndroidManifest.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="hu.javaforum.android">
  <application>
    <activity android:name=".HelloActivity" android:label="HelloActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>

```

Szükségünk lesz még egy *layout* mappára a *resources* alatt, benne egy *main.xml* nev állománnyal:

main.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent">
  <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello, JavaForum!"/>
</LinearLayout>

```

Kell egy *values* könyvtár a *resources* alatt, benne pedig egy *strings.xml* nev fájl szükséges:

strings.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
    <string name="app_name">HelloJavaForum</string>
</resources>
```

Több nem is kell, már csak maga a Java program, amelyet a `src/main/java/hu/javaforum/android` könyvtár alá kell elhelyeznünk *MainActivity.java* néven:

HelloActivity.java

```
package hu.javaforum.android;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class HelloActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        TextView textView = new TextView(this);
        textView.setText("Hello, JavaForum.hu!");
        setContentView(textView);
    }
}
```

A projekt már fordítható is (eltte állítsuk be a `JAVA_HOME` és az `ANDROID_HOME` környezeti változókat):

Parancssor

```
$ export JAVA_HOME=/opt/jdk1.6.0/
$ export ANDROID_HOME=/home/auth.gabor/android-sdk-linux_86
$ mvn clean install deploy android:deploy
[INFO] Scanning for projects...
[INFO] -----
[INFO] Building HelloJavaForum demo application
[INFO]    task-segment: [clean, install, deploy]
[INFO] -----
[INFO] [clean:clean]
[INFO] Deleting directory /home/auth.gabor/Android/HelloJavaForum/target
[INFO] artifact junit:junit: checking for updates from javaforum
[INFO] [android:generate-sources]
[INFO] ANDROID-904-002: Found aidl files: Count = 0
[INFO] ANDROID-904-002: Found aidl files: Count = 0
[INFO] Copying local resource files to combined resource directory.
[INFO] /home/auth.gabor/android-sdk-linux_86/platforms/android-3/tools/aapt [package, -m, -J, /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/r, -M, /home/auth.gabor/Android/HelloJavaForum/src/main/manifest/AndroidManifest.xml, -S, /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/combined-resources/res, -I, /home/auth.gabor/android-sdk-linux_86/platforms/android-3/android.jar]
[INFO] [resources:resources]
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 3 resources
[INFO] skip non existing resourceDirectory /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/extracted-dependencies/src/main/resources
[INFO] [compiler:compile]
[INFO] Compiling 2 source files to /home/auth.gabor/Android/HelloJavaForum/target/classes
[INFO] [jar:jar]
[INFO] Building jar: /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.jar
[INFO] [android:dex]
[INFO] /home/auth.gabor/android-sdk-linux_86/platforms/android-3/tools/dx [--dex, --output=/home/auth.gabor/Android/HelloJavaForum/target/classes.dex, /home/auth.gabor/Android/HelloJavaForum/target/android-classes]
[INFO] [resources:testResources]
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/auth.gabor/Android/HelloJavaForum/src/test/resources
```

```
[INFO] [compiler:testCompile]
[INFO] No sources to compile
[INFO] [surefire:test]
[INFO] No tests to run.
[INFO] [android:apk]
[INFO] Copying local resource files to combined resource directory.
[INFO] /home/auth.gabor/android-sdk-linux_86/platforms/android-3/tools/aapt [package, -f, -M, /home/auth.gabor/Android/HelloJavaForum/src/main/manifest/AndroidManifest.xml, -S, /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/combined-resources/res, -I, /home/auth.gabor/android-sdk-linux_86/platforms/android-3/android.jar, -F, /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.ap_]
[INFO] /home/auth.gabor/android-sdk-linux_86/tools/apkbuilder [/home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.apk, -z, /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.ap_, -f, /home/auth.gabor/Android/HelloJavaForum/target/classes.dex, -rf, /home/auth.gabor/Android/HelloJavaForum/target/classes]
[ERROR] Using keystore: /home/auth.gabor/.android/debug.keystore
[INFO] [android:internal-pre-integration-test]
[INFO] [android:internal-integration-test]
[INFO] [install:install]
[INFO] Installing /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.apk to /home/auth.gabor/.m2/repository/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.apk
[INFO] Installing /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.jar to /home/auth.gabor/.m2/repository/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.jar
[INFO] [android:generate-sources]
[INFO] ANDROID-904-002: Found aidl files: Count = 0
[INFO] ANDROID-904-002: Found aidl files: Count = 0
[INFO] Copying local resource files to combined resource directory.
[INFO] /home/auth.gabor/android-sdk-linux_86/platforms/android-3/tools/aapt [package, -m, -J, /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/r, -M, /home/auth.gabor/Android/HelloJavaForum/src/main/manifest/AndroidManifest.xml, -S, /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/combined-resources/res, -I, /home/auth.gabor/android-sdk-linux_86/platforms/android-3/android.jar]
[INFO] [resources:resources]
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 3 resources
[INFO] skip non existing resourceDirectory /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/extracted-dependencies/src/main/resources
[INFO] skip non existing resourceDirectory /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/extracted-dependencies/src/main/resources
[INFO] [compiler:compile]
[INFO] Compiling 1 source file to /home/auth.gabor/Android/HelloJavaForum/target/classes
[INFO] [jar:jar]
[INFO] Building jar: /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.jar
[INFO] [android:dex]
[INFO] /home/auth.gabor/android-sdk-linux_86/platforms/android-3/tools/dx [--dex, --output=/home/auth.gabor/Android/HelloJavaForum/target/classes.dex, /home/auth.gabor/Android/HelloJavaForum/target/android-classes]
[INFO] [resources:testResources]
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/auth.gabor/Android/HelloJavaForum/src/test/resources
[INFO] [compiler:testCompile]
[INFO] No sources to compile
[INFO] [surefire:test]
[INFO] No tests to run.
[INFO] [android:apk]
[INFO] Copying local resource files to combined resource directory.
[INFO] /home/auth.gabor/android-sdk-linux_86/platforms/android-3/tools/aapt [package, -f, -M, /home/auth.gabor/Android/HelloJavaForum/src/main/manifest/AndroidManifest.xml, -S, /home/auth.gabor/Android/HelloJavaForum/target/generated-sources/combined-resources/res, -I, /home/auth.gabor/android-sdk-linux_86/platforms/android-3/android.jar, -F, /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.ap_]
[INFO] /home/auth.gabor/android-sdk-linux_86/tools/apkbuilder [/home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.apk, -z, /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.ap_, -f, /home/auth.gabor/Android/HelloJavaForum/target/classes.dex, -rf, /home/auth.gabor/Android/HelloJavaForum/target/classes]
[ERROR] Using keystore: /home/auth.gabor/.android/debug.keystore
[INFO] [android:internal-pre-integration-test]
[INFO] [android:internal-integration-test]
[INFO] [install:install]
[INFO] Installing /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.apk to /home/auth.gabor/.m2/repository/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.apk
[INFO] Installing /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.jar to /home/auth.gabor/.m2/repository/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.jar
[INFO] Installing /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.jar to /home/auth.gabor/.m2/repository/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.jar
```

```

[INFO] [deploy:deploy]
Uploading: http://nexus.javaforum.hu/nexus/content/repositories/releases/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.apk
12K uploaded (HelloJavaForum-1.0.0.apk)
[INFO] Retrieving previous metadata from releases
[INFO] Uploading repository metadata for: 'artifact hu.javaforum.android:HelloJavaForum'
[INFO] Uploading project information for HelloJavaForum 1.0.0
Uploading: http://nexus.javaforum.hu/nexus/content/repositories/releases/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.jar
9K uploaded (HelloJavaForum-1.0.0.jar)
Uploading: http://nexus.javaforum.hu/nexus/content/repositories/releases/hu/javaforum/android/HelloJavaForum/1.0.0/HelloJavaForum-1.0.0.jar
9K uploaded (HelloJavaForum-1.0.0.jar)
[INFO] -----
[INFO] Building HelloJavaForum demo application
[INFO] task-segment: [android:deploy] (aggregator-style)
[INFO] -----
[INFO] [android:deploy]
[INFO] /home/auth.gabor/android-sdk-linux_86/tools/adb [-e, install, -r, /home/auth.gabor/Android/HelloJavaForum/target/HelloJavaForum-1.0.0.apk]
[ERROR] 234 KB/s (12331 bytes in 0.051s)
pkg: /data/local/tmp/HelloJavaForum-1.0.0.apk
Success
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 16 seconds
[INFO] Finished at: Sat Apr 10 10:40:54 CEST 2010
[INFO] Final Memory: 32M/84M
[INFO] -----

```

A szövegáradatból kimazolázható, hogy a Maven2 letisztázta, lefordította, fémásolta az alkalmazást a tároli repóba, majd telepítette azt az emulátorba. Sajnos nem indítja el a telepítés után, ez egyelre hiányossága a pluginnak, ám ezzel egyelre együtt lehet élni.

1.3.3. Az els alkalmazás

A fejleszt környezetek által létrehozott minta alkalmazás helyett írjuk be a következ kis példaprogramot:

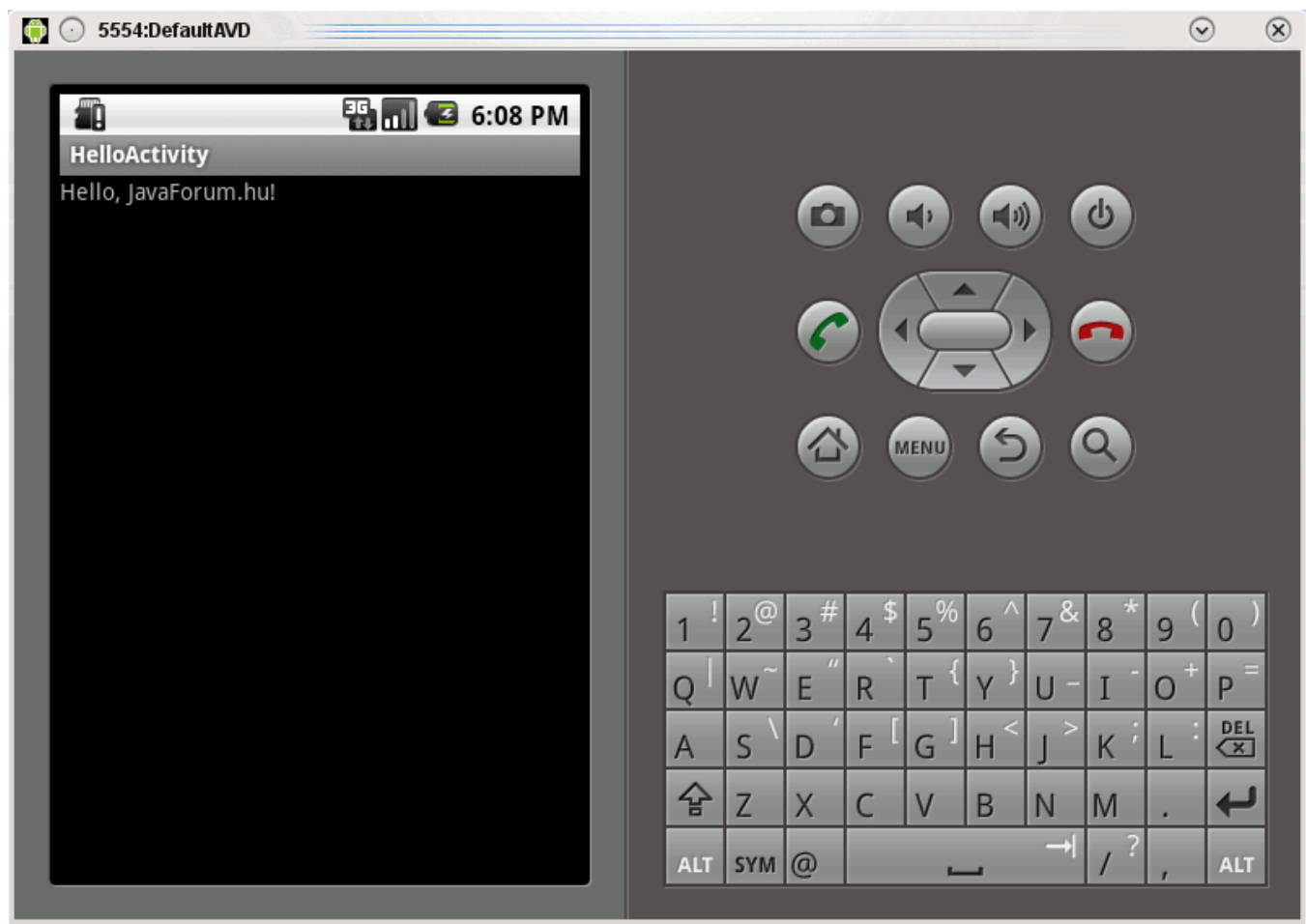
HelloActivity.java

```

package hu.javaforum.android;
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;
public class HelloActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        TextView textView = new TextView(this);
        textView.setText("Hello, JavaForum.hu!");
        setContentView(textView);
    }
}

```

A program egyszeren kiírja majd a *"Hello, JavaForum.hu"* szöveget, fordítsuk le, s futtassuk a szokott módon, ám az alkalmazás a már megnyitott és futó emulátorban fog megjelenni:



Ezzel el is készült az els - Android platformon futó - alkalmazásunk, amelyet akár fel is másolhatunk egy Android platformot futtató telefonra - ilyen egyszer programot bármelyik verzió képes futtatni... 😊