



Android: Les Bases

Assane SECK
Ingénieur-Informaticien



Sommaire

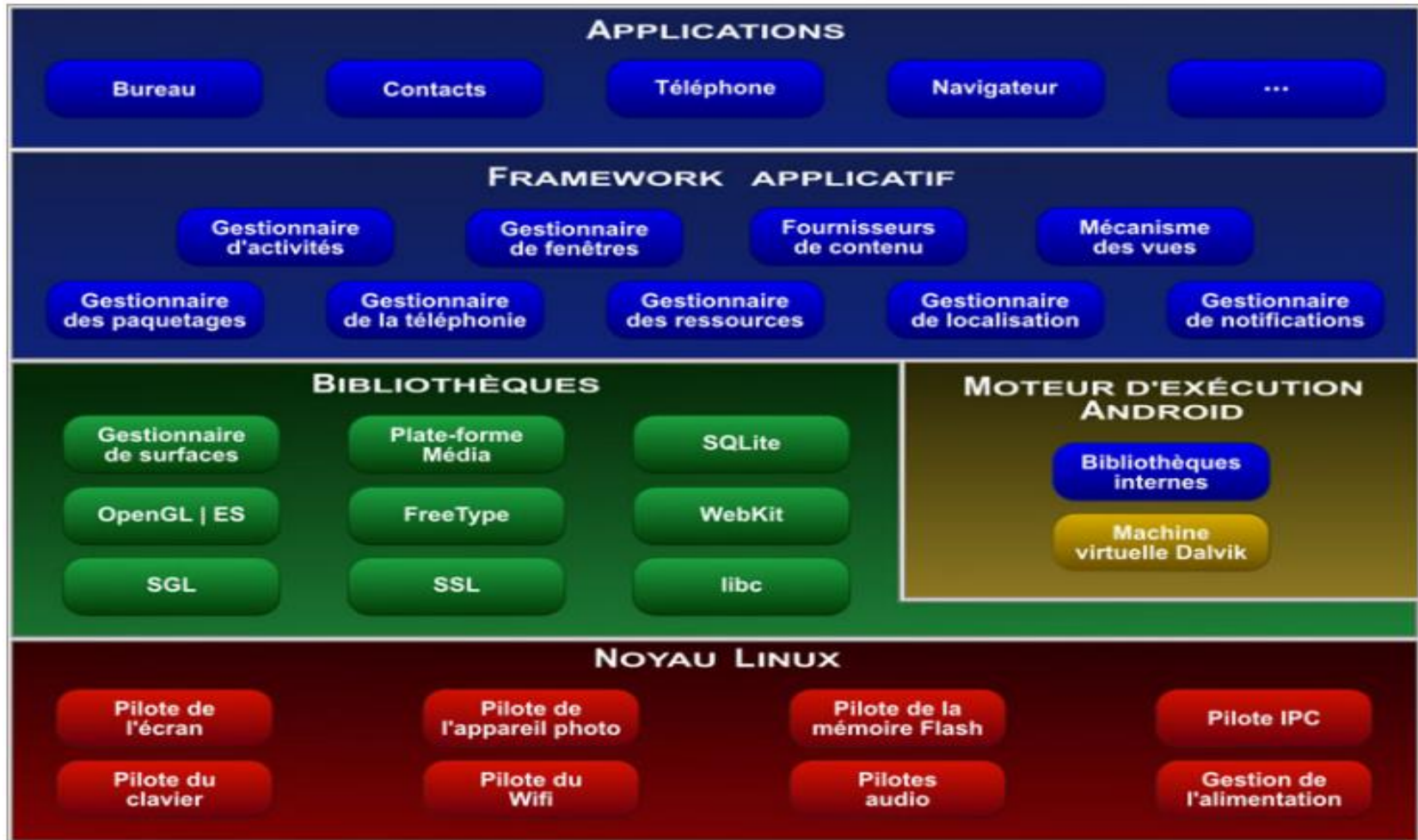
- ▶ Hitorique
- ▶ Architecture
- ▶ Avantage
- ▶ Dalvik VM / **Android Runtime**
- ▶ Composants
- ▶ Activity
- ▶ CONTENT PROVIDERS
- ▶ INTENTS
- ▶ SERVICES
- ▶ Structure projet Android
- ▶ AndroidManifest
- ▶ Versionning
- ▶ Emulateurs

Historique

- ▶ En 2003 était le nom d'une PME, rachetée par Google en 2005
- ▶ Objectif de OS plus intelligent: Interaction et Géolocalisation
- ▶ En Janvier 2007, Apple dévoilait l'iPhone
- ▶ En Novembre 2007 l'Open Handset Alliance (OHA) avec 35 Entreprises dont Google l'Initiateur
- ▶ Le logiciel vedette est appelé Android.



Architecture



Avantages

► **Open source**

Le contrat de licence pour Android respecte les principes de l'*open source*, c'est-à-dire que vous pouvez à tout moment télécharger les sources et les modifier selon vos goûts

► **Gratuit (ou presque)**

► **Facile à développer**

Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès

► **Facile à vendre**

Le *Play Store* (anciennement *Android Market*) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile

► **Flexible**

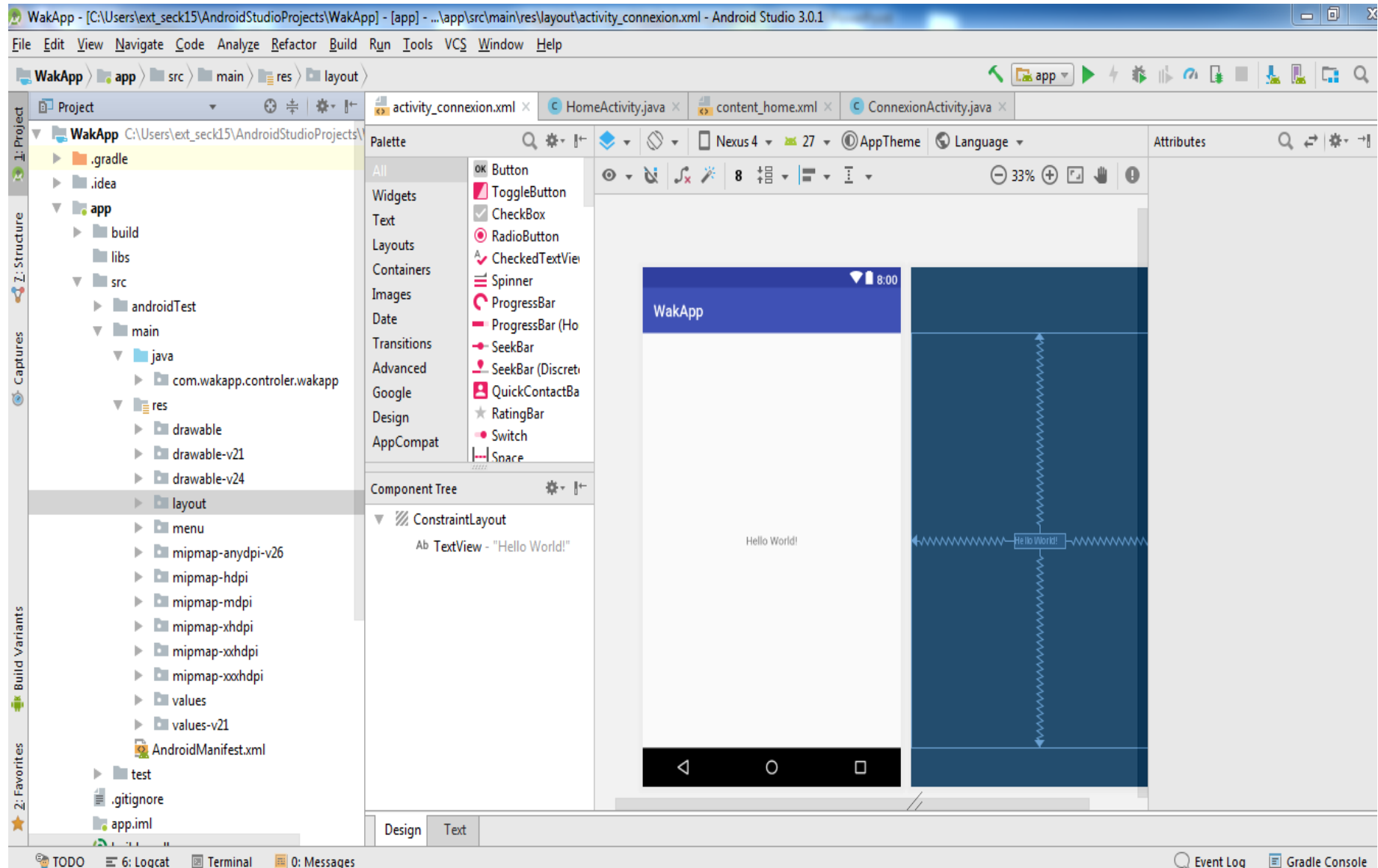
Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes. Les smartphones, les tablettes, la présence ou l'absence de clavier ou de *trackball*, différents processeurs... On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android !

► 5 **Ingénieurs**

Dalvik VM

- ▶ La plus part des téléphones portables ont des mémoires inférieures à des ordinateurs
- ▶ JVM optimisée pour mobile
- ▶ La Dalvik VM exécute de fichiers .dex, contrairement à des fichiers .class avec une JVM normale (Compilation à chaque execution)
- ▶ Dan Bornstein et son équipe chez Google
- ▶ L'ART (Android Runtime) compilation juste à l'installation
- ▶ Dalvik est un petit village au nord de l'Islande

Structure d'un projet Android

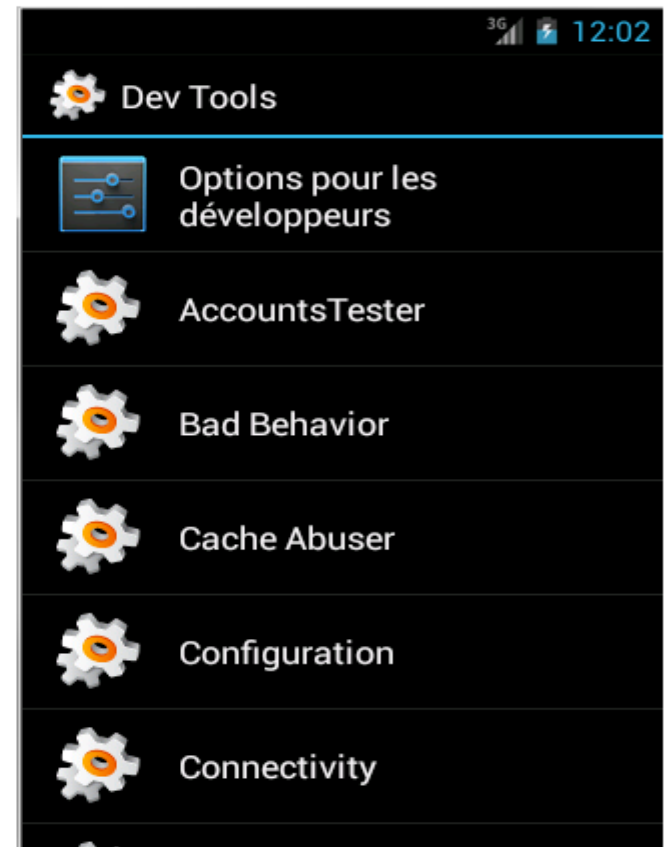
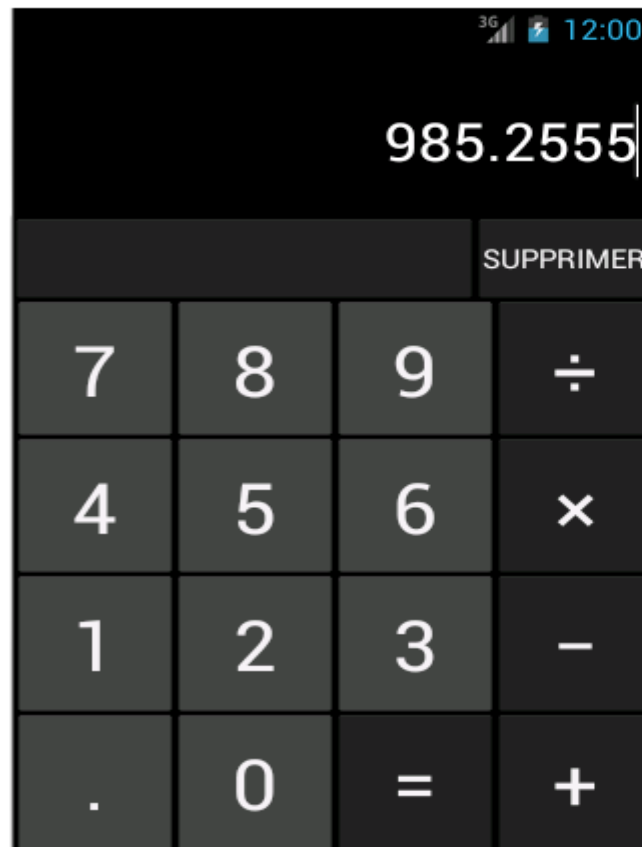


Composants

- ▶ Activity
- ▶ Content providers
- ▶ Intents
- ▶ Services
- ▶ Broadcast Receivers
- ▶ Autres
- ▶ Stockage, services système, réseau, multimédia, GPS, phone services,...

Activity

- ▶ Partie importante de l'interface graphique Android
- ▶ Peut être considéré comme un «dialogue» ou une «fenêtre»



CONTENT PROVIDERS

- ▶ Niveau d'abstraction pour données stockées sur le device
- ▶ Est accessible à travers différentes applications
- ▶ L'utilisation est recommandée

INTENTS

- ▶ Messages système qui forment la «colle» d'Android
- ▶ Un message que l'on passe à Android pour lui dire "je veux que tu fasses... quelque chose (Ouvrir une activité, etc)
- ▶ Peut être utilisé pour:
 - ▶ **Envoyer et recevoir des event ou actions**
 - ▶ **Changements de matériels, d'état, données entrantes, événement,...**
 - ▶ **Lancer des activités**
 - ▶ **Il est possible de réagir aux intents**

SERVICES

- ▶ Les activity ont une durée de vie courte
- ▶ Les services ont une durée de vie plus longue
- ▶ S'exécute indépendamment d'une Activity
- ▶ Exemples:
 - ▶ Mise à jour RSS
 - ▶ Lecture de musique en fond

Autres...

- ▶ Stockage: fichiers packagés avec l'application (icônes, fichiers d'aide,...),
- ▶ bases de données et fonctionnalités pour stockage de fichiers sur support SD ou autre.
- ▶ Réseau: les devices sont souvent connectés. Android dispose de différents moyens d'utiliser ceci au travers de sockets, HTTPClient ou le navigateur webkit.
- ▶ Multimédia: playback et enregistrement audio/video, prise de photos,...
- ▶ Localisation: GPS, réseau, passif...
- ▶ Services téléphoniques: appels, SMS, etc.

AndroidManifest.xml

- ▶ Fichier XML qui décrit l'application.
- ▶ Quels composants sont utilisés et comment sont-ils interconnectés
- ▶ Activities
- ▶ Services
- ▶ ...

bin/

- ▶ Contient l'application une fois compilée
- ▶ bin/classes/ contient les classes Java
- ▶ bin/classes.dex contient l'exécutable compilé depuis les classes Java
- ▶ bin/yourapp.ap_ contient les ressources au format ZIP
- ▶ bin/yourapp.apk est l'application effective.

- ▶ Code source généré par les outils Android
- ▶ Forme le lien entre les ressources et le code Java
- ▶ **R.java** : Ce fichier est automatiquement généré par le SDK Android à chaque pré-compilation.
- ▶ **BuildConfig.java**: Sera automatiquement mis à jour lors de la création de la version finale de l'Apk afin de désactiver les logs.

```
public static final class layout {  
    public static final int main=0x7f030000;  
}  
public static final class string {  
    public static final int app_name=0x7f050000;  
    public static final int main_no_items=0x7f050001;  
}  
public static final class style {  
    public static final int MyListView=0x7f060002;  
    public static final int MyTheme=0x7f060000;  
    public static final int SpecialText=0x7f060001;  
}
```


libs/

- ▶ Bibliothèques Java (JARs)
- ▶ Fichiers .so Utilisé principalement avec Ant
- ▶ Mettre toutes bibliothèques éventuelles dans ce répertoire pour une meilleure réorganisation

src/

- ▶ Code source Java de l'application
- ▶ Ce dossier contient les sources de votre application (code JAVA) et les packages.
- ▶ **com.example.projet**: un package de votre application. Vous pouvez avoir plusieurs packages dans votre application.
- ▶ **MainActivity.java** : Notre activité principale dans cet exemple.

res/

- ▶ Resources du projet
- ▶ Fichiers statiques package dans l'application
- ▶ `res/drawable-xxx/` : Contient vos images dans les différentes résolutions (basse, moyenne, haute et très haute).
- ▶ `res/layout/` : spécifications interface graphique
- ▶ `res/menu/` : spécifications XML pour les menus
- ▶ `res/values/` : strings, dimensions, couleurs et autres
- ▶ **ic_launcher.png** : L'icône de votre application.
- ▶ **assets** : Contient des données non internationalisées qui seront utilisées dans votre application (images, vidéos, licence...etc).

Qu'est-ce le manifest ?

- ▶ L'élément racine de l'application
- ▶ Déclaration d'activités, services,...
- ▶ Comment ces différents morceaux s'interconnectent
- ▶ Exemple: launcher

Exemple basique de manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.projet"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.projet.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Element root du manifest

- ▶ Package de l'application
- ▶ Forme la base de l'application
- ▶ Il est possible de référencer relativement les classes en utilisant un point
- ▶ Exemple : `.SampleActivity`
- ▶ Le package name doit être unique. L'Android Market n'accepte pas les applications qui ont le même package name.
- ▶ `<manifest`
- ▶ `xmlns:android="http://schemas.android.com/apk/res/android"`
- ▶ **`package="com.example.projet">`**
- ▶ ...
- ▶ `</manifest>`

Permissions, applications et instrumentations

- ▶ **uses-permission** : quelles sont les permissions que l'application utilise ?
- ▶ **permission** : de quelles permissions l'application a-t-elle besoin?
- ▶ **instrumentation** : code qui devrait être exécuté à certains moments clés
- ▶ (activities, logging, JUnit, etc.)
- ▶ **uses-library** : librairies supplémentaires
- ▶ **uses-sdk** : quel **SDK** est utilisé?
- ▶ dernier élément important: l'élément **application**

Permissions, applications et instrumentations

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.projet"

    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />

</manifest>
```


Element application

- ▶ Peut contenir différentes **activity**
- ▶ **android:name** classée, en partant du package name (avec un point) ou en utilisant le nom entier
- ▶ **android:label** titre pour l'activity
- ▶ **intent-filter**: sous quelle condition l'activity peut-elle être lancée
- ▶ **action**: La classe de traitement pour la condition
- ▶ **category**: Spécifie la catégorie de l'action

Element application

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.projet"

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.projet.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Element application

- ▶ Autres:
- ▶ **receivers** : des non-activités qui doivent être appelées à un moment clé, en fonction d'un intent. Exemple: réception d'un SMS.
- ▶ **providers** : Un **contentProvider** sert à stocker et récupérer des données et ainsi les rendre accessibles à toutes les applications. Partager des données entre différentes applications. Exemple: la gestion des contacts
- ▶ **services**: Si votre application souhaite effectuer des opérations ou des calculs en dehors de l'interaction utilisateur. Exemple: Surfer sur le net et lire son

Element uses-sdk

- ▶ Android évolue
- ▶ Chaque SDK a des nouvelles fonctions, classes, paramètres,...
- ▶ Ceux-ci ne peuvent pas être utilisés par des précédentes version SDK
- ▶ Ce tag assure que l'application est exécutée sur un device d'une certaine version ou plus haute
- ▶ **minSdkVersion:API** niveau minimum supporté
- ▶ **targetSdkVersion:API** niveau ciblé
- ▶ **maxSdkVersion:API** niveau maximum supporté (optionel)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.projet"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />

</manifest>
```

USES-SDK

Platform Version	API Level	VERSION_CODE	Notes
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD	
Android 2.2.x	8	FROYO	
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

Element uses-feature

- ▶ Décrit une caractéristique matérielle ou logicielle utilisée par l'application
- ▶ Nécessaire pour indiquer quels sont les pré requis pour une application
- ▶ Bluetooth, appareil photo,...
- ▶ Contient
- ▶ **android:name** : nom du feature
- ▶ **android:required** : si le feature doit être présent ou pas
Valeur par défaut de android:required est true

```
<uses-feature android:name="android.hardware.camera" android:required="false" />
```

VERSIONNER SON APPLICATION

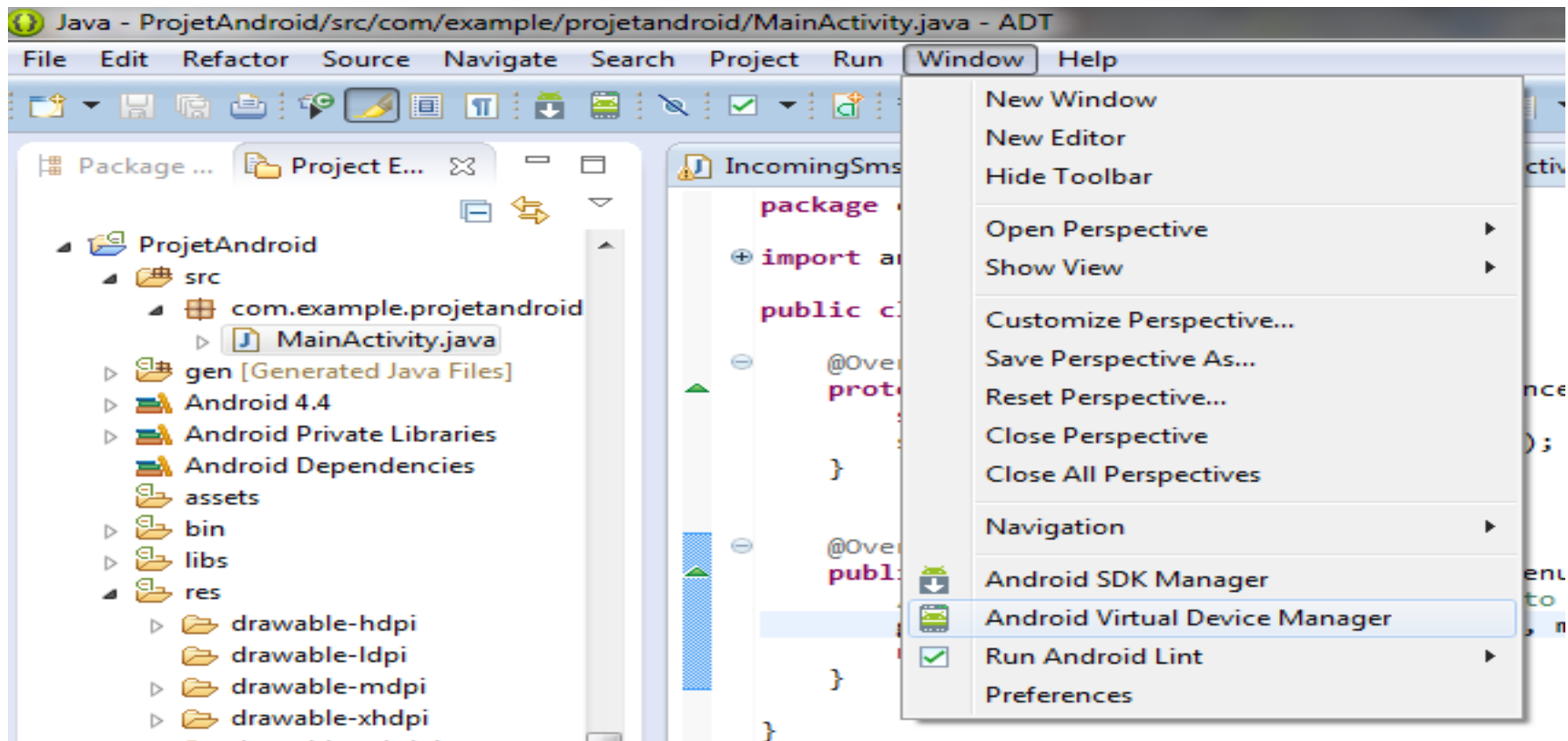
- ▶ Se fait sur l'élément manifest
- ▶ **android:versionCode** : Numéro integer incrémenté à chaque nouvelle version
- ▶ **android:versionName** : Nom pour cette version (alphanumérique)
- ▶ 1.1
- ▶ 2.0b
- ▶ ...

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.projet"
    android:versionCode="1"
    android:versionName="1.0" >

</manifest>
```

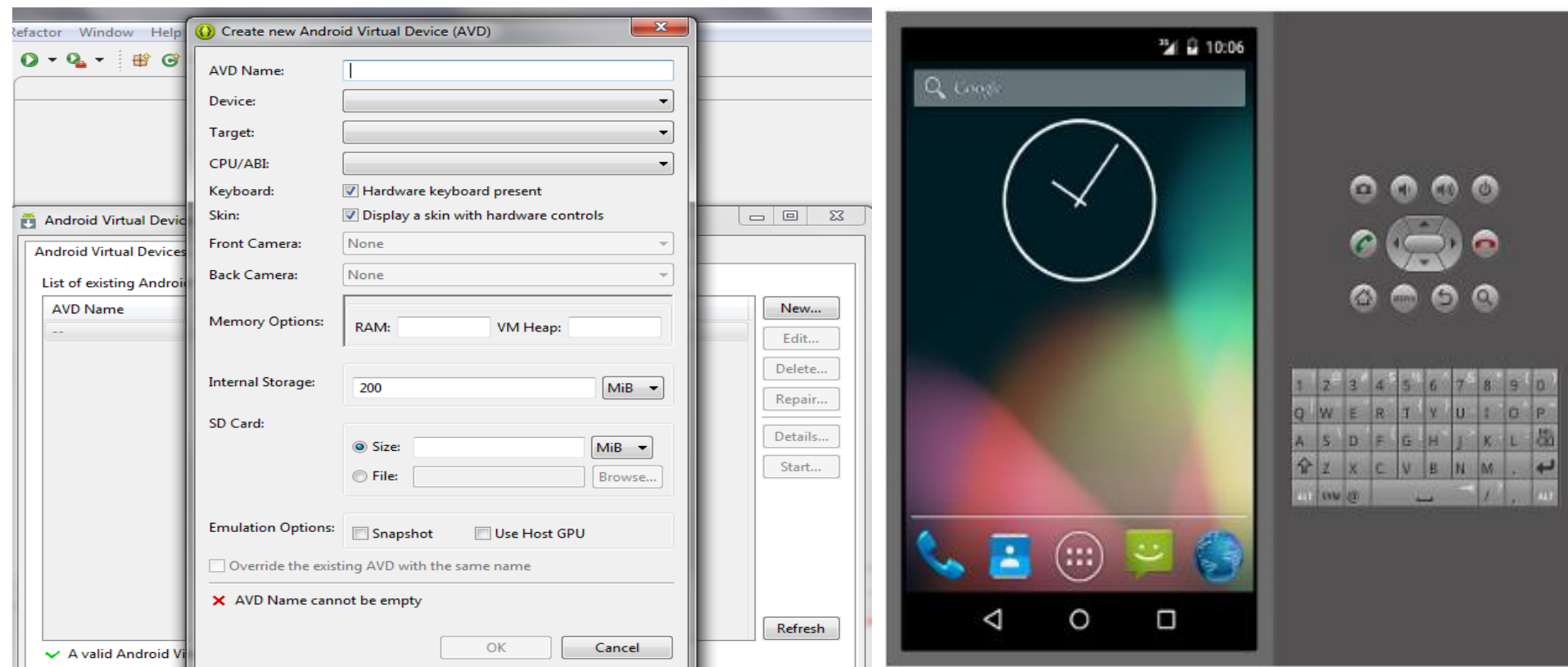
CIBLES/EMULATEURS

- ▶ AVD: «Android Virtual Device»
- ▶ Window-Android Virtual Device Manager



CIBLES/EMULATEURS

- ▶ AVD: «Android Virtual Device»
- ▶ Créez un AVD d'un API cible de votre choix



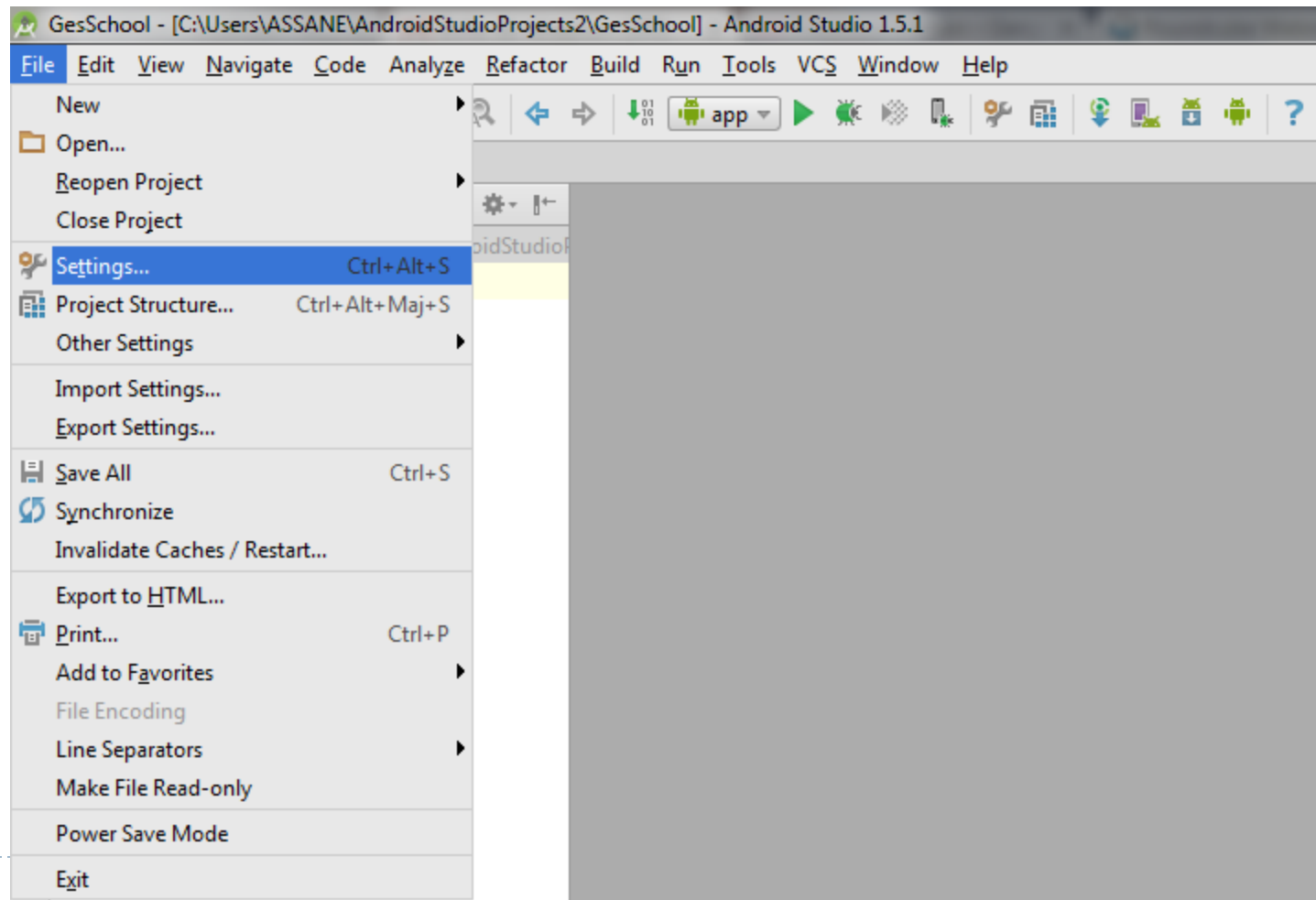
CIBLES/EMULATEURS: POINTS D'ATTENTION

- ▶ Tester sur un maximum de cibles différents
- ▶ Quand une nouvelle cible est disponible, testez votre application dessus
- ▶ Les émulateurs sont bien, mais parfois il faut mieux tester sur un vrai téléphone
- ▶ Vitesses différentes dans émulateur (dues aux caractéristiques de la machine et du type d'émulateur)

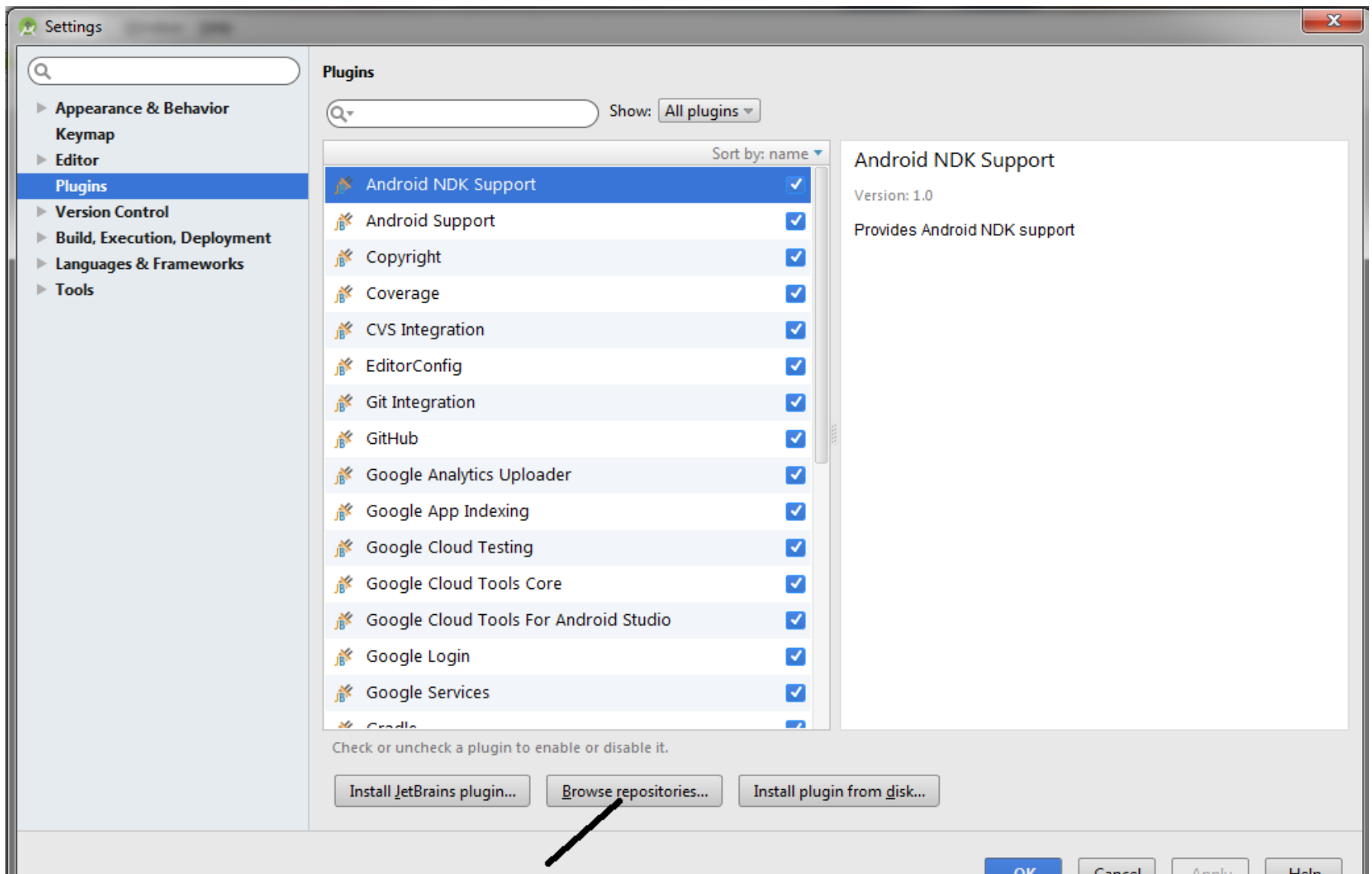
CREER UNE APPLICATION ANDROID

!

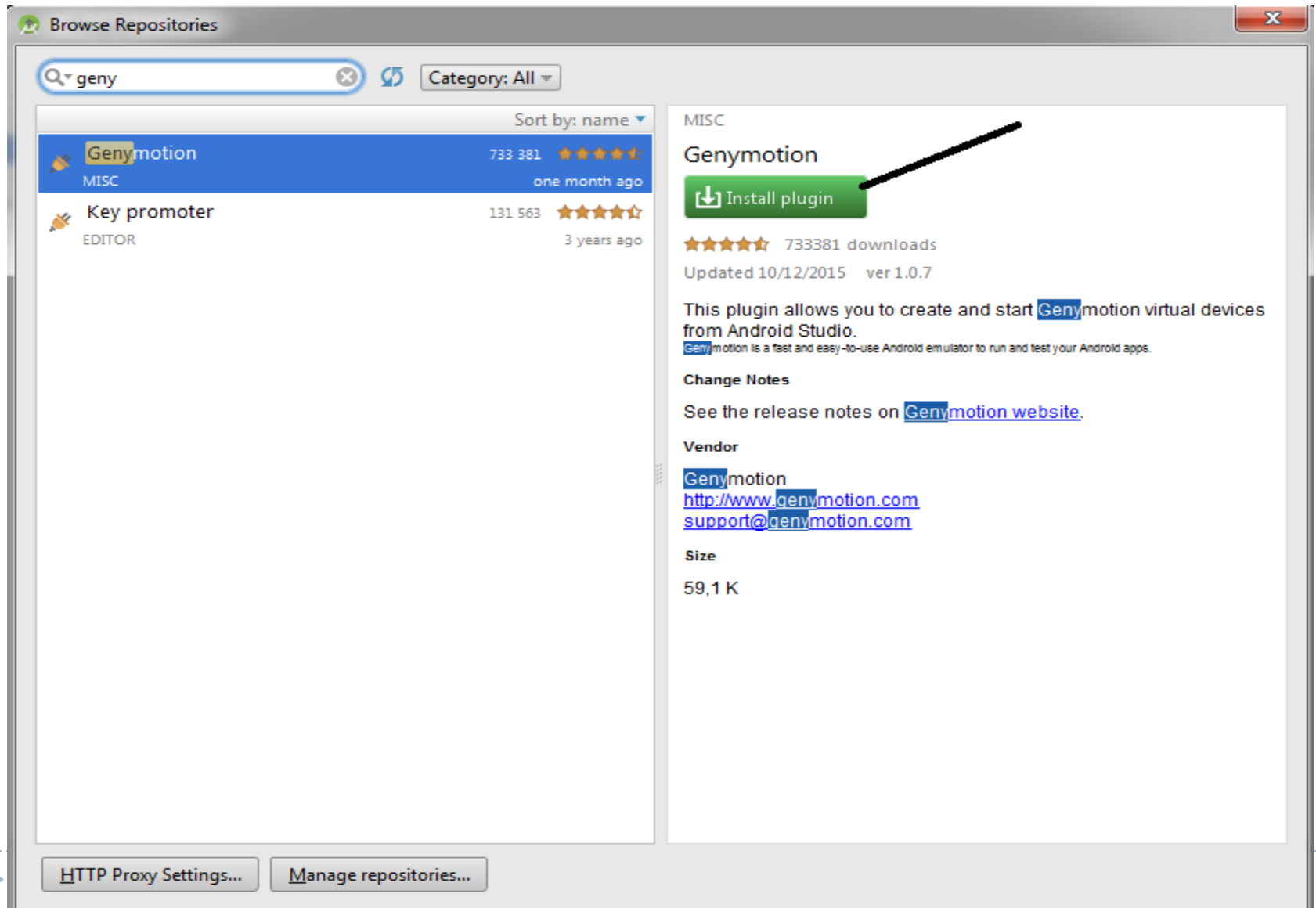
Genymotion



Genymotion



Genymotion



The screenshot shows the 'Browse Repositories' window in Android Studio. The search bar contains 'geny' and the category is set to 'All'. The results list 'Genymotion' (733,381 downloads, 5 stars, one month ago) and 'Key promoter' (131,563 downloads, 5 stars, 3 years ago). The 'Genymotion' plugin details are shown on the right, including the 'Install plugin' button, download statistics, update information, description, change notes, vendor information, and size.

Browse Repositories

Search: Category: All

Sort by: name

Plugin Name	Downloads	Rating	Updated
Genymotion	733 381	★★★★★	one month ago
Key promoter	131 563	★★★★★	3 years ago

Genymotion

MISC

[Install plugin](#)

★★★★★ 733381 downloads
Updated 10/12/2015 ver 1.0.7

This plugin allows you to create and start **Genymotion** virtual devices from Android Studio.
Genymotion is a fast and easy-to-use Android emulator to run and test your Android apps.

Change Notes

See the release notes on [Genymotion website](#).

Vendor

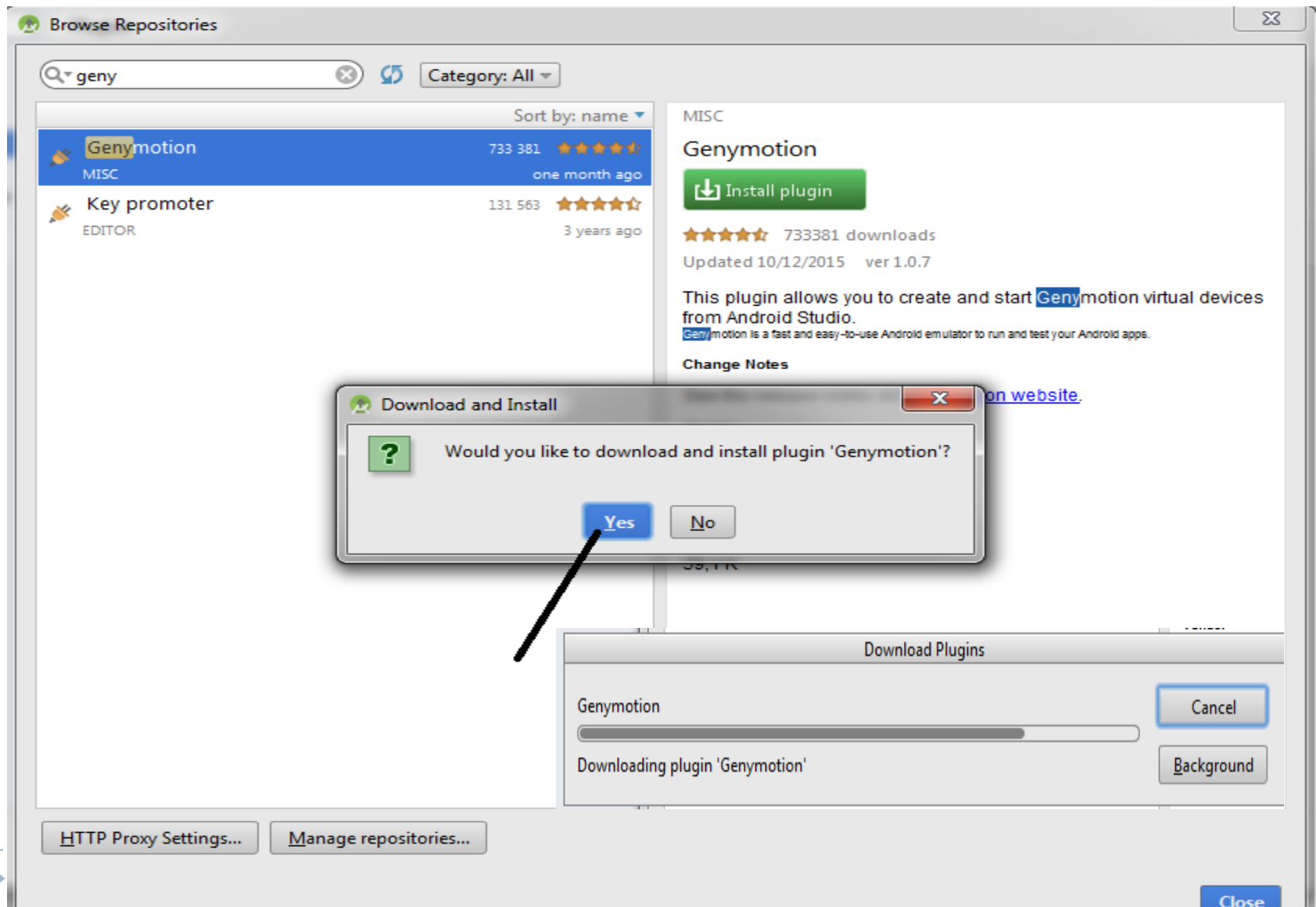
Genymotion
<http://www.genymotion.com>
support@genymotion.com

Size

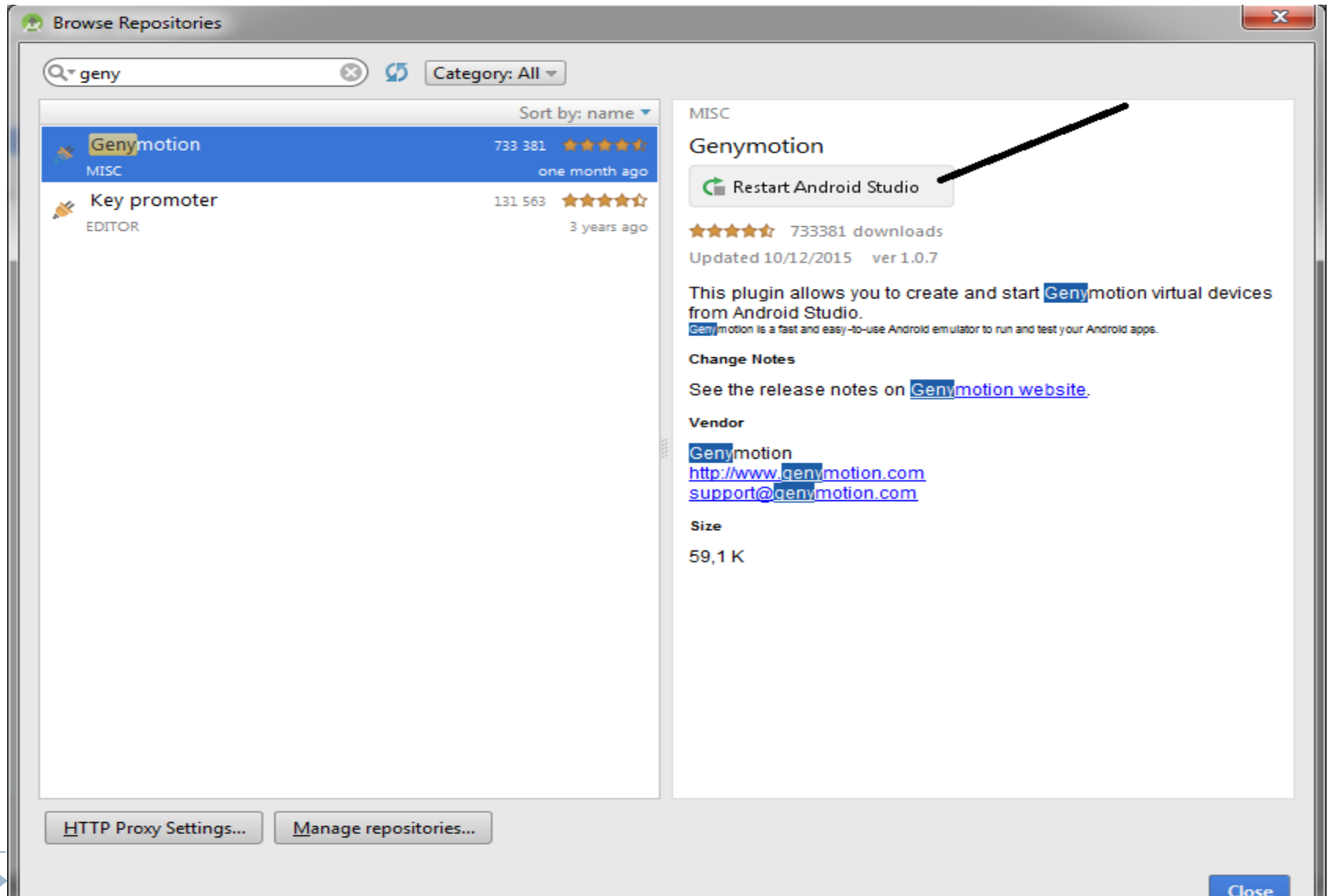
59,1 K

[HTTP Proxy Settings...](#) [Manage repositories...](#)

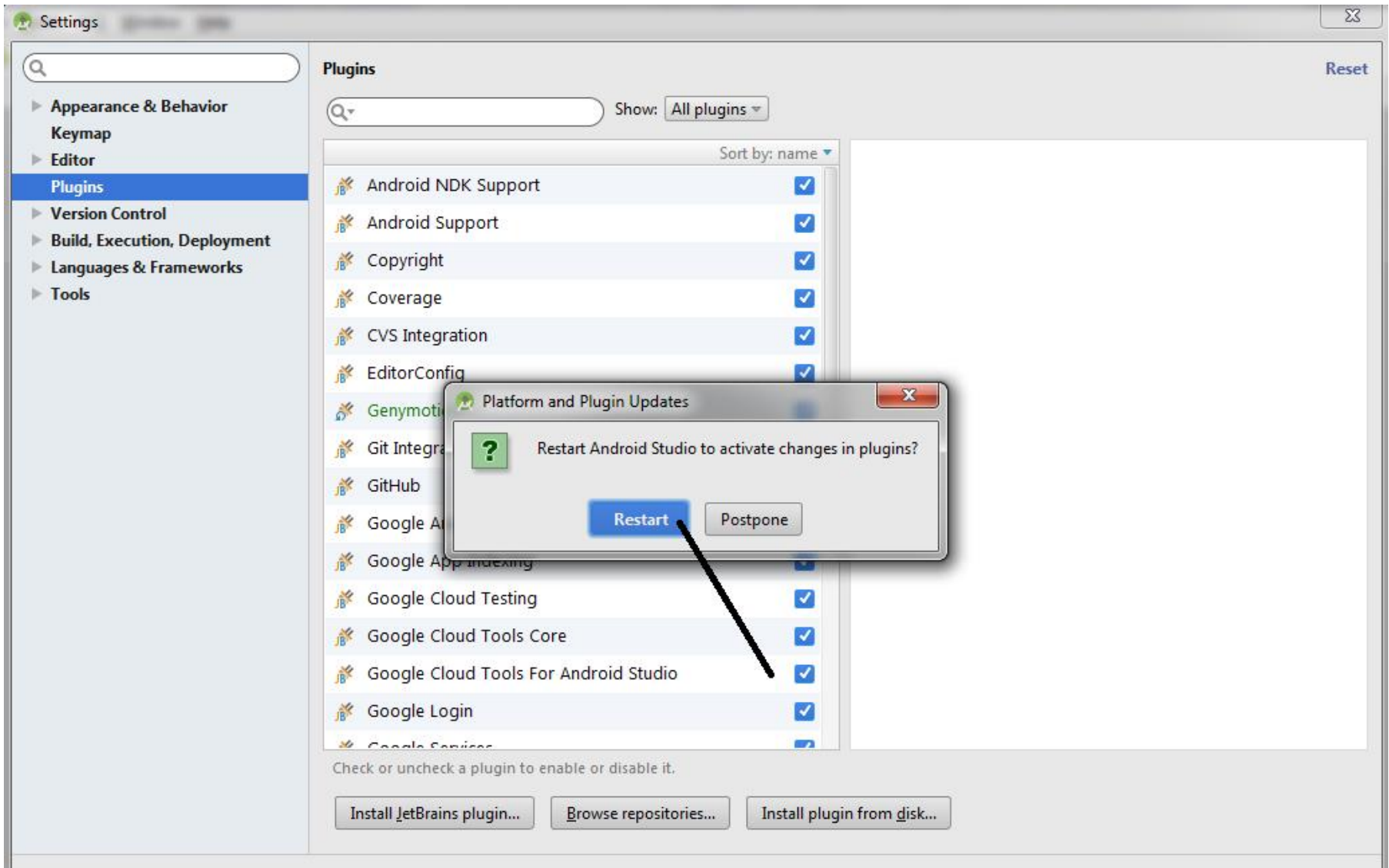
Genymotion



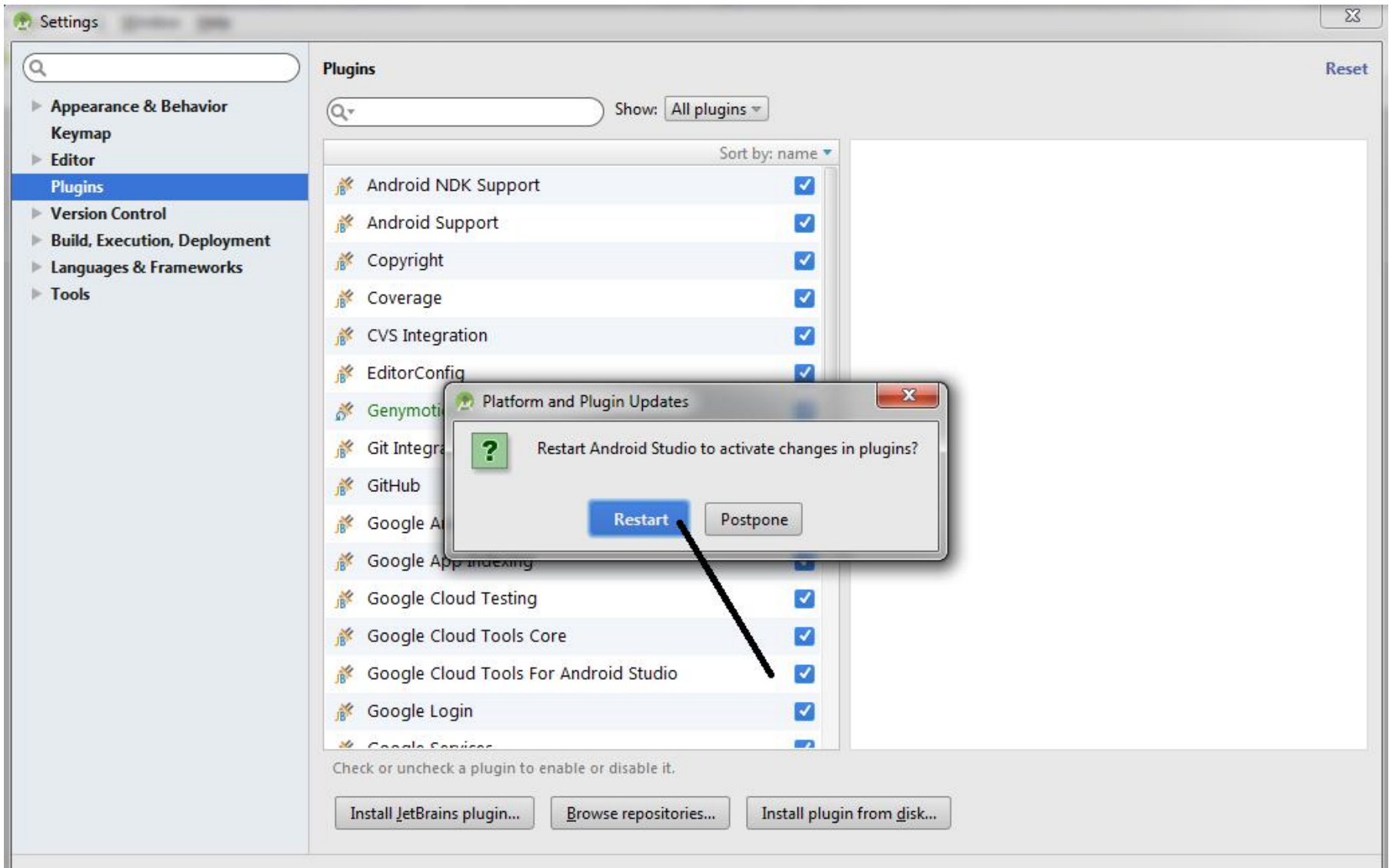
Genymotion



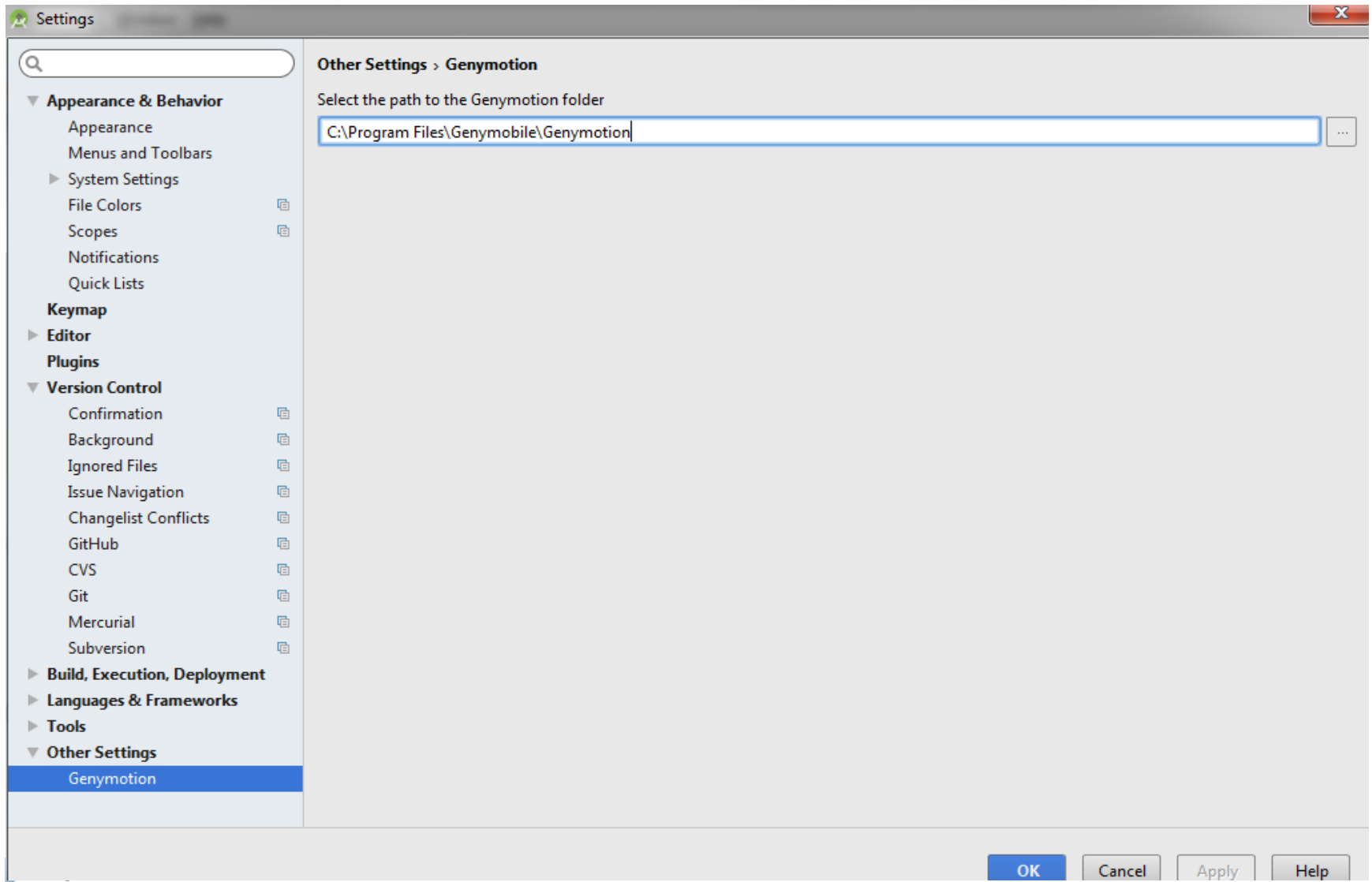
Genymotion



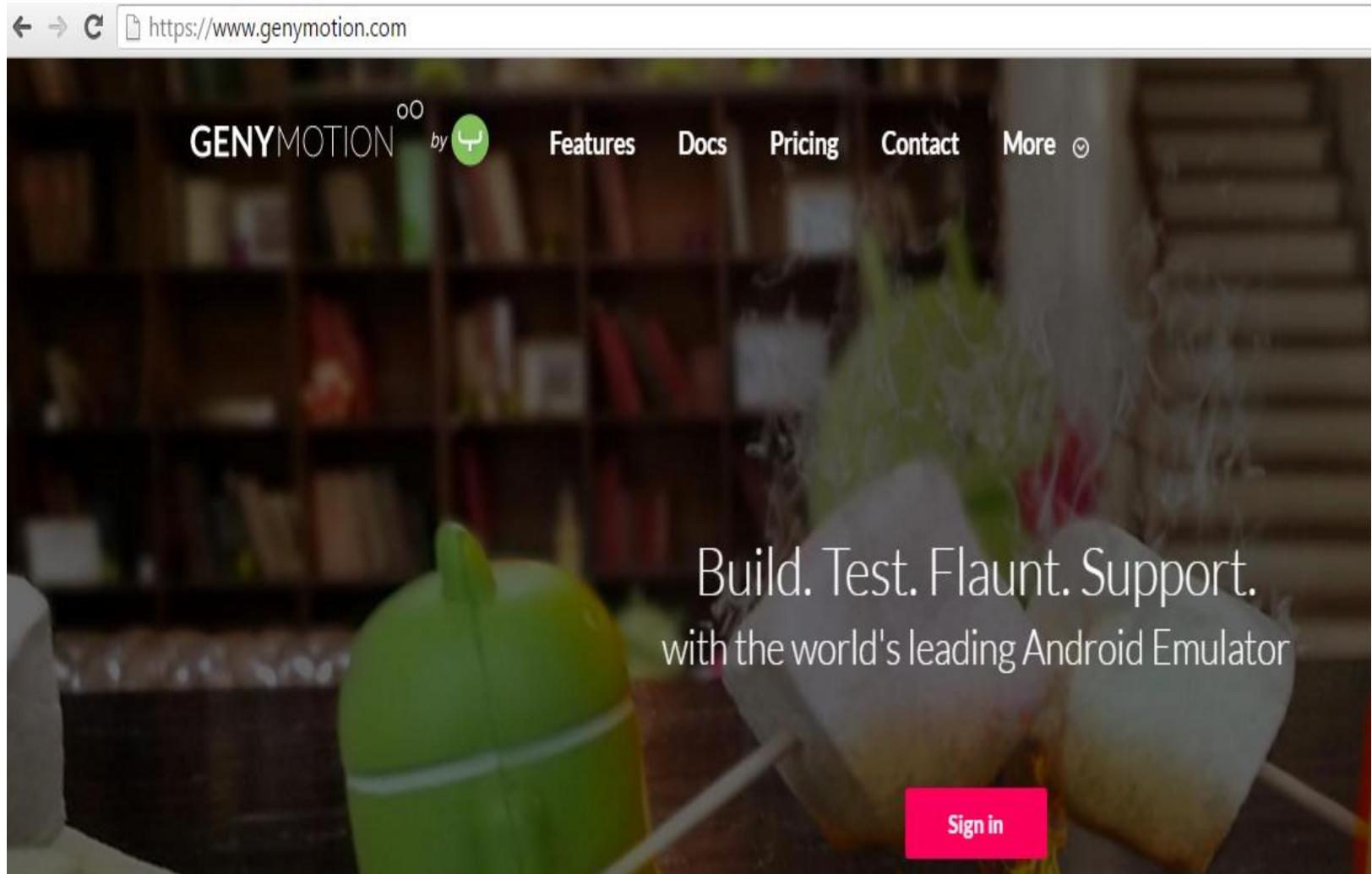
Genymotion



Genymotion



Genymotion



Genymotion

GENYMOTION^{OO} by 

Features

Docs

Pricing

Contact

More 

Sign in

Username

Password

[Forgotten your password?](#)

Sign in

Create an account



Genymotion

Sign up

Username

Personal use



Email Address

Development



Password

☐ Get our latest news & updates

☒ I accept terms of the [privacy statement](#)

Create an account

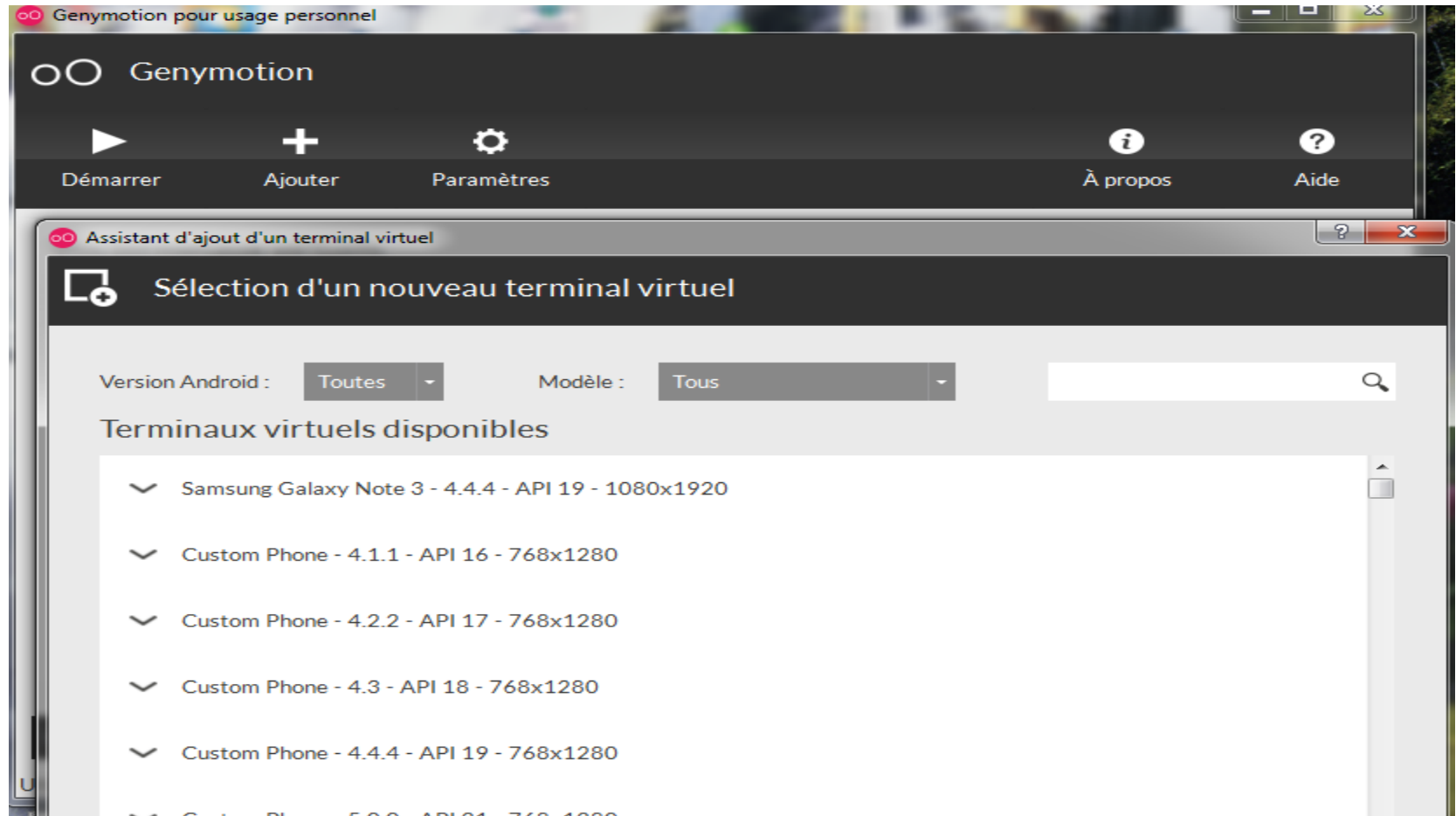
Genymotion

Activé le compte Genymotion depuis son email de création de compte

Lancé le genymotion qui était sur le Bureau



Genymotion



Apk location

