

Ministère de l'Enseignement Supérieur, de la Recherche Scientifique et de la Technologie

Institut Supérieur des Etudes Technologiques de Kébili



Travaux pratiques N°1: Mise en place d'un environnement de développement mobile

<i>Unité d'Enseignement : Département Technologies de l'Informatique</i>	<i>Elément Constitutif : Atelier Développement Mobile</i>
<i>Section : DSI3</i>	<i>Durée : 3h</i>

Objectif de TP :

Au cours de ce TP, on se propose de mettre en place l'environnement de développement mobile sous Android : installation d'Android Studio, configuration de SDK, création et lancement de l'émulateur Android.

1. Introduction à la programmation avec Android

Android est un système d'exploitation embarqué open-source pour smartphones, tablettes tactiles, PDA, smartwatches et terminaux mobiles, basé sur Linux.

Depuis le rachat d'Android par Google en 2005, ce système d'exploitation mobile est en constante évolution grâce au support fourni par un consortium composé d'entreprises très diverses (Samsung, HTC, Sony, LG...). Pendant les trois dernières années, le nombre d'applications disponibles pour les utilisateurs d'Android sur Play Store a doublé pour atteindre 2 millions applications.

1.1. Installation et outils

Pour créer des applications pour Android, il est nécessaire d'installer les éléments suivants :

- **JDK** (*Java Development Kit*) : Environnement de développement Java, qui permet de compiler et exécuter des applications écrites en Java.
- **Android Studio** : Environnement de développement spécialisé dans le développement d'applications Android.
- **Android SDK** : *Android Software Development Kit* : Le SDK fournit une API et un ensemble d'outils pour le développement d'applications sur Android. Il contient principalement un émulateur (**AVD** pour *Android Virtual Device*) qui permet de modéliser un appareil mobile réel en définissant les options logicielles et matérielles désirées. Le SDK est disponible en téléchargement pour les plateformes Linux, Mac et Windows à l'adresse suivante : <http://developer.android.com/sdk/index.html>

1.2. Installation d'Android Studio

Android Studio est un environnement de développement écrit en java. En outre, les applications développées au sein de cet environnement sont basées sur le langage java. Pour ces raisons, il est nécessaire d'installer le JDK qui permet de compiler des programmes Java. La JDK contient également une **JRE** (*Java Runtime Environment*), l'environnement d'exécution permettant d'exécuter des programmes écrits en Java.

Il est possible de télécharger JDK à partir du site suivant :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Pour installer Android Studio, il suffit d'aller sur <https://developer.android.com/studio/index.html> et de télécharger la version correspondante à votre plateforme.

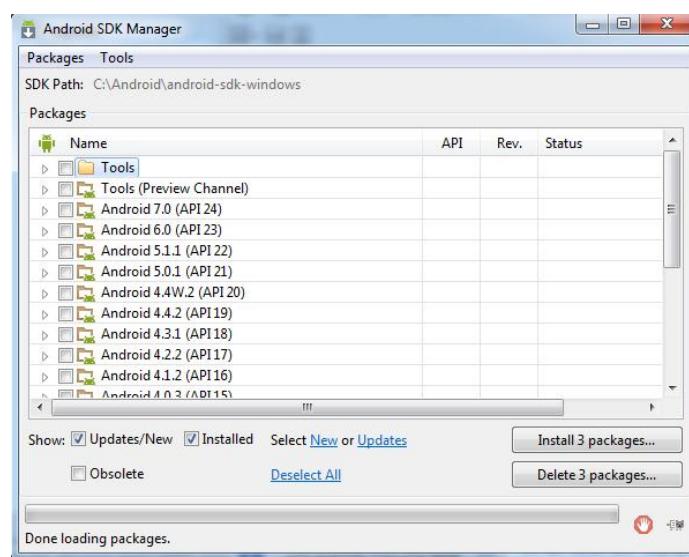
Il est possible de récupérer Android Studio sous la forme d'un fichier compressé. Une fois ce fichier décompressé, allez vers le répertoire *bin* et lancer le fichier *studio*.



1.3. Installation du SDK

Le SDK Android est donc un ensemble d'outils que met à disposition Google afin de permettre le développement des applications mobiles pour Android. Pour installer le SDK Android, il suffit d'aller sur <https://developer.android.com/studio/index.html> et de télécharger la version correspondante à votre plateforme.

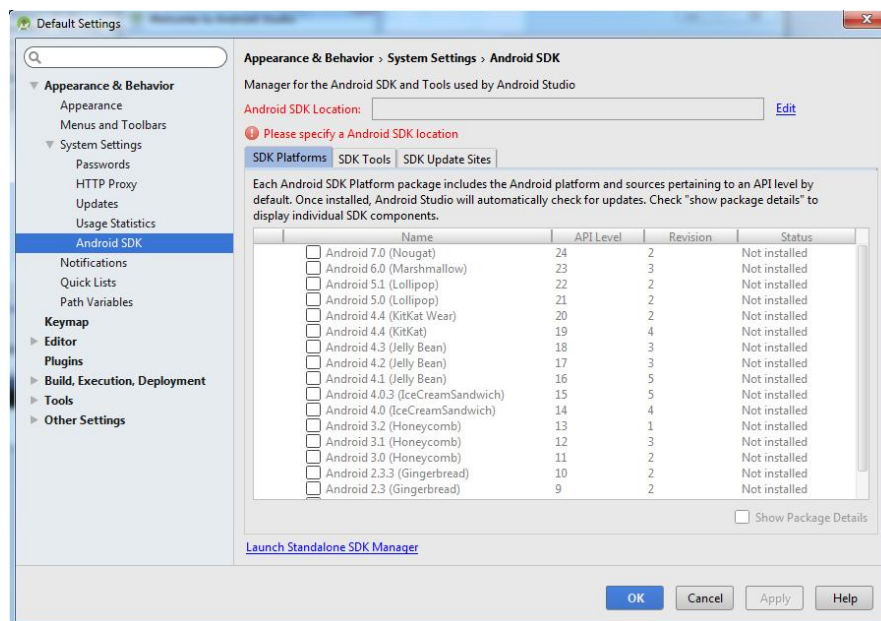
Il est possible de récupérer le SDK Android sous la forme d'un fichier compressé. Une fois ce fichier décompressé, exécutez le fichier *SDK Manager*. Le gestionnaire du SDK est lancé :



L'étape suivante est donc d'installer les plateformes Android nécessaires. Il suffit de sélectionner les plateformes qu'on désire installer et cliquer sur *Install Packages...*

1.4. Configuration de SDK sous Android Studio

La première étape à faire est d'indiquer le chemin de répertoire SDK dans Android Studio. Pour ce faire, cliquez sur *Configure -> SDK Manager*, ensuite spécifiez l'emplacement de SDK Android :

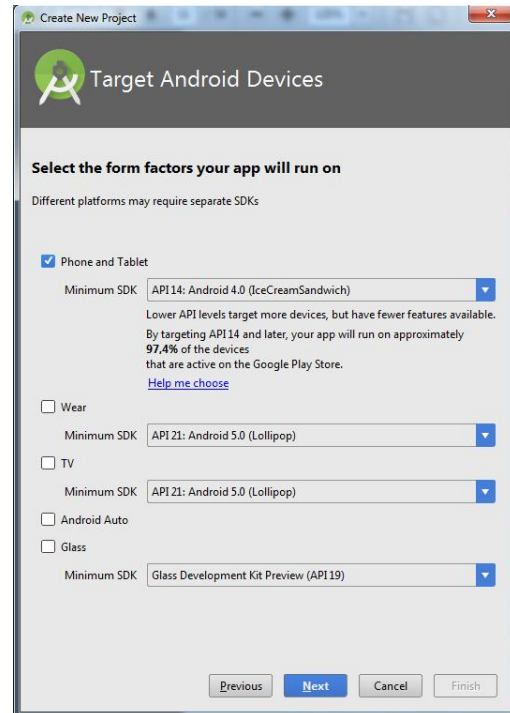
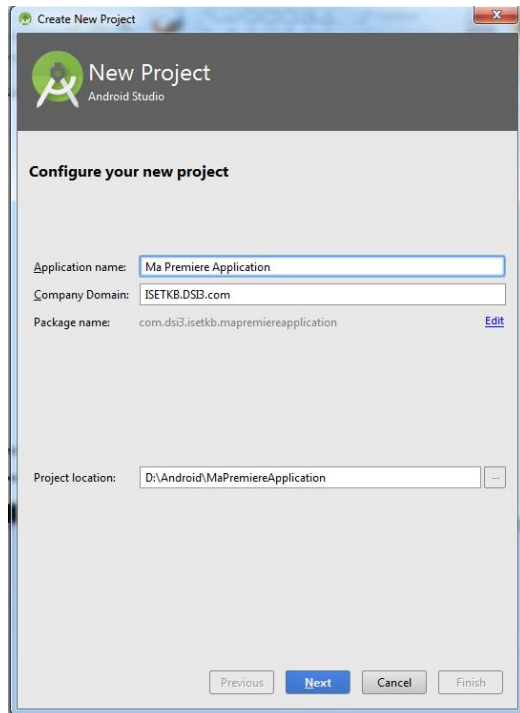


Une fois le SDK est configuré, il est possible de démarrer Android Studio.

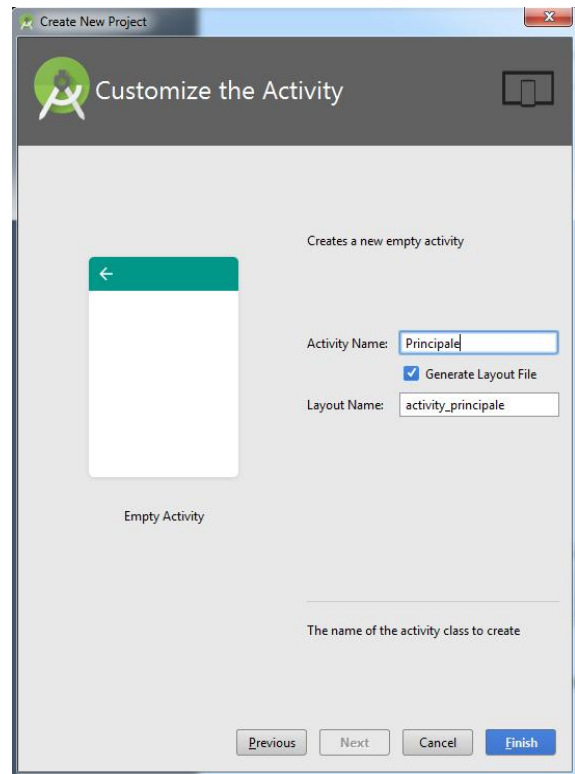
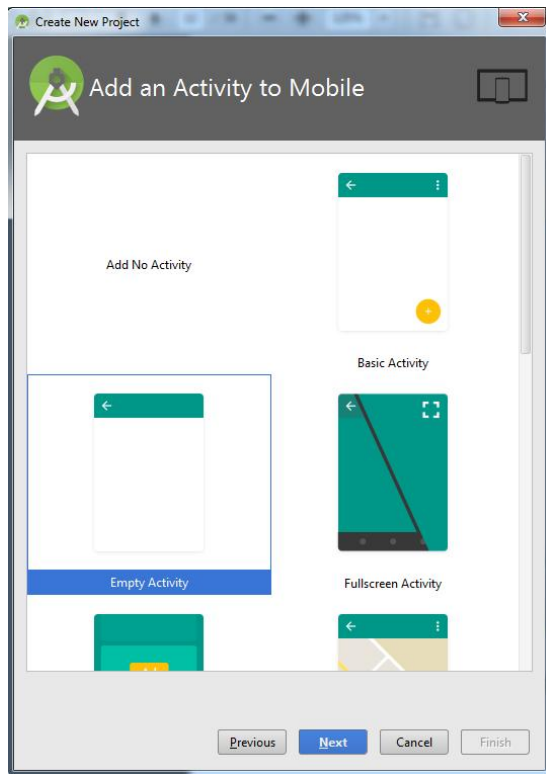
2. Ma première application Android

Pour créer un projet Android :

- Dans le menu *Quick Start*, sélectionnez *Start a new Android Studio Project*, et renseignez les informations comme indiqué dans la figure suivante :



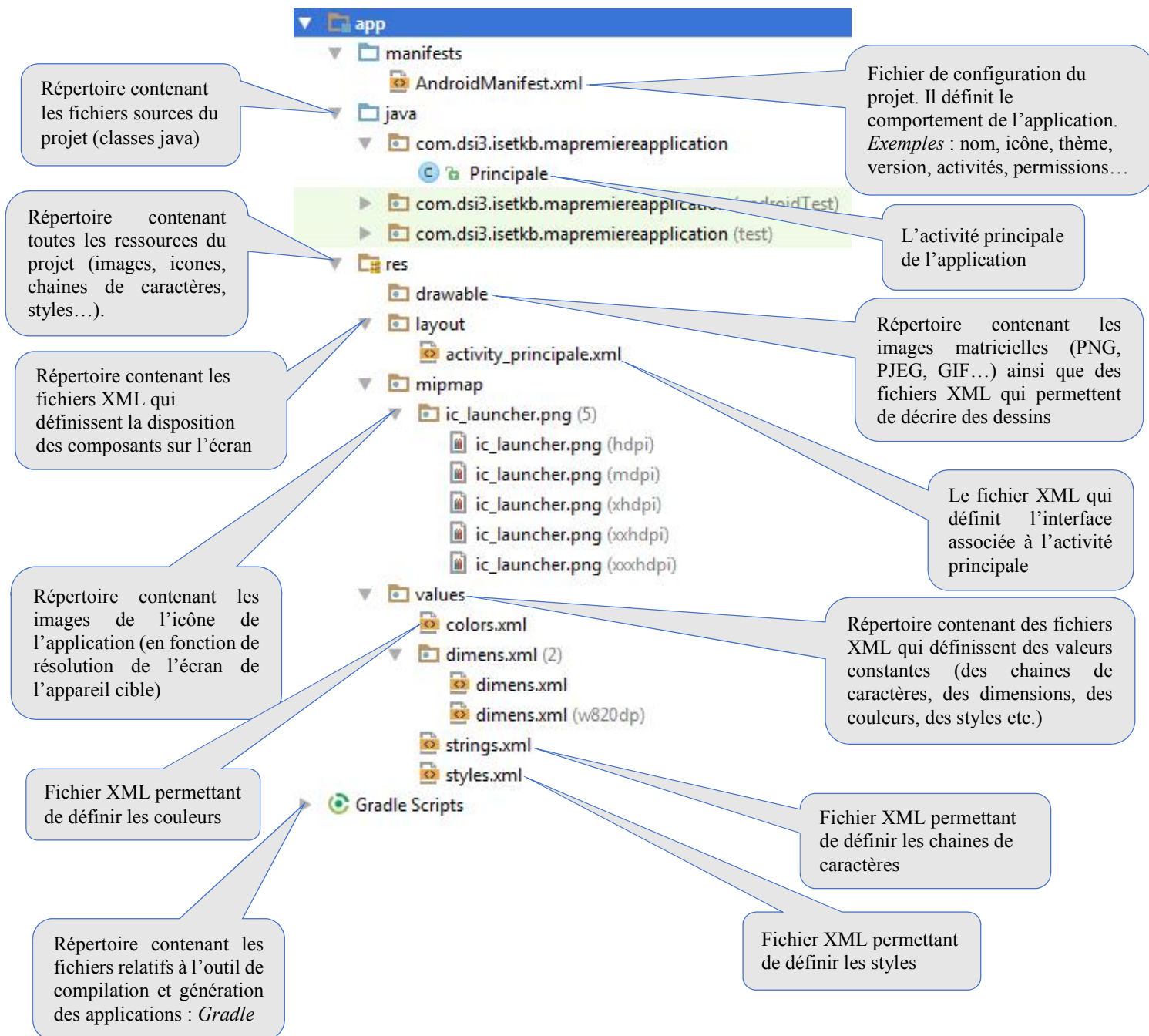
- Cliquez sur *Next* puis remplissez les champs :
 - ✓ **Application name** : c'est le nom qui va apparaître dans la liste des applications sur l'appareil et dans le Play Store.
 - ✓ **Company domain** : c'est un qualifiant qui apparaîtra dans le nom du package
 - ✓ **Package name** : il est utilisé comme identifiant de l'application, il permet de considérer différentes versions d'une application comme étant une même application. Il doit être unique parmi tous les packages installés sur le système.
 - ✓ **Minimum required SDK** : c'est la version Android la plus ancienne sur laquelle l'application peut tourner. Il faut éviter de remonter trop en arrière, ça réduirait les fonctionnalités que vous pourriez donner à votre application.
- Cliquez sur *Next*. Nous arrivons à la création d'une activité (un écran avec une interface graphique). Sélectionnez *Empty Activity* et cliquez *Next*.



- Cliquez sur *Finish*, le projet est créé.

Vous pourriez choisir l'utilisation de *fragments*, mais pour faire simple nous poursuivrons sans fragments. Chaque activité dispose d'un *layout* qui définit la façon dont les composants seront disposés sur l'écran. Une activité peut être divisée en portions (ou fragments) chacune ayant son propre *layout*. La notion de fragment a été introduite pour favoriser la réutilisabilité de morceaux d'activité (un fragment peut être défini une fois et réutilisé dans plusieurs activités).


Voyons Maintenant l'arborescence d'un projet Android :

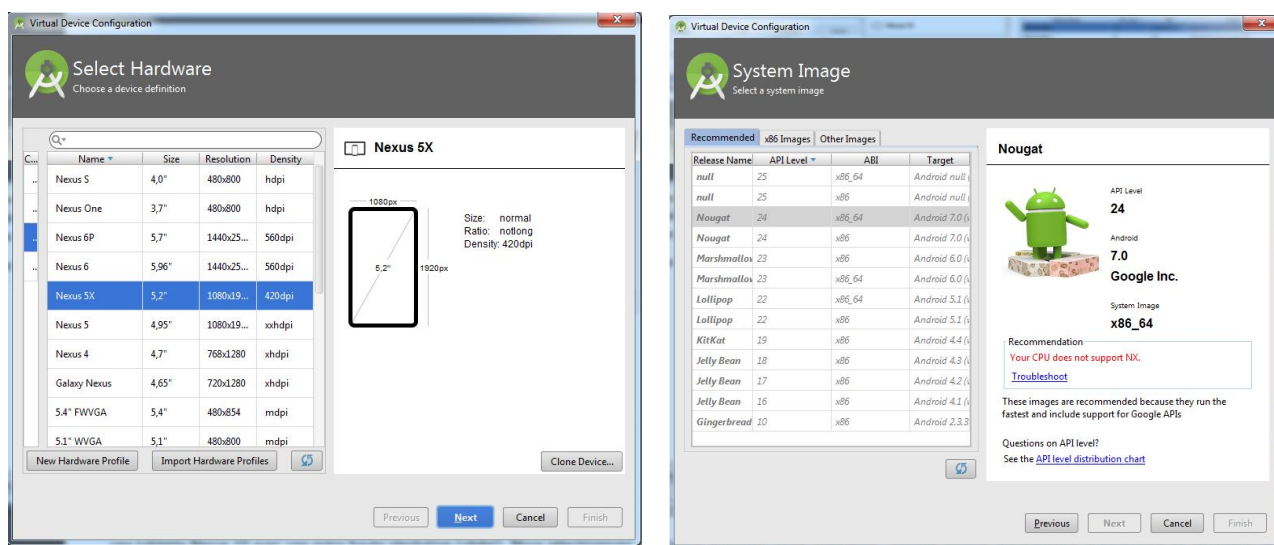


3. Exécution de l'application


3.1. Sur un émulateur (AVD)

Un émulateur permet de reproduire le comportement d'un appareil réel d'une façon virtuelle. L'utilisation d'un émulateur nous évite d'avoir à charger à chaque fois l'application dans un appareil pour la tester. On pourra ainsi lancer l'application dans l'IDE et elle s'exécutera sur un appareil virtuel appelé *Android Virtual Device* (AVD) qui émule le comportement d'un téléphone, une tablette ou autre.

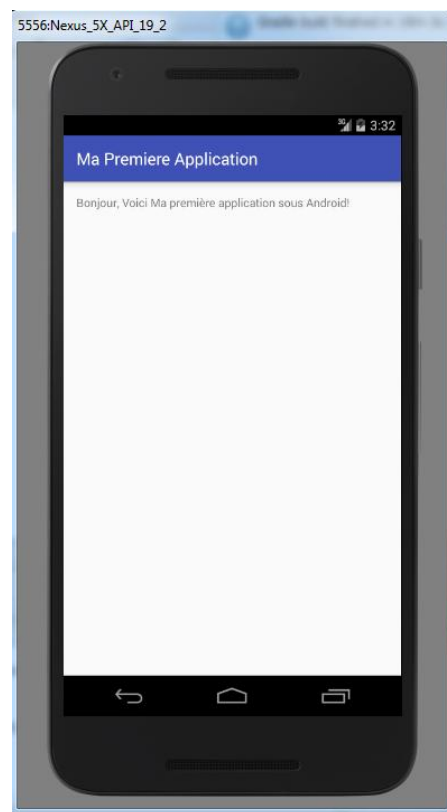
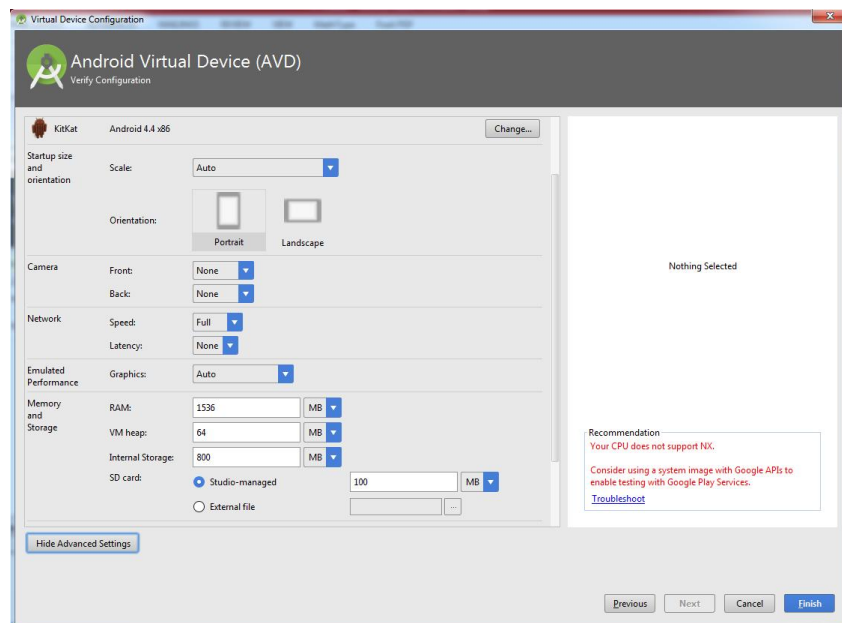
Pour configurer un émulateur, allez dans *Tools > Android > AVD Manager* ou bien cliquer sur . Un émulateur existe par défaut, mais vous pourriez rajouter d'autres appareils en cliquant sur *Create Virtual Device*. Ensuite sélectionnez le processeur qui sera émulé. Il est possible de télécharger d'autres images systèmes avec d'autres versions Android.



En cliquant sur *Next*, vous avez ensuite la possibilité de configurer d'autres paramètres. Pour accéder aux paramètres avancés de l'appareil émulé, cliquer sur *Show Advanced Settings*. Vous pouvez modifier la taille de mémoire vive (RAM), mémoire de stockage interne, VM heap, taille de carte SD...

Pour lancer l'exécution sur l'émulateur, appuyez sur le bouton d'exécution  et sélectionnez l'émulateur sur lequel vous souhaitez lancer l'application. L'émulateur se lance, ça peut prendre

quelques minutes soyez patients. Rassurez-vous, vous n'aurez pas à le relancer à chaque fois que vous compilez votre projet, laissez-le ouvert et à chaque fois que vous compilez et relancez votre application, elle pourra être chargée dans l'émulateur en cours.



Remarque :

Au cours de la sélection de processeur (*X86, ARM, MIPS...*) à émuler, si vous disposez d'un processeur Intel qui prend en charge la technologie de virtualisation, cette dernière est assez souvent désactivée par défaut dans le bios de votre ordinateur, il faut choisir l'image Intel X86 pour assurer le lancement rapide de l'AVD. Dans ce cas, pour accélérer la vitesse de l'émulation vous devez installer le package *Intel HAXM (Hardware Accelerated Execution Manager)*, disponible sous l'adresse suivante :

<https://software.intel.com/en-us/android/articles/intel-hardware-accelerated-execution-manager>

Si non, vous pouvez sélectionner l'architecture ARM ou MIPS. Dans ce cas, l'émulateur AVD sera plus lent.

Une autre alternative est d'utiliser [Genymotion](#) : un émulateur Android rapide et gratuit pour l'utilisation personnelle. Dans ce cas, il faut installer le plugin nécessaire dans Android Studio.

3.2. Sur un appareil réel

Connectez l'appareil par câble USB à l'ordinateur et installez le pilote si nécessaire. Activez l'option de débogage USB sur votre appareil (dans les paramètres, sous développement ou option de développement). Lancez l'application depuis Android Studio comme précédemment. Si on vous demande de choisir l'appareil, sélectionnez *Choose a running device*, puis votre téléphone ou tablette. Android Studio installera l'application sur votre appareil et la lancera.

Une fois que votre application est compilée, un fichier *.apk* est créé dans le dossier *app\build\outputs\apk* de votre répertoire de travail. C'est l'exécutable de votre application. C'est ce fichier que vous devez déployer pour distribuer votre application. Le contenu de ce fichier peut être inspecté à l'aide de n'importe quel logiciel standard de compression/ décompression de fichiers.