

Tuto 8

Site: lms.univ-cotedazur.fr
Cours: Realite virtuelle - EIMAD919
Livre: Tuto 8

Imprimé par: Theo bonnet
Date: vendredi 28 février 2020, 14:59

Table des matières

1. Déployer le projet sur un smartphone

1.1. Sous macOS

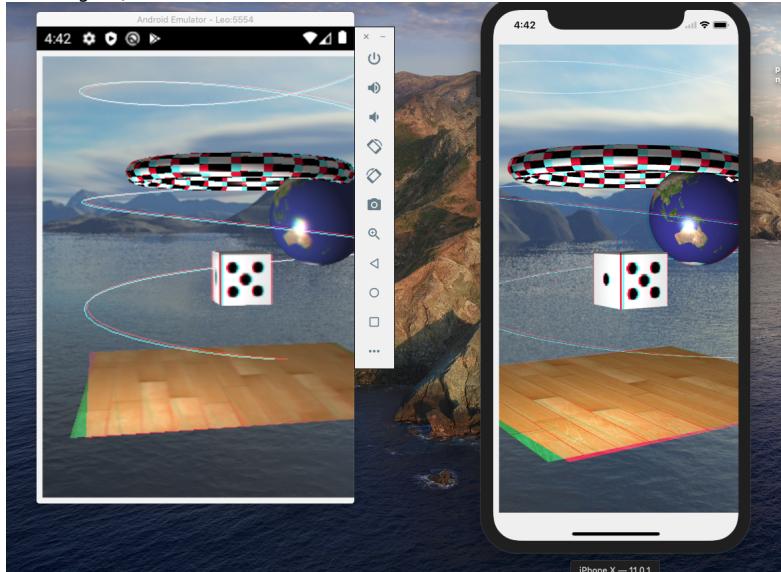
1.2. Sous Windows

1.3. Sous Linux

2. Mode paysage

1. Déployer le projet sur un smartphone

Le but de cette dernière session est de voir comment déployer une application OpenGL sur un smartphone en utilisant la technologie Qt.



Notre projet de base est celui de la session 5 : avec la caméra stéréoscopique mais sans la bibliothèque assimp. Le projet de départ est dans le final du TP5 avec quelques modifications pour le rendre compatible avec OpenGL ES :

- plus aucun appel à la fonction `glPolygonMode` qui n'est pas compatible avec OpenGL ES ;
- dans tous les shaders, toutes les variables doivent avoir une déclaration de précision : j'ai choisi donc de mettre `highp` partout.
- dans l'interface graphique, il n'y a plus de contrôles latéraux : la totalité de l'écran est occupée par le `RVWidget`.

A part ces modifications, le code du TP5 est totalement compatible avec iOS et Android, une fois que l'on a bien configurés les *kits* correspondants dans Qt.

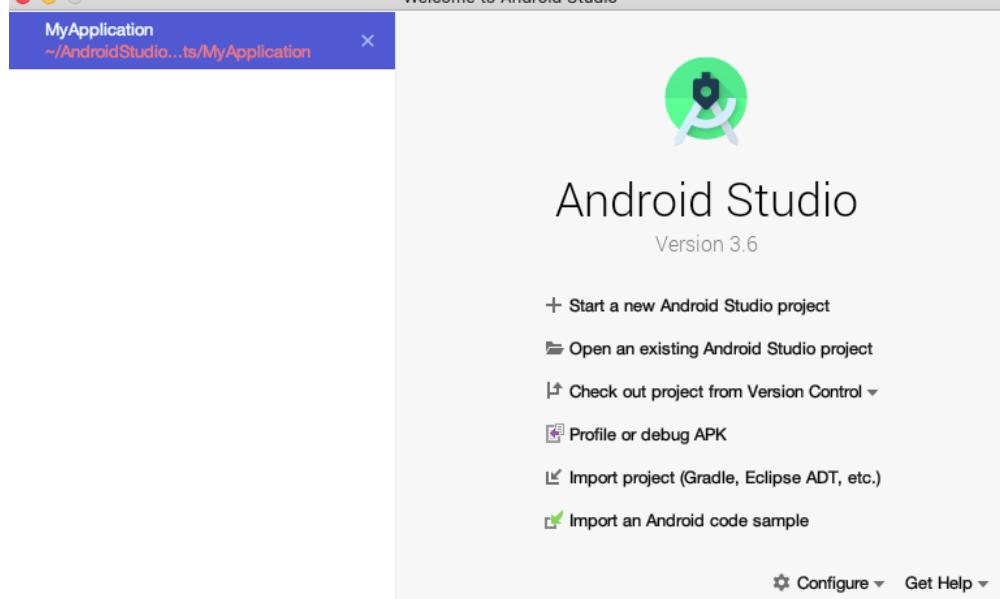
1.1. Sous macOS

Déployer sur android

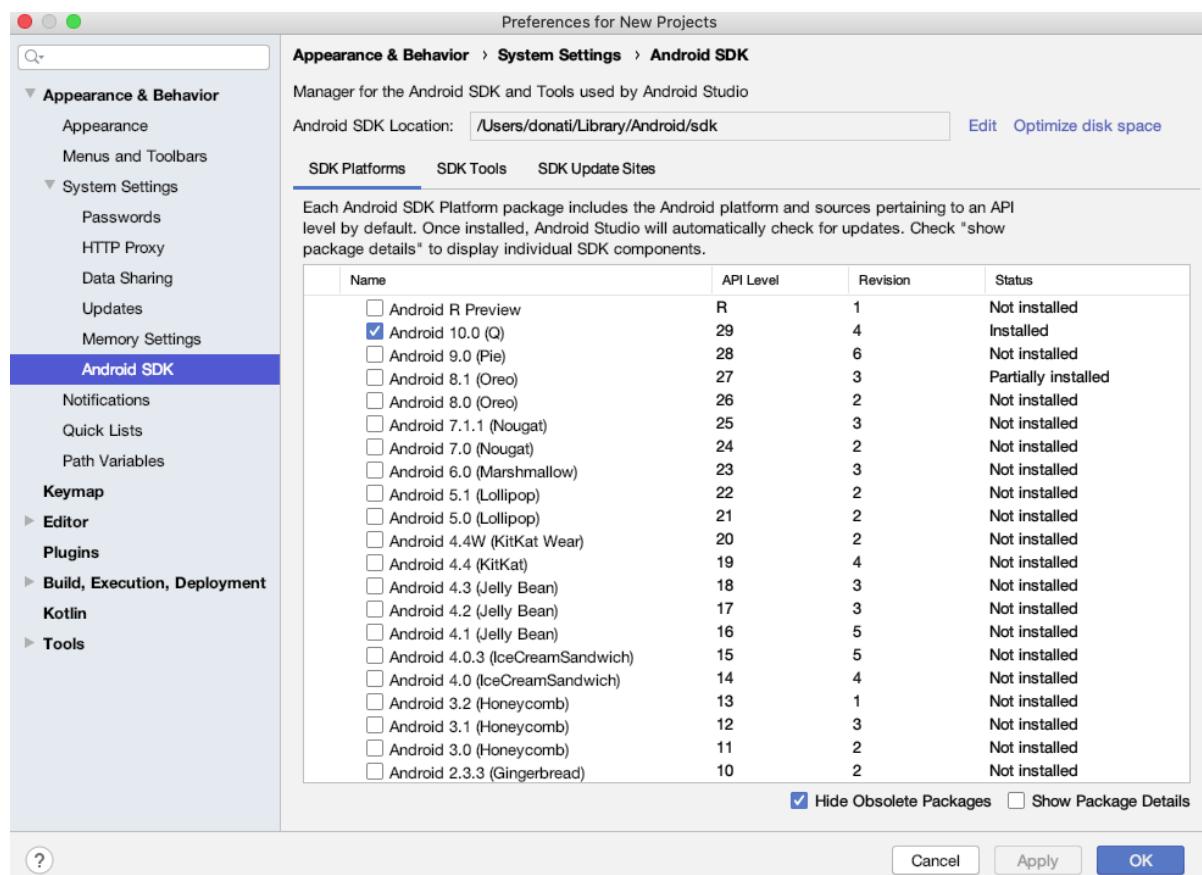
Les étapes sont décrites sur le site de Qt à <https://doc.qt.io/qt-5/android-getting-started.html>
Les éléments nécessaires sont :

- Les **Android SDK Tools** qui sont disponibles en installant [Android Studio](#)
- Le **Android NDK** (pour Native Development Kit) qui permet de développer en C/C++ sous Android. L'installation va aussi se faire via Android Studio
- Le [Java SE Development Kit \(JDK\)](#).

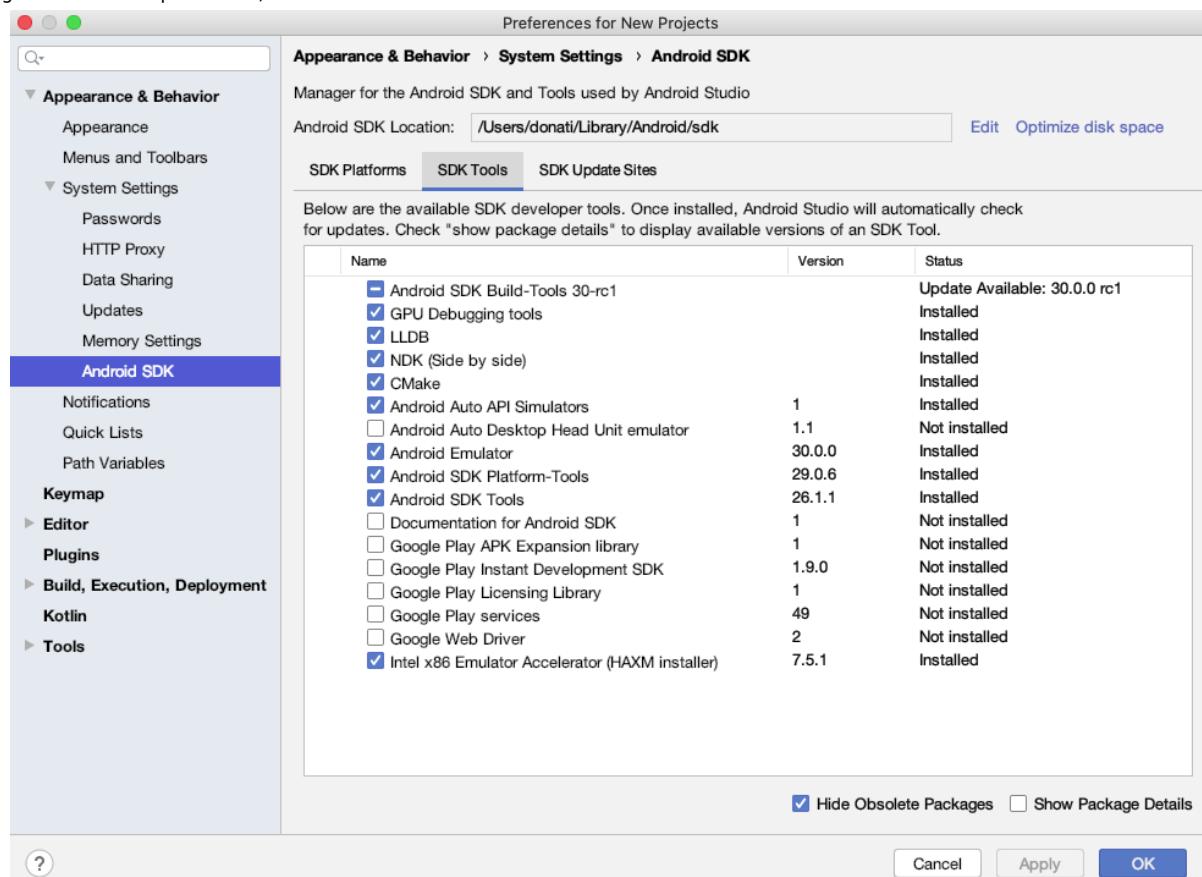
Après avoir lancé **Android Studio**, sur la page de garde, cliquer sur le bouton `Configure` et choisir `SDK Manager` :



- dans la rubrique `SDK Platforms`, sélectionner et installer la dernière version stable du SDK (Android 10.0 (Q) - API 29)

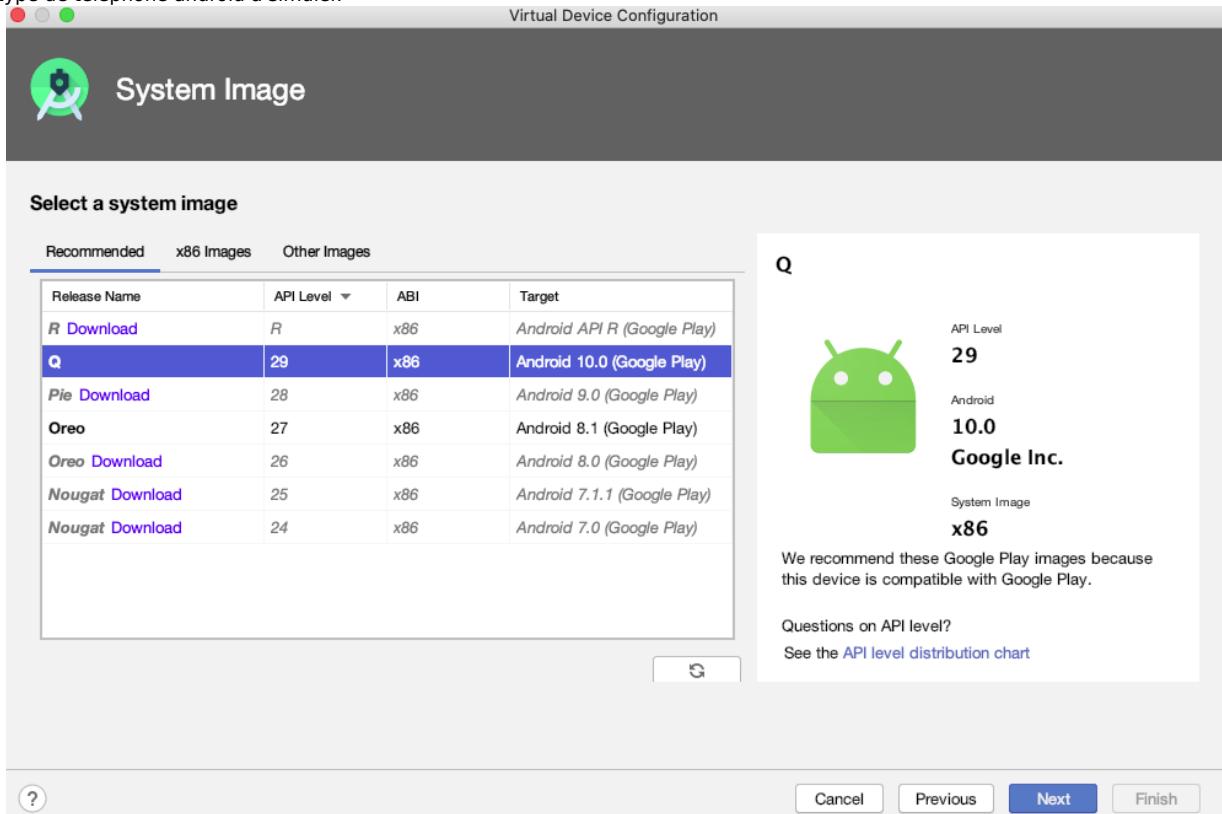


- dans la rubrique **SDK Tools**, cocher et installer **Android SDK Tools**, **NDK (Side by side)** et **Android Emulator** (pour le reste garder les cases par défaut). **Attention :** bien noter le chemin où le SDK Android sera installé.

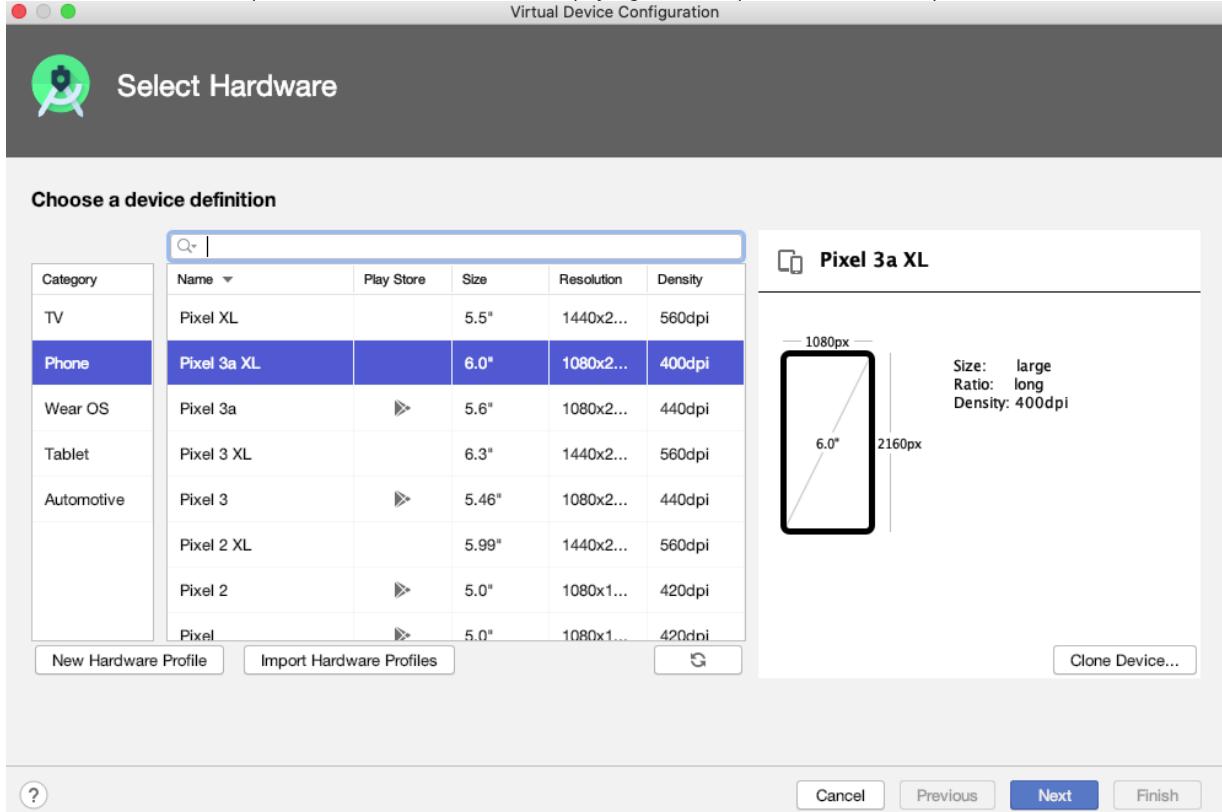


Toujours à partir de la page de lancement de Android Studio, dans **Configure** choisir **AVD Manager** qui va permettre de choisir les

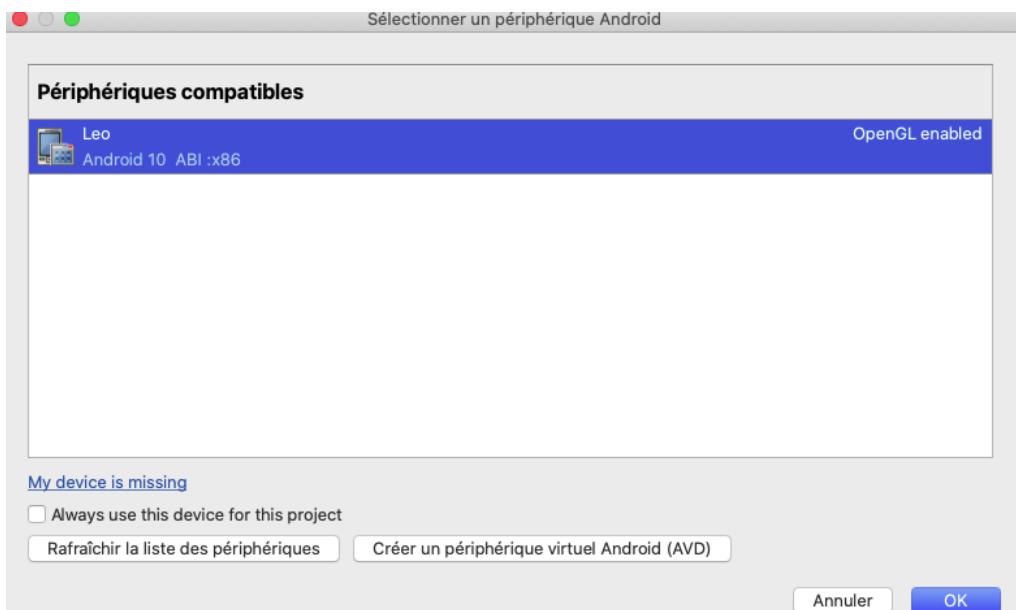
téléphones à simuler dans l'émulateur (AVD = Android Virtual Device). Avec le bouton + Create Virtual Device on peut choisir le type de téléphone android à simuler.



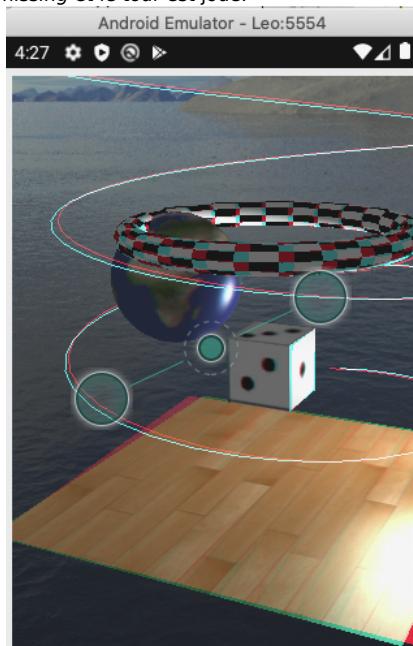
La chose importante c'est que ce soit un téléphone compatible avec la version 29 de Android et une architecture x86. On peut aussi choisir, à ce moment, que l'émulateur démarre en mode paysage (landscape) au lieu du mode portrait.



On n'a plus besoin d'Android Studio qui peut être fermé maintenant.
Ensuite, après toutes ces installations (y compris le JDK), il faut intégrer toutes ces informations dans **QtCreator**.
Ouvrir donc le projet du Tuto8 : si tout va bien (c'est rare) alors parmi les kits de configuration pour le projet vous pouvez choisir des Kits pour Android et en particulier **Android for x86** et alors en compilant le projet vous pouvez choisir (ou créer) un périphérique Android.



qui peut être soit simulé soit un périphérique réel connecté à votre ordinateur (si vous ne le voyez pas cliquer sur *My device is missing* et le tour est joué).

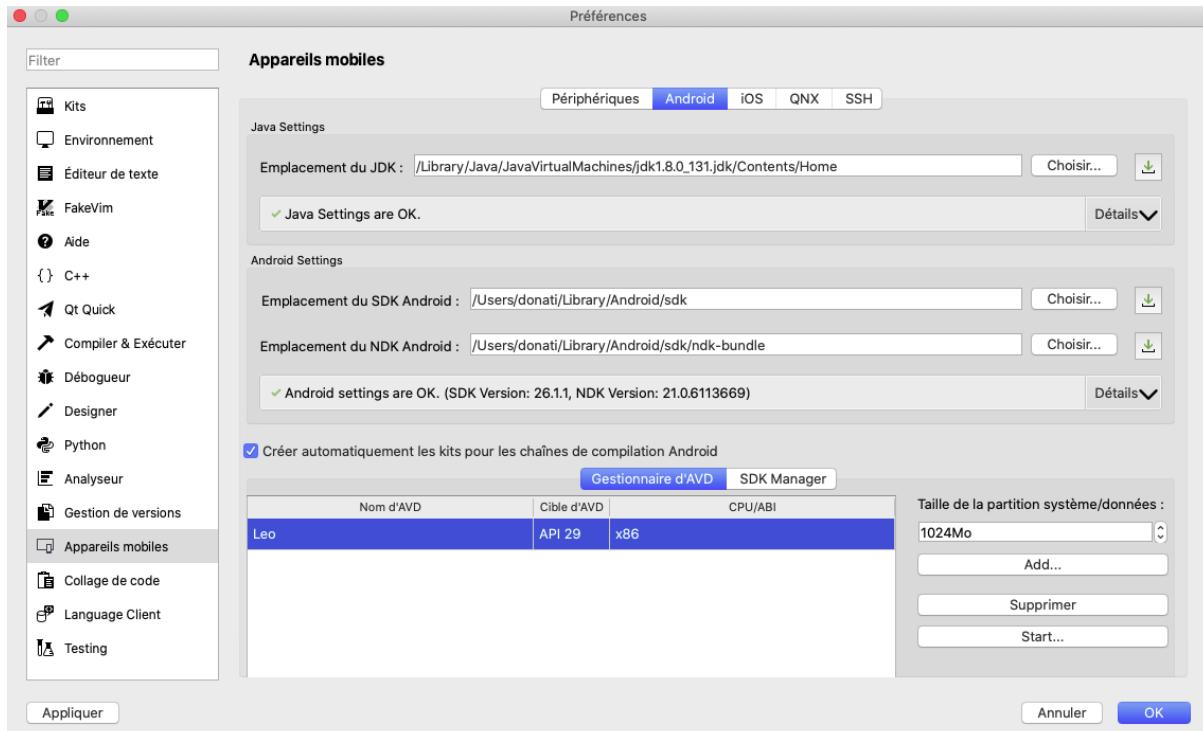


Si le kit Android ne vous est pas proposé ou si le kit n'est pas opérationnel, voici les étapes à suivre : dans `qtcreator/Preferences` choisir la rubrique `Kits`

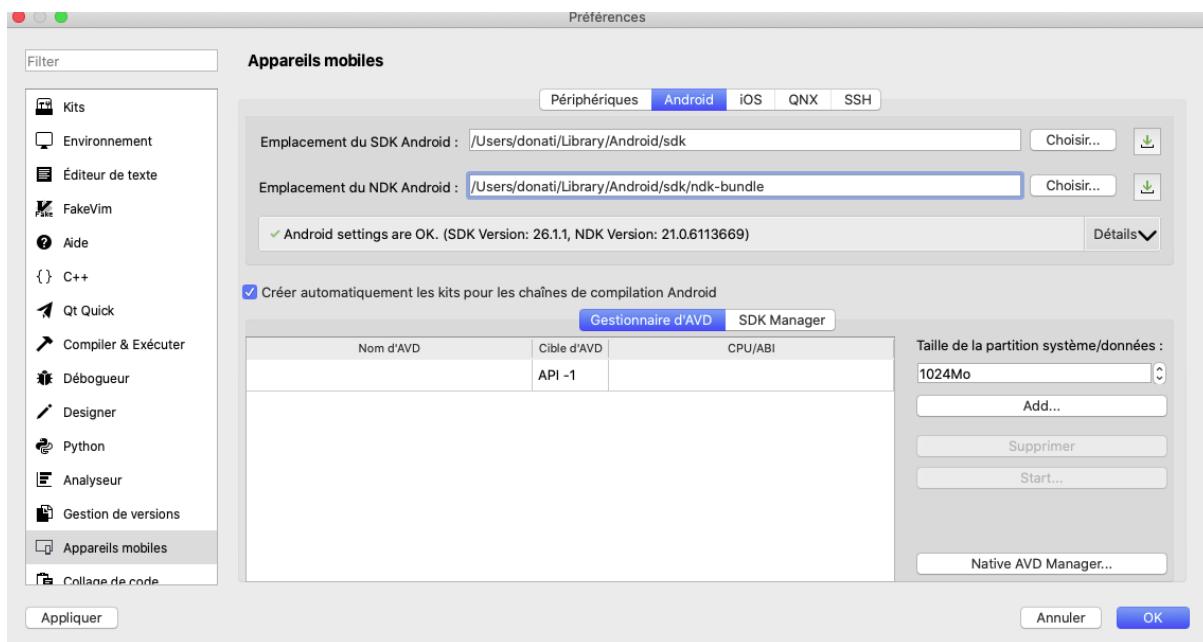
- Si les kits Android ne sont pas proposés (x86, arm64 ou armv7) il faut lancer le Qt Maintenance Tool avec l'option `Check for Updates` dans le menu `qtcreator`. Dans Qt Maintenance Tool il faut choisir l'option `Ajouter ou supprimer des composants` pour installer les kits Android



- Si les kits sont bien là, alors dans la rubrique Appareils mobiles puis Android il faut informer Qt de l'emplacement du JDK, du SDK Android et du NDK Android jusqu'à ce que l'interface indique que *Java Settings are OK* et *Android Settings are OK*.



Plus bas, dans la même page vous avez un gestionnaire d'AVD ou vous pouvez créer la machine à simuler.



Normalement c'est tout ce qu'il faut faire pour déployer le projet sur la plateforme Android choisie.

Déployer sur iOS

Il n'y a aucun composant supplémentaire à installer à parti du moment où vous avez xcode (et donc le simulateur intégré dans xcode).

Ouvrir donc le projet du Tuto8 : si tout va bien alors parmi les kits de configuration pour le projet vous pouvez choisir des Kits pour iOS et pour le simulateur iOS.

Si vous n'avez pas de iPhone ou iPad il faut choisir le simulateur évidemment.

En revanche si les kits ne vous sont pas proposés alors il faut lancer le Qt Maintenance Tool avec l'option Check for Updates dans le menu QtCreator . Dans Qt Maintenance Tool il faut choisir l'option Ajouter ou supprimer des composants pour installer les kits iOS.

Avec les kits installés et sélectionnés vous pouvez donc compiler le projet qui sera déployé sur le simulateur de iPhone que vous avez choisi.

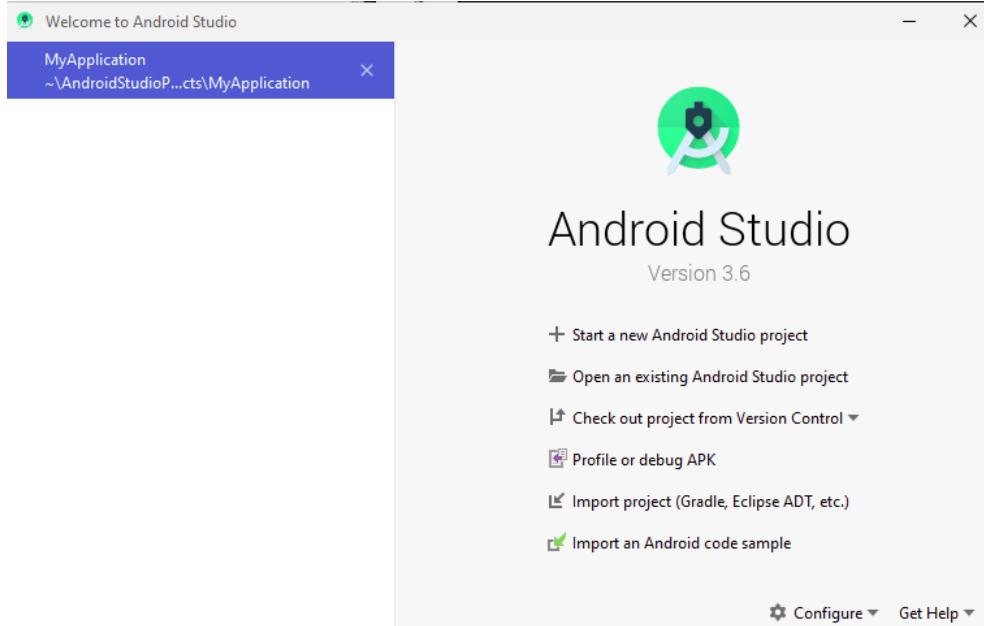
1.2. Sous Windows

Déployer sur android

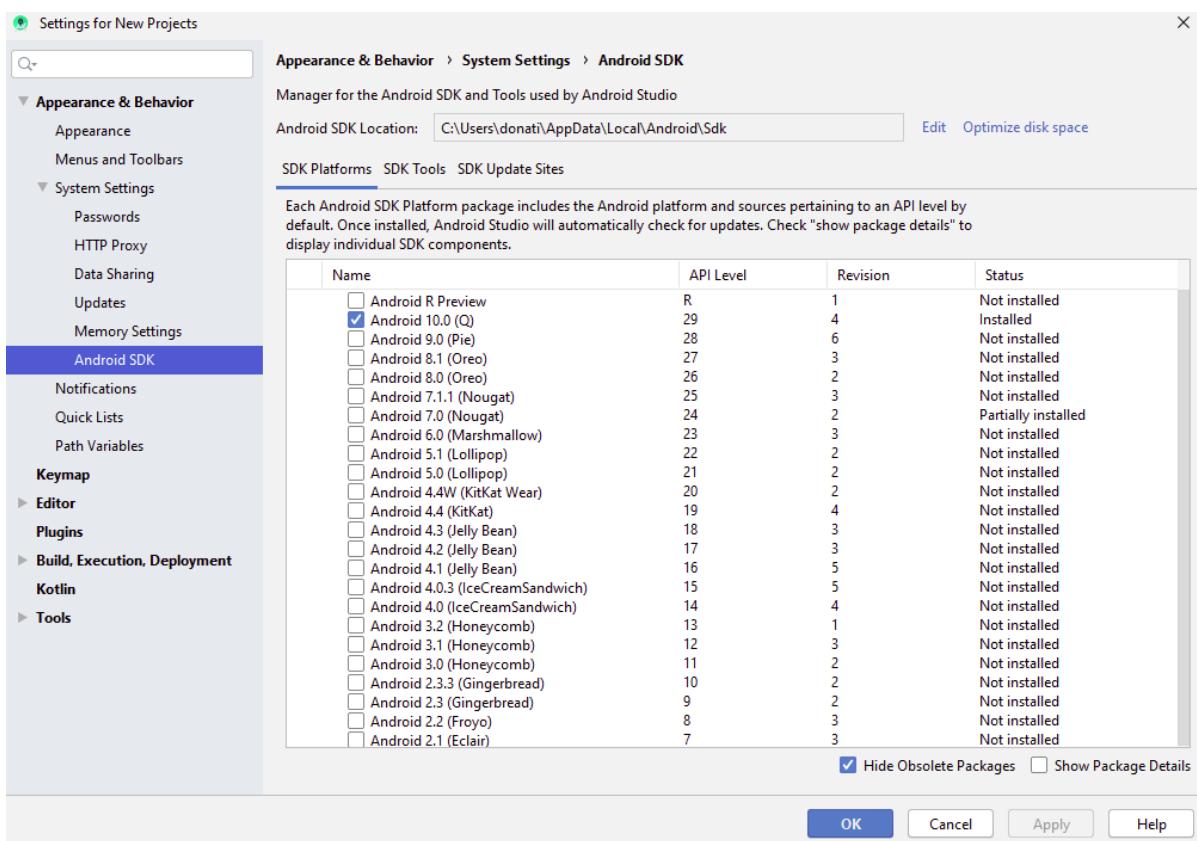
Les étapes sont décrites sur le site de Qt à <https://doc.qt.io/qt-5/android-getting-started.html>
Les éléments nécessaires sont :

- Les **Android SDK Tools** qui sont disponibles en installant [Android Studio](#)
- Le **Android NDK** (pour Native Development Kit) qui permet de développer en C/C++ sous Android. L'installation va aussi se faire via Android Studio
- Le [Java SE Development Kit \(JDK\)](#).

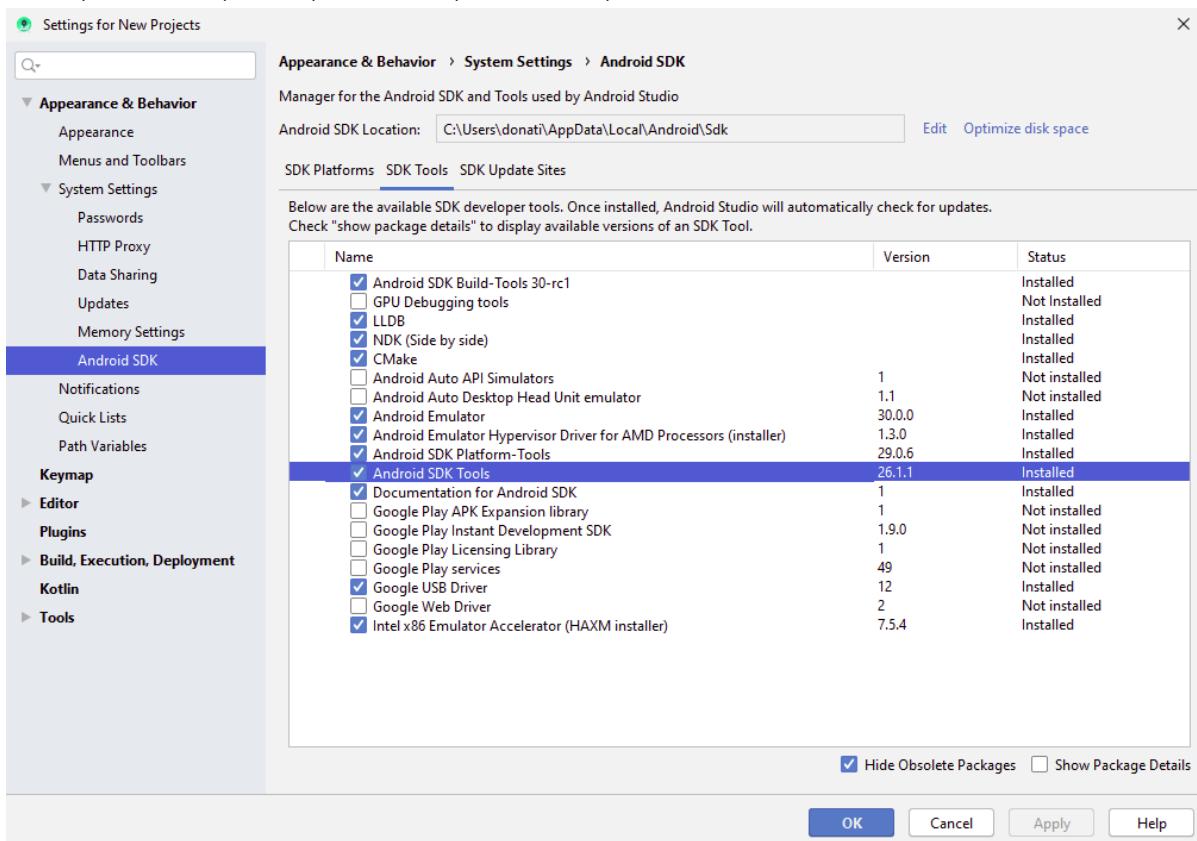
Après avoir lancé **Android Studio**, sur la page de garde, cliquer sur le bouton `Configure et choisir SDK Manager` :



- dans la rubrique `SDK Platforms`, sélectionner et installer la dernière version stable du SDK (Android 10.0 (Q) - API 29)

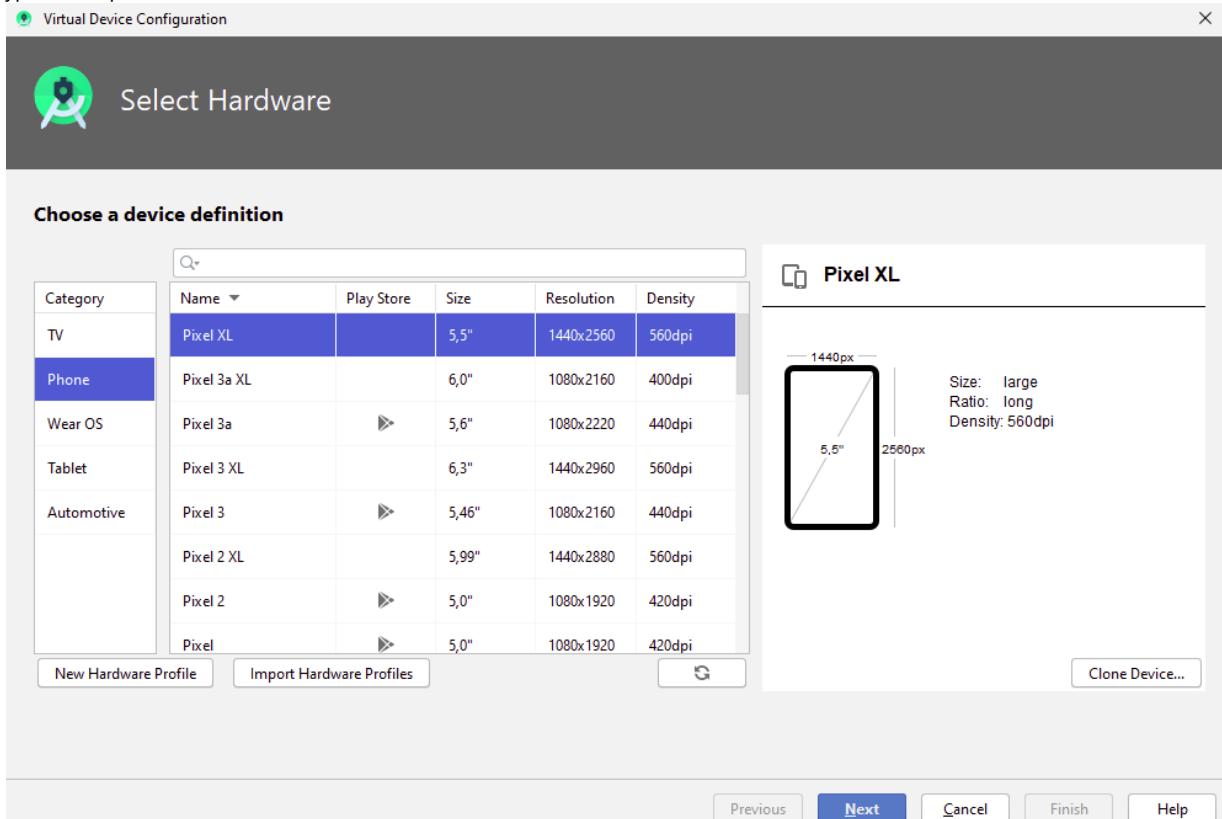


- dans la rubrique **SDK Tools**, cocher et installer **Android SDK Tools**, **NDK (Side by side)** et **Android Emulator** (pour le reste garder les cases par défaut). **Attention :** bien noter le chemin où le SDK Android sera installé. Et ne pas oublier de sélectionner le driver pour utiliser le port USB pour communiquer avec le téléphone Android.

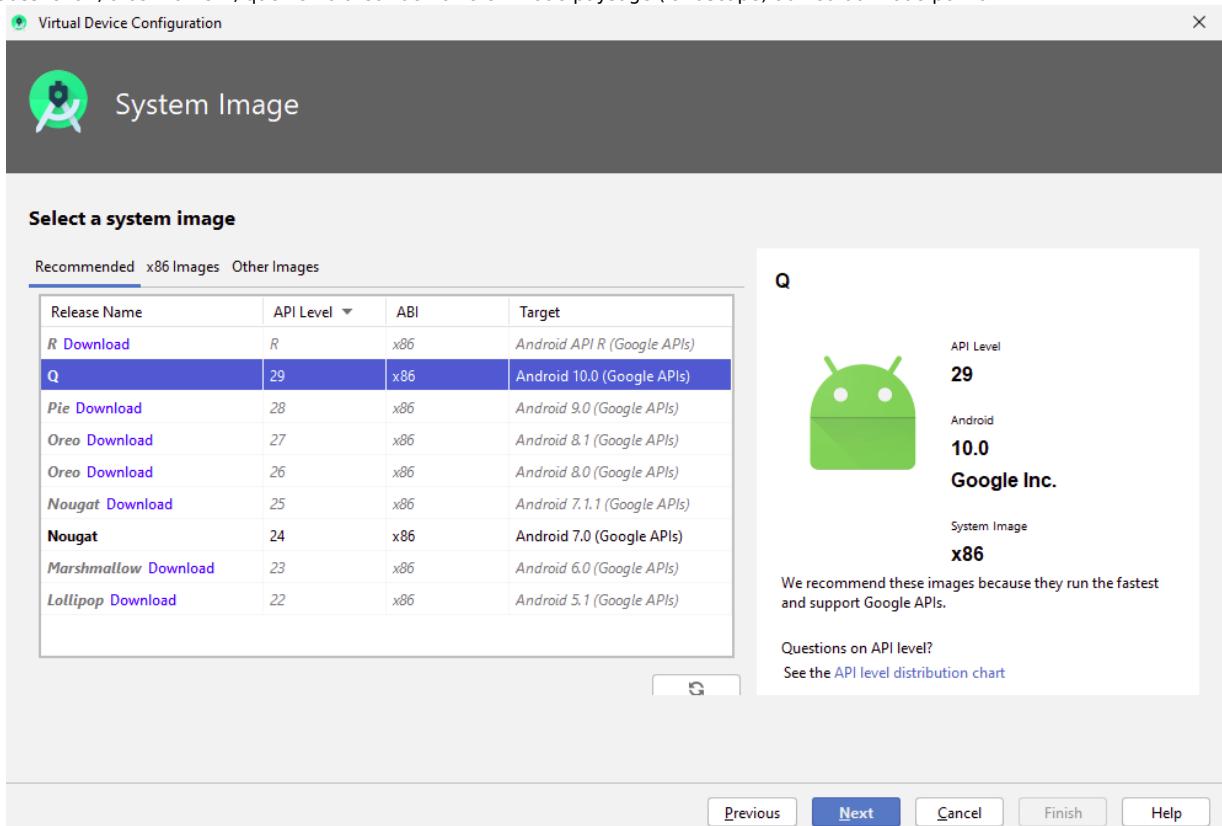


Toujours à partir de la page de lancement de Android Studio, dans **Configure** choisir **AVD Manager** qui va permettre de choisir les téléphones à simuler dans l'émulateur (AVD = Android Virtual Device). Avec le bouton **+ Create Virtual Device** on peut choisir le

type de téléphone android à simuler.



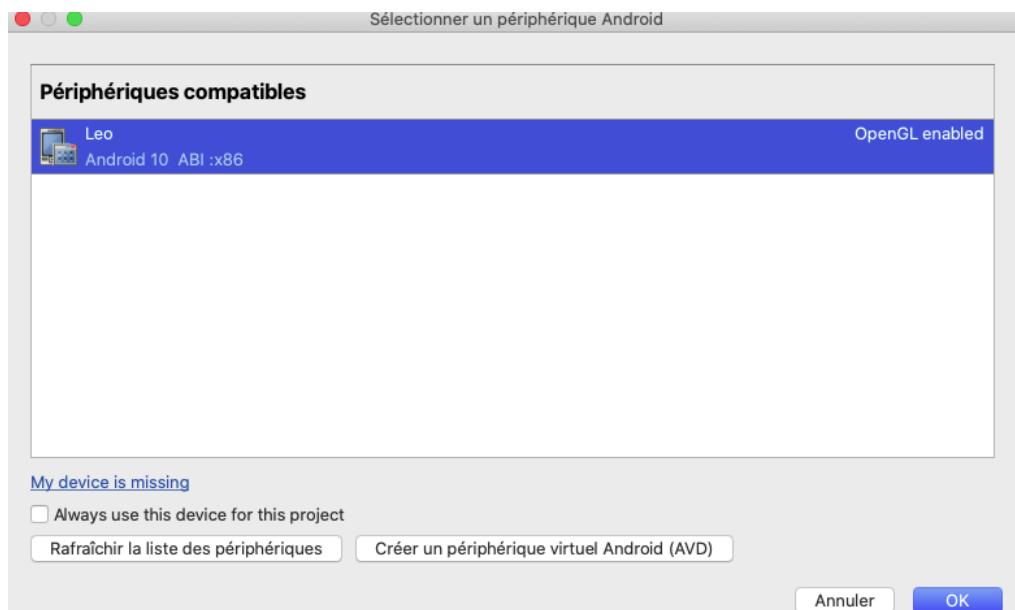
La chose importante c'est que ce soit un téléphone compatible avec la version 29 de Android et une architecture x86. On peut aussi choisir, à ce moment, que l'émulateur démarre en mode paysage (landscape) au lieu du mode portrait.



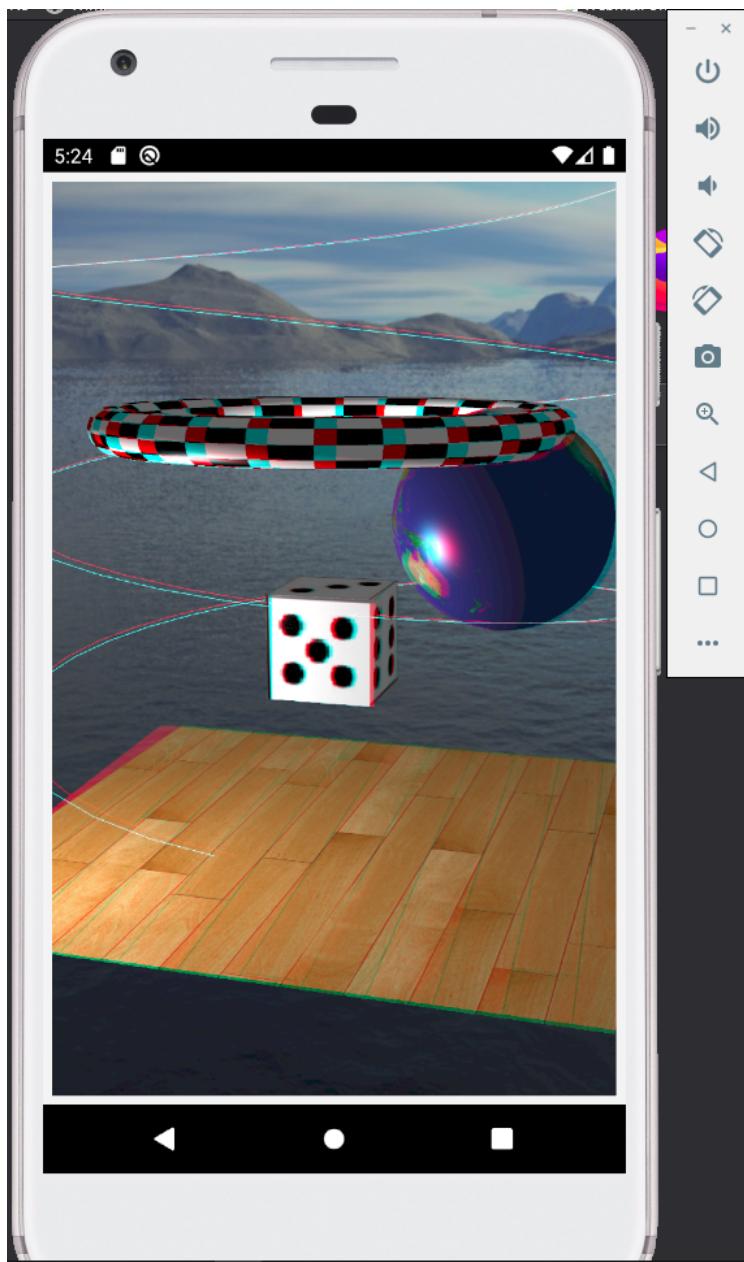
On n'a plus besoin de Android Studio qui peut être fermé maintenant.

Ensuite, après toutes ces installations (y compris le Jdk), il faut intégrer toutes ces informations dans **QtCreator**

Ouvrir donc le projet du Tuto8 : si tout va bien (c'est rare) alors parmi les kits de configuration pour le projet vous pouvez choisir des Kits pour Android et en particulier **Android for x86** et alors en compilant le projet vous pouvez choisir (ou créer) un périphérique Android

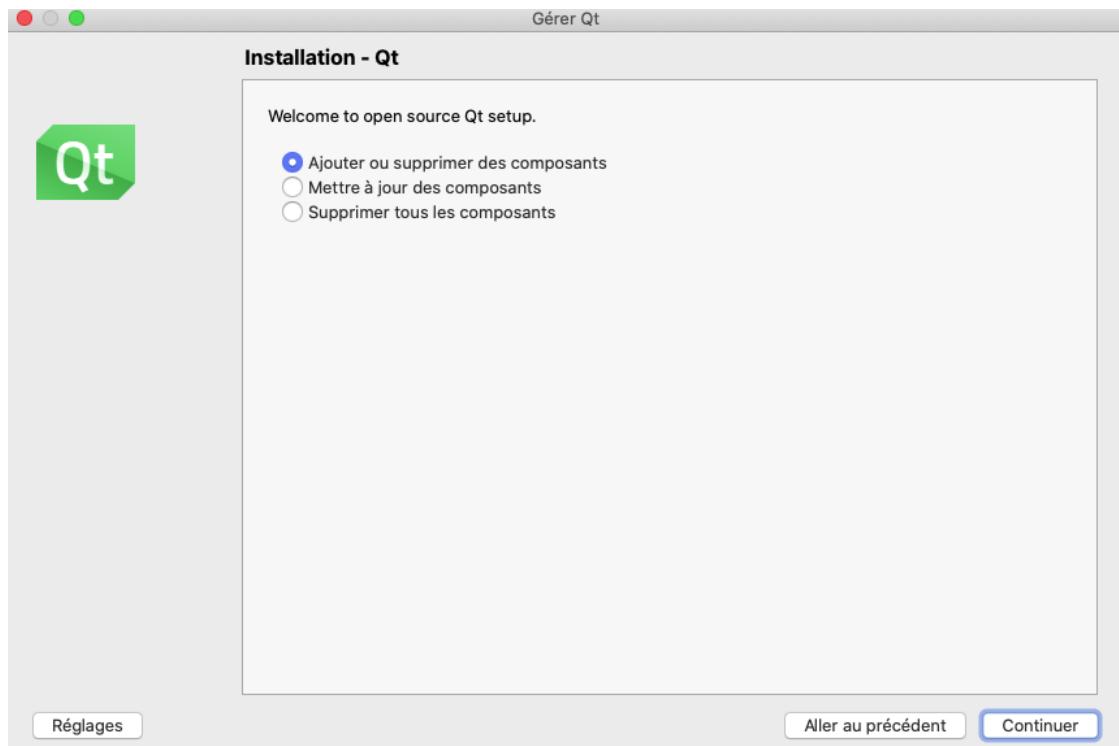


qui peut être soit simulé soit un périphérique réel connecté à votre ordinateur (si vous ne le voyez pas cliquer sur *My device is missing* et le tour est joué).

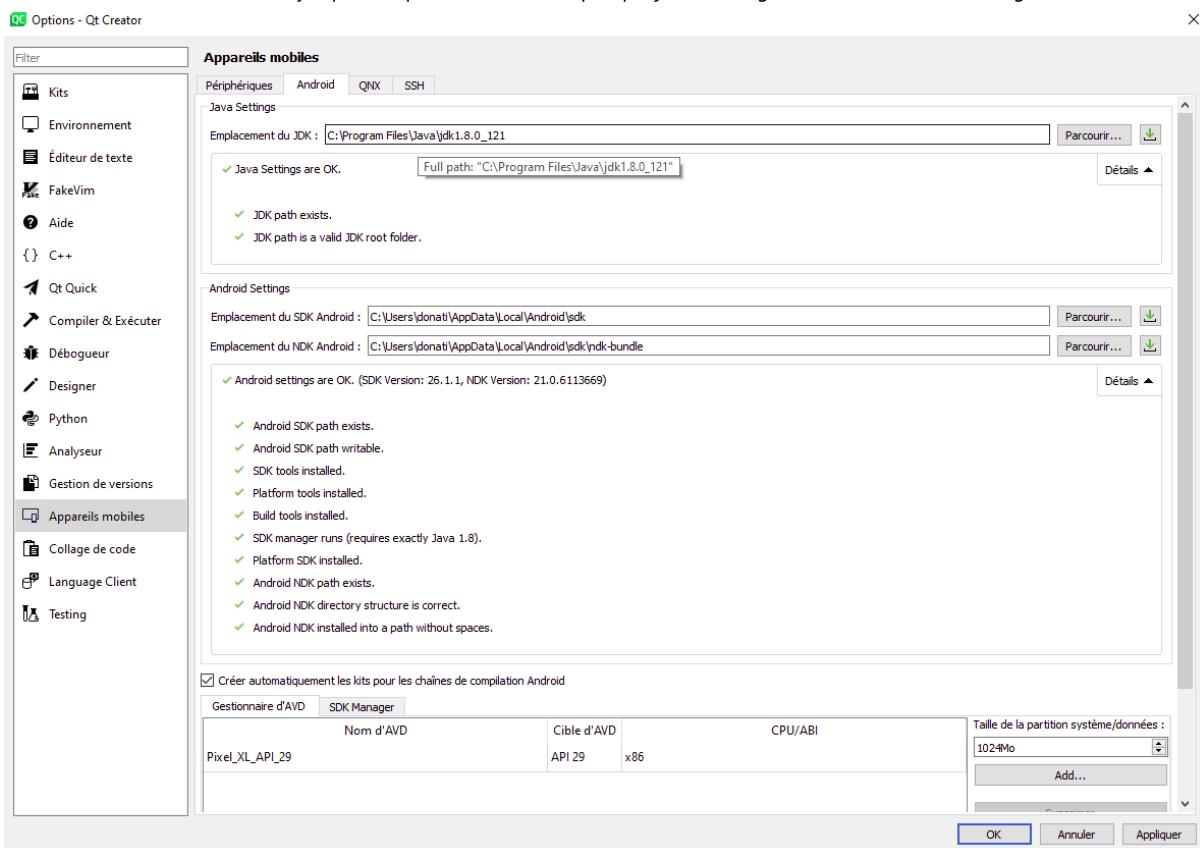


Si le kit Android ne vous est pas proposé ou si le kit n'est pas opérationnel, voici les étapes à suivre : dans Outils/Options choisir la rubrique Kits

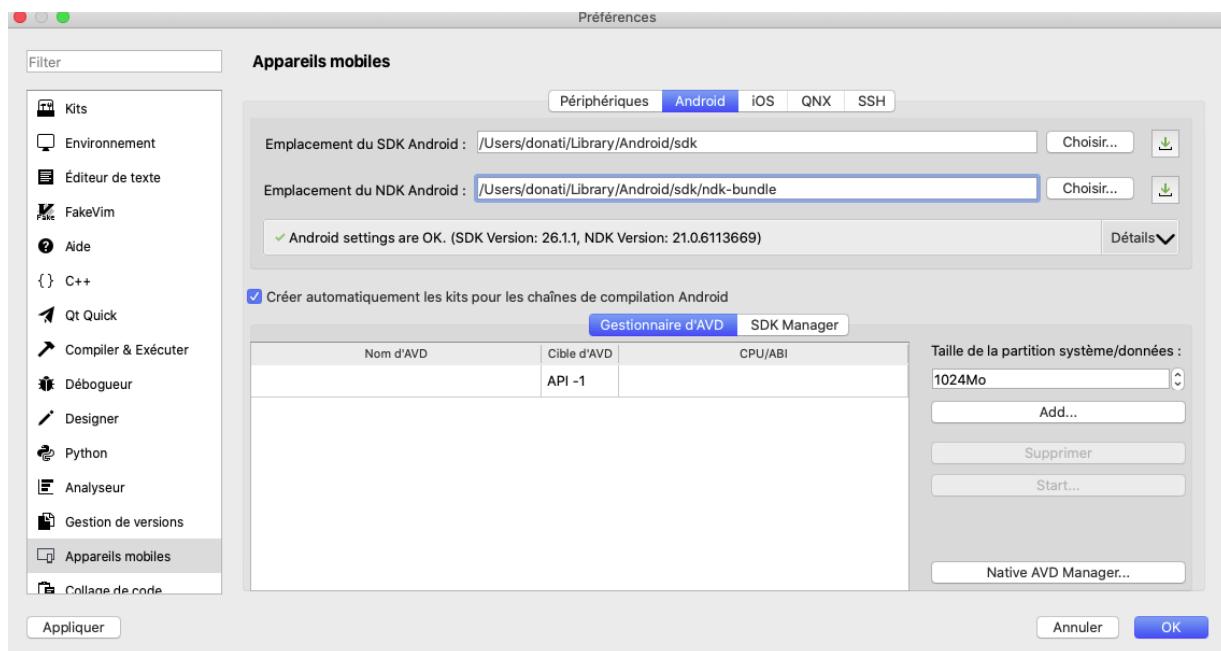
- Si les kits Android ne sont pas proposés (x86, arm64 ou armv7) il faut lancer le Qt Maintenance Tool avec l'option Check for Updates dans le menu qtcreator . Dans Qt Maintenance Tool il faut choisir l'option Ajouter ou supprimer des composants pour installer les kits Android



- Si les kits sont bien là, alors dans la rubrique Appareils mobiles puis Android il faut informer Qt de l'emplacement du JDK, du SDK Android et du NDK Android jusqu'à ce que l'interface indique que *Java Settings are OK* et *Android Settings are OK*.



Plus bas, dans la même page vous avez un gestionnaire d'AVD ou vous pouvez créer la machine à simuler.



Normalement c'est tout ce qu'il faut faire pour déployer le projet sur la plateforme Android choisie.

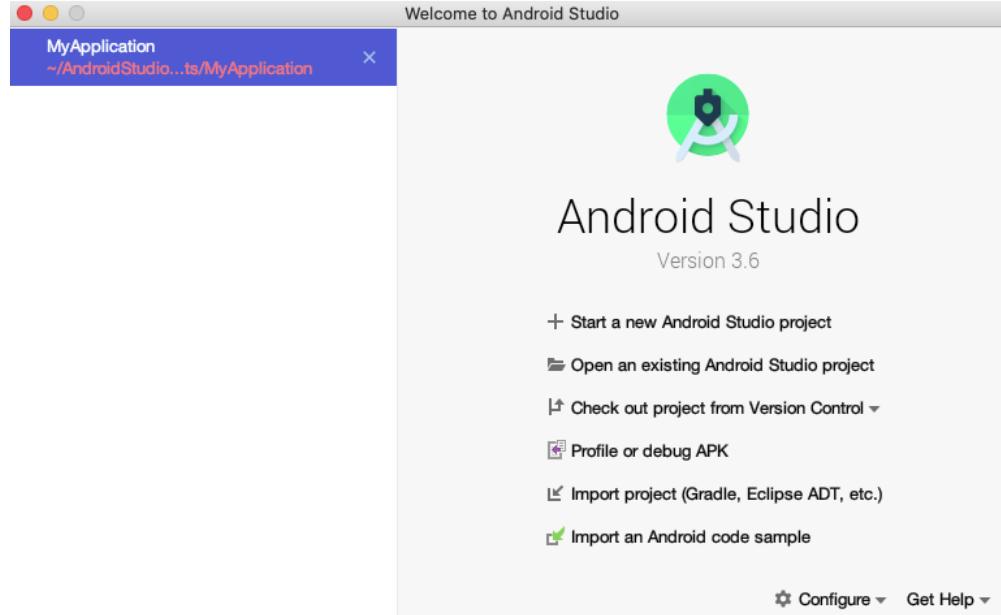
1.3. Sous Linux

Déployer sur android

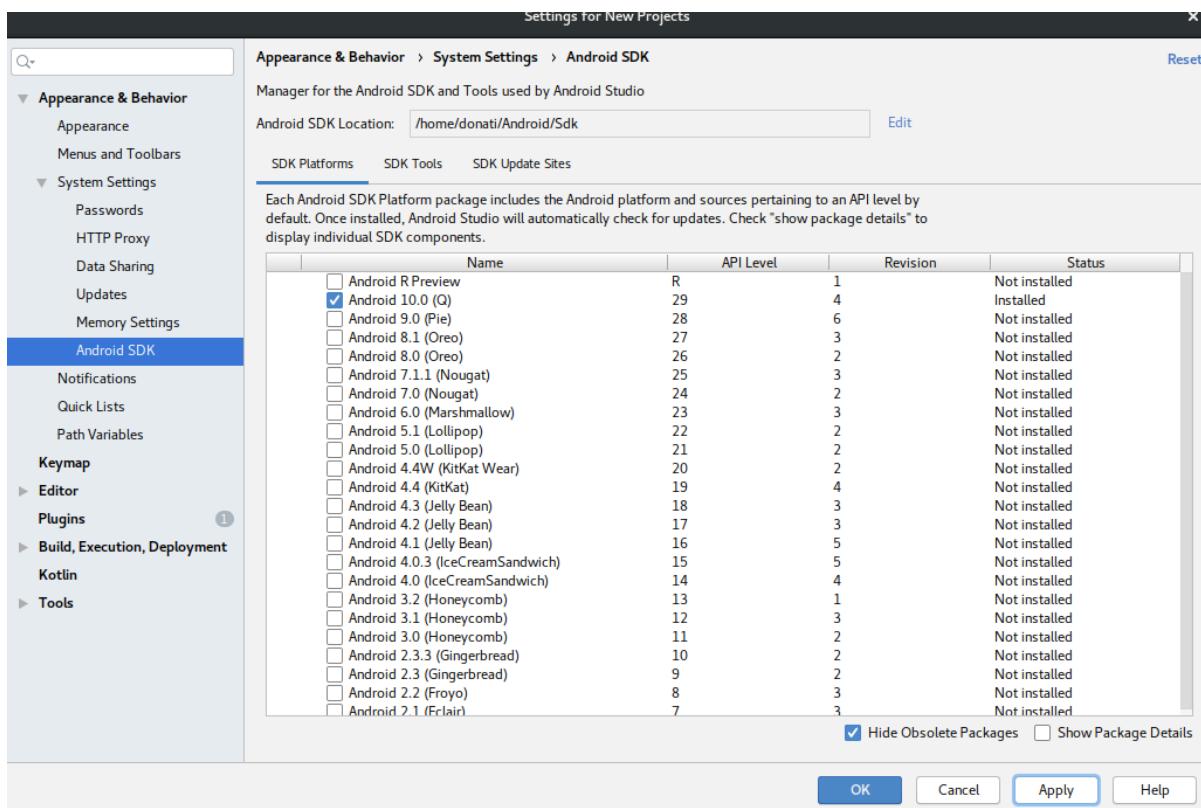
Les étapes sont décrites sur le site de Qt à <https://doc.qt.io/qt-5/android-getting-started.html>
Les éléments nécessaires sont :

- Les **Android SDK Tools** qui sont disponibles en installant [Android Studio](#)
- Le **Android NDK** (pour Native Development Kit) qui permet de développer en C/C++ sous Android. L'installation va aussi se faire via Android Studio
- Le [Java SE Development Kit \(JDK\)](#).

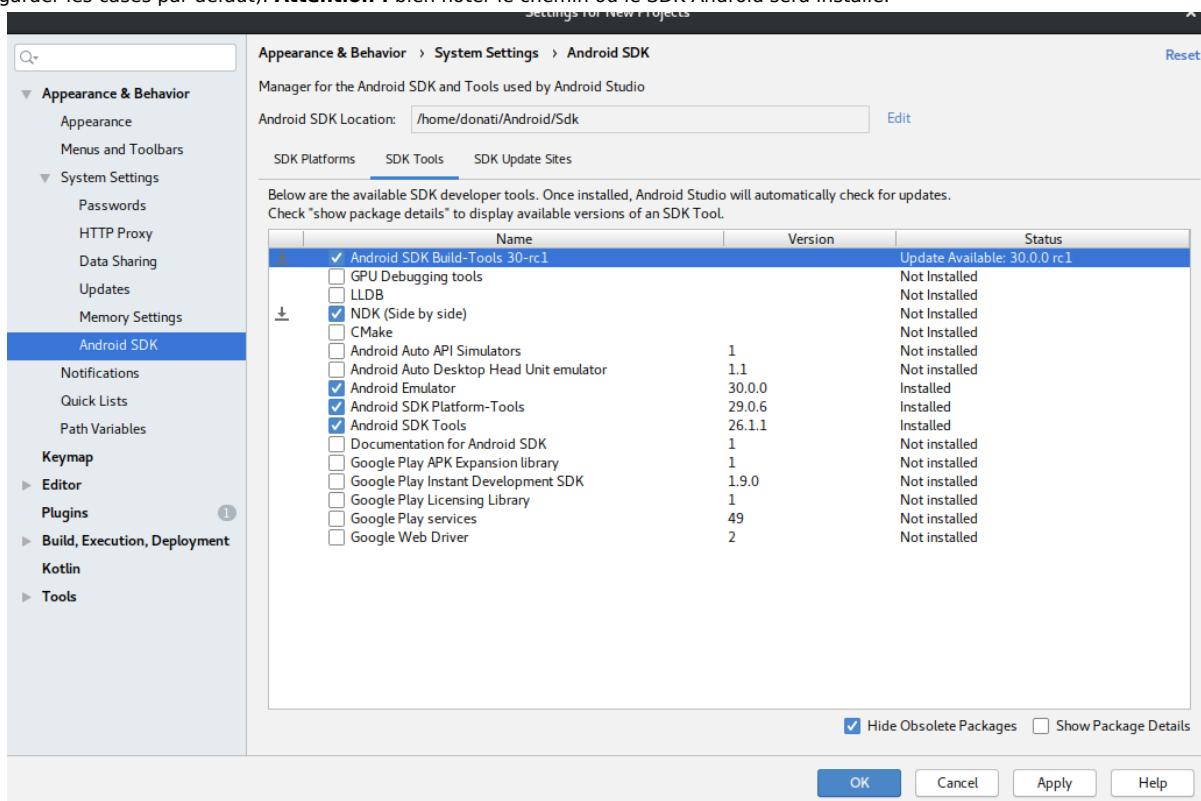
Après avoir lancé **Android Studio**, sur la page de garde, cliquer sur le bouton `Configure` et choisir `SDK Manager` :



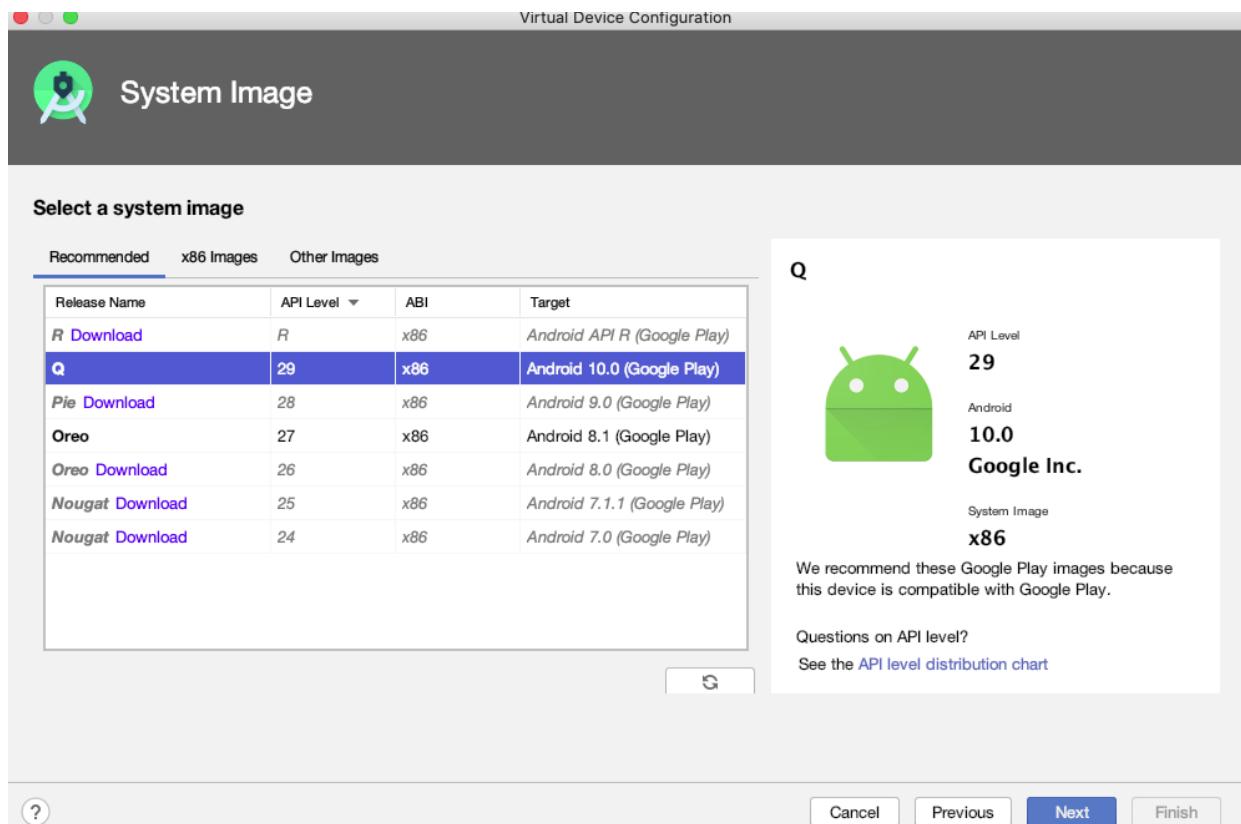
- dans la rubrique `SDK Platforms`, sélectionner et installer la dernière version stable du SDK (Android 10.0 (Q) - API 29)



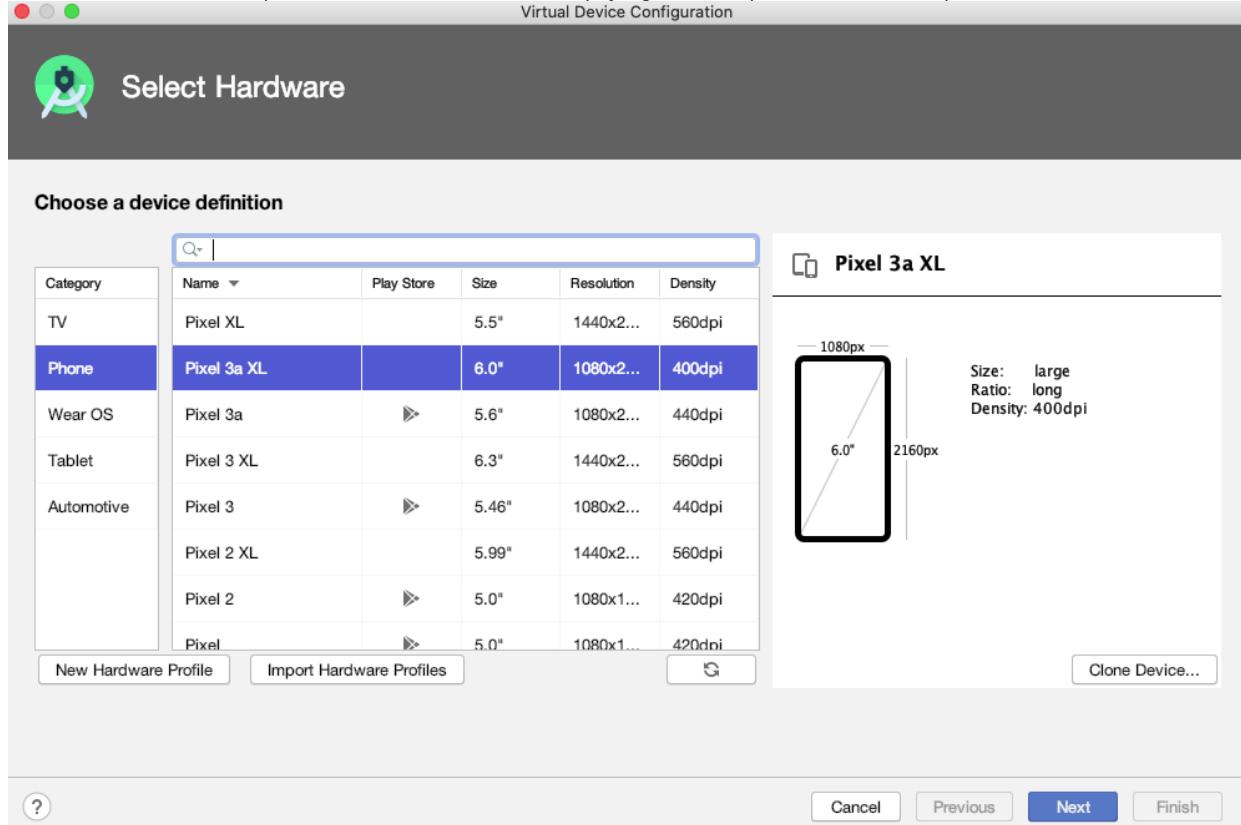
- dans la rubrique **SDK Tools**, cocher et installer **Android SDK Tools**, **NDK (Side by side)** et **Android Emulator** (pour le reste garder les cases par défaut). **Attention :** bien noter le chemin où le SDK Android sera installé.



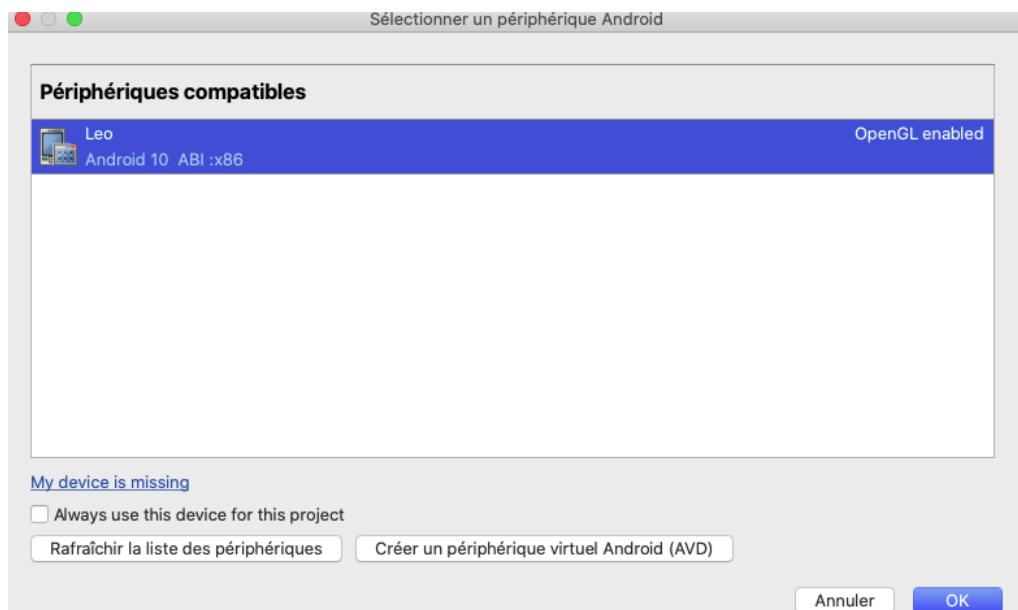
Toujours à partir de la page de lancement de Android Studio, dans **Configure** choisir **AVD Manager** qui va permettre de choisir les téléphones à simuler dans l'émulateur (AVD = Android Virtual Device). Avec le bouton **+ Create Virtual Device** on peut choisir le type de téléphone android à simuler.



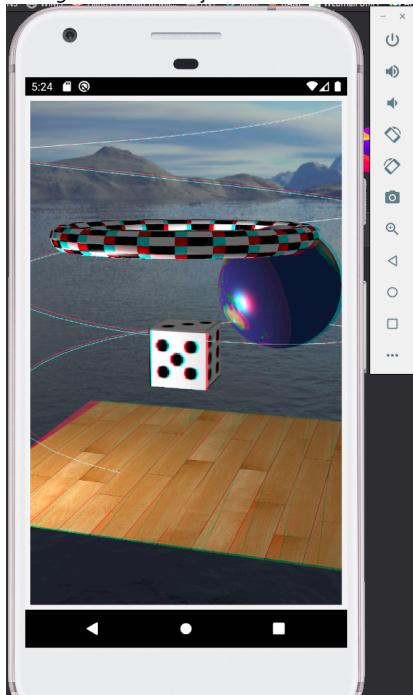
La chose importante c'est que ce soit un téléphone compatible avec la version 29 de Android et une architecture x86. On peut aussi choisir, à ce moment, que l'émulateur démarre en mode paysage (landscape) au lieu du mode portrait.



On n'a plus besoin d'Android Studio qui peut être fermé maintenant.
 Ensuite, après toutes ces installations (y compris le JDK), il faut intégrer toutes ces informations dans **QtCreator**
 Ouvrir donc le projet du Tuto8 : si tout va bien (c'est rare) alors parmi les kits de configuration pour le projet vous pouvez choisir des Kits pour Android et en particulier **Android for x86** et alors en compilant le projet vous pouvez choisir (ou créer) un périphérique Android



qui peut être soit simulé soit un périphérique réel connecté à votre ordinateur (si vous ne le voyez pas cliquer sur *My device is missing* et le tour est joué).

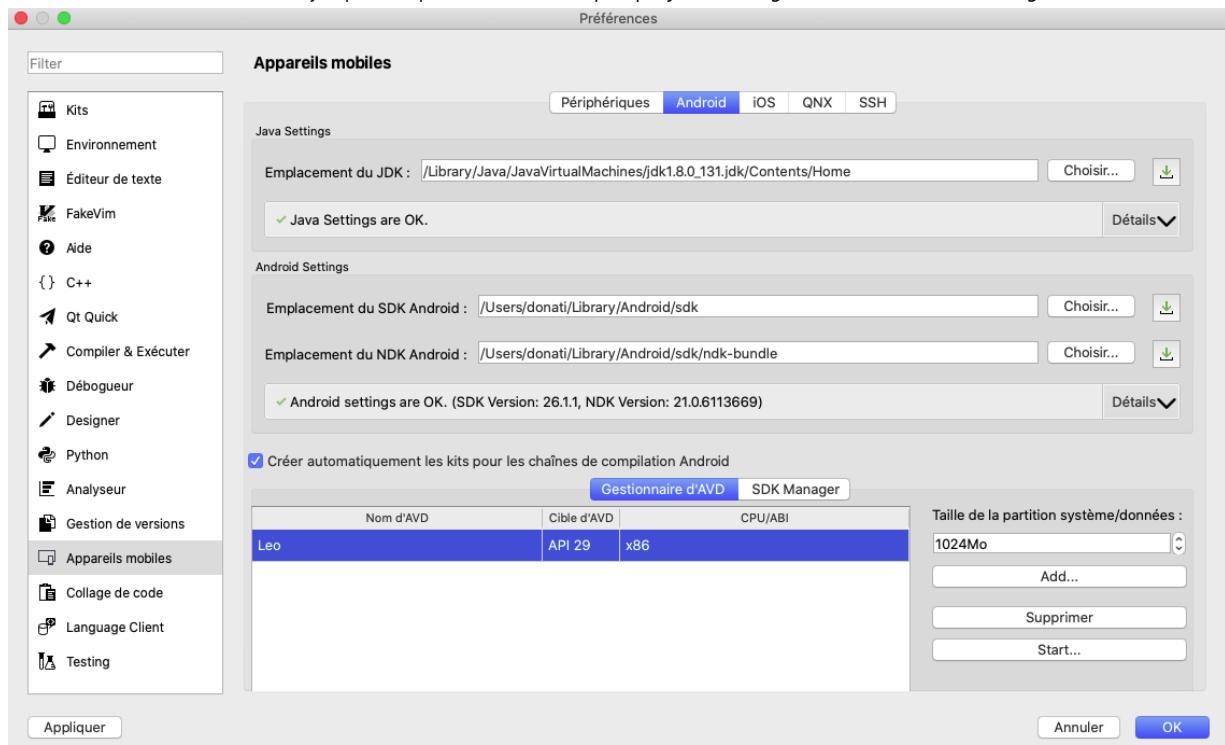


Si le kit Android ne vous est pas proposé ou si le kit n'est pas opérationnel, voici les étapes à suivre : dans Outils/options... choisir la rubrique Kits

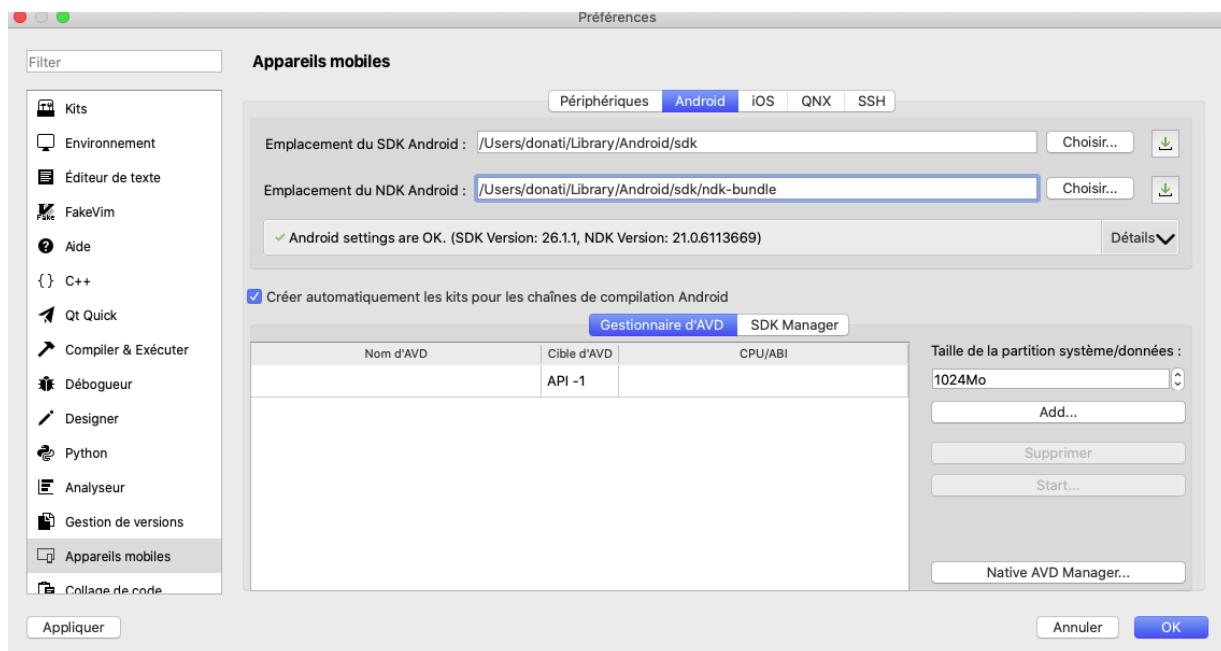
- Si les kits Android ne sont pas proposés (x86, arm64 ou armv7) il faut lancer le Qt Maintenance Tool avec l'option Check for updates dans le menu qtcreator . Dans Qt Maintenance Tool il faut choisir l'option Ajouter ou supprimer des composants pour installer les kits Android



- Si les kits sont bien là, alors dans la rubrique Appareils mobiles puis Android il faut informer Qt de l'emplacement du JDK, du SDK Android et du NDK Android jusqu'à ce que l'interface indique que *Java Settings are OK* et *Android Settings are OK*.



Plus bas, dans la même page vous avez un gestionnaire d'AVD ou vous pouvez créer la machine à simuler.



Normalement c'est tout ce qu'il faut faire pour déployer le projet sur la plateforme Android choisie.

2. Mode paysage

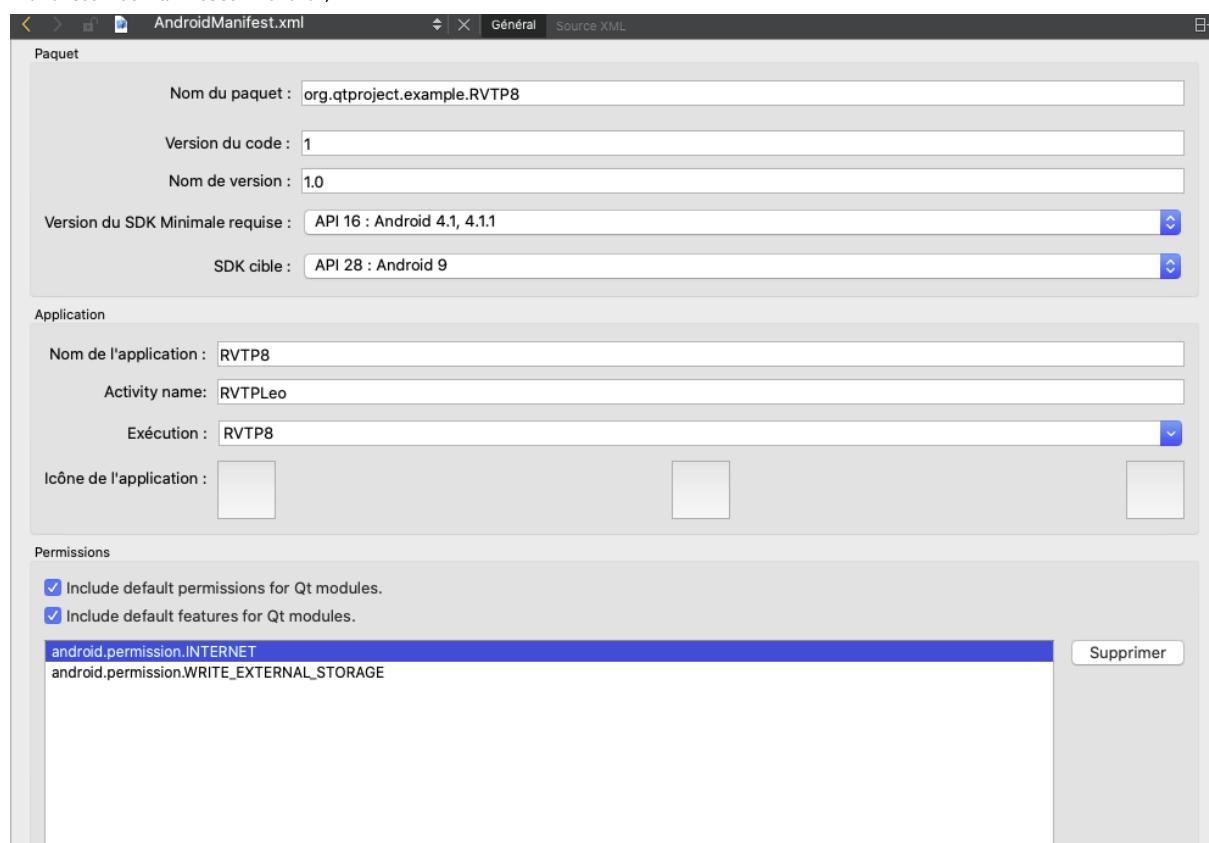
Si vous passez du mode portrait en mode paysage dans le simulateur, le rendu reste orienté le long de l'axe du téléphone au lieu d'être recalculé pour la nouvelle orientation.

Pour que dès le démarrage, l'application se lance avec un rendu en mode paysage, il faut éditer le fichier `AndroidManifest.xml` qui contient tous les réglages spécifiques à l'application Android :

- Aller chercher le fichier `Androidmanifest.xml` que Qt a créé dans le répertoire de `build` et copier-le à l'endroit où se trouve votre code ;
- Dans le projet ajouter le fichier `Androidmanifest.xml` avec Clic-droit puis Ajouter des fichiers existants ;
- Pour que ce fichier soit utilisé lors de la compilation par Qt au lieu du fichier créé dans le `build` il faut ajouter dans le fichier `rvtuto8.pro` la ligne

```
ANDROID_PACKAGE_SOURCE_DIR = $$PWD
```

- Pour éditer le fichier `Androidmanifest.xml` dans la zone **Other files** de QtCreator, faites un Clic-droit puis Ouvrir avec .../Editeur de manifeste Android ;



- En passant en mode Source XML il faut changer `android:screenOrientation="landscape"`

