

## Séance 5a: Préparer l'environnement de travail Android — Hello World —

### Installation des outils

#### A propos de ce tutoriel

Il existe plusieurs bons tutoriels pour l'installation du SDK <sup>1</sup> Android et d'un environnement de développement qui permettra de l'utiliser. Le présent tutoriel pourra donc être complété par les instructions d'installation du site web officiel d'Android <sup>2</sup>, ou encore des extraits du livre *Hello, Android* <sup>3</sup>. Il y a aussi d'autres tutoriels en ligne assez bien conçus comme par exemple celui de Lars Vogel <sup>4</sup> qui a l'air assez complet et à jour.

Pour utiliser le SDK Android vous avez le choix entre plusieurs alternatives mais certaines d'entre elles rendront les choses beaucoup plus faciles que d'autres. Vous pouvez installer un environnement de développement préconfiguré et optimisé pour la programmation pour Android, configurer vous même un environnement de développement existant ou encore choisir de travailler directement en ligne de commande. Pour ce cours nous considérerons que vous utilisez l'environnement *Android Studio*.

*Android Studio* repose sur l'environnement de développement IntelliJ de JetBrains préconfiguré et optimisé pour Android et uniquement pour Android. Nous choisissons de travailler avec cet environnement car l'ancien plug-in *Android ADT* pour Eclipse n'est plus en développement actif depuis 2014 et ne connaîtra donc plus d'évolution alors que des fonctionnalités rendant encore plus aisée que dans les environnements moins récents la programmation pour Android sont et seront de plus en plus présentes dans *Android Studio*.

Cependant, puisque *Android Studio* est en évolution constante, des mises à jours périodiques seront rendues disponibles par les développeurs d'*Android Studio*. Pour mettre à jour manuellement le logiciel aller à *Help > Check for updates* (sur Mac aller à *Android Studio > Check for updates*).

Puisque Android pour *Android Studio* "compile sous java", vous aurez au préalable à installer le *Java Standard Edition Development Kit* (JDK). <sup>5</sup>

---

1. Software Development Kit (boîte à outils pour le développeur d'applications Android)

2. <http://developer.android.com/sdk>

3. Hello, Android. The Pragmatic Programmer. Ed Burnette. Quatrième édition.

4. <http://www.vogella.de/articles/Android/article.html>

5. Au minimum la version JDK 7 est requise si vous voulez développer des applications pour Android 5.0. Le JDK 8 ne semble pas encore parfaitement supporté par Android, par contre. A utiliser avec caution donc. Si vous ne savez pas quelle version de Java est actuellement installée sur votre système, exécutez la commande `javac -version` dans un terminal pour vérifier.

LSINF1225	Séance 5a: Préparer l'environnement de travail Android — Hello World —	Dest : étudiants
2016-02-26		Auteur : KM,DM,CZM,XC

## Installer le JDK 7

Rendez vous sur le site d'Oracle<sup>6</sup> et téléchargez la version du JDK qui correspond à votre système, après avoir accepté le 'License Agreement' de Oracle.

Exécutez le fichier téléchargé et suivez les instructions d'installation (Utilisateurs de Linux : voir l'annexe 1).

## Installer Android Studio

Rendez vous sur le site d'Android<sup>7</sup> et téléchargez le paquet *Android Studio* qui correspond à votre système d'exploitation.

### Utilisateurs de Mac

Exécutez le fichier d'installation téléchargé et suivez soigneusement les instructions données sur le site. Après installation, lors du lancement de l'application installée, autorisez l'exécution au cas où vous recevez une notification de votre système d'exploitation.

### Utilisateurs de Windows

Exécutez le fichier .exe téléchargé et suivez les instructions d'installation. Autorisez l'exécution au cas où vous recevez une notification de votre système d'exploitation. (Cf. Figure 1, par exemple.) Vous pourrez ensuite lancer *Android Studio* à partir de votre menu 'Démarrer'.

### Utilisateurs d'une distribution Linux (par exemple Ubuntu)

- Placez l'archive téléchargée à l'endroit où vous voudrez que se trouve *Android Studio* dans votre ordinateur.
- Extrayez l'archive.
- Ensuite ouvrez un terminal et exécutez la commande :  
`cd moncheminversandroidstudio/android-studio/bin` qui vous permet de vous déplacer vers le répertoire de ce nom.
- Exécutez la commande `./studio.sh` pour démarrer *Android Studio*.

Si vous trouvez qu'il est assez embêtant de devoir à chaque fois se déplacer dans le répertoire sus mentionné afin de lancer *Android Studio*, vous pourrez rajouter le chemin vers *Android Studio* aux variables d'environnement de votre système ce qui vous permettra de le lancer à partir de n'importe quel emplacement en entrant simplement la commande `studio.sh`

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

7. <http://developer.android.com/sdk>

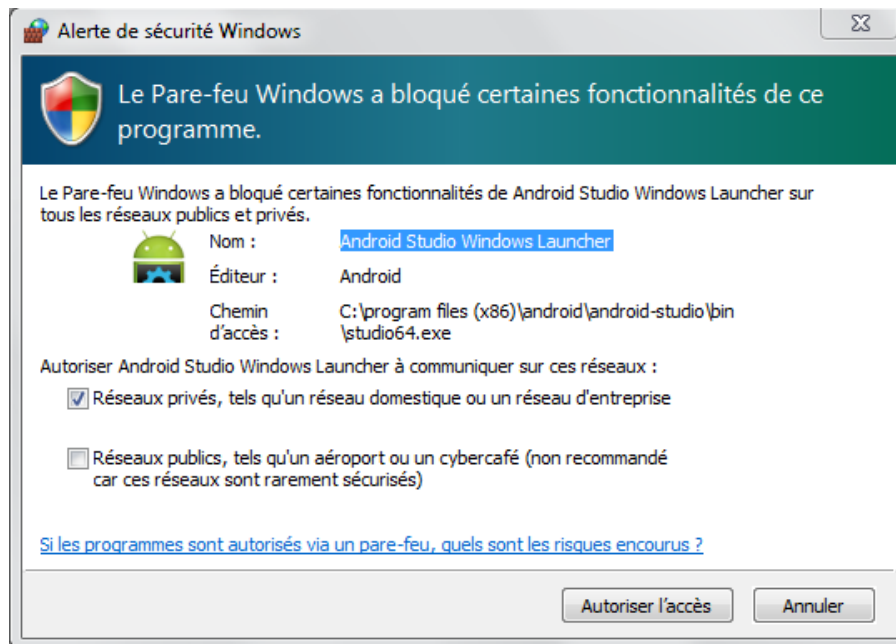


FIGURE 1 – Blocage exécution Windows

## Equiper Android Studio avec les bons outils

Une fois *Android Studio* lancé pour la première fois (et après avoir suivi éventuellement les étapes du *Android Studio Setup Wizard*), la fenêtre de démarrage rapide apparaît (cf. Figure 2). Dans cette fenêtre sélectionnez 'Configure' puis 'SDK Manager'.

Le SDK manager (cf. Figure 3) vous permet de sélectionner les APIs que vous voudriez installer sur votre PC. Certains composants sont présélectionnés. Il s'agit soit de composants dont on ne peut dans tous les cas pas se passer, soit de composants qu'Android voudrait particulièrement promouvoir.

Dans 'SDK Platforms' vous choisirez le niveau d'API le plus récent que vous voudrez installer (par exemple Android 6.0) en gardant cependant à l'esprit que les versions les plus récentes pourraient ne pas toucher un assez large public ou encore susciter des problèmes de rétro-compatibilité.

Dans 'SDK Tools' vous devez au minimum cocher les éléments suivants :

- Android SDK Build-tools
- Android SDK Tools
- Android SDK Platform-tools

Pour pouvoir profiter des APIs Google et d'autres APIs utiles cochez également :

- Google Repository

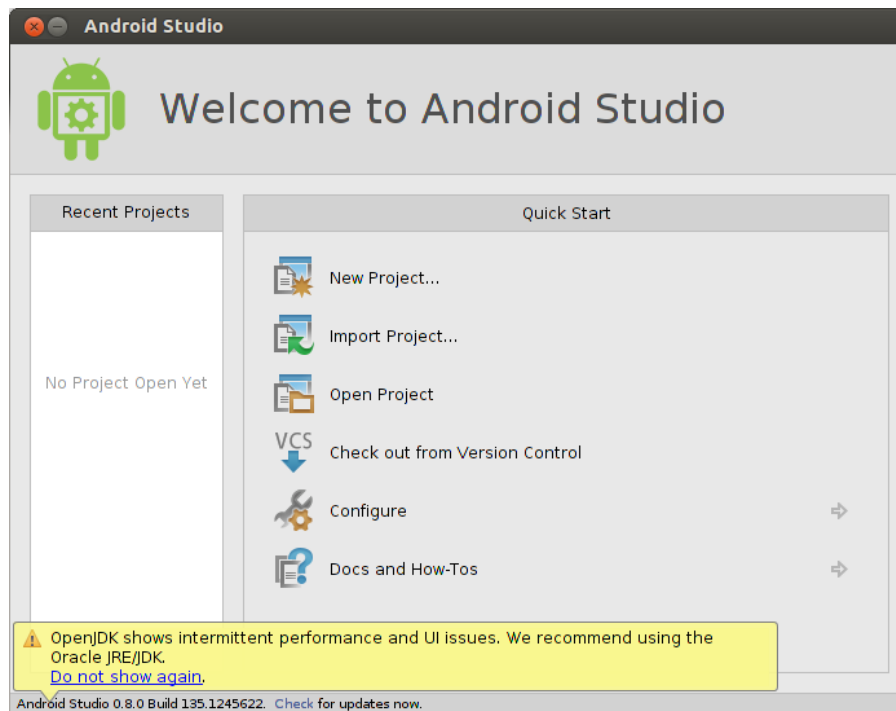


FIGURE 2 – Fenêtre de démarrage Android studio

- Google Play services
- Android Support Library

Cliquez sur 'Ok', et attendez que l'installation se fasse. Pour accéder à nouveau au SDK manager dans le futur, cliquez sur l'icône (cf. Figure 4) dans la barre d'outils de votre projet Android.

## ***Hello World : votre première application Android***

### **Créer un nouveau projet Android**

Pour créer un nouveau projet Android :

- A partir du menu principal 'Quick Start', créez un nouveau projet.
- Nommez votre application. Il est important de bien choisir le nom du package de l'application (cf. Figure 5). C'est ce nom qui servira à représenter de façon unique votre application. Cliquez sur 'Next'.
- Un 'Form factor' (cf. Figure 6) est le type d'appareil pour lequel votre application va être créée. Dans ce cours nous ciblerons plutôt les tablettes et smartphones. Laissez sélectionné donc 'Phone and Tablet'

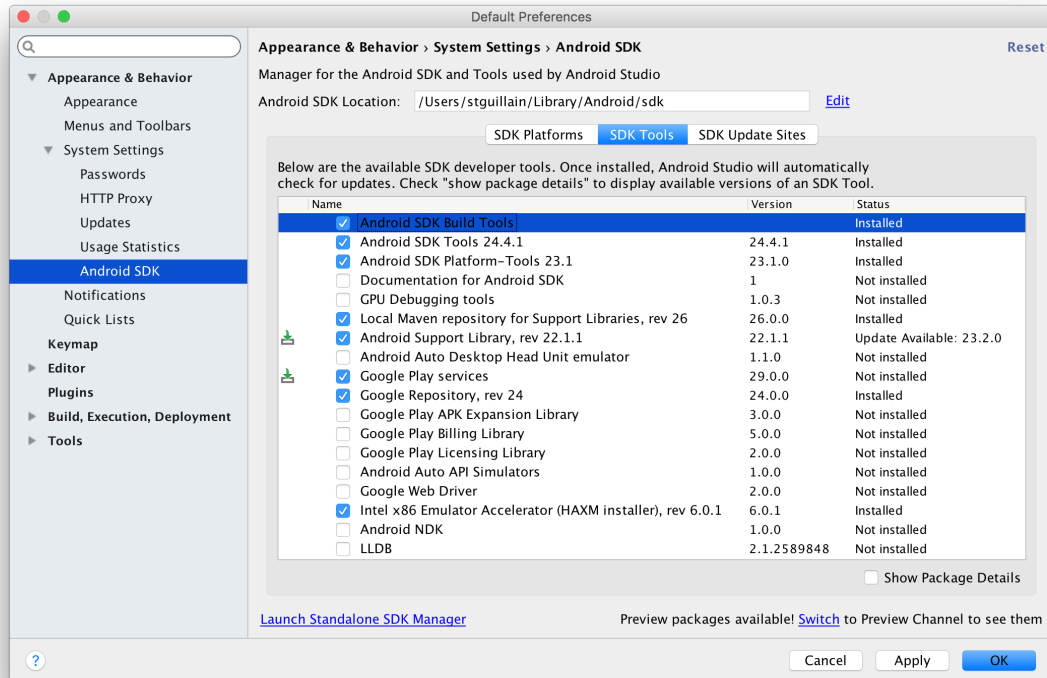


FIGURE 3 – Android SDK Manager



FIGURE 4 – Icône SDK Manager

puis choisissez le SDK minimum pour lequel fonctionnera votre application. Nous conseillons de laisser sélectionnée l'API 15. Cliquez sur 'Next'.

- Android vous aide à démarrer en créant pour vous un modèle d'une première activité (cf. Figure 7). Vous pourrez choisir le modèle qui correspondra le plus au style de votre application lorsque vous débuterez votre projet. Pour l'heure, vous pouvez choisir une simple 'Blank Activity' et cliquer sur 'Next'.
- Configurez l'activité créée par défaut par Android dans votre projet. En accord avec le principe de "Convention plutôt que configuration" Android déduit automatiquement le nom du layout XML et celui du titre de l'activité que vous créez, en vous permettant tout de même de modifier ces conventions si nécessaire. Lorsque vous débuterez votre projet, nous vous conseillons de donner à cette activité par

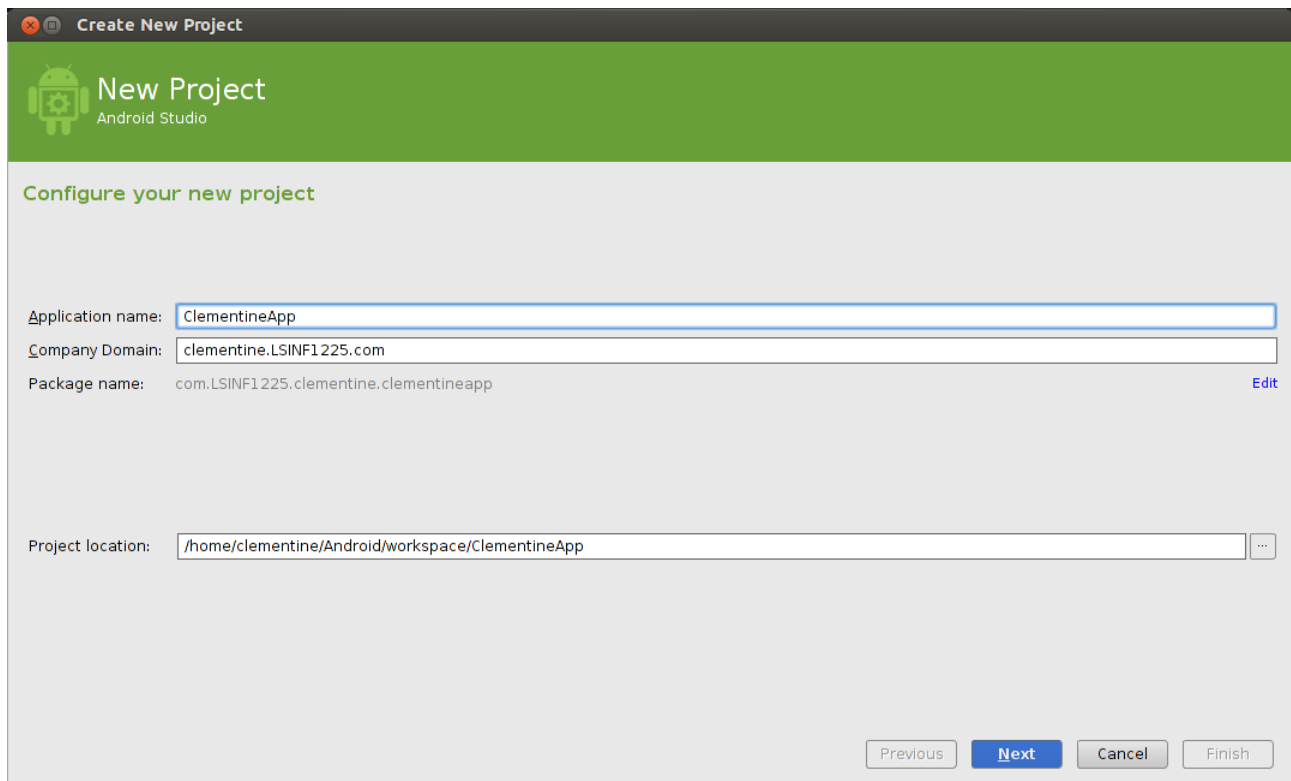


FIGURE 5 – Dialogue 1 de création d'un nouveau projet

défaut un nom du type 'MainActivity' afin qu'elle puisse faire office d'écran de démarrage de votre application. (Cf. Figure 8.)

- Cliquez sur 'Finish' puis attendez que le projet se construise (cela peut prendre un temps considérable).

## Créer un AVD

Un appareil virtuel Android (AVD) est un simulateur d'OS Android dans lequel il est possible d'exécuter des applications Android. Une fois le/les SDK Android installé(s) et un projet Android crée (cf. la section suivante) vous pouvez créer un AVD comme ceci :

- Dans la barre d'outils cliquez sur l'icône du AVD Manager (cf. Figure 10).
- Si la liste des AVD est vide (cf. Figure 11) ou si vous souhaitez créer un nouvel AVD compatible avec votre application cliquez sur 'Create Virtual Device...' (cf. Figure 12).
- Sélectionner le Hardware (par exemple une téléphone Nexuse One) que vous voulez simuler, puis cliquez sur 'Next'.
- Sélectionner l'image système que vous désirez simuler sur ce Hardware (par exemple Lollipop API Level 21 Android 5.0) et cliquez sur Download si nécessaire pour télécharger cet image. Attendez

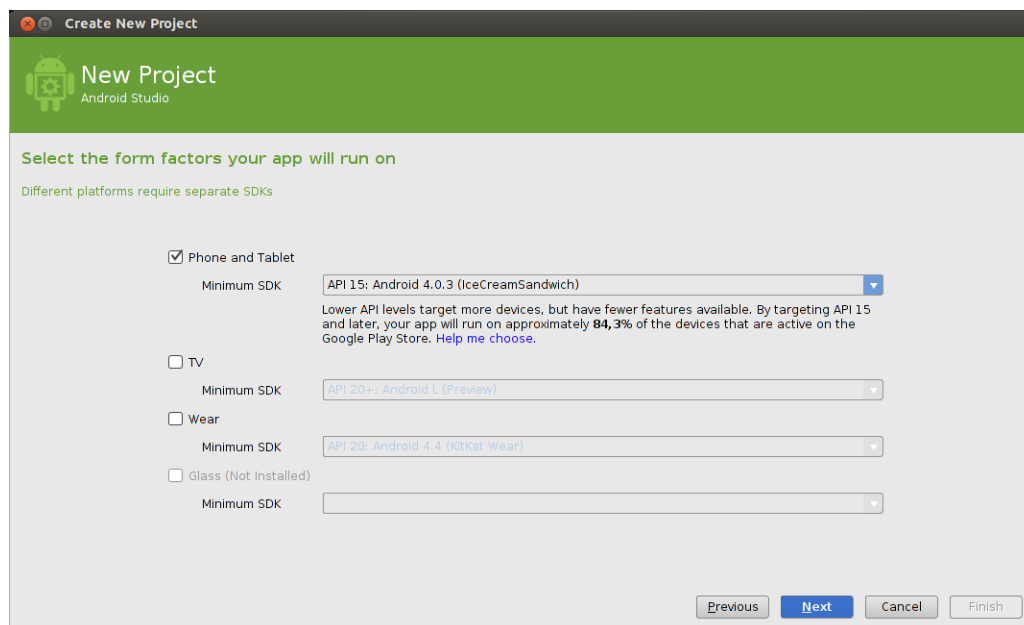


FIGURE 6 – Dialogue 2 de création d'un nouveau projet

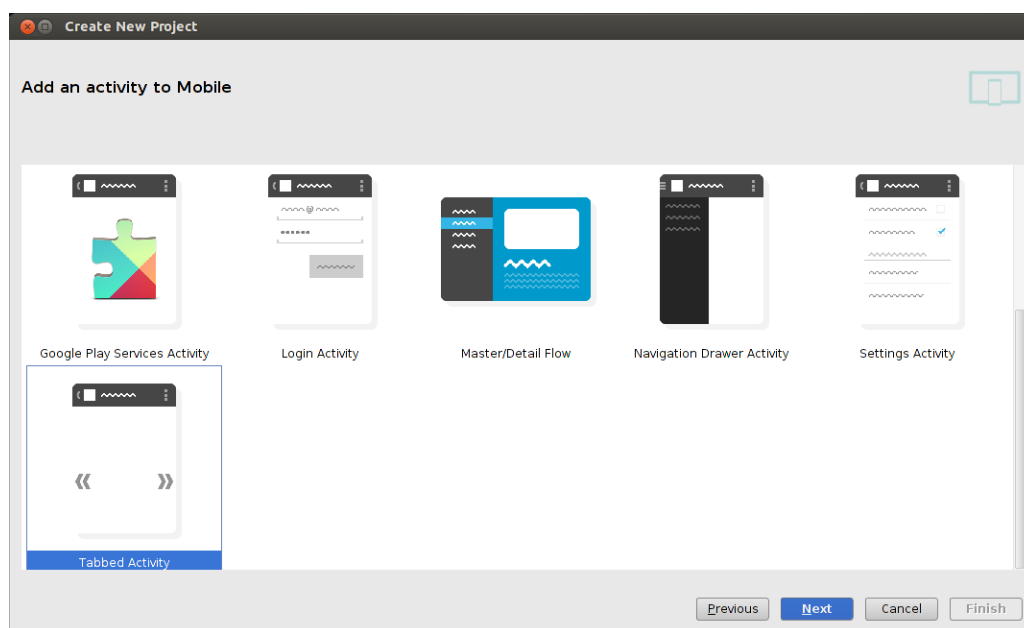


FIGURE 7 – Dialogue 3 de création d'un nouveau projet

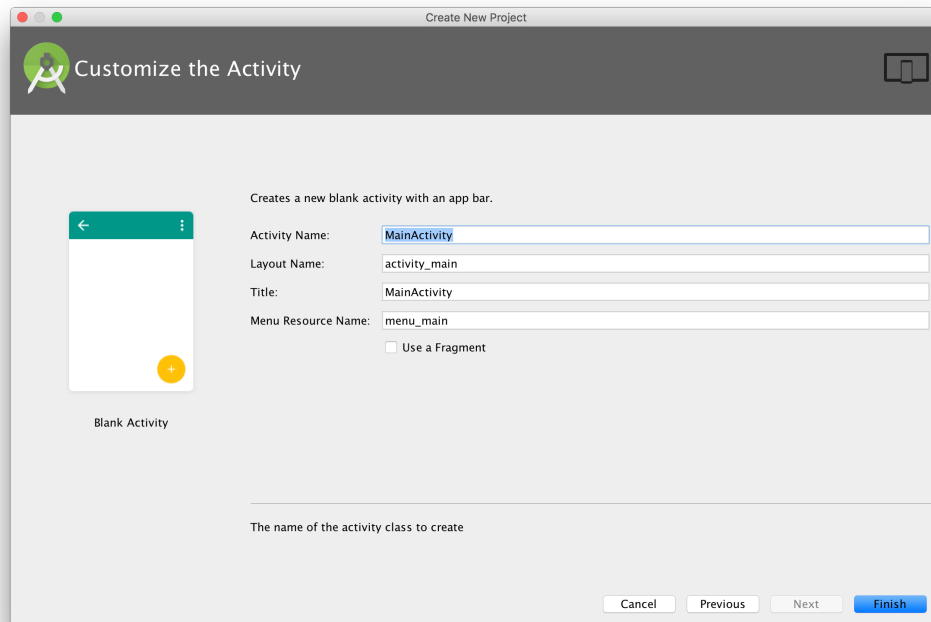


FIGURE 8 – Dialogue 4 de création d'un nouveau projet

l'installation, ensuite cliquez sur 'Finish'.

- Dans la fenêtre de création (cf. Figure 12) choisissez un nom parlant pour votre nouvel AVD qui vous aidera à vous rappeler le contexte dans lequel il sera utilisé (version d'Android, type appareil...).
- Cliquez sur 'Finish'. Le AVD est maintenant sauvegardé. Cet AVD vous permet normalement d'exécuter vos applications Android sur émulateur. Cependant, en pratique ce dispositif est extrêmement lent à démarrer et à fonctionner. Le plus aisé et efficace pour vous sera sans doute de tester votre application à l'aide d'un vrai appareil Android (voir l'annexe 2). Un autre alternatif est d'utiliser un émulateur de GenyMotion <https://www.genymotion.com/> qui est beaucoup plus efficace. Lorsque tout ceci est fait, vous pouvez déjà avoir un aperçu statique de ce dont à l'air l'écran qui correspond à votre activité par défaut dans le volet 'Preview' de l'outil d'édition des fichiers XML. (Cf. Figure 9.)

## Exécuter un projet Android

Pour tester le projet que vous venez de créer :

- Allez à Run > Edit Configurations. Dans l'encadré 'Target Device' sélectionnez 'Show chooser dialog' (cf. Figure 13). Cette option vous permettra de choisir entre lancer l'application sur un émulateur ou



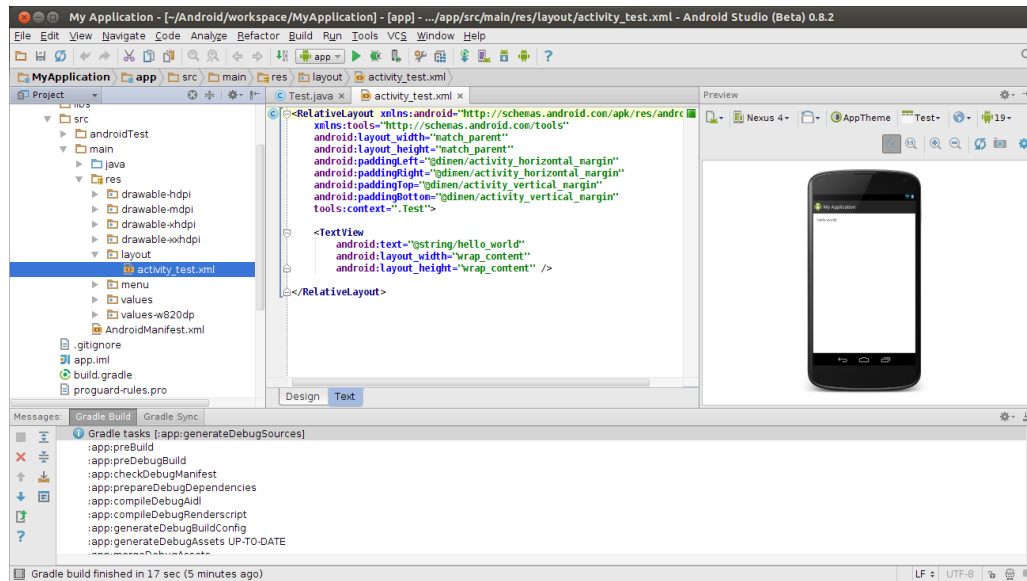


FIGURE 9 – Nouveau Projet crée



FIGURE 10 – Icône AVD Manager

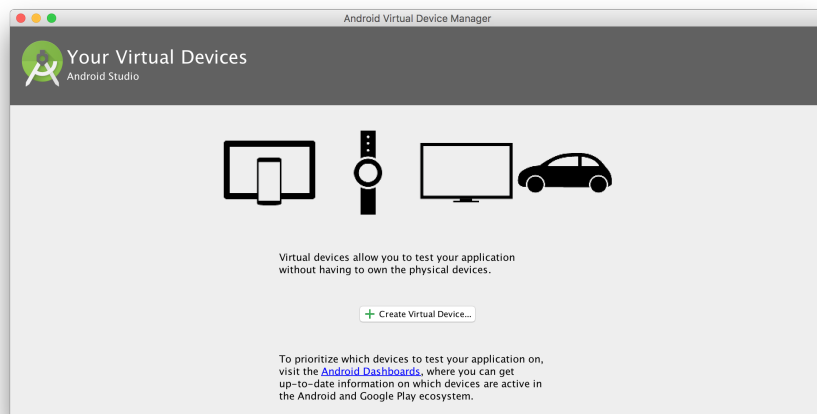


FIGURE 11 – Liste des AVD

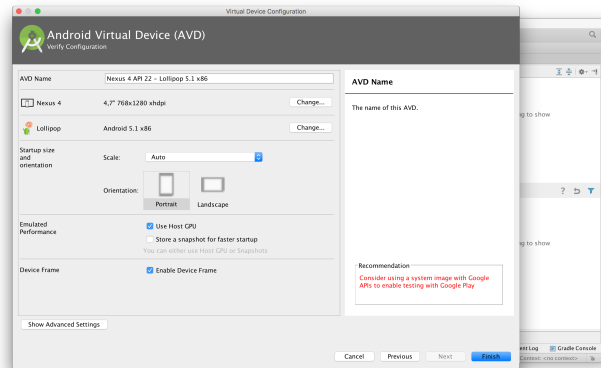


FIGURE 12 – Menu de création d'un AVD

directement sur un appareil Android connecté. Cliquez 'OK'.

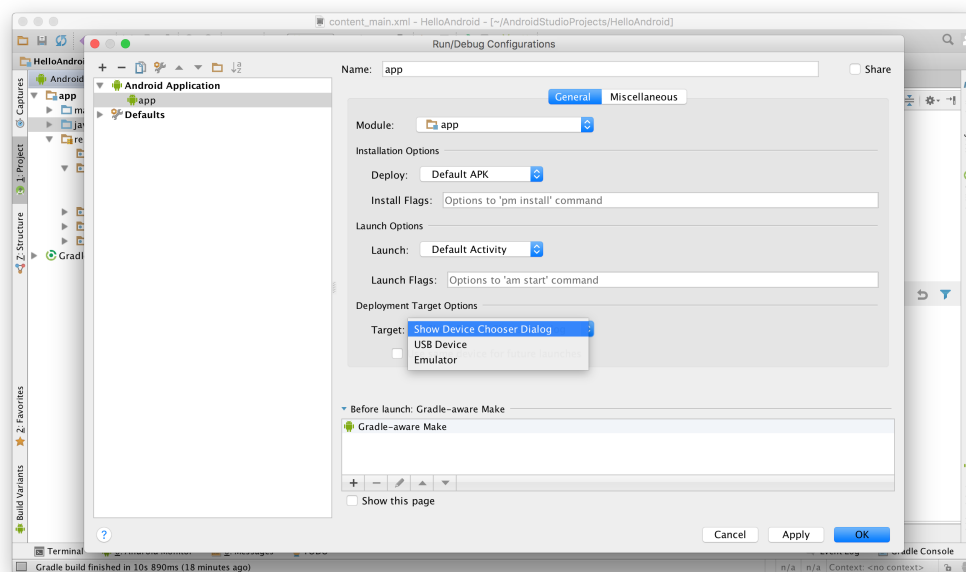


FIGURE 13 – Configuration de l'exécution

- Cliquez sur l'icône 'Run' de la barre d'outils. Dans le dialogue qui apparait sélectionnez soit un appareil Android connecté, soit une AVD existant et cliquez sur 'OK'. Un exécutable de votre application sera alors crée et lancé sur le dispositif choisi (il faut là aussi être patient).

LSINF1225	Séance 5a: Préparer l'environnement de travail Android — Hello World —	Dest : étudiants
2016-02-26		Auteur : KM,DM,CZM,XC

## Problèmes

Voici quelques problèmes que vous pourriez rencontrer lors de l'installation d'*Android Studio* et des indications pour vous aider à les surmonter. Si ces indications ne vous aident pas à surmonter vos soucis d'installation ou si vous rencontrez d'autres soucis, n'hésitez pas à contacter votre assistant/tuteur. (Après avoir consulté d'autres tutoriels sur Internet.)

— *Sur Windows, un appareil Android est connecté mais n'est pas détecté par Android*

→ Installer "universal ADB plug-in manager" ici  
<https://plus.google.com/103583939320326217147/posts/BQ5iYJEaaEH>

— *Un message du type "échec du redémarrage de ADB" apparaît*

→ Dans un terminal, se rendre à `/android-studio/sdk/platform-tools` et exécuter

```
adb kill-server
adb start-server
```

Si aucun message d'erreur n'apparaît cela signifie que adb a redémarré avec succès.

## Annexe 1 : Installation de Java sur les distributions Linux

Référence : <http://stackoverflow.com/questions/17474963/android-studio-tools-jar-file-is-not-present-in-classpath>

— Vérifier que le JDK est correctement installé.

```
dpkg --get-architecture | grep -i jdk
```

ii	icedtea-7-jre-jamvm:amd64	7u55-2.4.7-1ubuntu1~0.13.10.1
ii	openjdk-7-jre:amd64	7u55-2.4.7-1ubuntu1~0.13.10.1
ii	openjdk-7-jre-headless:amd64	7u55-2.4.7-1ubuntu1~0.13.10.1

Il n'y a que l'openJDK.

— Installer le JDK. Nous ajoutons un nouveau répertoire aux répertoires de confiance connus de notre système.

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update && sudo apt-get install oracle-jdk7-installer
```

<b>LSINF1225</b>	<b>Séance 5a: Préparer l'environnement de travail Android</b> <b>— Hello World —</b>	<b>Dest : étudiants</b>
<b>2016-02-26</b>	<b>Auteur : KM,DM,CZM,XC</b>	

- Après l'installation activer le JDK  
`update-alternatives --display java`
- Vérifier que le système utilise Java JDK7  
`java -version`  
  
`java version "1.7.0\_65"`  
`Java(TM) SE Runtime Environment (build 1.7.0\_65-b17)`  
`Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04, mixed mode)`
- Vérifier la version du compilateur  
`javac -version`
- Ajouter JAVA\_HOME à la variable d'environnement dans le fichier /etc/environnement en ajoutant  
`JAVA_HOME=/usr/lib/jvm/java-7-oracle` à la fin du fichier.

## **Annexe 2 : Comment lancer son application sur un appareil Android**

- Activer le mode 'Debug' de l'appareil Android : retrouver parmi les options de réglages les 'Options pour les développeurs' puis activer le débogage USB.  
 Attention : le mode 'Debug' est caché à partir d'Android 4.2.
- Procédez de la même façon que pour lancer votre application sur un AVD en sélectionnant dans le dialogue l'appareil Android que vous aurez au préalable connecté à votre ordinateur.