

## TP1\_android

### Introduction et installation d'Android

#### 1 Introduction

Android est un système d'exploitation pour téléphone portable de nouvelle génération développé par Google. Celui-ci met à disposition un kit de développement (SDK) basé sur le langage Java. Ce TP va vous expliquer les étapes d'installation du SDK et des différents IDE pour le développement pour Android. Il explique aussi les étapes de création d'un projet d'application Android.

#### 2 Installation

##### 2.1 Installation du SDK

Basé sur le langage Java, le SDK Android nécessite d'avoir un **compilateur JAVA**. Un JDK (5 ou 6) doit être installé sur la machine de travail, sinon vous pouvez vous connecter au site suivant pour télécharger et installer le JDK :

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

Le SDK Android est disponible sur le site

<http://developer.android.com/sdk/index.html>

- Sous windows :  
Vous n'avez qu'à suivre les instructions de l'installateur et retenir le répertoire du SDK, on aura besoin pour le paramétrage de l'IDE de développement.
- Sous Linux :  
Editez votre fichier ~/.bash\_profile ou ~/.bashrc et cherchez la ligne définissant le PATH afin d'y ajouter le chemin <android\_sdk>/tools. Si cette ligne n'existe pas, ajoutez celle-ci: export PATH=\${PATH}:<android\_sdk>/tools. Vous pouvez également taper directement cette commande dans une fenêtre Shell mais celle-ci ne sera alors valide que pour ce shell (et ses fils). Il vous faudra donc ressaisir la commande si vous fermez votre shell.

## 2.2 Eclipse

Il y a quatre solutions pour développer simplement avec les outils fournis par Google pour son SDK Android, soit utiliser l'IDE Eclipse, soit utiliser Netbean, soit utiliser le ADT Bundle et enfin soit utiliser la nouveau outil Android Studio. Tous les TP d'android seront traité avec le ADT Bundle qui est un outil qui incorpore son Eclipse paramétré avec son ADT et SDK. Dans ce TP, nous allons présenter l'installation de l'IDE Eclipse et sa configuration pour l'étendre au développement d'android. De même, nous allons présenter l'outil de développement le ADT Bundle.

### 2.2.1 Installation d'Eclipse

Si Eclipse est installé sur votre machine : **C:\Program Files\eclipse** tant mieux, sinon, connectez- vous à l'adresse <http://www.eclipse.org/downloads/> et téléchargez la version « Eclipse IDE for Java Developers ». Extrayez l'archive et exécutez Eclipse.

Si vous êtes derrière un proxy avec ou sans authentification vous devez paramétrer Eclipse : Eclipse/Windows > Preferences > general > Network Connections

Active Provider : Manual

Remplissez les variables HTTP et HTTPS

## 2.3 Installation de l'ADT

Sachez que Google fournit un plugin pour cet IDE nommé Android Développement Tools (ADT). Voici la marche à suivre pour installer ce plugin sur Eclipse :

- Démarrez Eclipse puis sélectionnez le menu Help > Software Updates
- Dans la fenêtre qui vient de s'ouvrir, cliquez sur l'onglet Available Software.
- **Au cas où vous avez téléchargé ADT** ou que vous avez une version récente de l'ADT, cliquez sur archive et indiquez le chemin où se trouve le fichier, sinon cliquez sur Add Site et taper dans la boîte de dialogue qui apparaît un nom (par exemple Android Plugin) et 'taper l'URL <https://dl-ssl.google.com/android/eclipse/>. Appuyez sur le bouton OK.
- Dans l'onglet Available Software, sélectionnez Developer Tools et cliquez sur Install...
- Finissez l'installation puis redémarrez Eclipse.

Le **Plug-in ADT** vous permet entre autres, de compiler, débbugger, tracer, packager, tester et analyser vos applications Android. Il s'appuie sur le **SDK Android** pour bon nombre de ces actions mais l'avantage est qu'il intègre la plupart de celles-ci au sein d'Eclipse, facilitant grandement les différentes phases de l'élaboration d'applications Android. Il apporte également deux perspectives propres au développement Android, lesquelles sont **DDMS** (Dalvik Debug Monitor Server) et **Hierarchy View**, que l'on peut activer à



l'aide de la barre d'outils représentée ici :

Ces perspectives constituent deux contextes très complets pour le débogage et l'analyse de vos applications, afin de traquer les erreurs de codage.

Le plug-in ajoute également des raccourcis vers deux des outils du SDK Android, que sont **Android SDK Manager** et **Android Virtual Device Manager**, matérialisés par la présence des boutons suivants :



Ceux-ci vous permettent respectivement d'installer l'ensemble des composants du SDK Android et des "add-on" Google, ainsi que de gérer les différents émulateurs de terminaux mobiles.

## 2.4 Paramétrage d'Eclipse pour incorporer le SDK

Une fois Eclipse redémarré, sélectionnez le menu Window > Preferences... (ou Eclipse > Preferences si vous êtes sous Linux).

- Sélectionnez Android dans le panel de gauche.
- Indiquez le chemin où vous avez installé le SDK Android (bouton Browse pour parcourir le système de fichier).
- Appuyez sur le bouton OK.

## 3 Installation de l'ADT Bundle

Pour installer l'ADT Bundle, il suffit de télécharger le fichier zip se trouvant à l'adresse suivante :

[https://dl.google.com/android/adt/22.6.2/adt-bundle-windows-x86\\_64-2014321.zip](https://dl.google.com/android/adt/22.6.2/adt-bundle-windows-x86_64-2014321.zip)

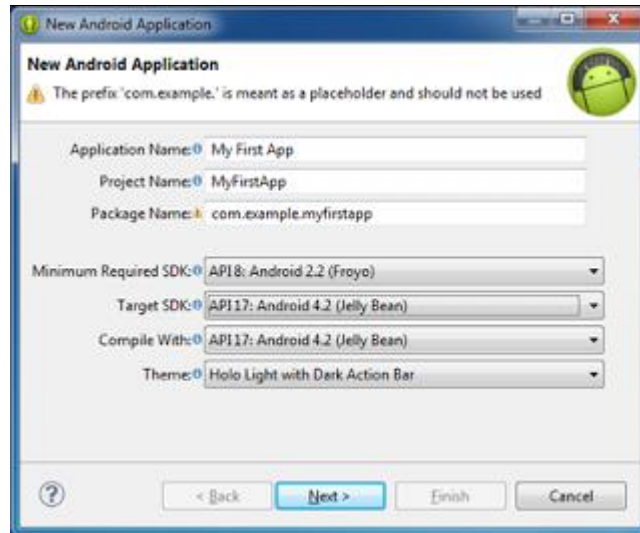
Ce fichier est pour Windows 64 bits, vous pouvez trouver la même version pour le Windows 32 bits et Linux. Une fois ce fichier est dézippé il suffit de démarrer Eclipse et de faire le paramétrage du SDK qui viens avec le fichier zippé.

De même, vous pouvez aller au site suivant télécharger l' ADT Bundle et suivre les instructions d'installation.

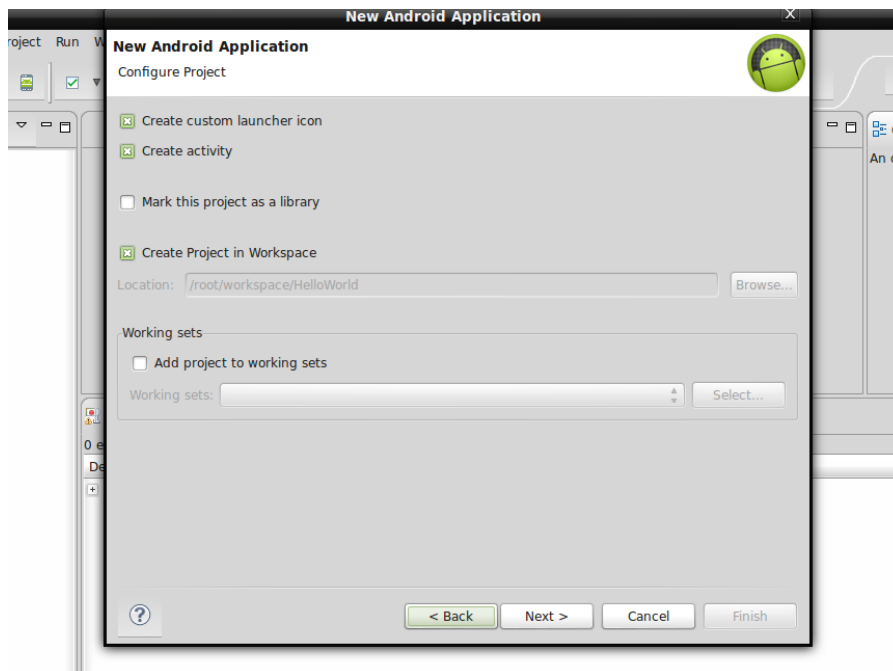
<http://www.technotalkative.com/adt-bundle-a-single-step-to-setup-android-development-environment/>

## 4 Première application avec Android : Helloworld

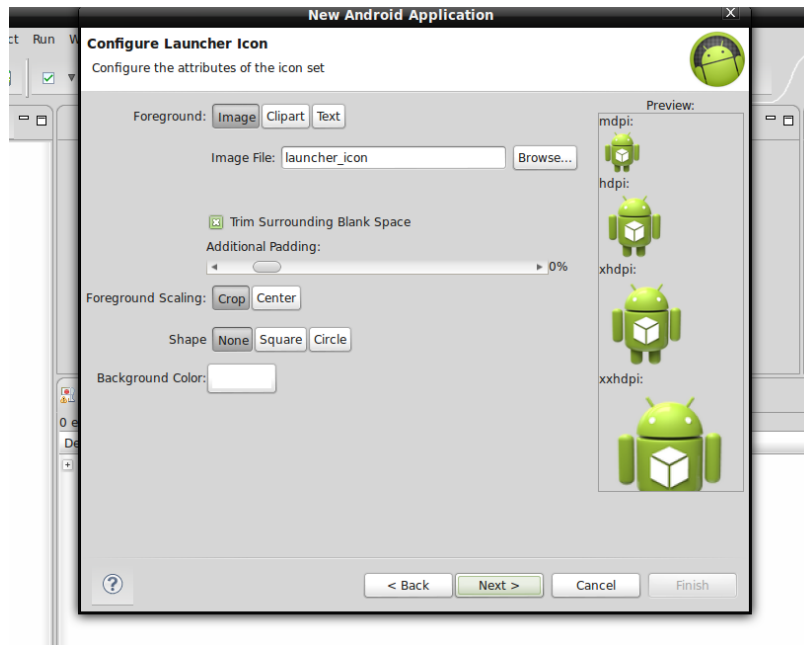
Pour développer une application avec ADT Bundle, commencez par sélectionner le menu File > New > Android Application Project.



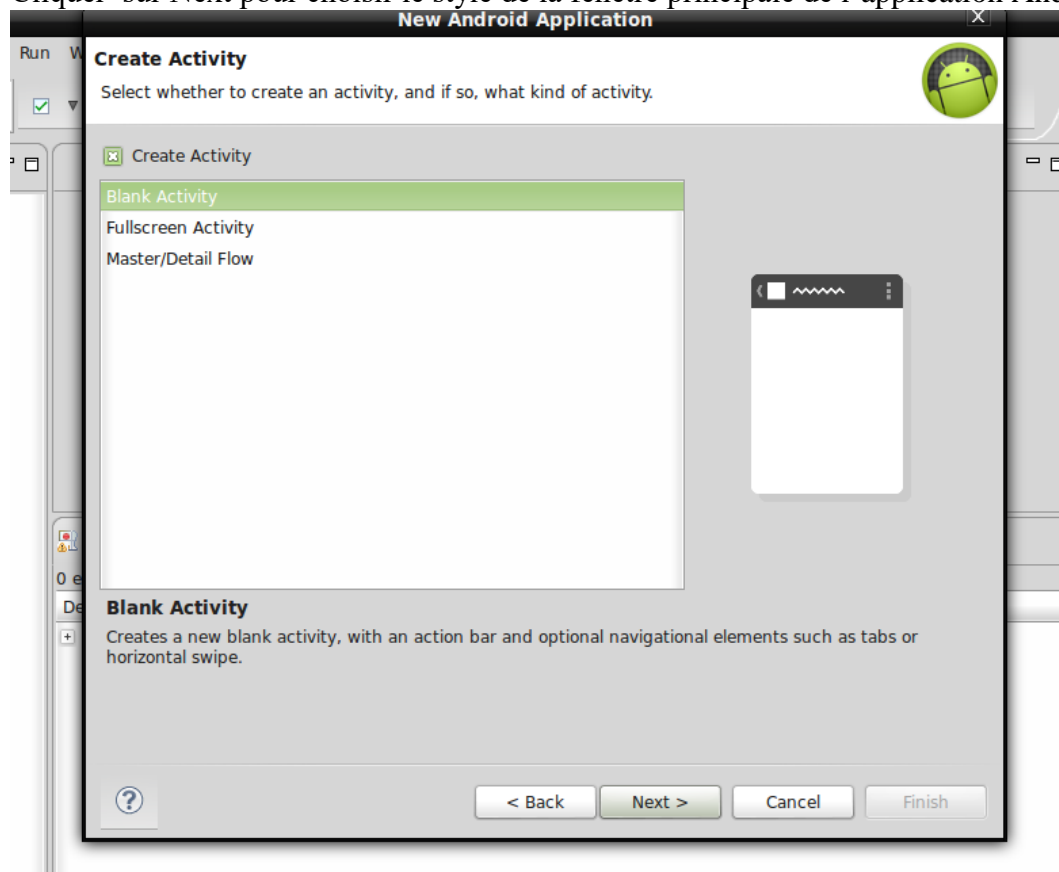
Dans le champ Application Name, saisir le nom de l'application; il doit commencer par un majuscule ; celui-ci se répète automatiquement au niveau des champs Project Name et Package Name, vous n'avez pas besoin de les changer. Les champs Minimum Required SDK et Target SDK donnent la version de SDK minimale et maximale pour le fonctionnement et développement de l'application. Le champ compile with donne la version SDK avec la quelle sera compilée l'application et enfin le champ Theme donne le style des 'fenêtres' qui seront utilisées par l'application. Appuyer sur Next.



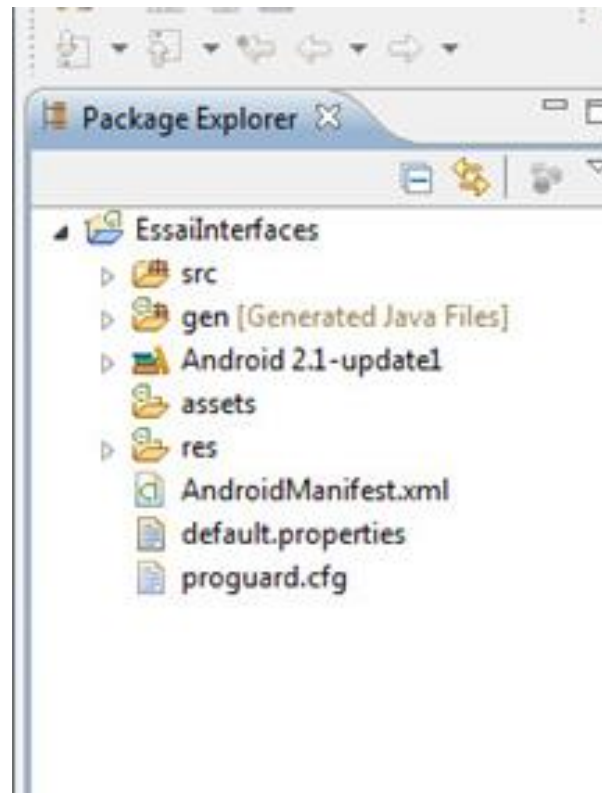
Laisser les sélections telles qu'elles sont et appuyer une autre fois sur Next.



La fenêtre ci-dessus vous permet de configurer l'icône de lancement de l'application. Cliquer sur Next pour choisir le style de la fenêtre principale de l'application Android.



Choisissez pour commencer le style Blank Activity. Cliquer sur Next pour donner un nom à cette fenêtre principale le nom généralement fourni est 'Main\_Activity'. Appuyer sur Finish et attendre qu'Android vous crée le projet de l'application. Une fois créé, le projet possède plus au moins la structure arborescente suivante :



**Le répertoire src** contient le code source java de l'application. **Le répertoire gen** contient les codes autogérés par Android lors de la construction de l'application. **Le répertoire**

**Android 2.1-update1**

montre la version d'Android utilisée pour la compilation. **Le**

**répertoire res** contient toutes les ressources utilisées par l'application, il est composé des sous répertoires suivants :

- res/drawable pour les images et icônes. Ces derniers peuvent être en hdpi ou high-resolution dot per inch (pour les images et icônes de haute qualité ), ou en ldpi (pour les icônes et images de mauvaise qualité), ou enfin en mdpi ( pour les images et icônes de qualité moyenne). Nous pouvons trouver également les répertoires xhdpi ou xxhdpi destinés à contenir des images de très hautes résolution.
- res/layout pour la description en XML de la composition de l'interface graphique.
- res/menu pour la description en XML des menus.
- res/values pour la définition des constantes, des messages, ....

Nous pouvons également trouver **Le répertoire bin** qui contient le code de l'application compilée. **Le répertoire lib** contient les bibliothèques nécessaires pour l'exécution de l'application. **Le répertoire assets** qui contient simplement des données qui seront chargées sur le mobile lors de la compilation de votre projet, cela peut, par exemple, être des fichiers texte décrivant la licence de l'application ou des fichiers audio ou vidéo. Enfin, le fichier Manifeste.xml est le point de départ de toute application Android. Il regroupe les éléments de configuration de l'application.

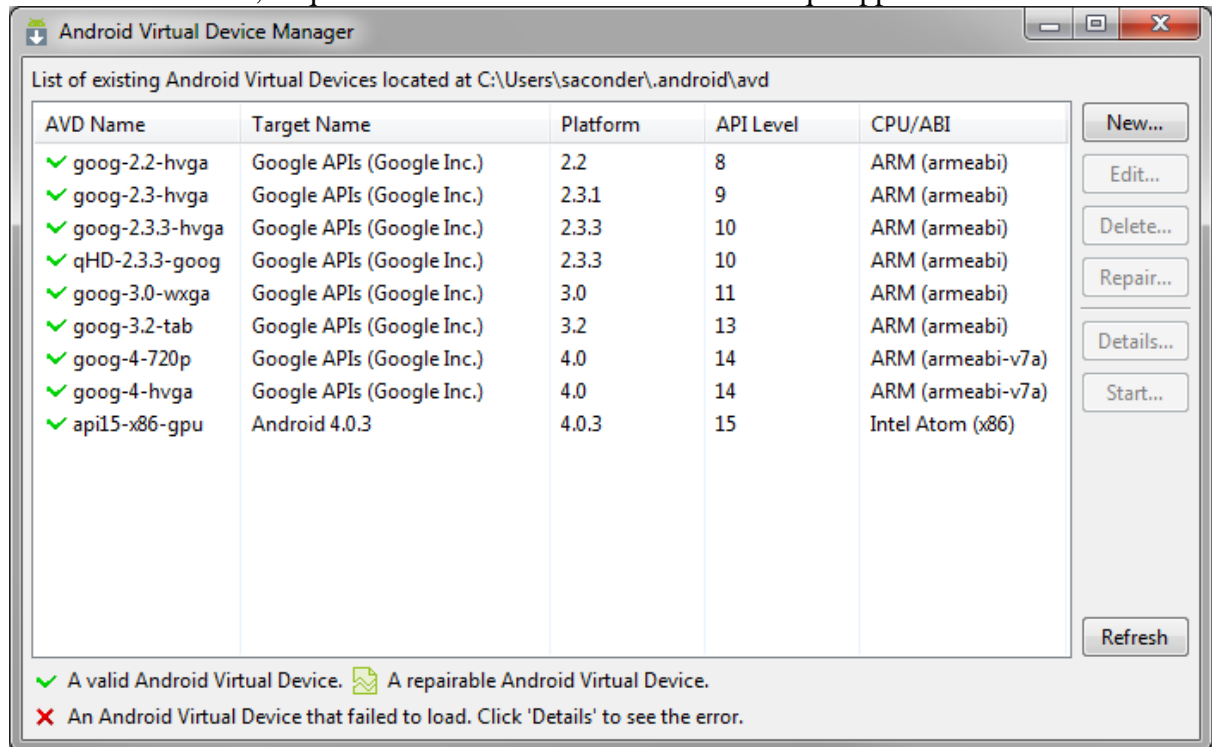
Maintenant, pour voir le mode de fonctionnement de l'application sur un téléphone portable, nous devons créer une machine virtuelle qui émule un Smartphone.

## 5 Prise en main de l'émulateur (AVDM)

L'Android Virtual Device Manager permet de créer des machines virtuelles servant à exécuter les applications Android. Pour le lancer allez au menu Windows et choisissez AVDM.

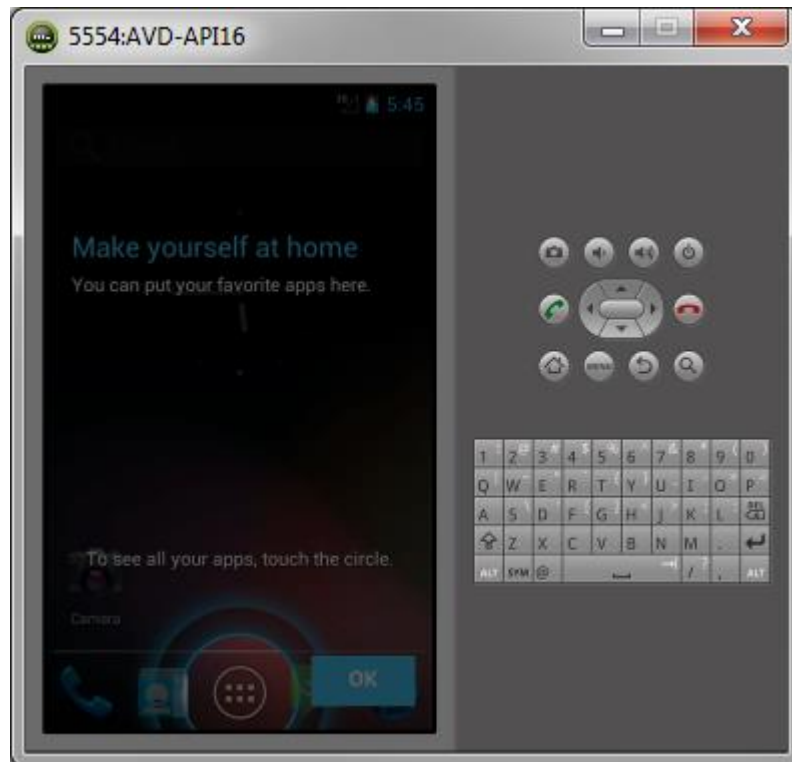
### 5.1 Créer un émulateur : Un Android Virtual Device (AVD)

Pour créer un AVD, cliquez sur le bouton New dans la fenêtre qui apparait ci-dessous.

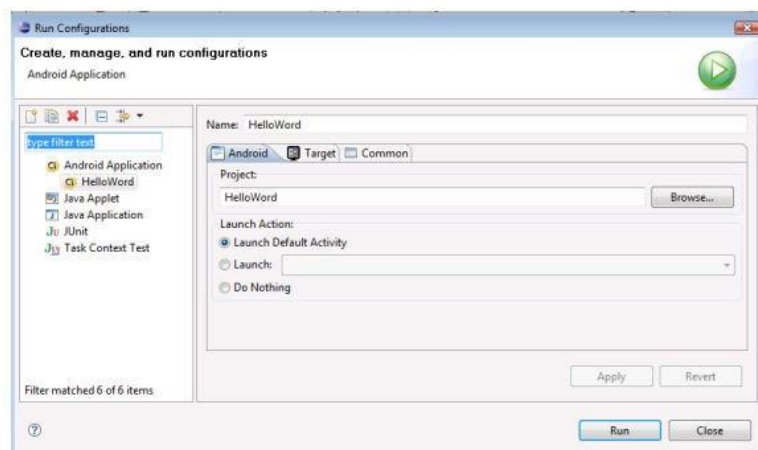


Une fenêtre de paramétrage de l'AVD s'affiche. Commencer par saisir le nom de votre machine virtuelle. Choisissez ensuite le type de votre device. Une fois ce dernier choisi, la version d'Android et le type de CPU sont automatiquement sélectionnés. Le champ Skin permet de sélectionner la résolution de l'écran. Vous pouvez selon vos besoins définir un front/back caméra. Si l'application n'a pas besoin de ceci il est préférable de ne pas les définir. Le memory option permet de définir la taille RAM de votre device, le internal storage définit la taille du disque dur du portable. De plus elle permet de définir la taille de mémoire maximale à partir de laquelle l'Android démarrera son Garbage Collector. Le AVD vous permet aussi de créer une machine virtuelle contenant éventuellement un SD card et vous permet de définir sa taille.

Une fois fini, il suffit de cliquer sur le bouton create AVD et votre émulateur sera créé comme suit :



Pour exécuter l'application, sélectionnez le menu Run > Run Configurations. Dans la boîte de dialogue qui s'ouvre, double cliquez sur l'entrée Android Application. Un nouveau lanceur nommé New\_configuration doit apparaître, sélectionnez le projet créé grâce au bouton Browse dans le champ Project.



Dans l'onglet « Target », cocher l'émulateur qu'on vient de créer et lancer l'exécution par le bouton Run. .