

## Tutoriel d'installation sur eclipse (plugin)

### Eclipse

Télécharger et installer eclipse

### SDK Eclipse

Télécharger le plug-in android

<http://developer.android.com/sdk/installing/installing-adt.html>

### libgdx

Télécharger libgdx

Générer un projet Desktop

### Gradle

Installer le plug in gradle dans eclipse

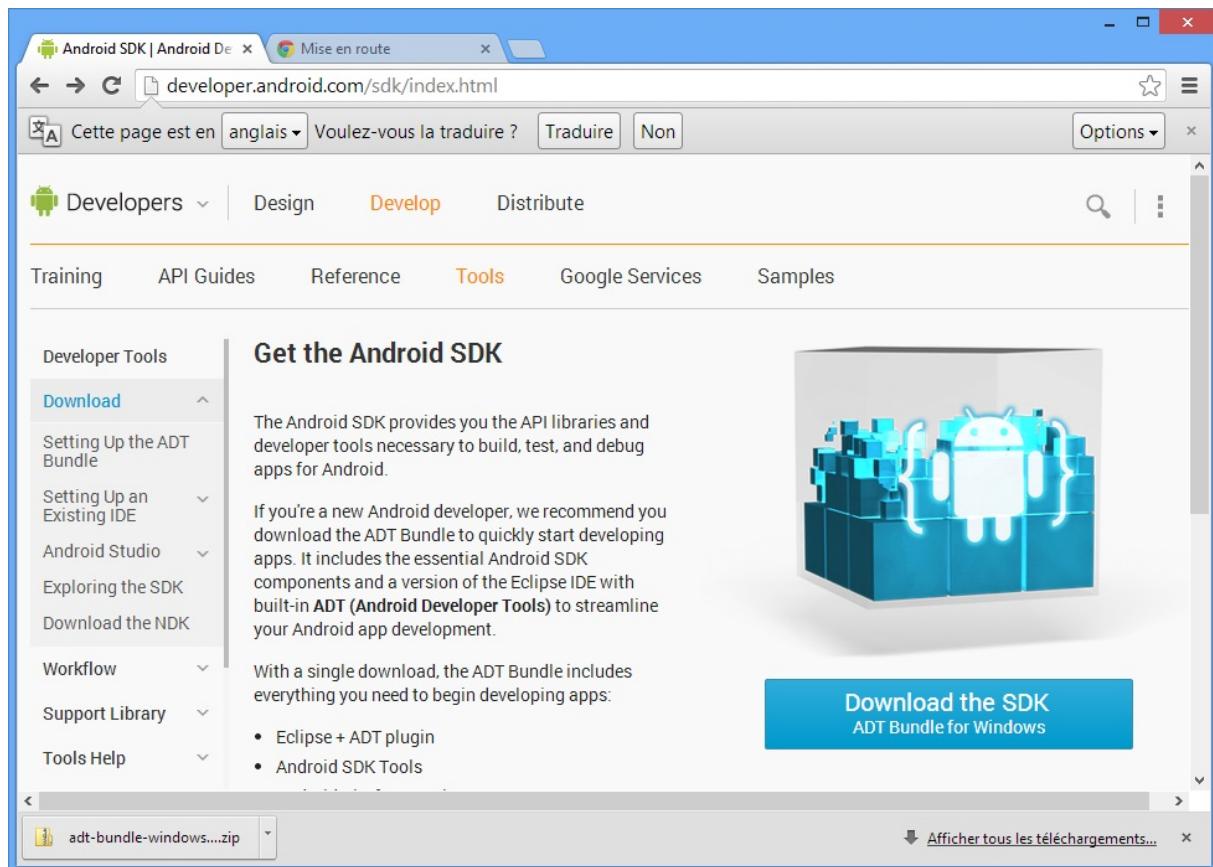
<https://github.com/spring-projects/eclipse-integration-gradle/#eclipse-integration-gradle>

Importer le projet libgdx dans eclipse

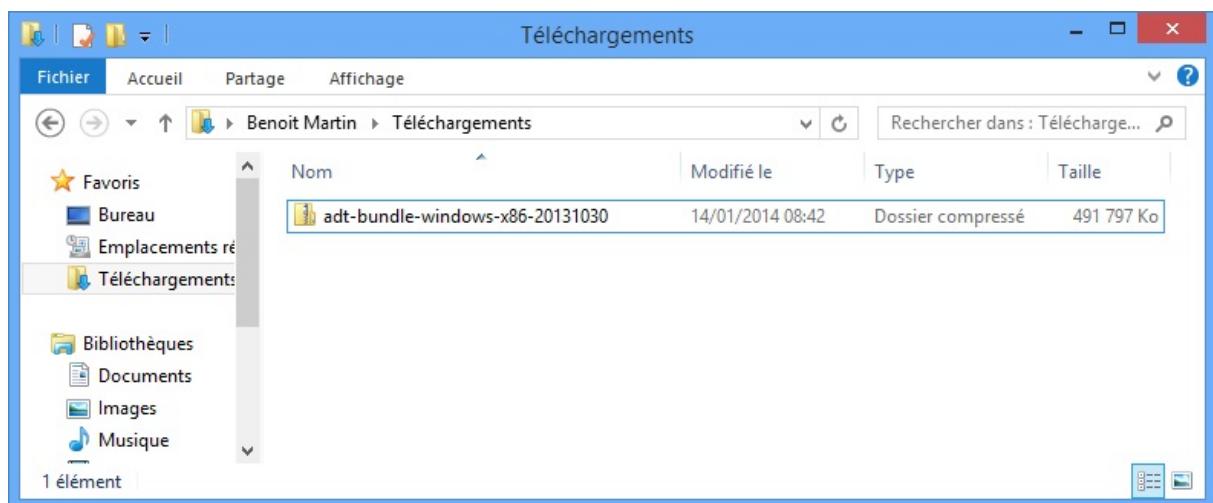
## Tutoriel d'installation d'ADT et de libgdx

### ADT

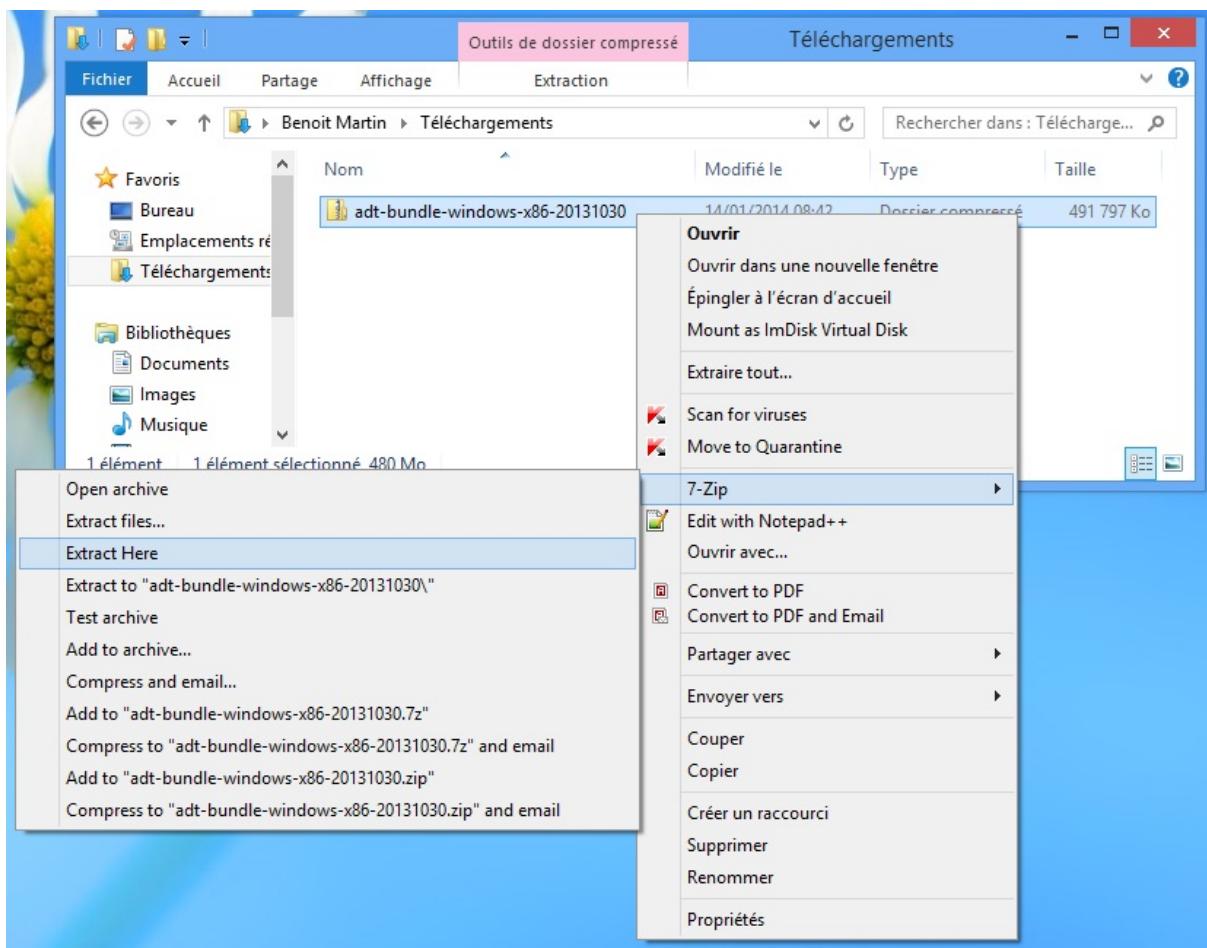
Télécharger le bundle ADT :



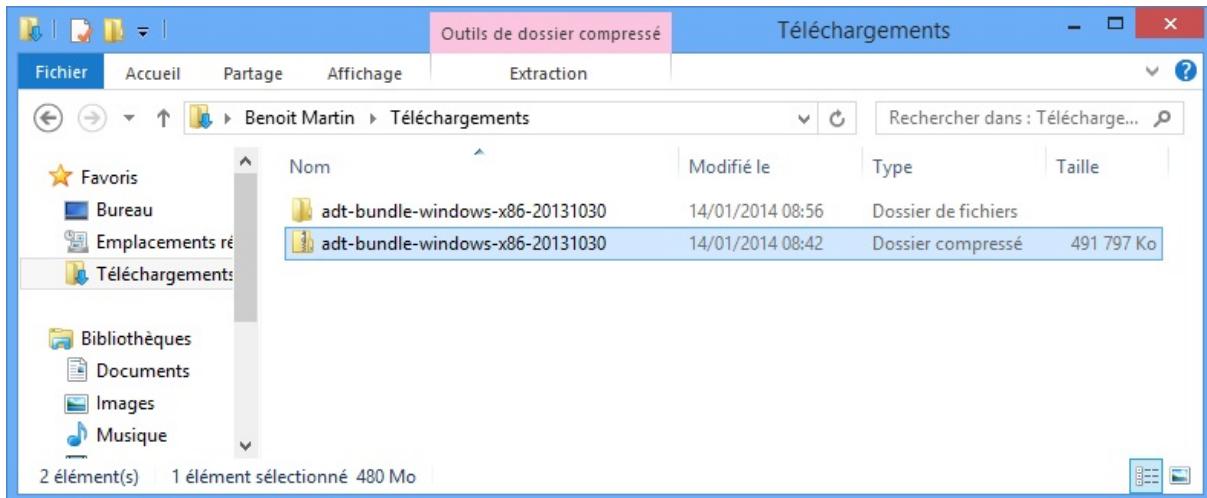
Le fichier se trouve dans le dossier Téléchargements :



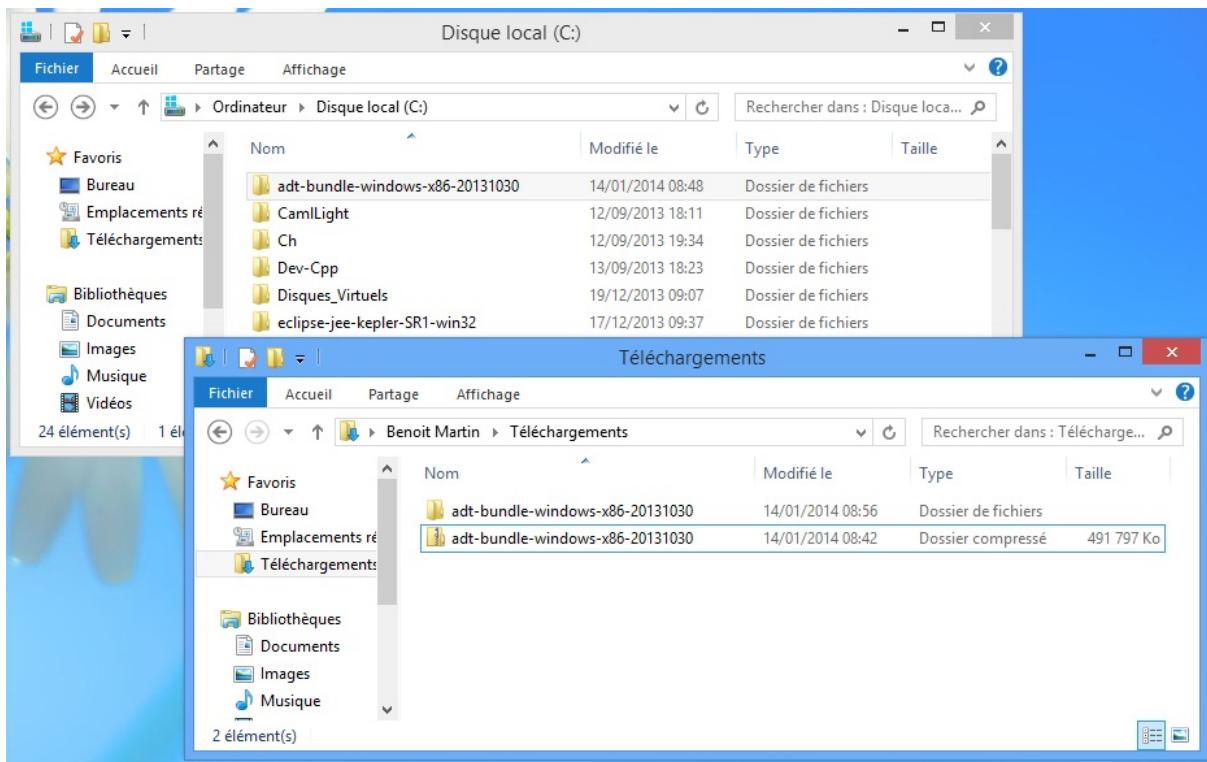
Décompresser l'archive :



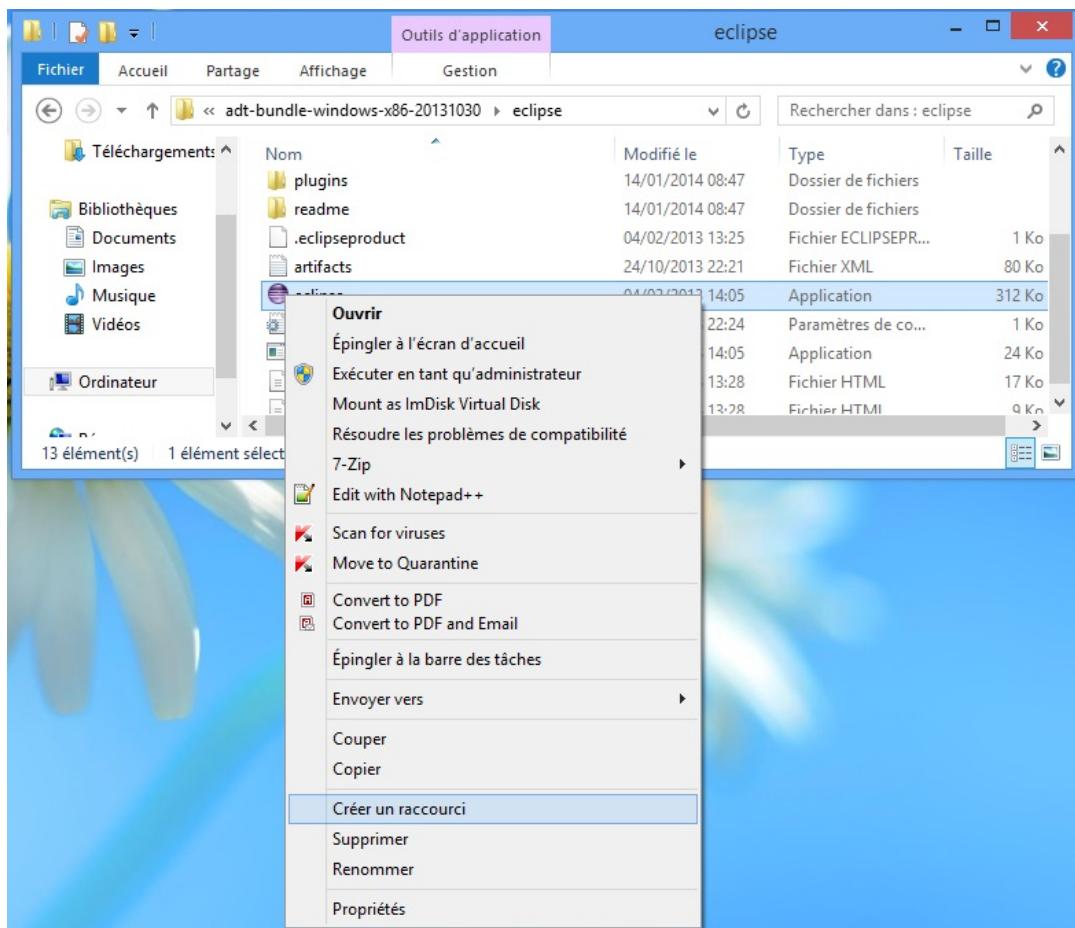
Résultat :



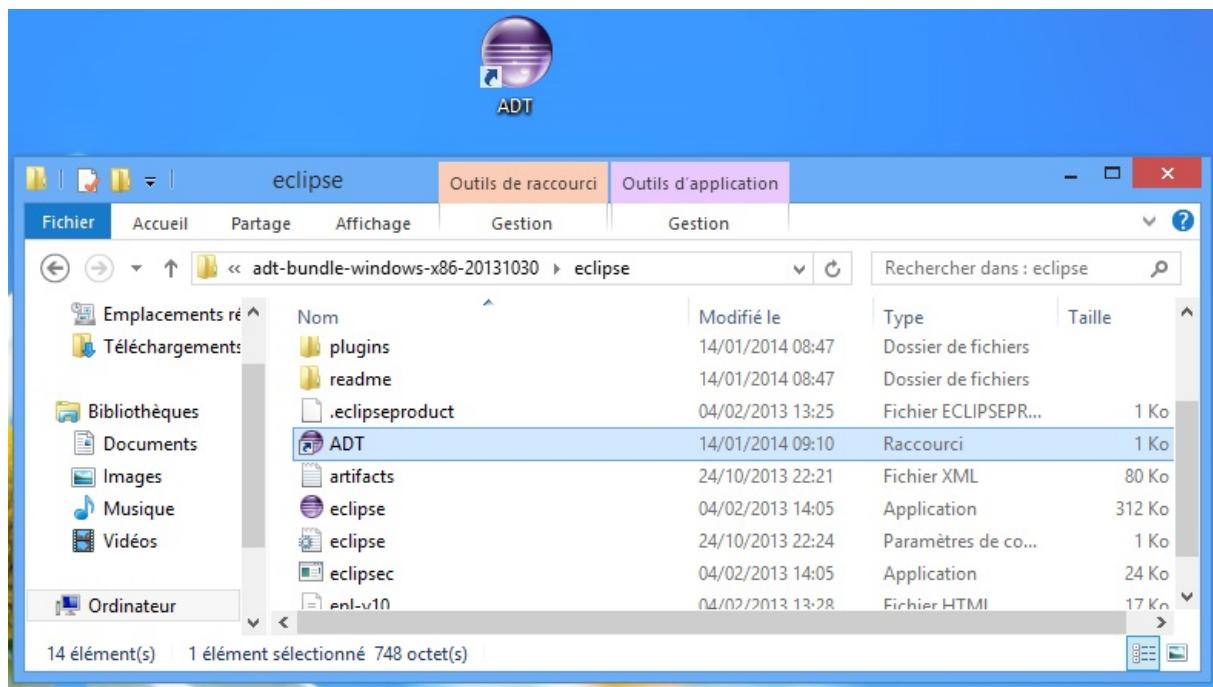
Déplacer le dossier décompressé vers "C:" :



Créer un raccourci que vous appellerez ADT :



Déplacer ce raccourci sur votre bureau :



## libGDX

Télécharger libGDX :

The screenshot shows a web browser window with the URL [libgdx.badlogicgames.com/download.html](http://libgdx.badlogicgames.com/download.html). The page has a dark theme with white text.

**Release Builds**

Release builds are versioned, e.g. 0.9.9, and are published periodically, usually every few months. They represent stable snapshots of libgdx. Note that new bug fixes and features are never backported to old libgdx releases.

[Download Latest Release](#)

[Old Releases](#)

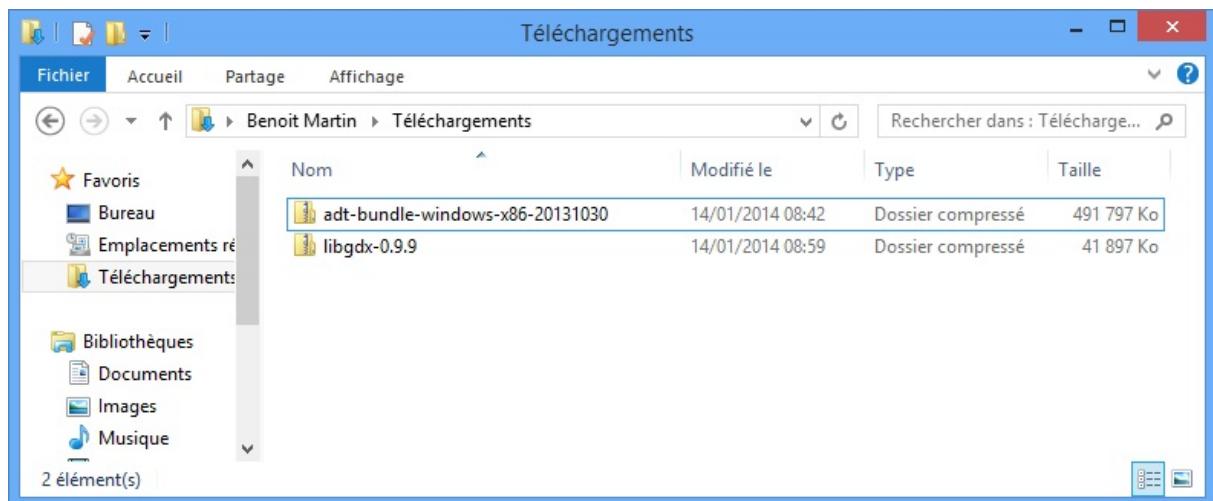
**Nightly Builds**

Nightly builds are created every night, and on every change in the Git repository, by our build server. They contain the latest and greatest bug fixes and features. Nightlies are usually very stable, just like our master branch on Github.

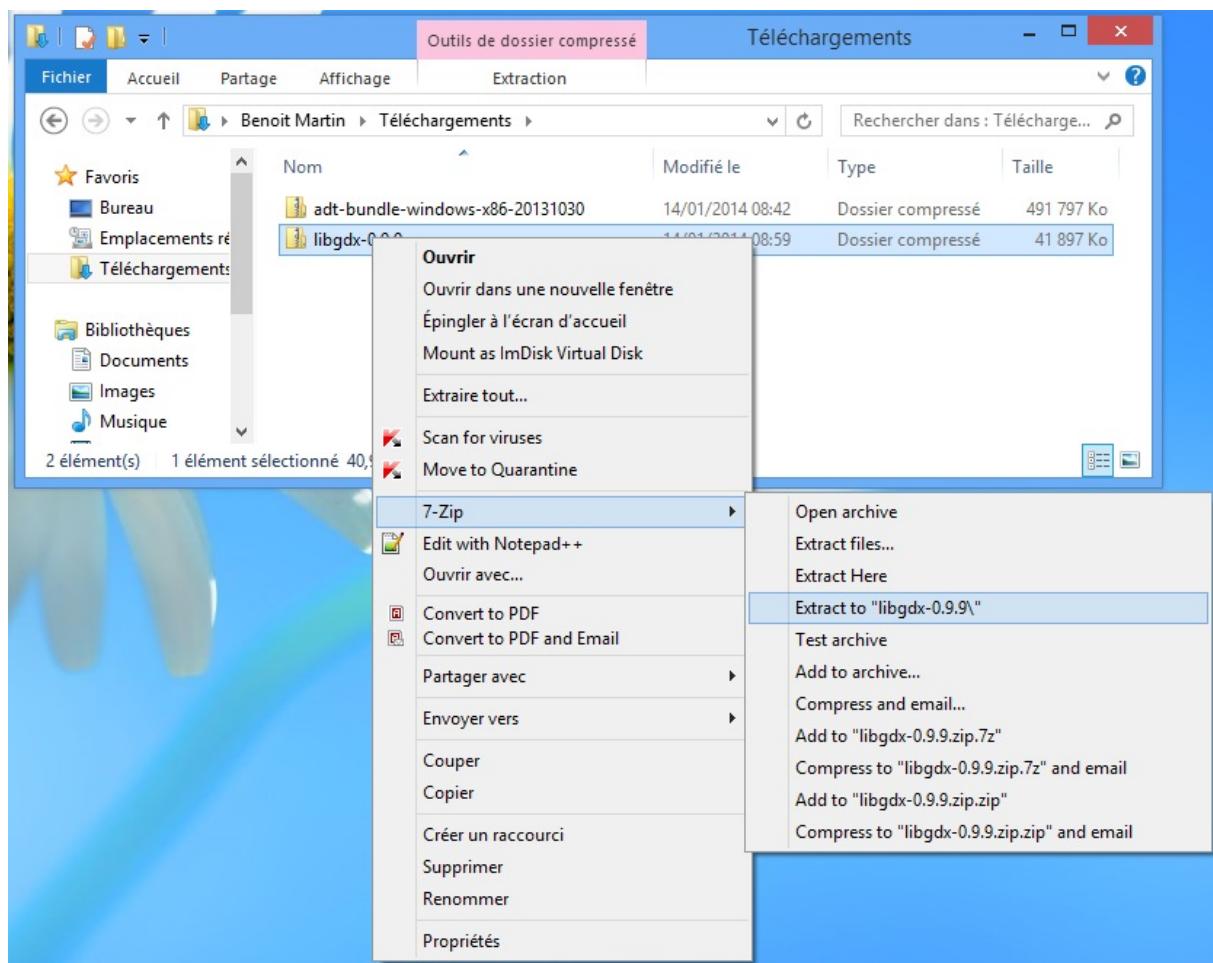
[Download Nightly](#)

At the bottom of the browser window, there are download progress bars for 'libgdx-0.9.9.zip' (10,5/40,9 Mo, 17 second...) and 'adt-bundle-windows...zip'. There is also a link 'Afficher tous les téléchargements...'.

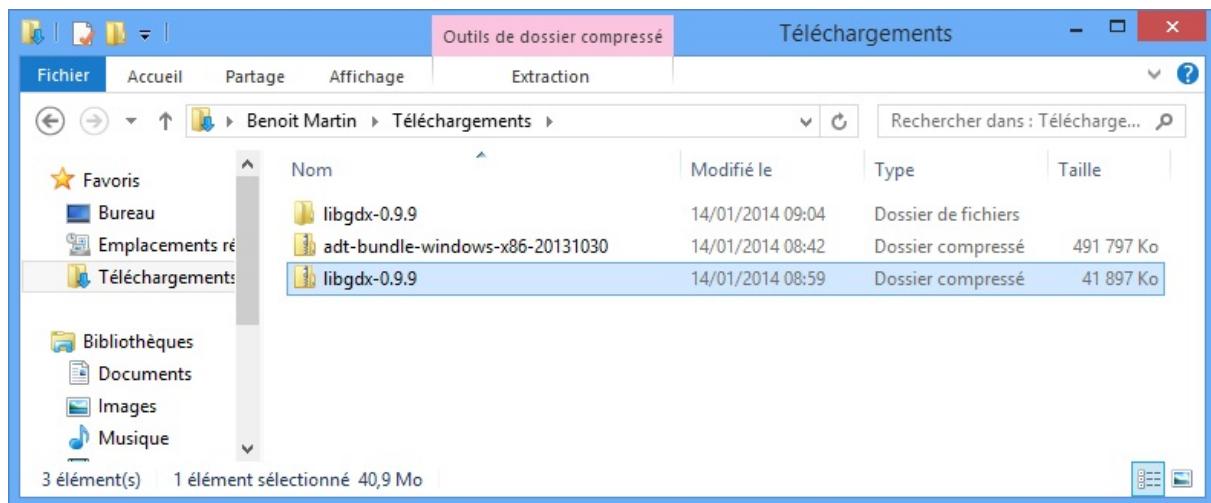
Le fichier se trouve dans le dossier Téléchargements :



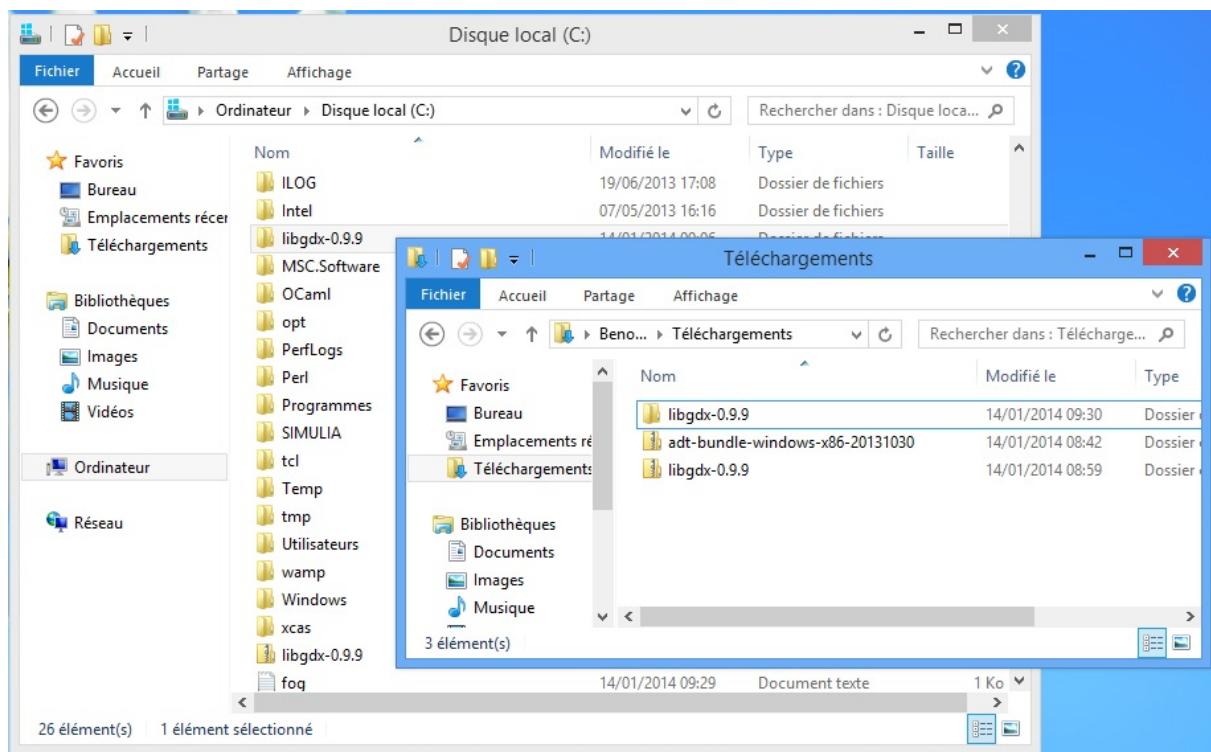
Décompresser l'archive :



Résultat :

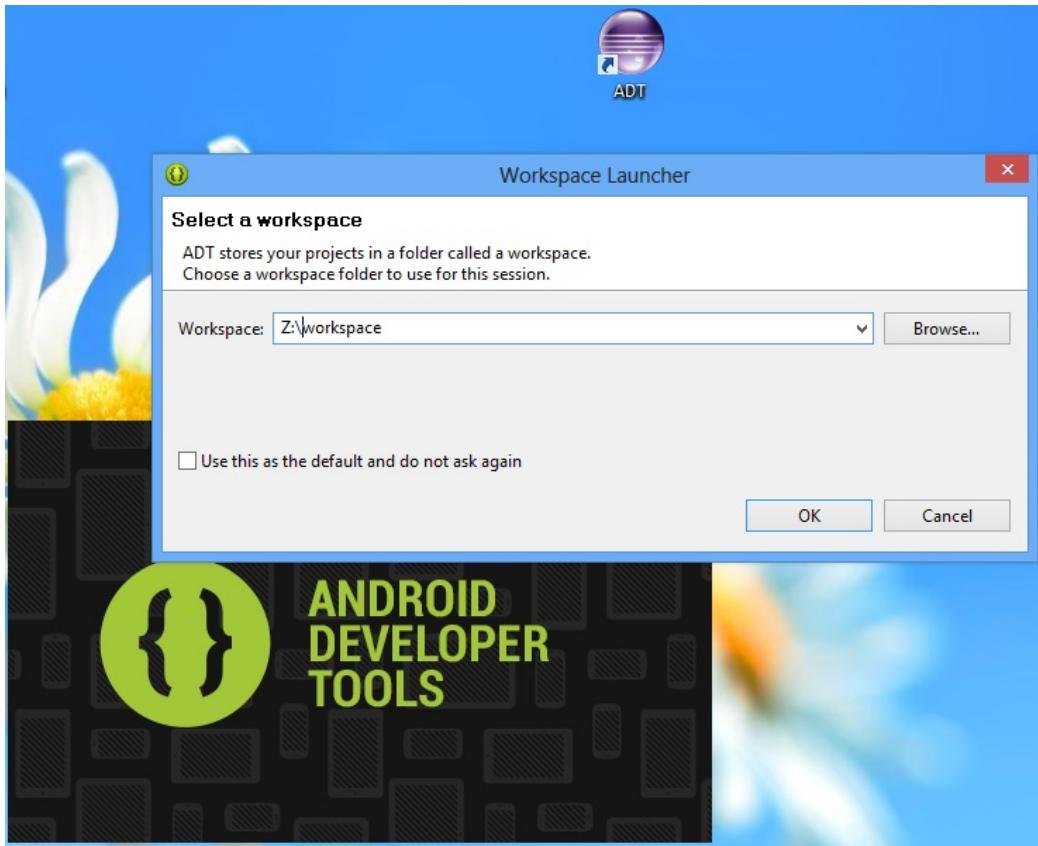


Déplacer le dossier décompressé et l'archive vers "C:" :



## Lancement d'ADT

Double-cliquez sur l'icône ADT :



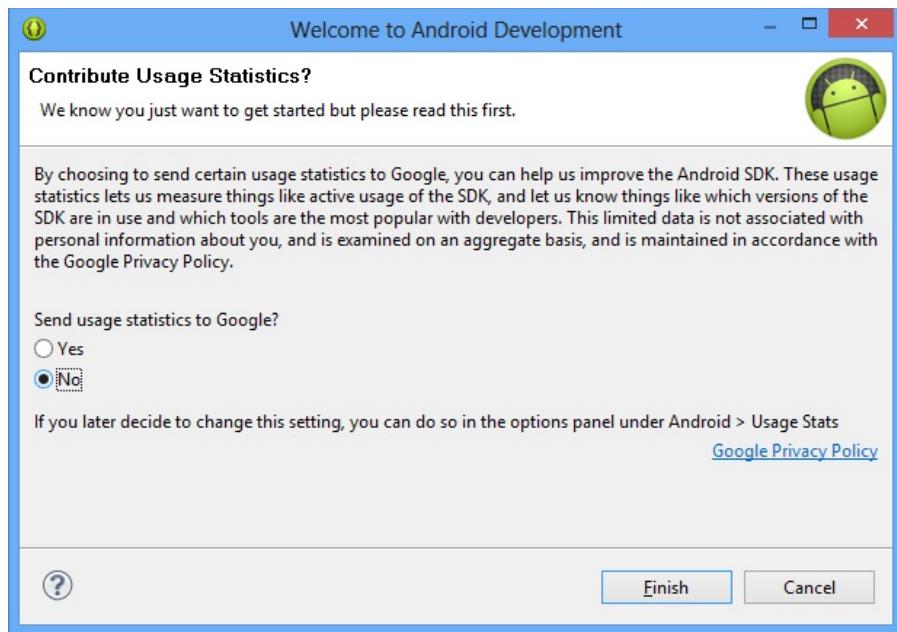
Si l'exécution échoue avec « failed ... », il convient de modifier le fichier « eclipse.ini » du répertoire de l'installation de l'ADT. Pour affecter moins de mémoire à la JVM, modifiez la ligne :

-Xmx768m

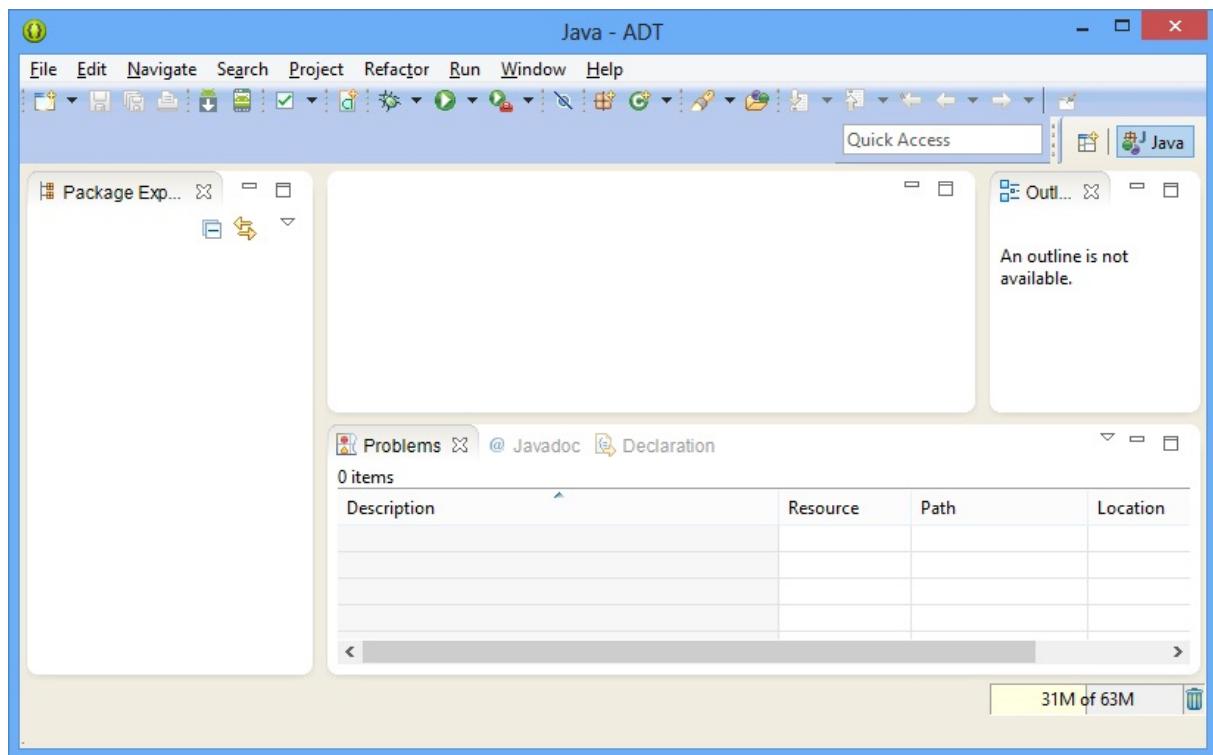
en

-Xmx512m

A cette étape, choisissez un répertoire pour votre espace de travail (workspace). Je vous propose de choisir « Z:\workspace » et de valider. Une fenêtre demande l'autorisation d'envoyer des statistiques chez Google. Cochez « No » puis validez.

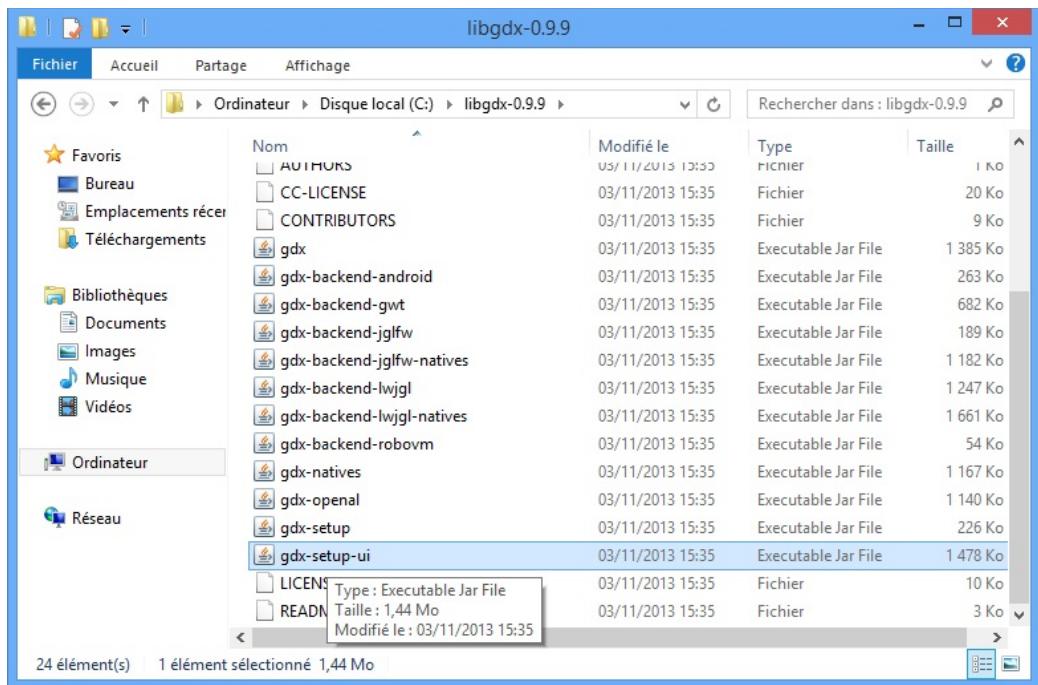


Résultat, un environnement vide :



## Création d'un projet libGDX

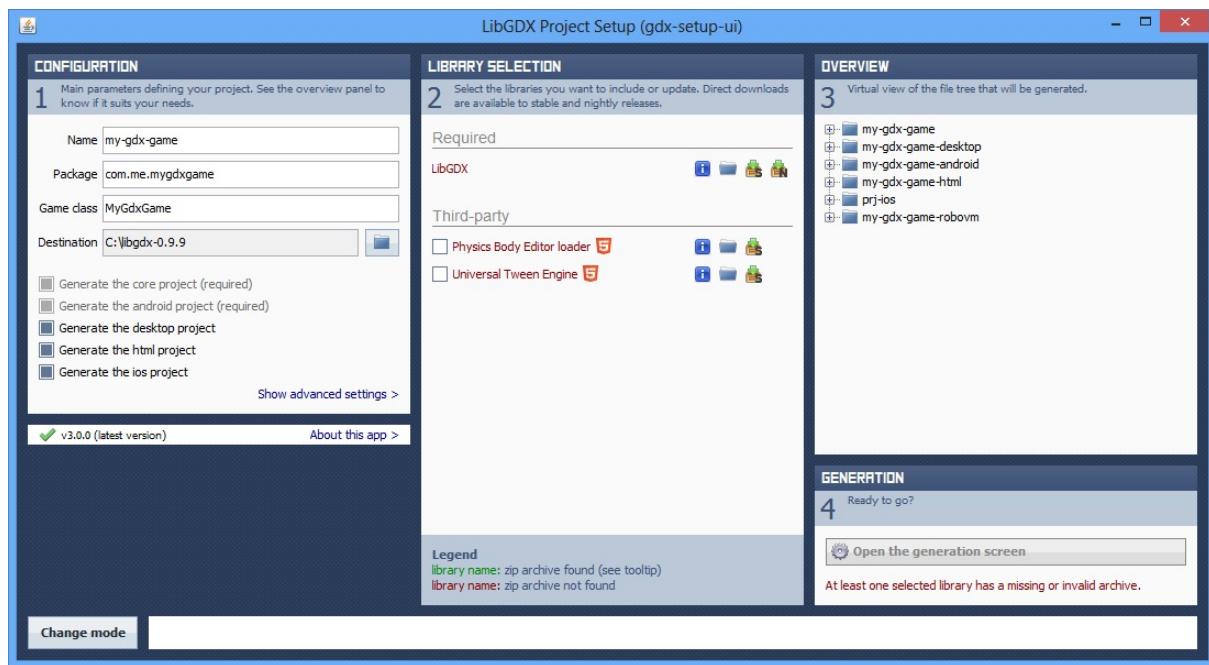
Lancement de l'application de paramétrage :



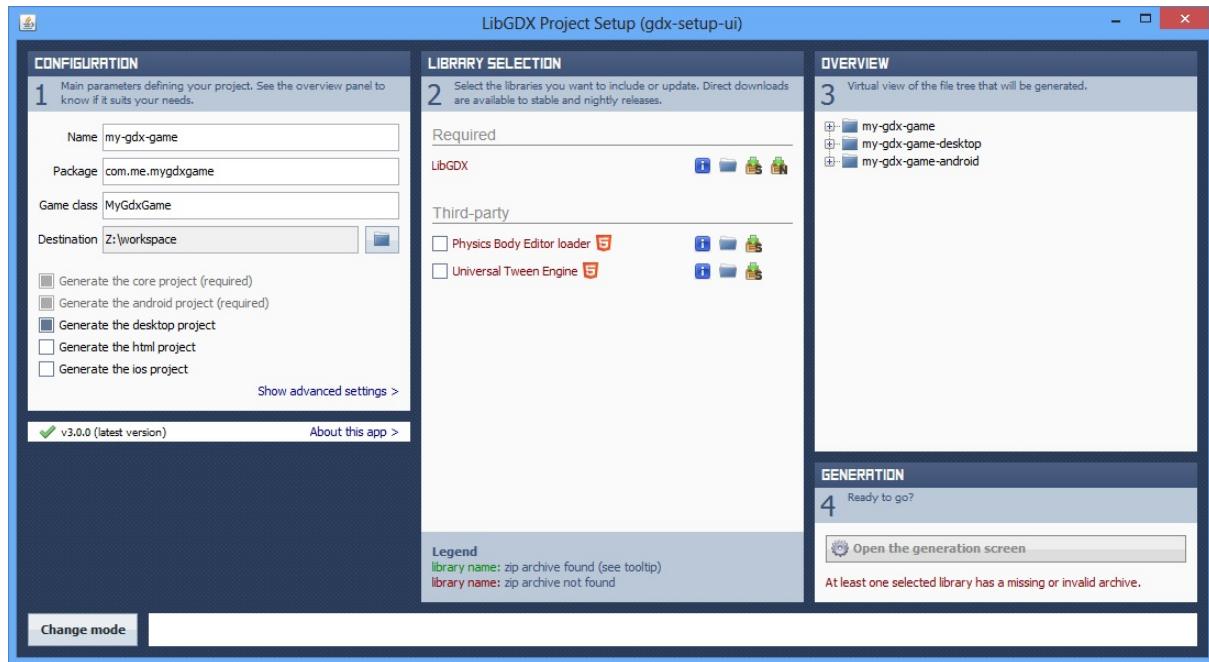
Résultat :



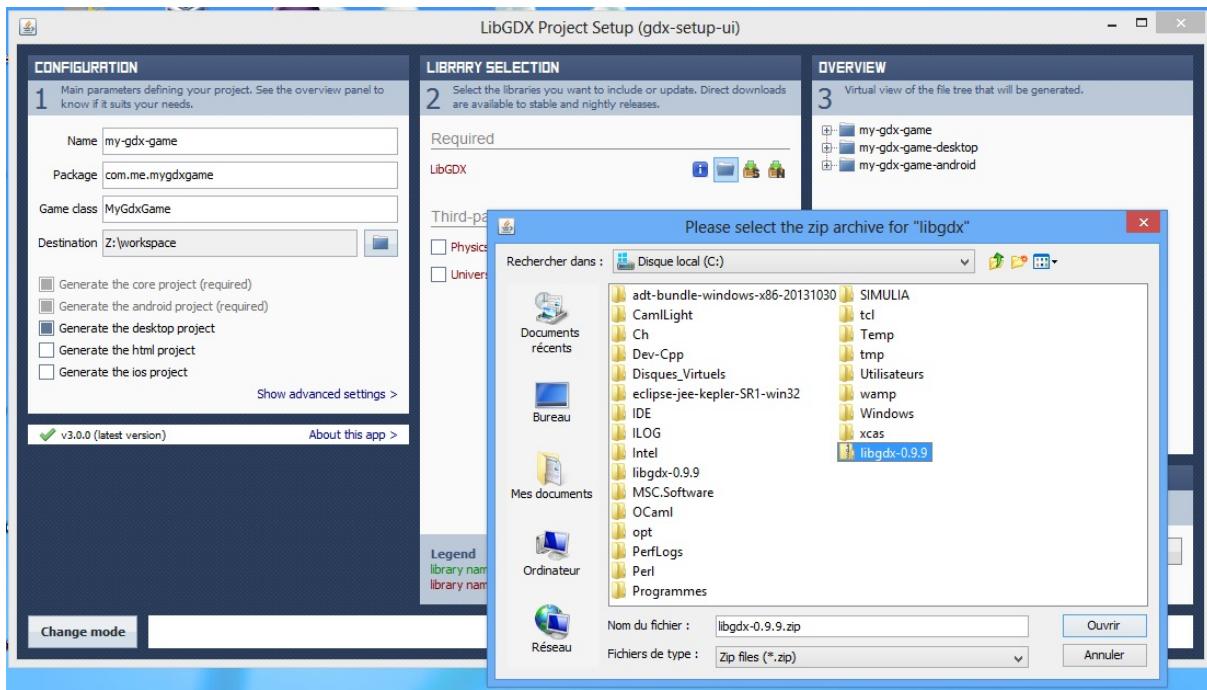
Create ... :



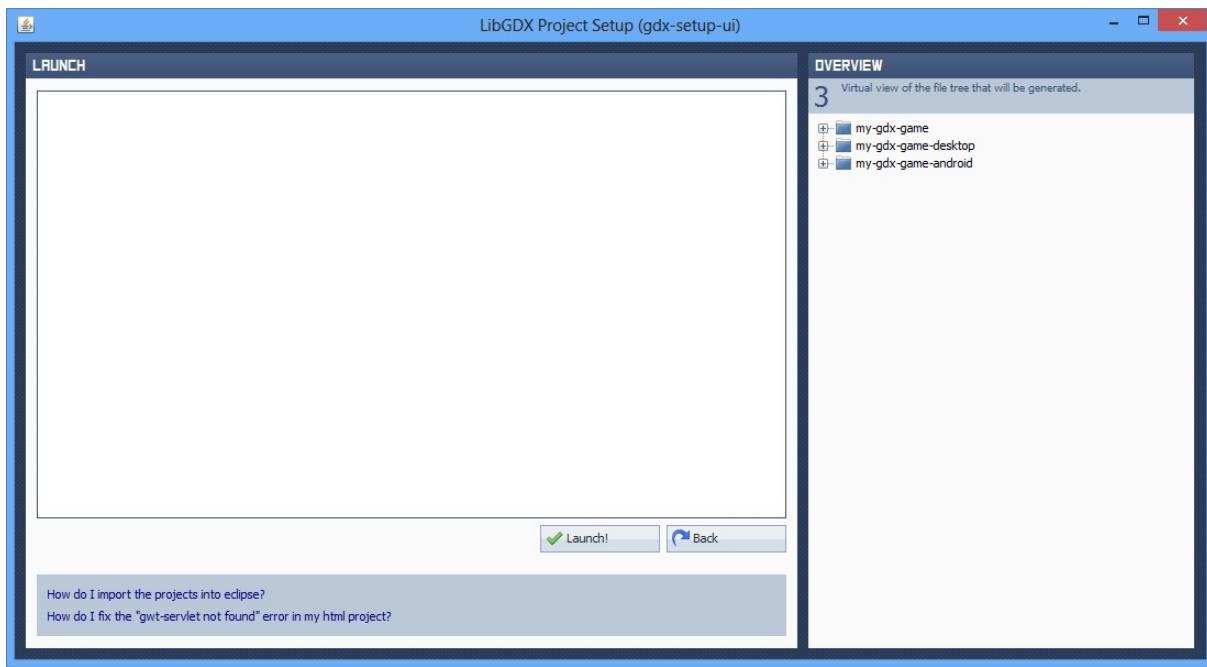
Modifiez la Configuration. On désire un projet Android et un projet Desktop :



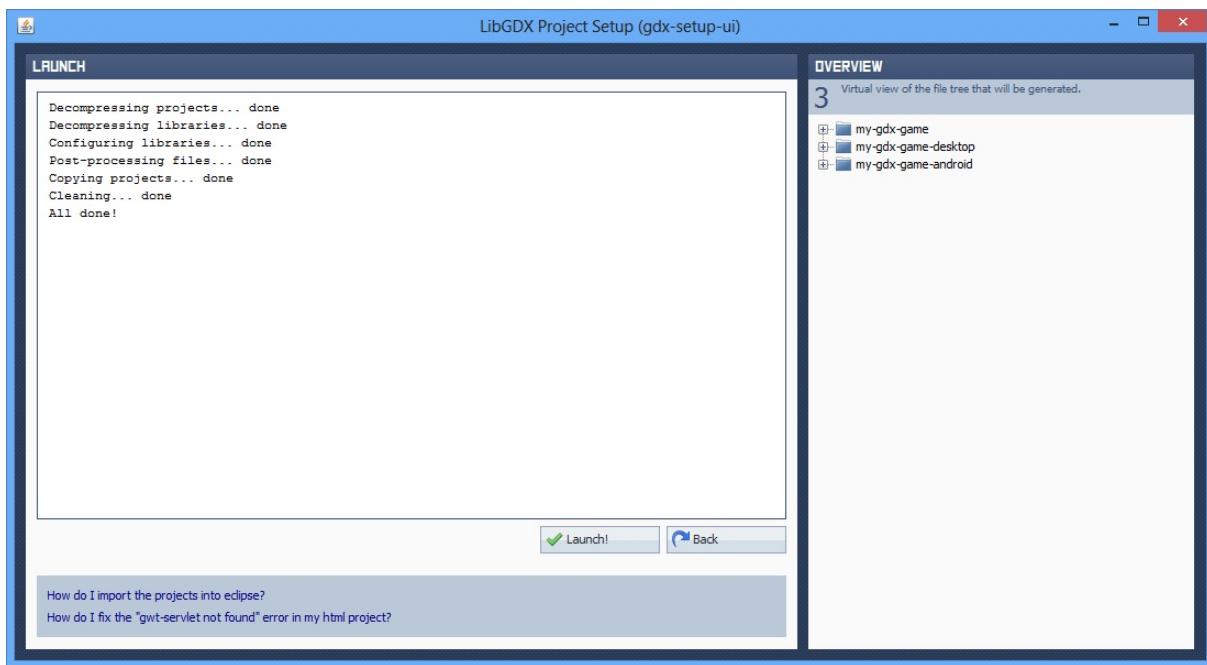
Il faut indiquer où se trouve la librairie libGDX. Il faut donc lui indiquer l'archive que l'on a mis sur « C : » :



On peut maintenant générer le projet (bouton « Open the generation screen ) :

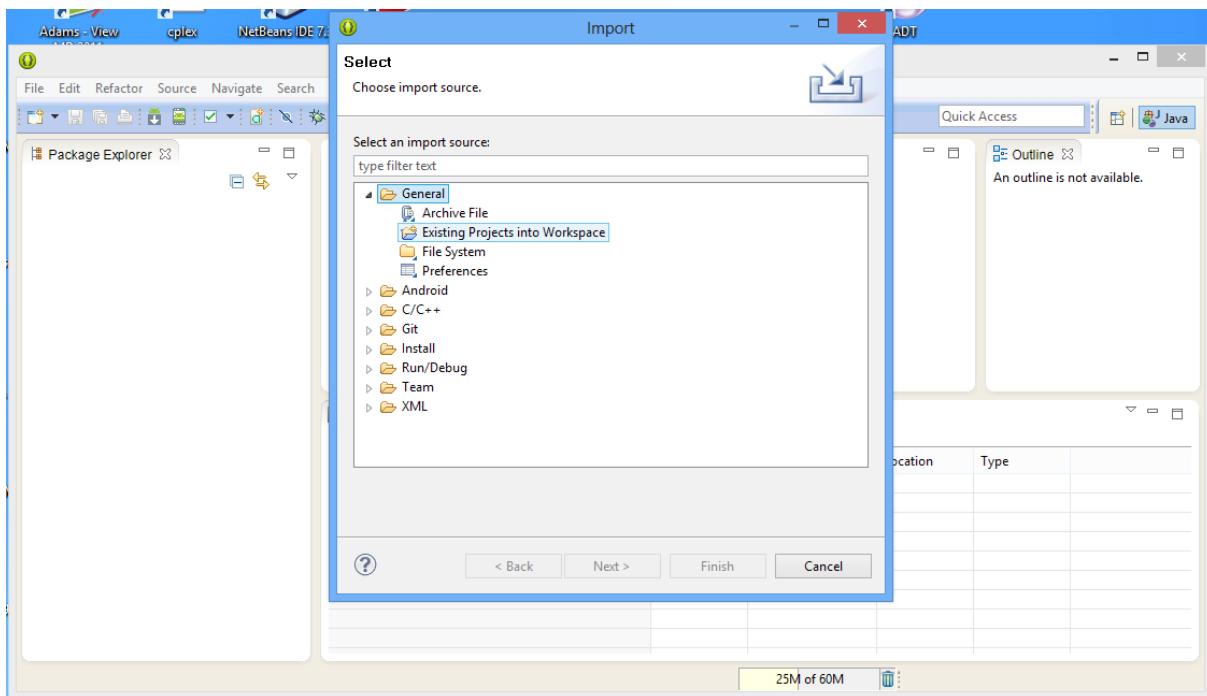


Résultat :

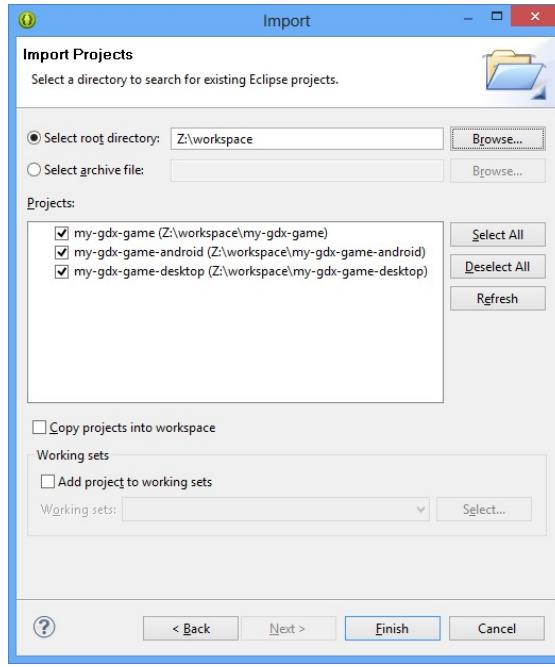


## Importer les projets dans l'ADT

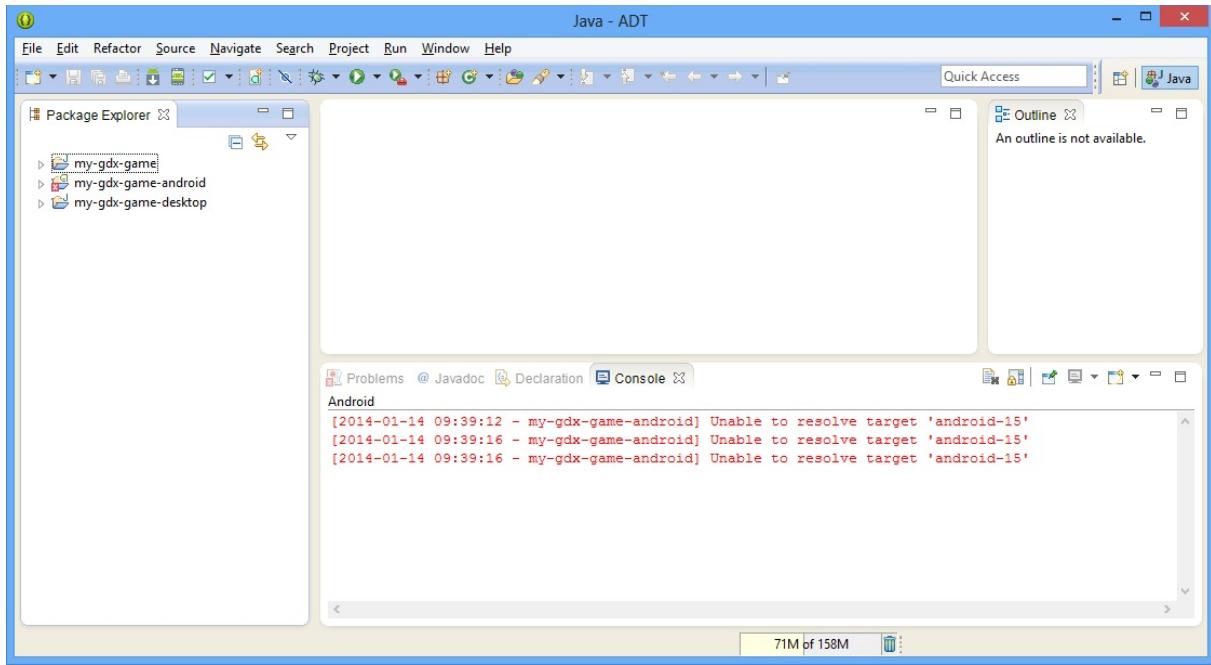
Importez les projets par le menu « File / Import ... / Existing Projects into Workspace » :



Sélectionnez les trois projets :



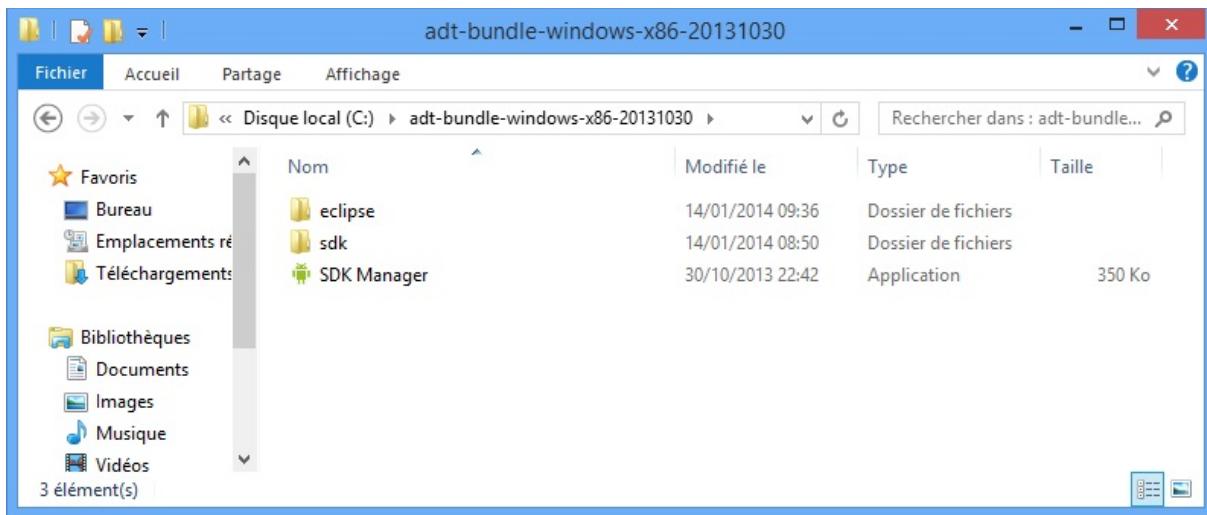
Résultat :



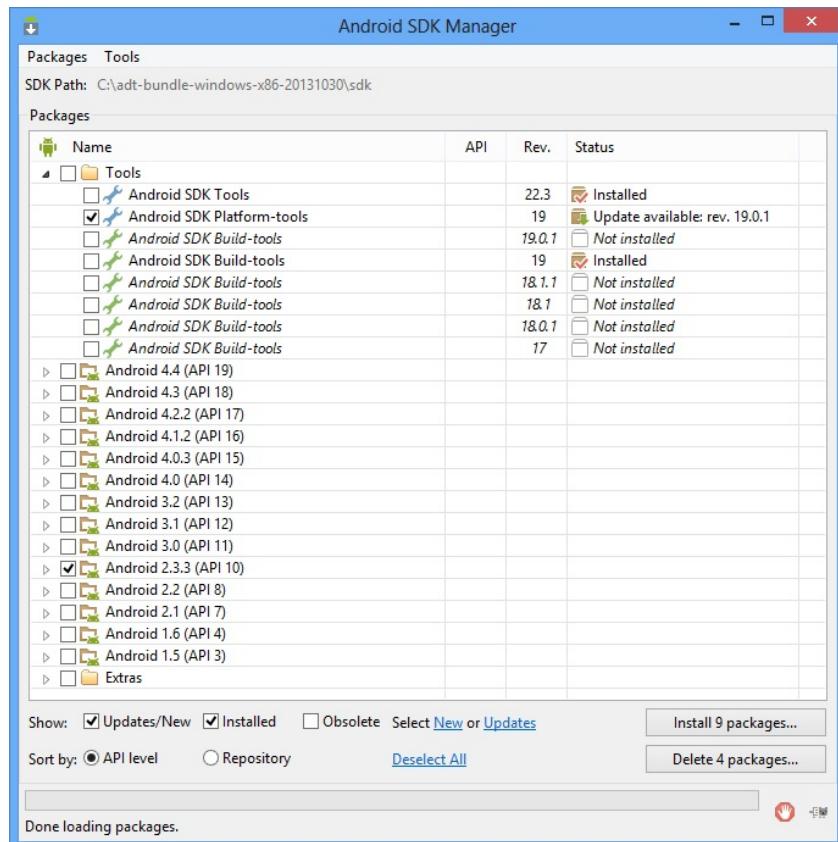
Une erreur est indiquée car l'installation de l'ADT n'a pas installé l'API utilisée par les projets générés pour libGDX. Avant de modifier l'API requise, on va voir comment ajouter d'autres API à notre ADT.

## SDK Manager

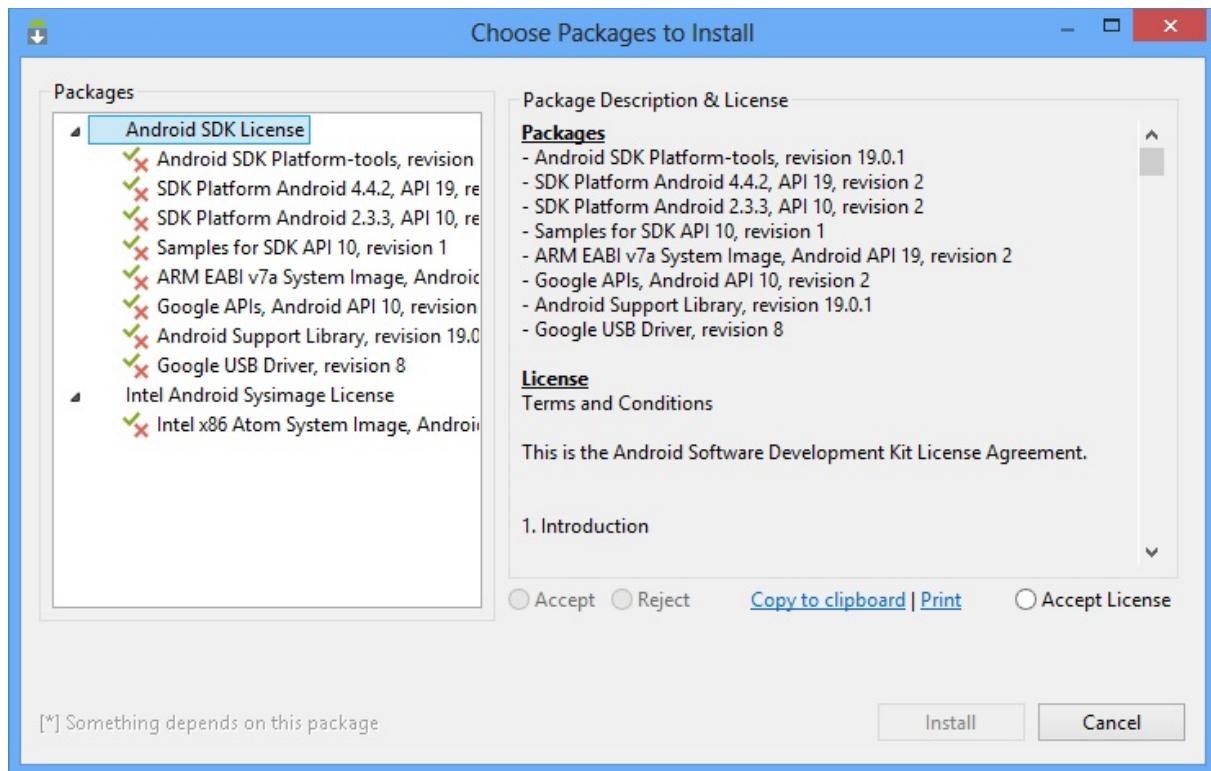
Exécutez SDK Manager. **Vous devez l'exécuter en Administrateur !**



Une fenêtre de choix apparaît. Vous constaterez que par défaut « Android 4.4 » est partiellement coché (voir à l'intérieur). Cochez également « Android 2.3.3 (API 10) » dans la fenêtre qui apparaît :



Installez x packages ... :



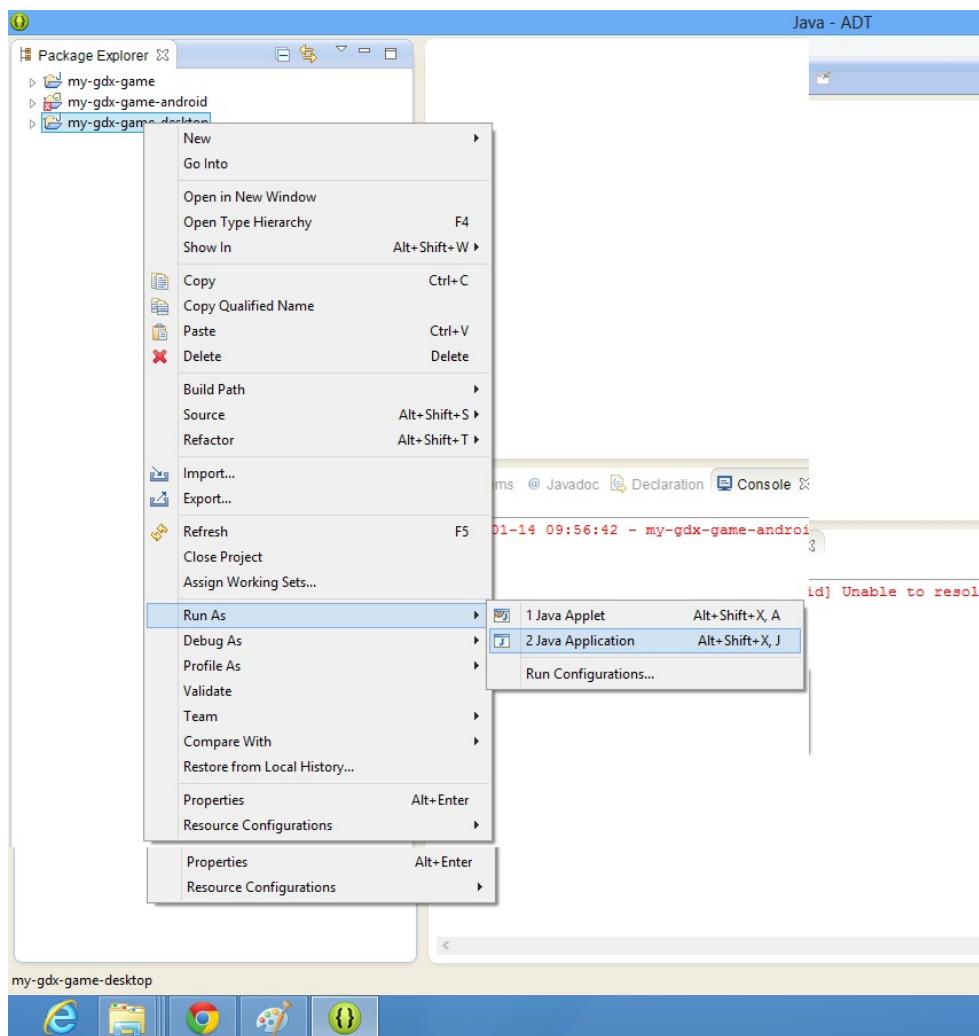
Vous devez accepter (Accept License) puis Install. L'installation se fait. A la fin vous pouvez fermer le SDK Manager.

## Tester le projet libGDX

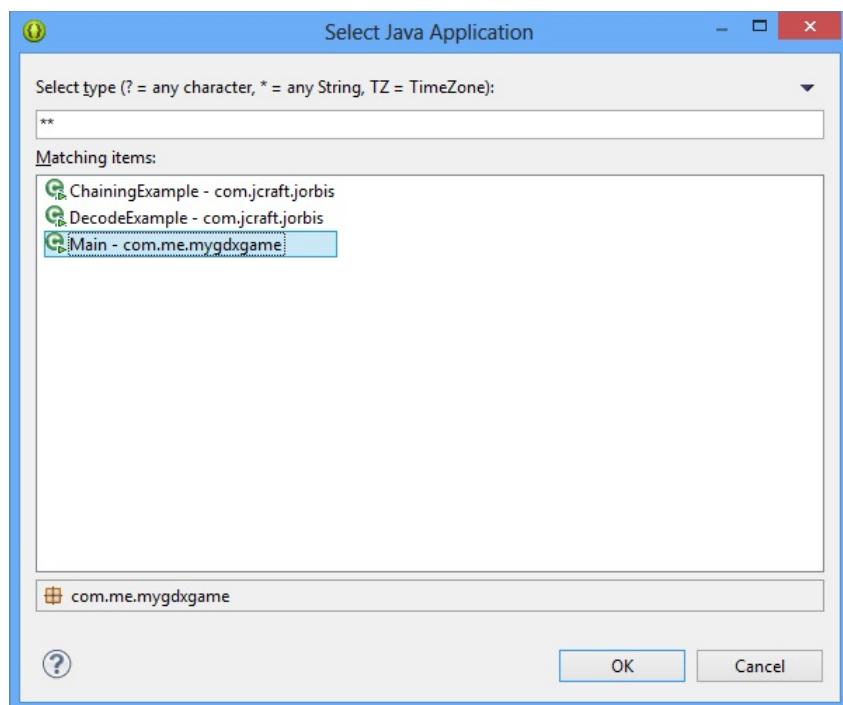
Pour prendre en compte les nouvelles API, il faut quitter ADT et le relancer ...

## Exécuter le projet pour le desktop

Choisir « Run As / Java Application » :



Choisir Main :

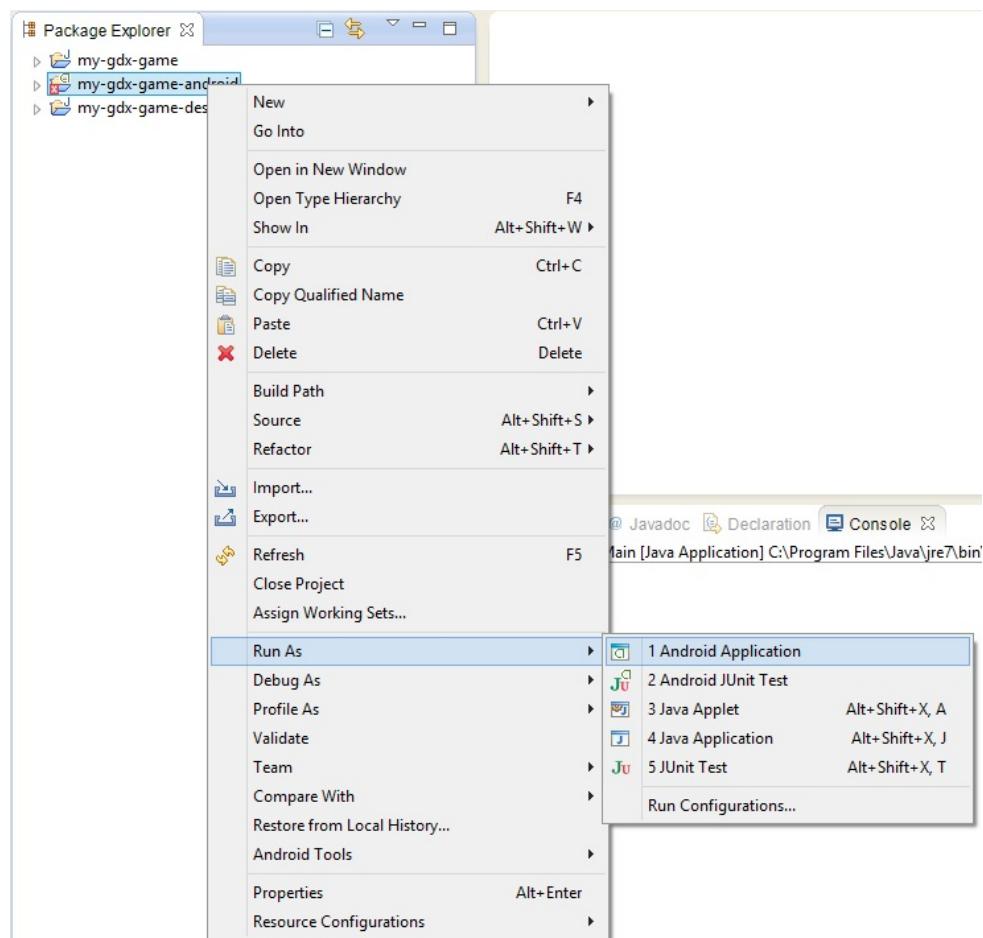


Résultat :



## Exécuter le projet pour Android

Choisir « Run As / Android Application » :



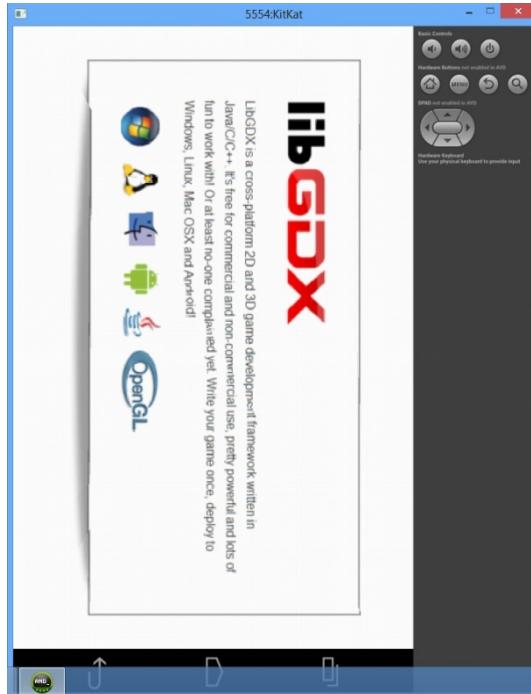
Si tout se passe bien vous obtenez ceci dans la console :

Problems @ Javadoc Declaration Console

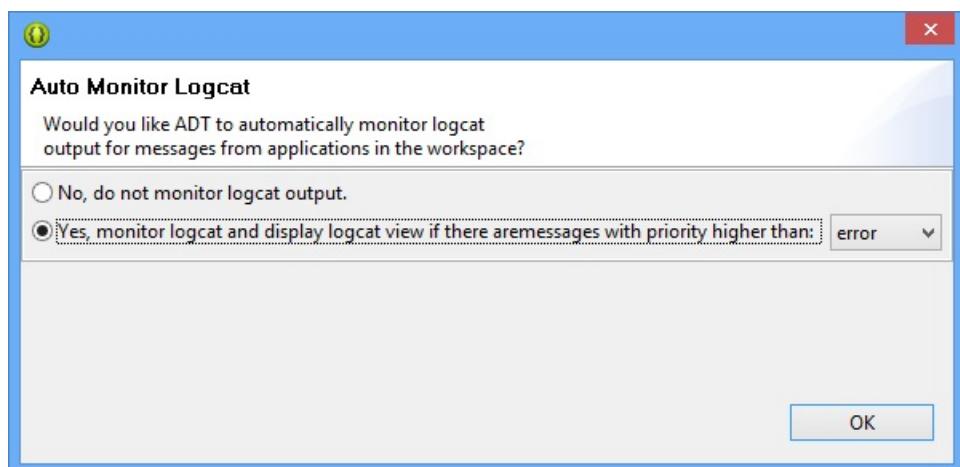
Android

```
[2014-01-14 10:13:52 - SDK Manager] hw.sensors.proximity=yes
[2014-01-14 10:13:52 - SDK Manager] hw.battery=yes
[2014-01-14 10:13:52 - SDK Manager] disk.dataPartition.size=200M
[2014-01-14 10:13:52 - SDK Manager] hw.audioInput=yes
[2014-01-14 10:13:52 - SDK Manager] hw.sensors.orientation=yes
[2014-01-14 10:13:52 - SDK Manager] hw.camera.front=none
[2014-01-14 10:13:52 - SDK Manager] hw.gps=yes
[2014-01-14 10:13:52 - SDK Manager] skin.dynamic=yes
[2014-01-14 10:13:52 - SDK Manager] hw.keyboard=yes
[2014-01-14 10:13:52 - SDK Manager] vm.heapSize=64
[2014-01-14 10:13:52 - SDK Manager] hw.ramSize=768
[2014-01-14 10:14:16 - my-gdx-game-android] -----
[2014-01-14 10:14:16 - my-gdx-game-android] Android Launch!
[2014-01-14 10:14:16 - my-gdx-game-android] adb is running normally.
[2014-01-14 10:14:16 - my-gdx-game-android] Performing com.me.mygdxgame.MainActivity activity launch
[2014-01-14 10:14:16 - my-gdx-game-android] Automatic Target Mode: launching new emulator with compatible AVD 'KitKat'
[2014-01-14 10:14:16 - my-gdx-game-android] Launching a new emulator with Virtual Device 'KitKat'
```

Puis sur l'émulateur :



N'oubliez pas d'autoriser la fenêtre de log « LogCat » :

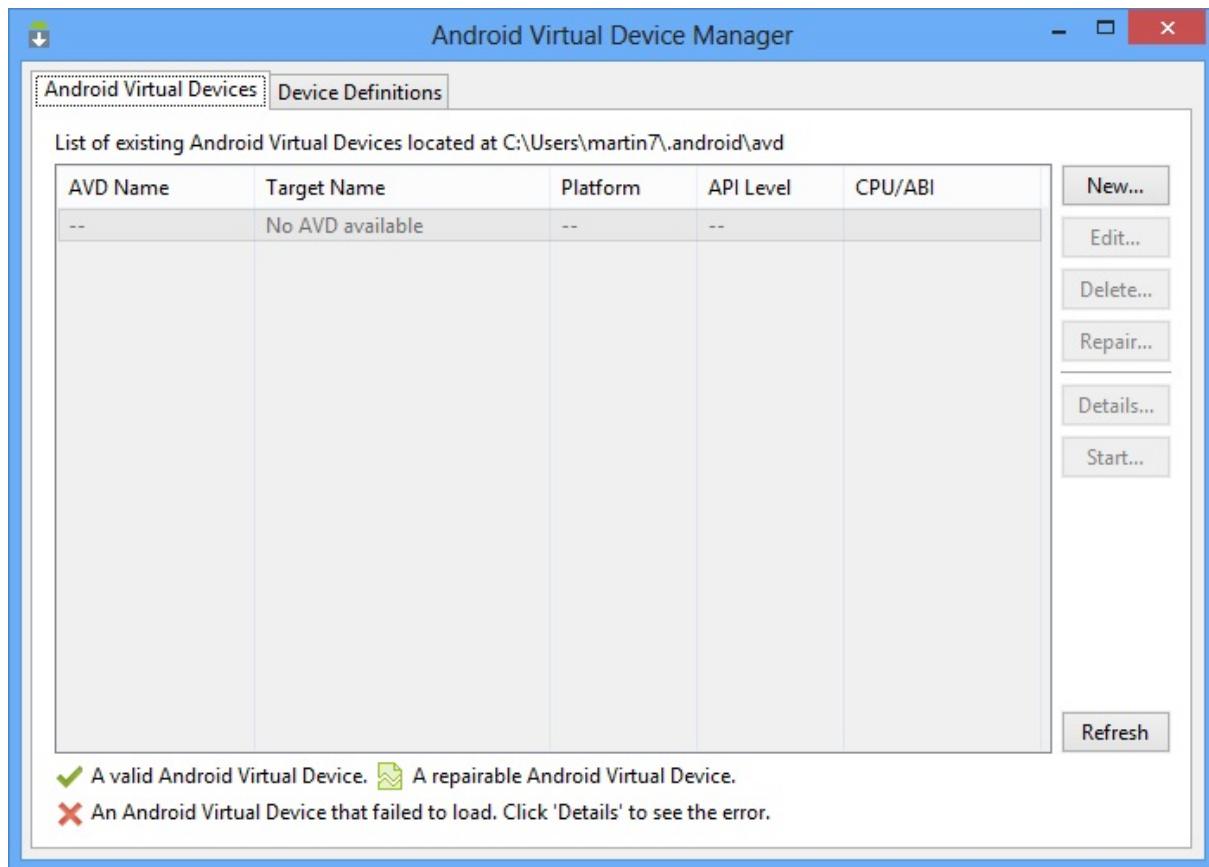


Mais il est probable que vous ayez reçu une erreur ... car il faut un émulateur ou un dispositif physique capable d'exécuter le code généré. Dans ce cas faites « Yes » et passez au paragraphe suivant avant de refaire cette procédure pour exécuter le projet.

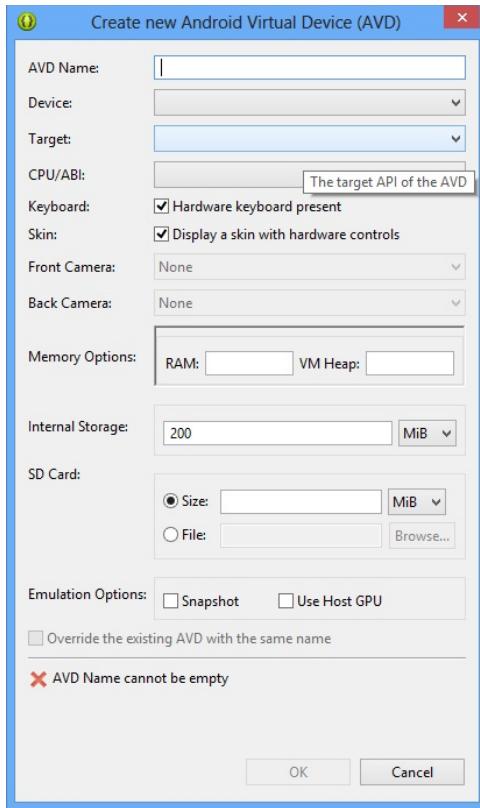


## Définir un nouvel « Android Virtual Device »

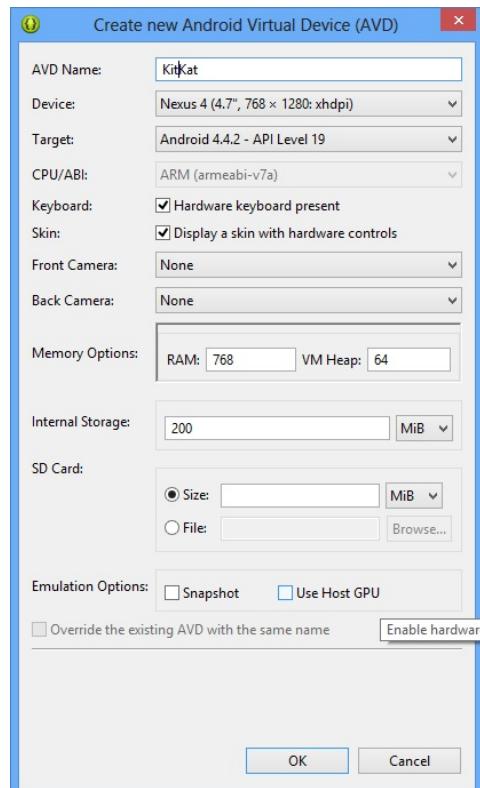
Résultat :



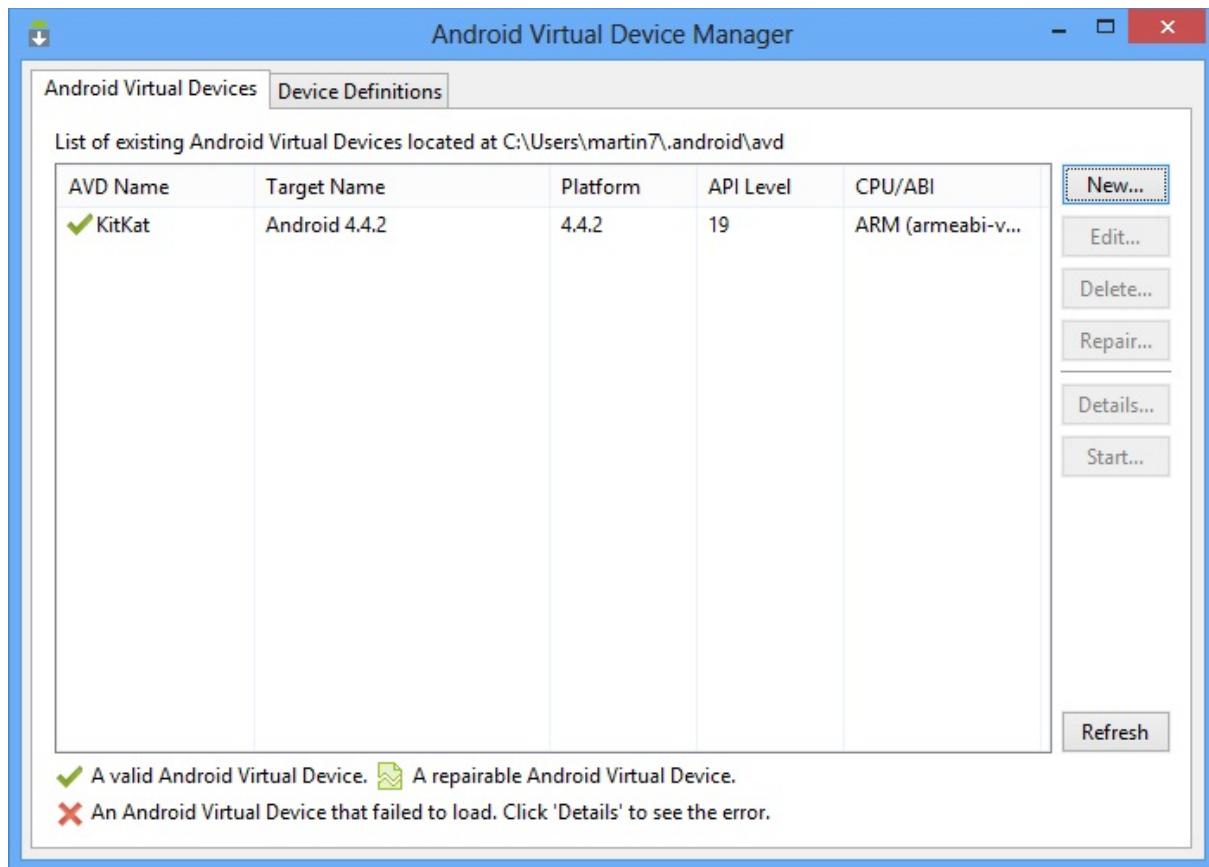
New ... :



Remplir les champs. Je fais un émulateur pour 4.4.2 que j'appelle KitKat :



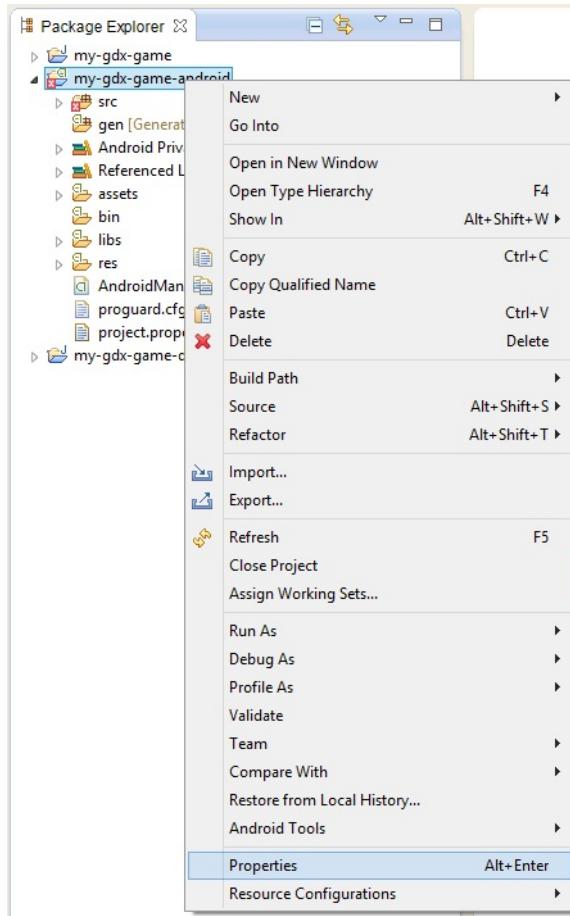
OK ... l'émulateur s'ajoute à la liste :



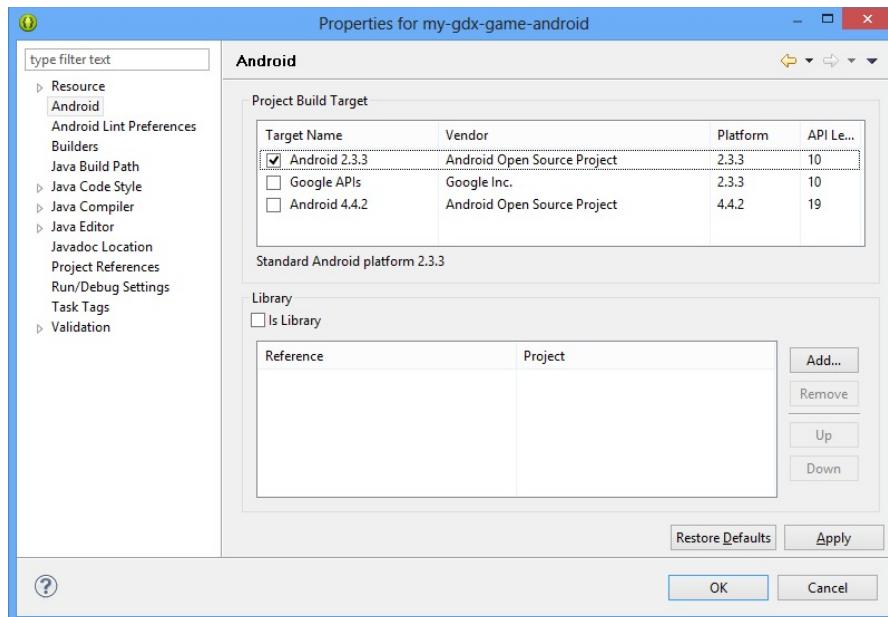
## Changer la cible du projet

Nous avons utilisé la version 4.4.2 d'Android (API 19) qui est la dernière née. Mais il est également possible de changer la cible pour utiliser une autre API. Un programme fait pour l'API 19 ne fonctionnera pas sur l'API 10. Mais l'inverse est vrai.

Entrez dans les propriétés du projet Android :



Dans la rubrique Android, vous pouvez choisir la cible désirée. ADT liste uniquement les API installées avec le SDK Manager. Choisissez « Android 2.3.3 » pour faire un essai :



Faites « OK » et préparez un émulateur avec cette API (suivre la même procédure que pour l'émulateur KitKat). Appelez-le GingerBread ...

Lancez l'application Android ... vous devez obtenir une erreur. En effet la cible est supérieure à la version de l'API demandée dans le fichier de configuration. Pour cela, ouvrez le fichier « *AndroidManifest.xml* » et allez dans l'onglet « *AndroidManifest.xml* » :



Attribuez la valeur 10 à `android:minSdkVersion` et `android:targetSdkVersion`. Puis retirez « `| screenSize` » à `android:configChanges`.

Sauvegardez et réexécutez. Maintenant l'application doit s'exécuter correctement avec l'émulateur GingerBread. Cette application fonctionne à présent sur tous les dispositifs disposant au moins de la version 2.3.3.