

TP Todo list

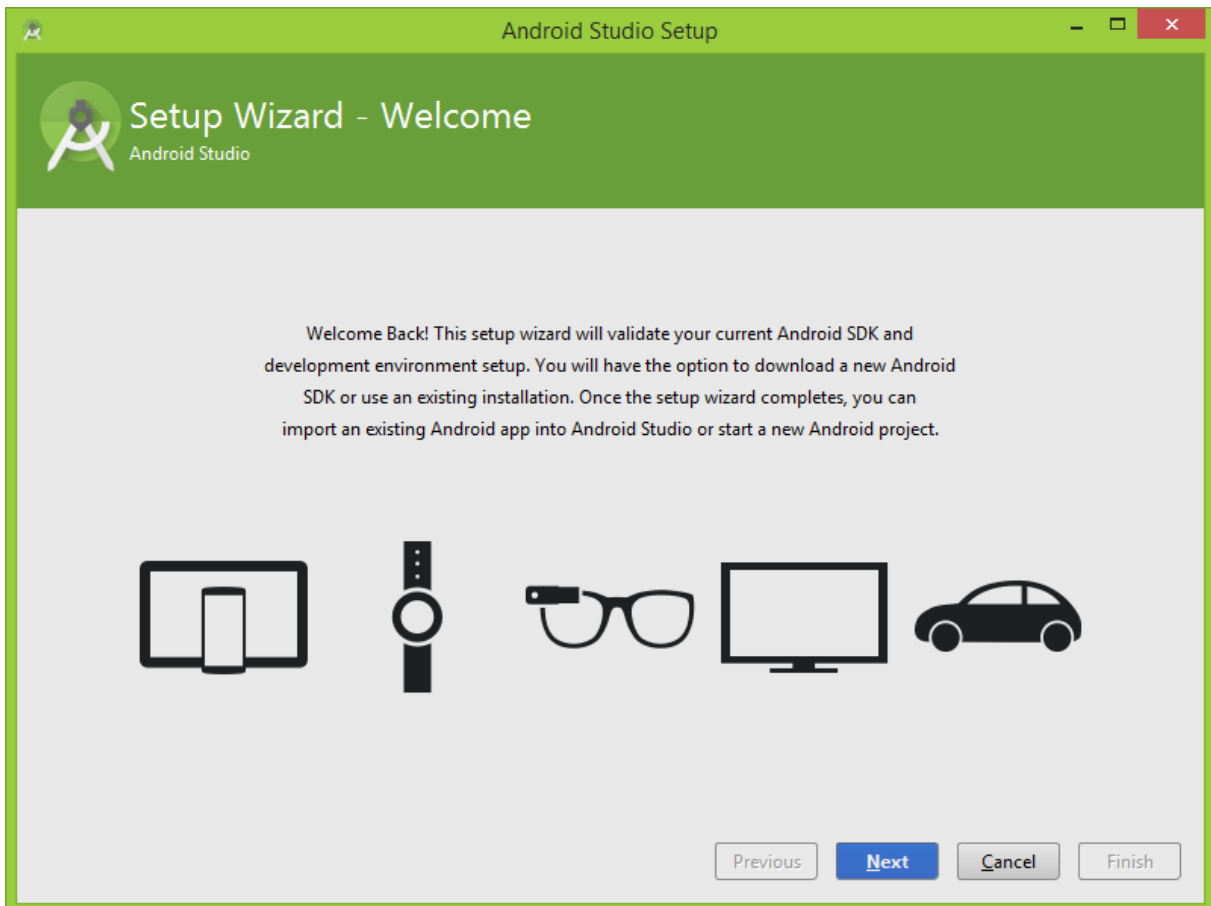
But du TP : Création d'une TodoList (petit programme pour lister des tâches à faire)

Tous vos projets seront à héberger sur un compte GitHub qui vous servira éventuellement de portfolio par la suite. Pour savoir comment partager un projet sur GitHub depuis android studio, veuillez suivre ce [lien](#).

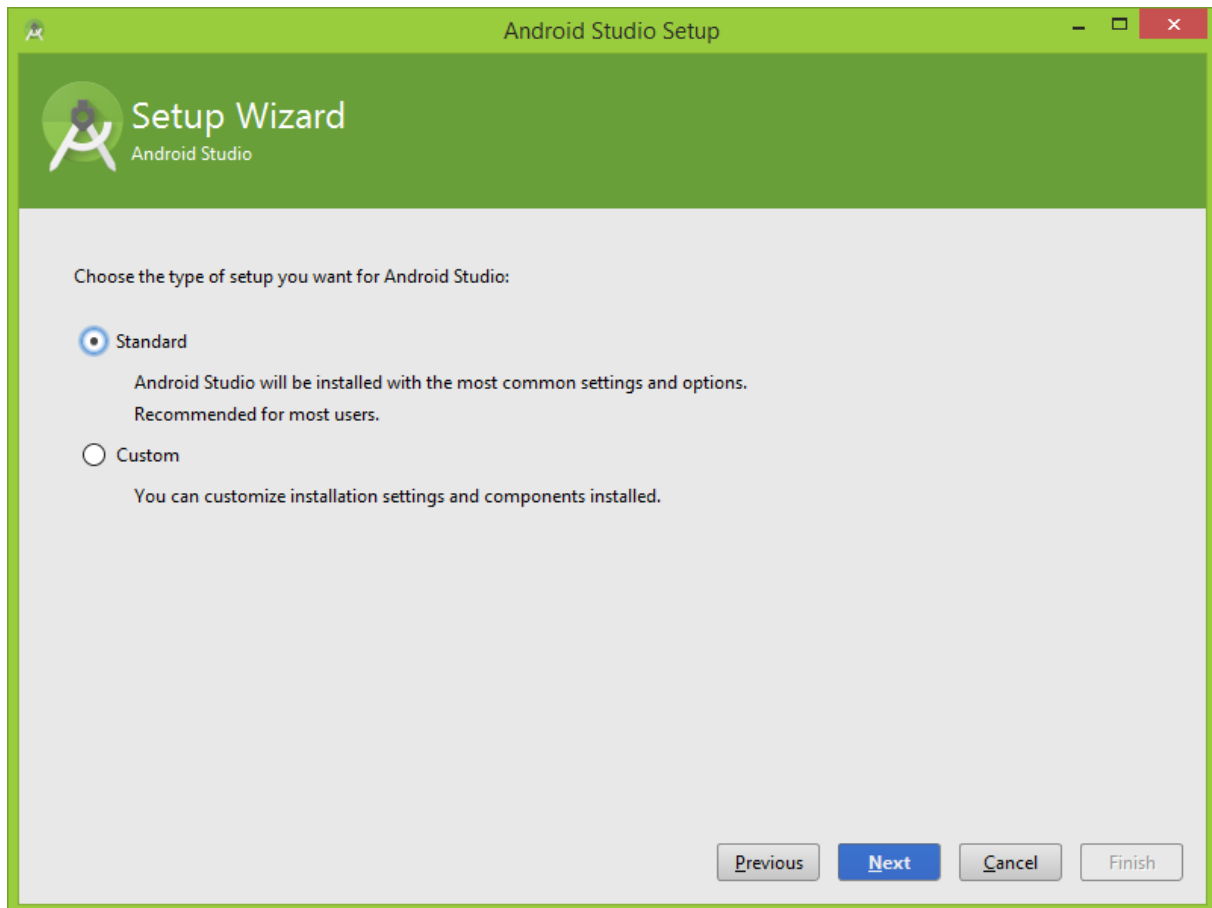
I. Création de l'application

A. Démarrage d'android studio

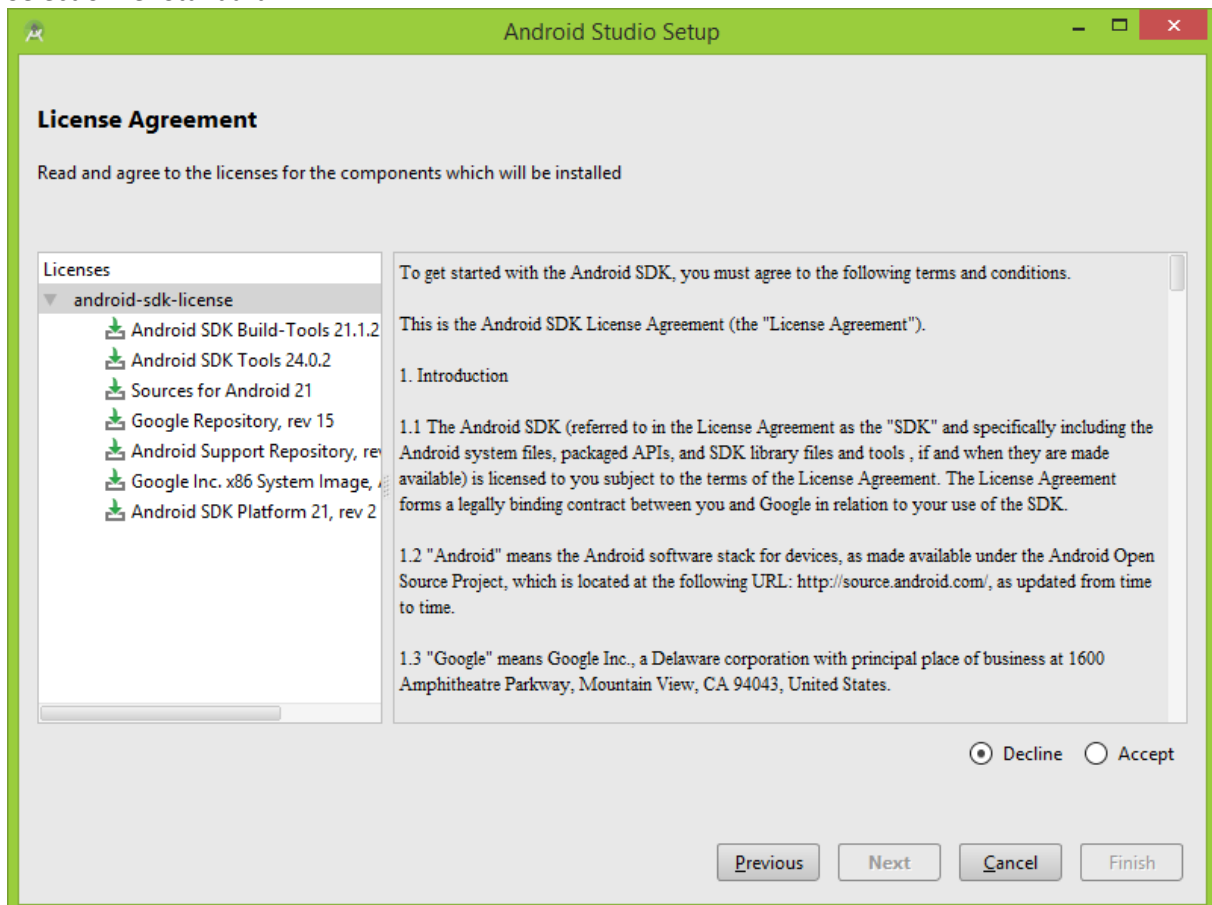
Démarrez l'IDE Android Studio.



Cliquer sur Next.

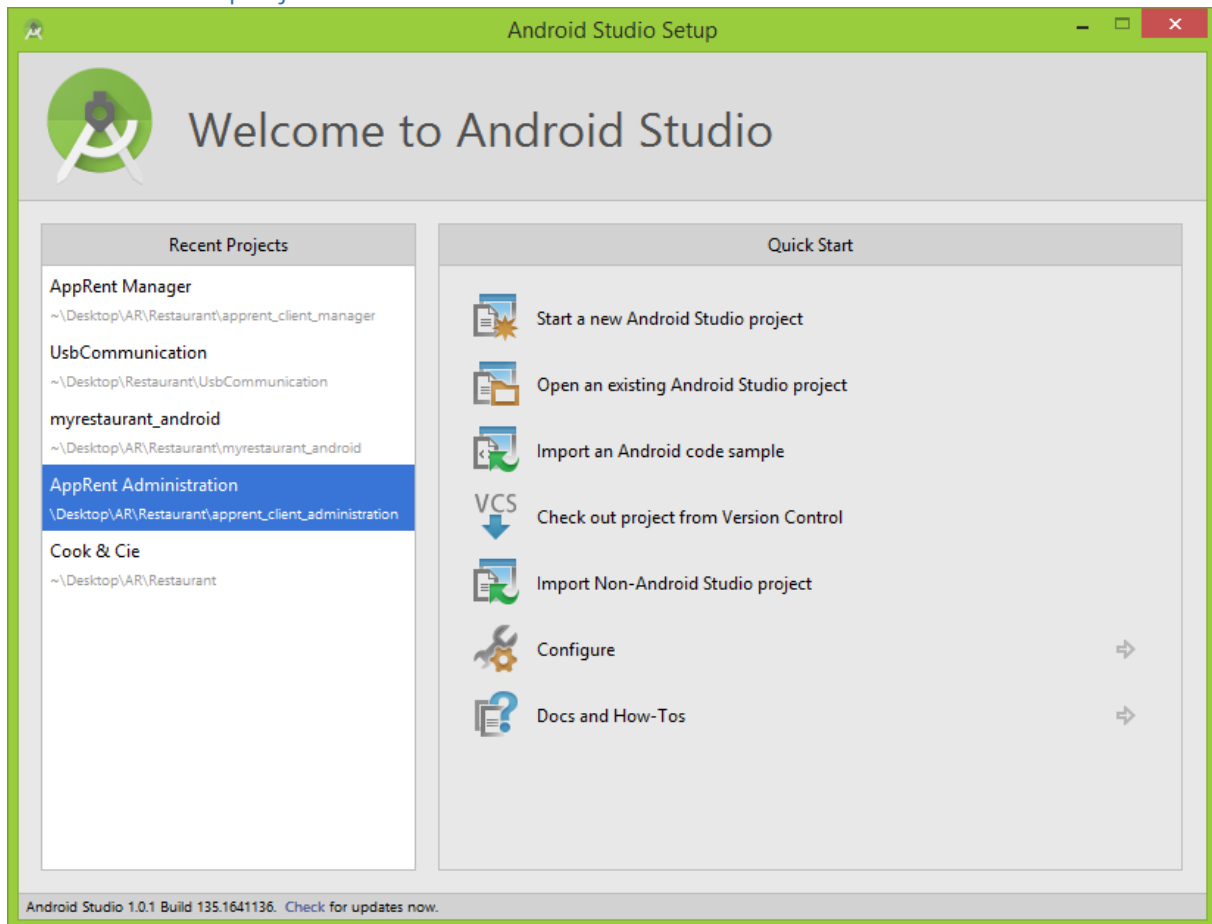


Sélectionner Standard



Cocher « Accept » puis Finish.
Le SDK s'installe alors.

B. Création du projet



Sélectionner « Start a new Android Studio project » afin de créer une application.
Remplir les différentes infos comme notées ci-dessus :

- Application name : MyToDoList
- Company domain : nom.prenom
- Package name : nom.prenom.mytodolist

Cliquer sur Next.

Sélectionner « Phone and Tablet » puis comme minimum SDK « API 13 » puis Next.

Sélectionner « Add no activity » puis Finish.

II. Création des classes et ressources

A. Création de l'activité principale

Créer une nouvelle classe TodoActivity depuis le panel à gauche qui représente l'arborescence du projet en faisant un clic droit sur le dossier app/java/nom.prenom.mytodolist/ (Attention ne pas le faire dans le dossier comportant « (androidTest) »).

Cette classe doit hériter d'Activity.

Modifier l'AndroidManifest.xml en rajoutant l'activity dedans (page 2 de l'android Cheat Sheet).

B. Création du layout correspondant

Créer un dossier « layout » dans le dossier « MyToDoListProject/MyToDoList/src/main/res ». (Clic droit, new, Android Resource Directory).

Créer un layout task_layout dans ce dossier comportant une TextView, un EditText, un bouton « Vider la liste » et un bouton « Valider ». (Clic droit, new, Layout resource file avec comme root element un « Linear Layout », puis glisser et déposer les éléments dans le layout). Définir des id pour les divers composants.

C. Liaison du layout et de l'activité

Redéfinir le onCreate dans la MainActivity, puis lier le layout à l'activité. (Page 5 de la Cheat Sheet)
Récupérer les divers composants du layout dans l'activité (findViewById()).

D. Création du listener

Création des listeners sur les boutons pour gérer leurs clics.

Au clic sur le bouton valider, le texte de l'editText doit être rajouté à la TextView. Au clic sur le bouton vider la liste, la TextView doit être vidée.

III. Compiler et tester son application avec l'émulateur

Si vous avez un appareil android passez au III bis.

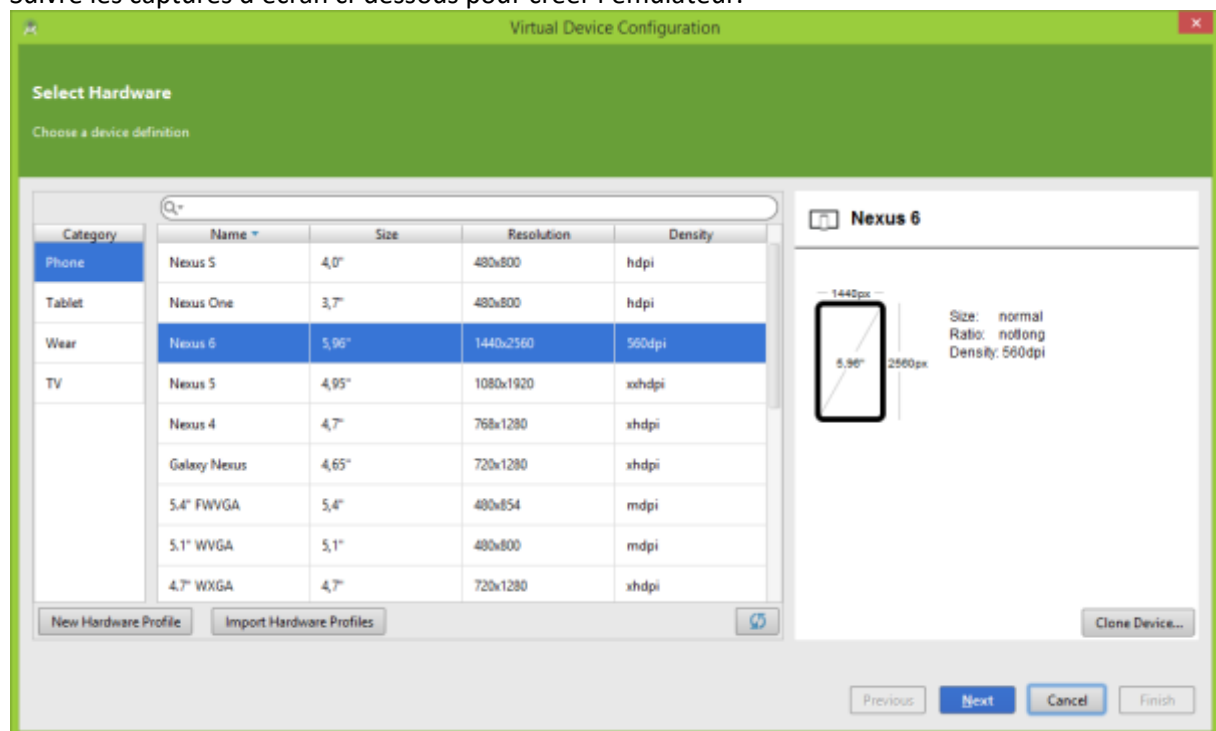


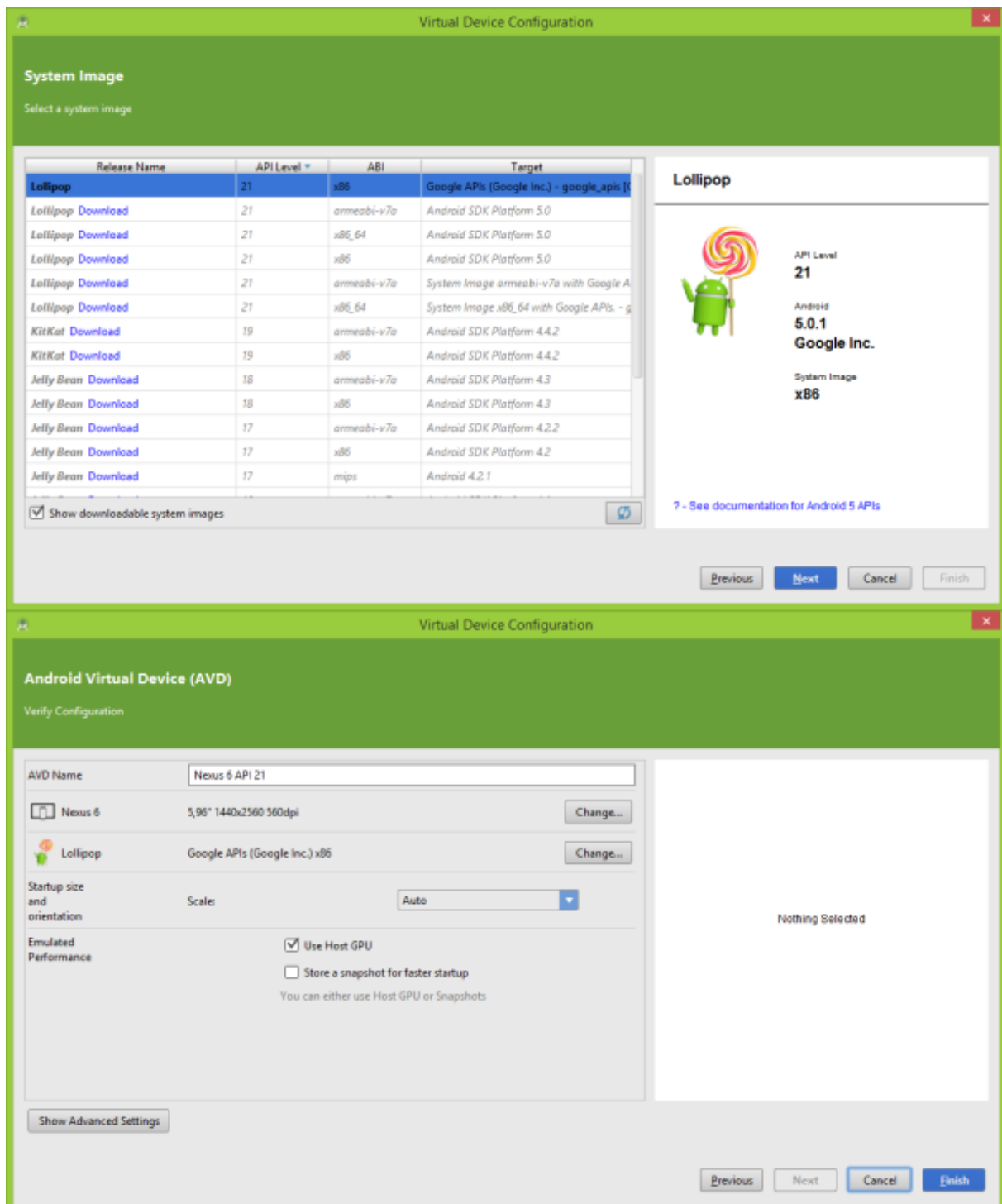
Cliquer sur le bouton « Run »

Appuyer sur « ... »

Appuyer sur « Create Virtual Device... »

Suivre les captures d'écran ci-dessous pour créer l'émulateur.





Cliquer sur l'émulateur dans la liste puis cliquer sur « Run » sur l'émulateur nouvellement créée.

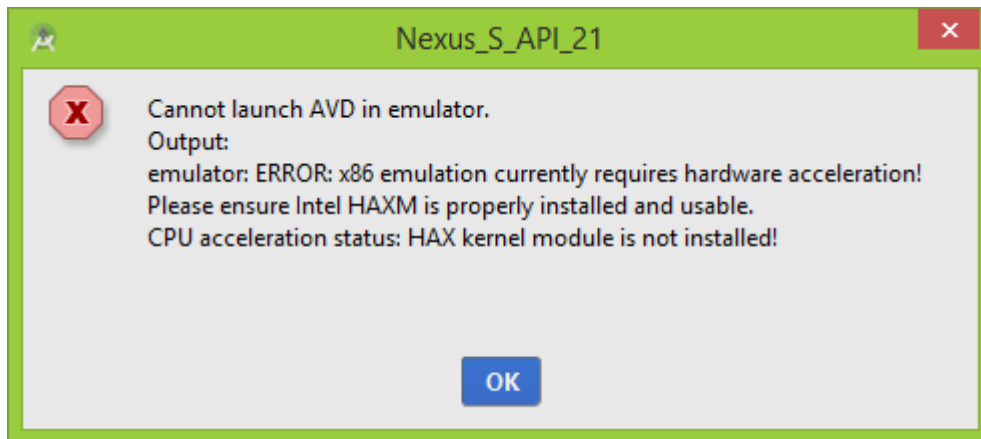
Dans Android virtual device doit maintenant être affiché « Nexus 6 API 21 »

Si l'erreur ci-dessous apparaît, veuillez installer l'intel x86 Emulator Accelerator, celui-ci est situé dans le dossier

{SDK_FOLDER}\extras\intel\Hardware_Accelerated_Execution_Manager\intelhaxm.exe.

SDK_FOLDER correspondant au répertoire d'installation du SDK android. L'adresse de ce répertoire est trouvable dans android studio depuis File->Project Structure-> Android SDK Location.

Si le répertoire intel n'existe pas, depuis le SDK manager (Android Studio,Tools,Android,SDK Manager) installez le package qui se nomme « Intel x86 Emulator Accelerator (HAXM installer) ». Le répertoire intel doit alors exister.



Cocher « Use same device for future launches » puis cliquer sur OK.

III. Compiler et tester son application avec un appareil android

A. Activation du mode déboguage sur l'appareil

Allez dans les paramètres de l'android et dans « Option pour les développeurs » et cochez « Mode déboguage ».

Il est possible que le menu « Option pour les développeurs » n'apparaisse pas directement. Sur certains appareils, la manipulation pour faire apparaître ce menu est d'aller dans le menu « A propos du téléphone » et d'appuyer 7 fois sur « Numéro du build ».

Si cette manipulation ne fonctionne pas, une simple recherche avec « enable debugging *nom du modèle* » devrait pouvoir vous donner la réponse.

B. Ajout du driver

Si le portable n'est pas directement reconnu, il vous est peut-être nécessaire d'installer le driver de celui-ci, pour cela 2 possibilités existent :

-Soit le driver est compris dans le package « Usb driver » du SDK manager d'android (Android studio->Tools->Android->SDK Manager, installer dans Extras « Google usb driver »)

-Soit le driver est disponible sur le site du constructeur.

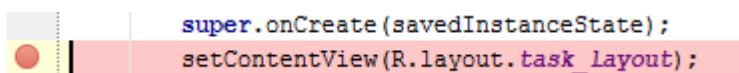
C. Lancement de l'application




Cliquer ensuite sur Run dans android studio et choisissez votre appareil dans la liste des appareils disponibles.

IV. Débugger l'application

Par moment pour débbugger l'application, il vous sera utile de mettre des points d'arrêts dans celle-ci (clic gauche dans la petite colonne à gauche d'une ligne)



Puis de démarrer l'application en mode debug via l'icône .

Le mode debug permet alors de voir le contenu des différentes variables courantes.

V. Stockage des valeurs

Afin de stocker les valeurs dans une ArrayList, vous utiliserez un singleton qui sera disponible dans l'app entière.

Pour plus d'infos : [Voir ici...](#)

VI. Affichage dans une listview

A partir de la cheatsheet fournie, et du cours (avant dernière slide), afficher les éléments dans une listview (un setAdapter devra aussi être fait lors du click sur les boutons valider/vider la liste.

VII. Pour les plus rapides

Ajouter une Activity lancée via un Intent pour rajouter des tâches plutôt que de le faire dans la TodoActivity.(Expliqué dans la cheat sheet)

Ajouter une checkbox « Important » pendant l'ajout d'un bouton. Stocker aussi cette valeur dans le singleton et faire que les éléments importants de la ListView soient affichés en rouge.