

TP 1

Introduction au SDK d'Android

Romain Raveaux

1 Introduction

Android est un système d'exploitation pour smartphone et tablette développé par Google. Celui-ci met à disposition un kit de développement (SDK) basé sur le langage Java. Ce TP va vous présenter un exemple d'utilisation au travers du développement d'une application de type "HelloWorld".

Pour ce TP, vous utiliserez la machine virtuelle intitulée tpdevmobile.

Démarrer la machine virtuelle TP Dev mobile.

Tout ce dont vous avez besoin pour ce TP se trouve sur </home/polytech/dev>

Vous trouverez **Eclipse** prêt à fonctionner (</home/polytech/eclipse>).

Vous trouverez Android Studio prêt à fonctionner (</home/polytech/android-studio>).

Créer un répertoire « **code** » pour y placer vos développements.

Les outils de développements sont déjà installés. Il est déconseillé de faire des mises à jour !!!! .

Les objectifs :

- Appréhender les outils du SDK de google.

- Développer une première application avec Eclipse+ADT

- Développer une première application avec Android Studio

2 Android SDK Manager

Android SDK manager est fourni par google est contient plusieurs éléments :

- Un compilateur JAVA pour Android (l'outil dx).
- Un logiciel de virtualisation pour les émulateurs Android
- Des exemples de projets avec des codes sources
- Un outil de communication entre l'ordinateur et l'émulateur
- Les images (iso) des différentes versions d'Android
- Un gestion de versions du SDK

Démarrer un terminal de commande et exécuter le gestionnaire de versions du SDK.

Le répertoire :~/dev/android-sdk-linux/tools et l'exécutable ./android &

Question 1 : Identifier les éléments installés sur votre machine ? Par contre ne rien mettre à jour cela prend du temps.

4 Prise en main de l'émulateur (AVD)

AVD est le nom de l'émulateur Android. Il existe un outil pour gérer les AVD. Cet outil s'appelle l'AVD Manager. Pour y accéder démarrer le SDK manager et aller dans le menu « Manage AVD ».

Question 2 : Créer un AVD et identifier les points configurables de l'AVD. A la fin, détruisez votre AVD

Noter qu'il y a des AVD génériques : Nexus, Android Wear, ..

Noter qu'il y a déjà une AVD de créer. Ce dernier s'appelle « testphone ».

Question 3 : Décrire la configuration de l'AVD « testphone » (version d'Android, matériel, RAM, ...)

**Question 4 : Lancer l'AVD « testphone » via le terminal de commande et l'outil « emulator »
~/dev/android-sdk-linux/tools\$./emulator**

Ne pas éteindre l'émulateur. Garder l'émulateur toujours opérationnel.

5. Les outils du SDK

- Allez regarder dans le répertoire <AndroidSDK>/platforms-tools (<AndroidSDK> = /home/polytech/dev/android-sdk-linux)
 - Lancer dans un terminal le fichier **adb**
 - A quoi sert **adb** ?
 - Arrêter et démarrer le server ADB. **Cela peut être pratique lorsque parfois Eclipse ne « voit » plus votre émulateur comment étant actif.**
 - Lister les périphériques connectés
 - Lancer la commande adb shell. Essayer quelques commandes Linux (ls, pwd, who, id, ...) . Essayer certains appels systèmes propre à Android :
 - o service call phone 2 s16 "+332....."
 - o service list
 - Regarder les autres commandes possibles avec ADB (Par exemple, l'installation d'APK, la désinstallation)
- Regarder le répertoire : /home/polytech/.android/avd
 - o Que contient ce répertoire ?

6. Première application avec Android : Helloworld

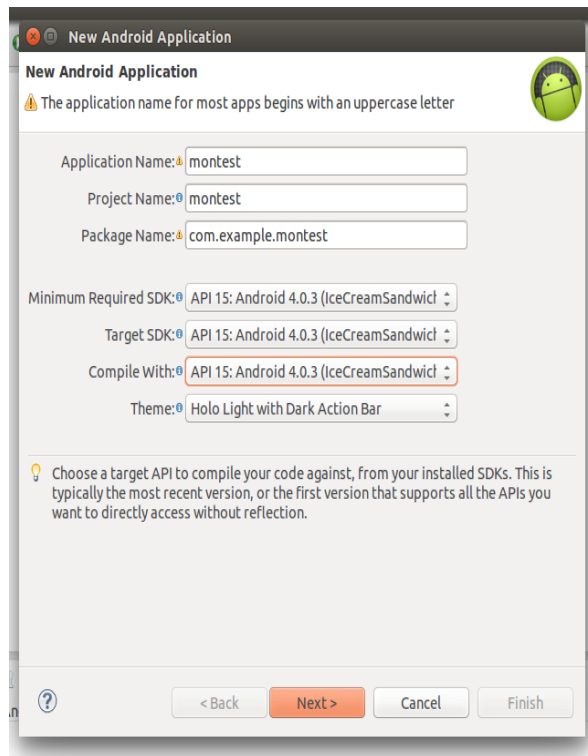
6.1 Développement avec le plugin Eclipse

Voici les étapes pour développer une première application avec Eclipse et le plugin Android Development Tools. Il faut déjà lancer eclipse. Pour se faire taper dans le terminal la commande suivante :

```
polytech@polytech-VirtualBox:~/dev/eclipse$ ./eclipse &
```

Question 5 : Commencez par sélectionner le menu File > New > Project... puis Android > Android project.

Dans un premier temps, laissez les paramètres par défaut, ne remplissez que ce qui est nécessaire (nom de l'application) .



Indiquez les propriétés du projet puis appuyez sur le bouton Next :

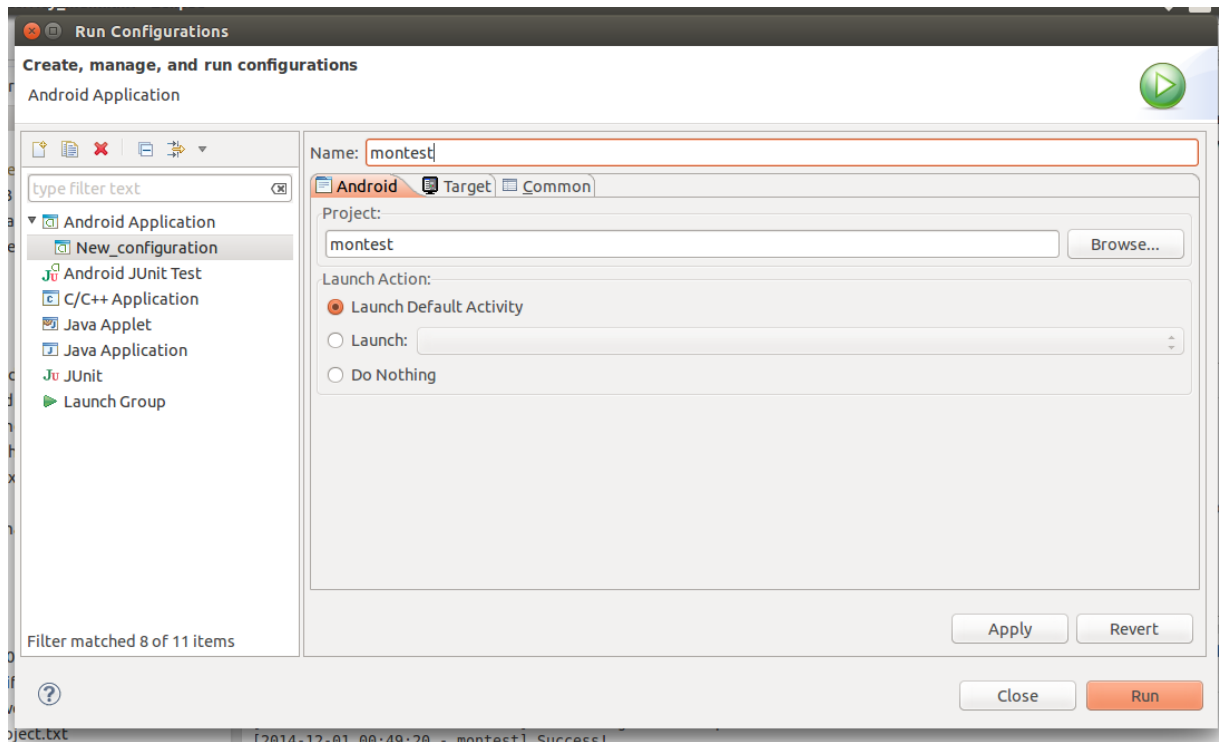
Compile with : Le code sera compilé avec les API de la version Android 4.0 dans votre cas.

Min SDK Version : C'est la version minimum d'Android nécessaire pour faire tourner votre application. A vous de vous assurer que vous n'utilisez pas de fonctions non-supportées par la version minimum d'Android que vous avez définie.

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.

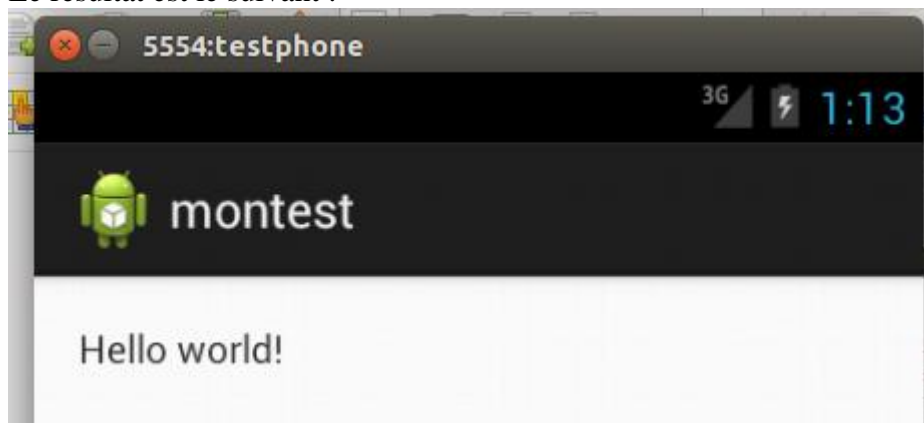
Un nouveau lanceur nommé New_configuration doit apparaître, sélectionnez le et renseignez le panel de droite :



Regarder l'onglet « Target » et choisissez votre AVD.

Il ne vous reste plus qu'à lancer l'application en appuyant sur le bouton Run.

Le résultat est le suivant :



6.2 Petite modification de l'application

Dans le package explorer sélectionner le fichier `com.example.montest.MainActivity.java`

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

```

Commençons notre HelloWorld en modifiant ce fichier afin d'afficher du texte à l'écran (nous détaillerons le code ensuite) :

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    TextView text = new TextView(this);
    text.setText("Salut");
    setContentView(text);
}

```

Dans Android, tous les composants graphiques (bouton, animation, champ texte, etc) sont basés sur la classe View.

Ainsi un champ texte est représenté par la classe TextView.

```
TextView tv = new TextView(this);
```

L'argument du constructeur de la classe TextView est une instance de la classe Context qui fournit entre autre des services tel que la récupération des ressources, des accès base de données et des préférences.

Comme notre classe MainActivity hérite d'Activity qui elle-même hérite de Context, nous pouvons passer la référence 'this' au TextView.

Une fois le TextView instancié, il faut lui spécifier le texte à afficher : `tv.setText("Salut");`

La dernière étape est d'afficher le TextView à l'écran : `setContentView(tv);`

7. La perspective DDMS (Dalvik Debug Monitor Server)

Dans Eclipse : Menu -> Windows -> Open Perspective -> Other -> DDMS

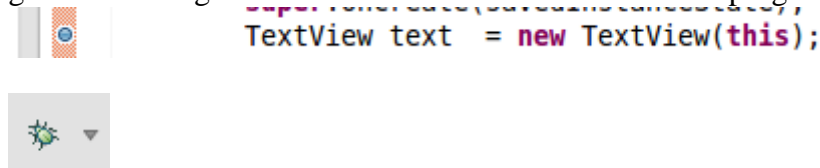
Regarder et indiquer l'utilité des Panels suivants :

- Devices
- Emulator Control
- Logcat
- File Explorer

a°) Pour écrire dans le Logcat des messages de debug il peut utiliser la commande suivant à la fin de la méthode onCreate :

`Log.d(""+this.getComponentName(), "ça marche");`

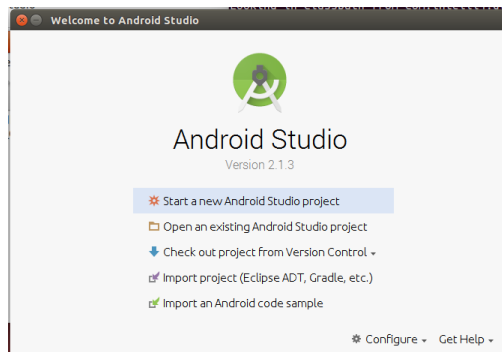
b°) Pour déboguer il suffit de mettre un point d'arrêt en double cliquant dans la colonne de gauche de la ligne souhaitée. Ensuite il faut exécuter le programme en mode débogue



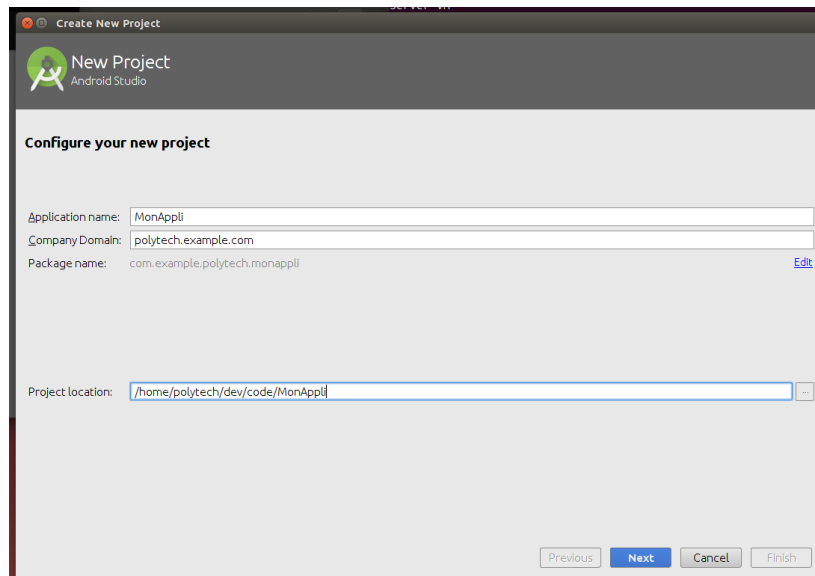
8 Première application avec Android Studio

Démarrer Android Studio en tapant la commande suivante dans le terminal :

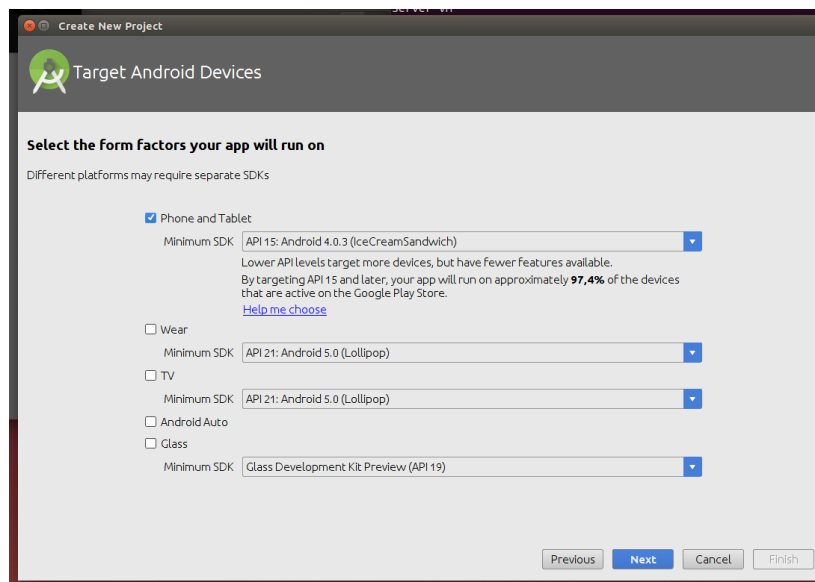
`dev/android-studio/bin/studio.sh`



L'application s'appellera MonAppli :



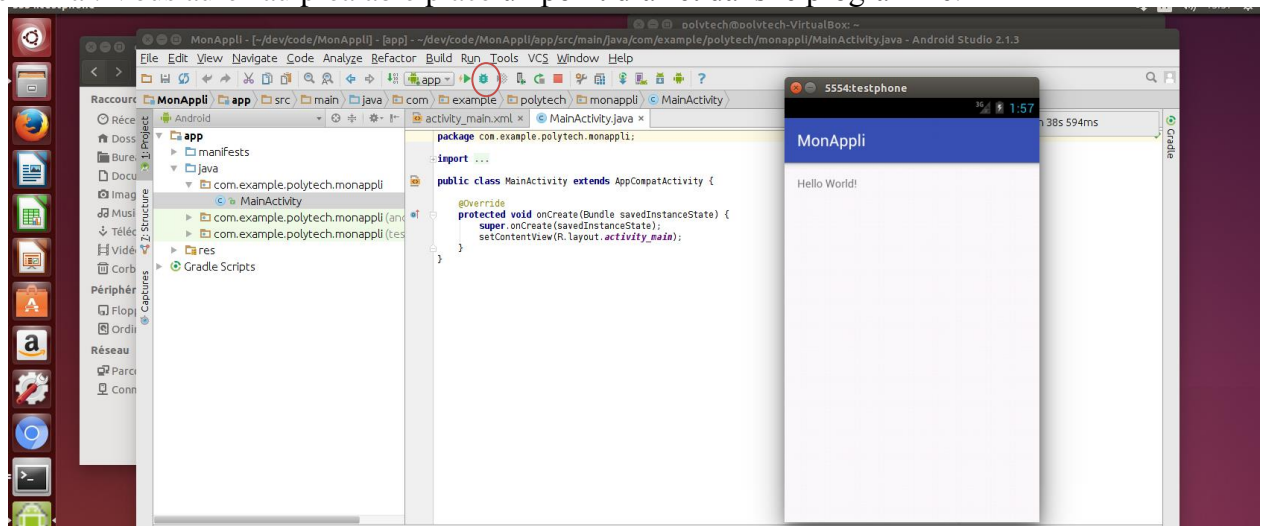
L'application serait compilée pour un terminal équipé de la version IceCreamSandwich.



8.1 °) Lancer l'application sur l'émulateur :

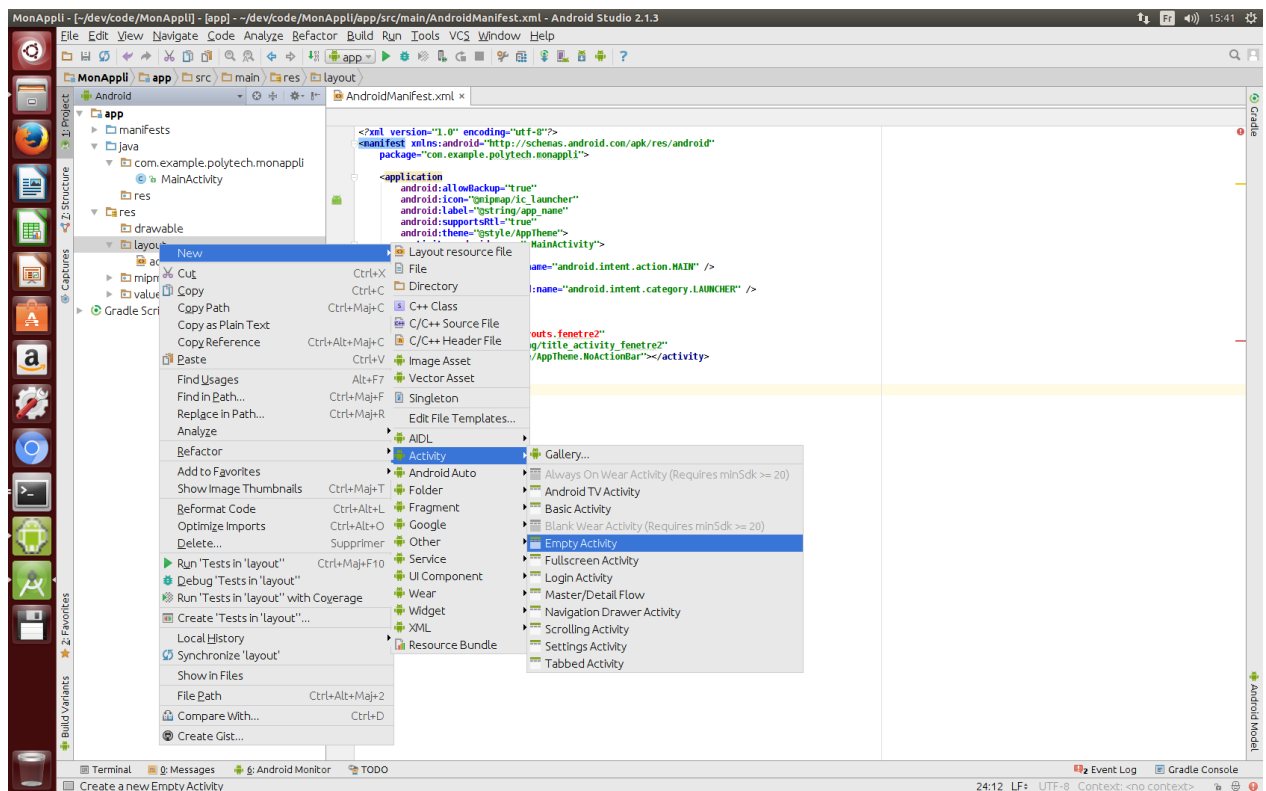


8.2°) Brancher la tablette HP pro slate 8 et lancer l'application en mode « debug » sur ce terminal. Vous aurez au préalable placé un point d'arrêt dans le programme.

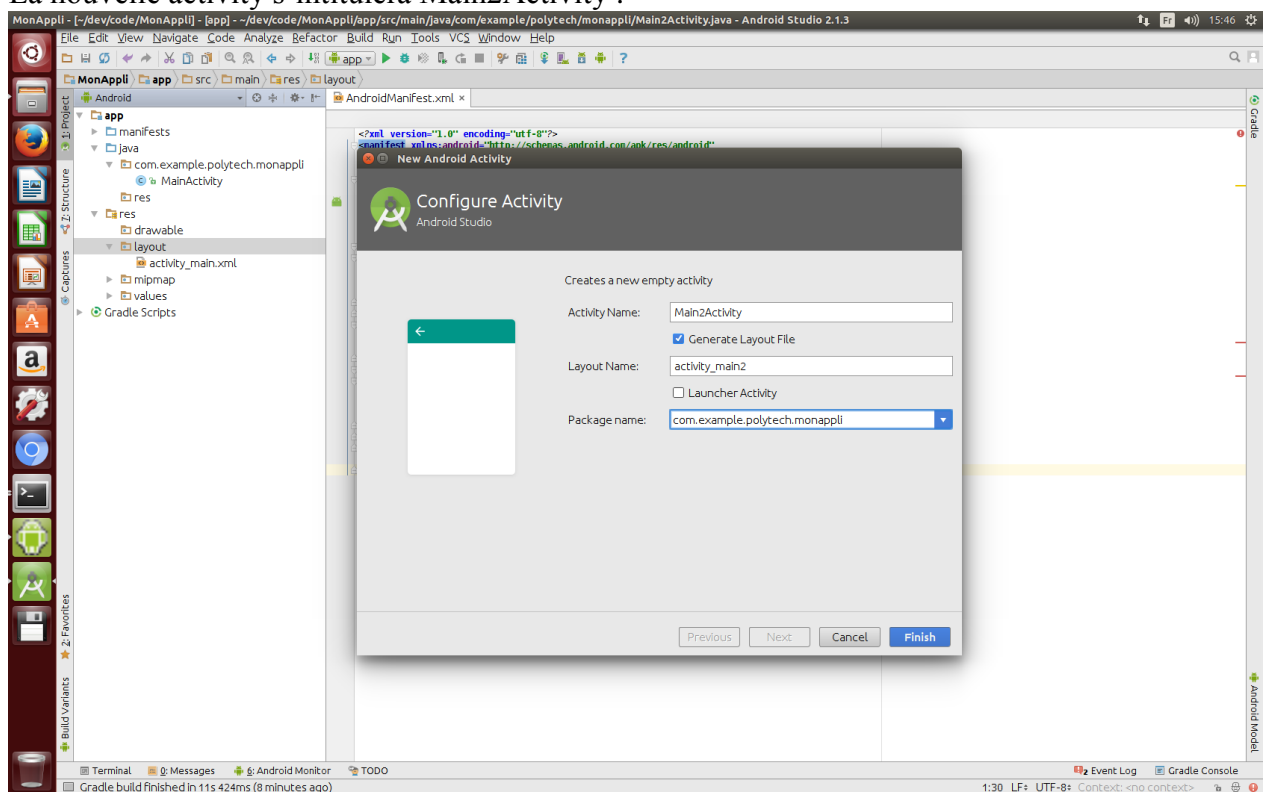


8.3 Créer une nouvelle fenêtre.

Créer une nouvelle Activity à l'aide de la figure suivante :



La nouvelle activité s'intitulera MainActivity :



8.4 Nouvelle fenêtre demarrage

En modifiant le manifest, faite en sorte que MainActivity soit la fenêtre lancée au demarrage de l'application.