

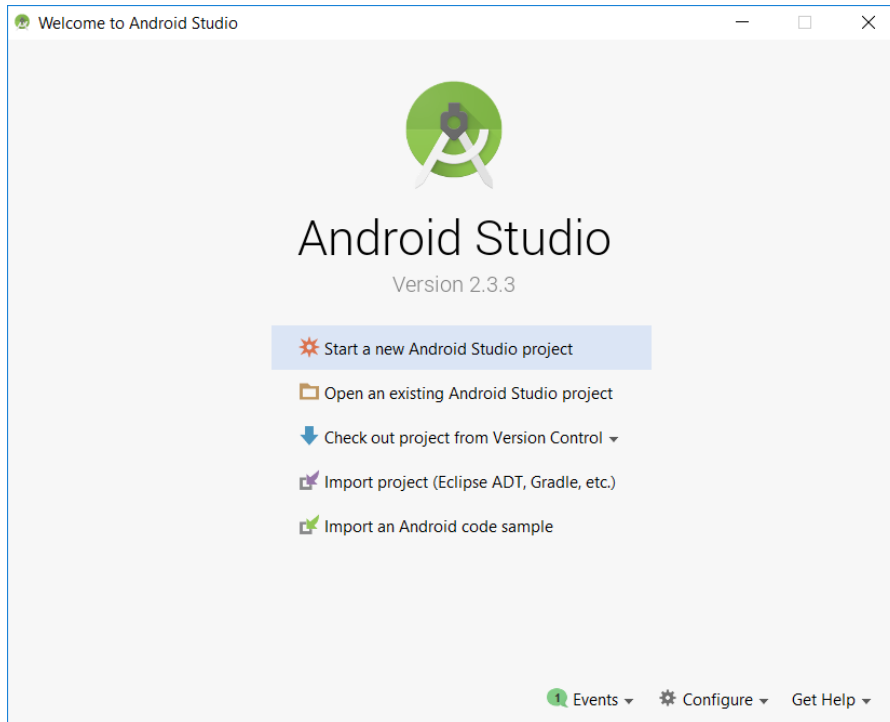


Guide pour les TP sur Android avec Android Studio



Créer un projet avec Android Studio

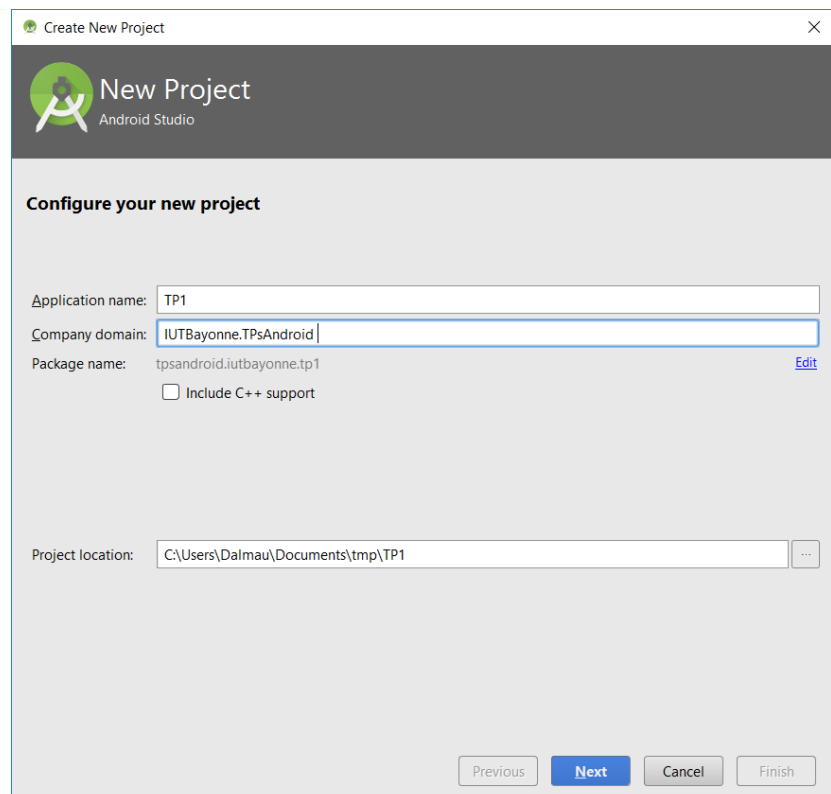
1. Lancer Android Studio
2. Choisir « Start a new Android Studio Project » :



ATTENTION : cette fenêtre reste ouverte ne la fermez pas.

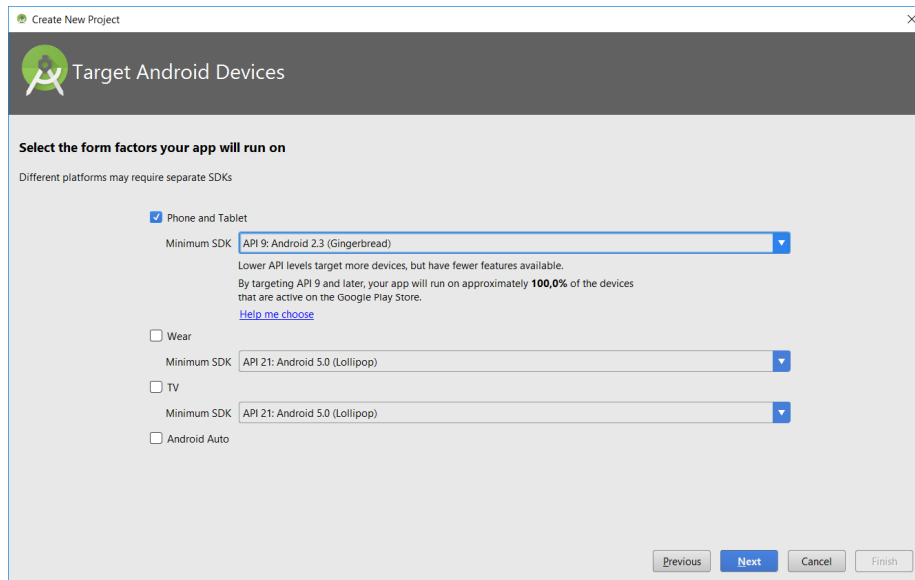
3 Configuration de l'application :

- Dans « Application name » mettre le nom de l'application
- Dans « Company Domain » mettre un nom unique qui identifiera votre application (ce nom doit au moins comporter 2 champs séparés par un point).
- Choisir le répertoire de travail

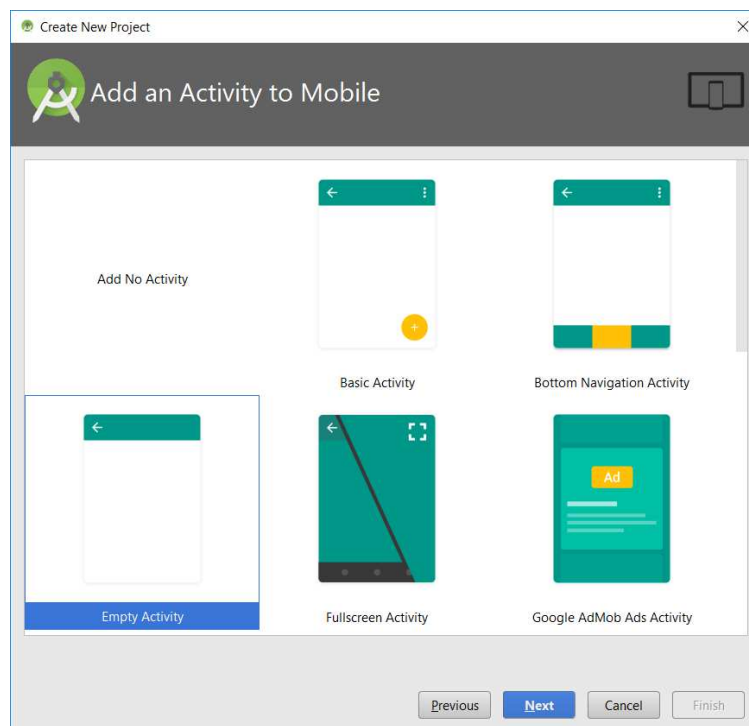


4 Choix du SDK :

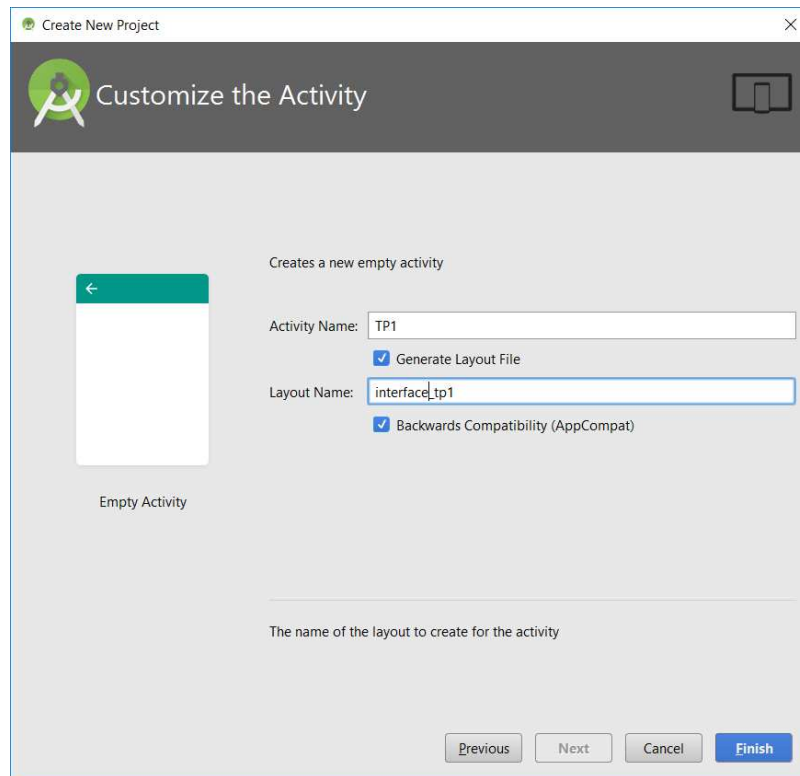
- Ne cocher que la case « Phone and Tablet » et choisir la version minimale de SDK (on peut aller jusqu'à API 9 mais il vaut mieux ne pas descendre sous 14)



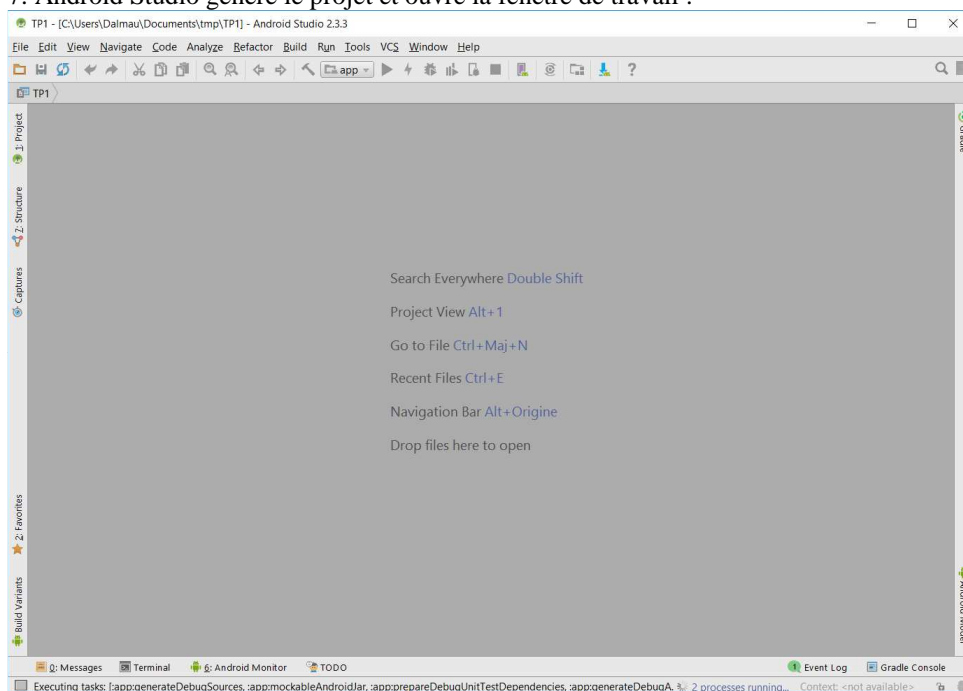
5 Modèle de l'application : choisir le modèle « Empty Activity »



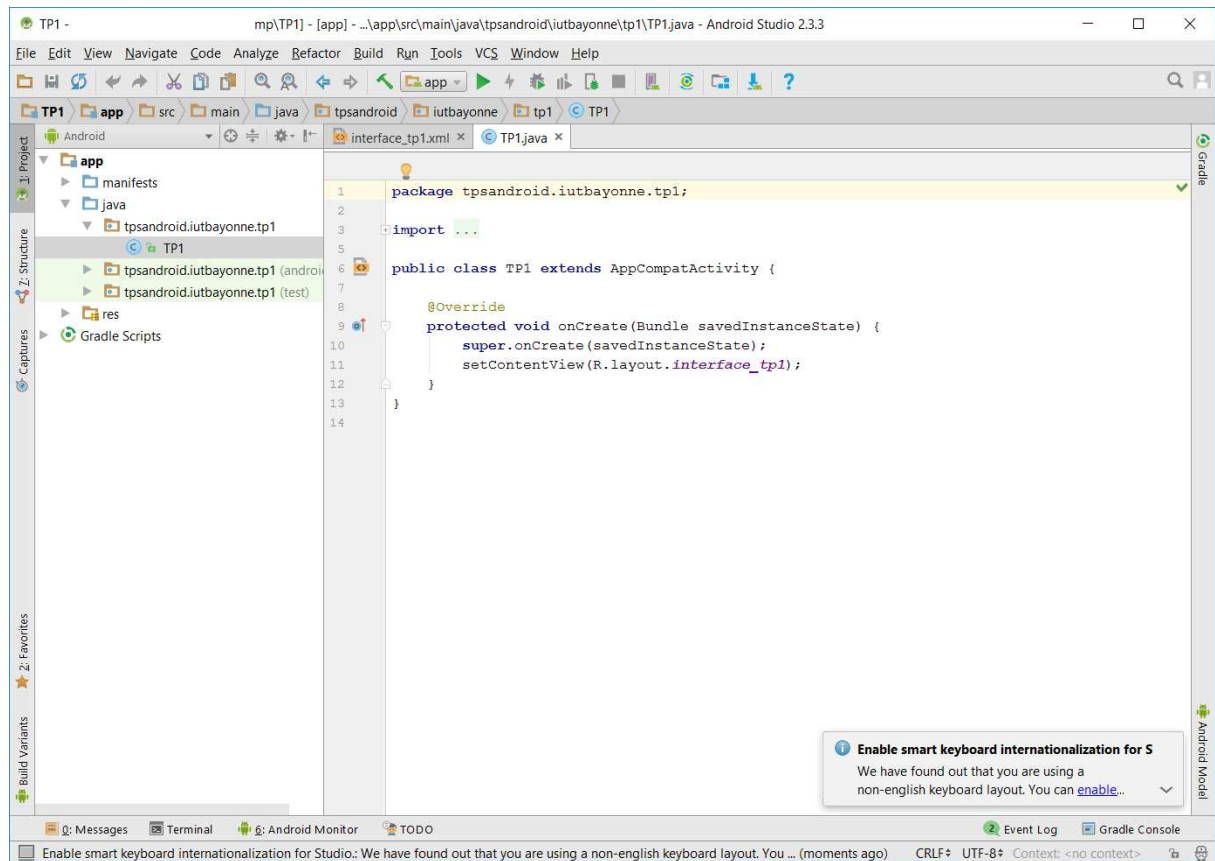
6. Choix du nom de votre activité (nom de la classe) et du nom du fichier XML contenant la description de son interface :



7. Android Studio génère le projet et ouvre la fenêtre de travail :



Choisir « Project View Alt 1 »



La classe créée hérite de **AppCompatActivity** et non de **Activity** pour assurer la compatibilité avec les versions antérieures. Elle ne contient qu'une méthode **onCreate** minimale qui met en place l'interface.

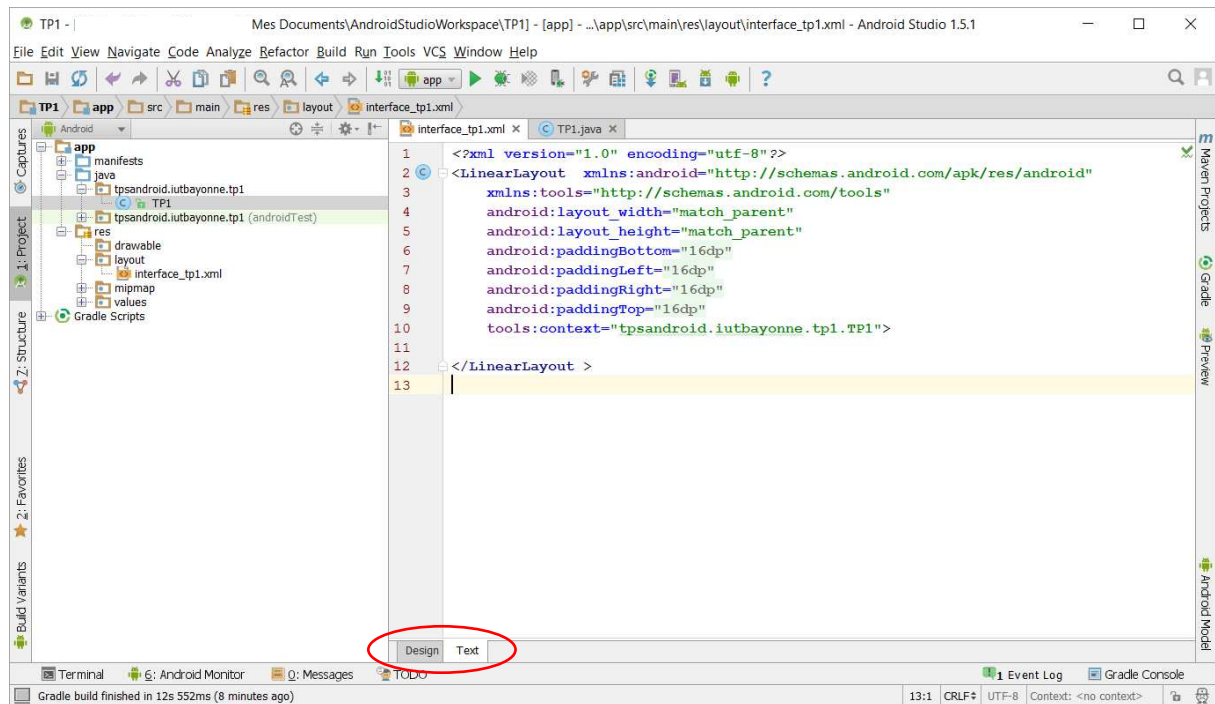
Le fichier XML d'interface créé se trouve dans `res/layout`. Il met en place un conteneur (**ConstraintLayout**) dans lequel est placé un **TextView** contenant la chaîne de caractères appelée "hello_world". Bien entendu ce fichier devra être modifié pour représenter l'interface de votre application. Son contenu est le suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="tpsandroid.iutbayonne.tp1.TP1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

9 Pour commencer, remplacer **RelativeLayout** par **LinearLayout** et supprimer toute la balise **TextView** :

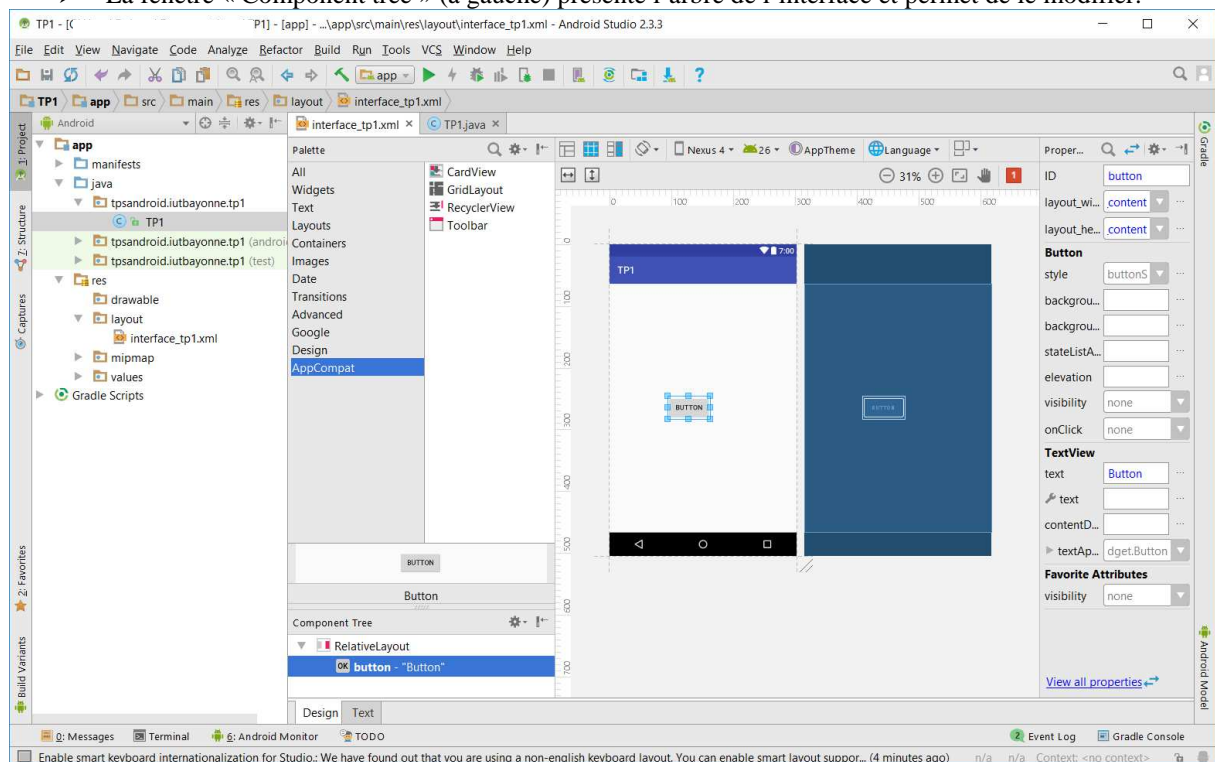


10 Ensuite il faudra y placer votre interface en modifiant ce fichier. Ceci peut se faire :

- En écrivant directement dans le fichier XML visible par l'onglet « Text » en bas
- Avec l'éditeur graphique accessible par l'onglet « Design » en bas

L'onglet « Design » propose :

- Une « Palette » où se trouvent les conteneurs (« Layouts ») et les composants d'interface « widgets »
- Ces éléments peuvent être mis en place par « glisser déposer » dans le terminal affiché.
- Lorsqu'un élément est sélectionné ses propriétés sont affichées à droite (« Properties ») et peuvent être modifiées.
- La fenêtre « Component tree » (à gauche) présente l'arbre de l'interface et permet de le modifier.



11 Android Studio crée également un manifeste minimal (AndroidManifest.xml) qui décrit l'application. Son contenu au démarrage est le suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="tpsandroid.iutbayonne.tp1">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".TP1">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

12 Les répertoires :

Le répertoire « java » contient un paquetage dont le nom est celui qui a été choisi au début et dans lequel se trouve la classe de l'activité.

Le répertoire « manifests » contient le fichier **AndroidManifest.xml** de votre activité.

- Le répertoire « res » est celui des ressources :
 - o drawable images
 - o layout interfaces en xml
 - o mipmap icones
 - o values définition de constantes
- Le répertoire « Gradle Scripts » contient les fichiers de commande utilisés par Android Studio pour compiler et lancer votre application. Normalement vous n'aurez pas à y toucher.

13. A partir de là vous pouvez commencer à créer votre propre application en :

- modifiant le fichier xml pour définir son interface
- modifiant le fichier de l'activité pour y mettre le code propre à l'application
- éventuellement modifiant de fichier AndroidManifest.XML en particulier si l'application a besoin d'autorisations particulières ou contient plusieurs activités
- modifiant le fichier strings.XML si vous devez définir d'autres valeurs
- ajoutant des ressources dans les sous répertoires du répertoire "res" si nécessaire.

Quelques commandes utiles

Ajouter un sous répertoire : clic droit sur le répertoire → New → Directory

Ajouter une classe : clic droit sur le paquetage → New → Java Class

Créer un fichier XML : clic droit sur le sous répertoire de res/layout → New → XML → Layout XML File

Compiler un projet Android

Pour compiler le projet aller dans le menu "Build" et choisir "Make Project".

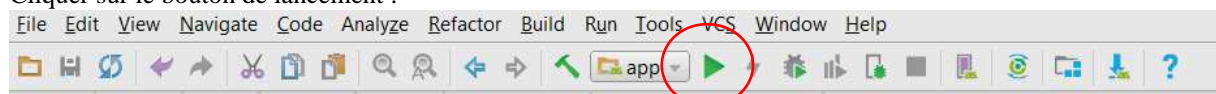
Il est parfois utile de provoquer une recompilation complète, pour cela aller dans le menu "Build" et choisir "Clean" puis aller dans le menu "Build" et choisir "Rebuild project".

Exécuter un projet Android

L'exécution de votre application peut se faire :

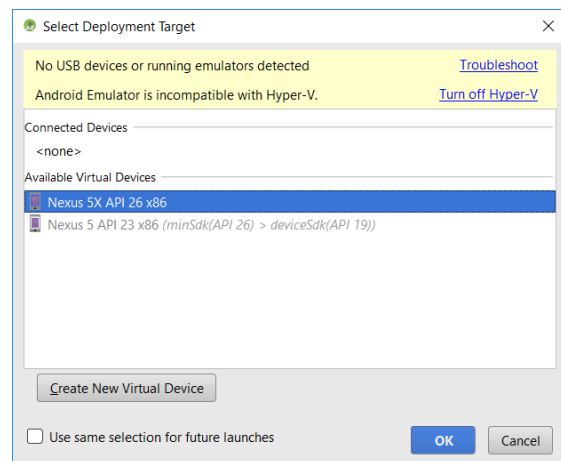
- sur un véritable terminal Android relié par un câble USB (à condition d'avoir le bon driver USB)
- sur un simulateur (terminal virtuel)

Cliquer sur le bouton de lancement :



Choisir un terminal connecté (s'il y en a) ou un terminal virtuel pour utiliser l'émulateur.

On peut également créer un nouveau terminal virtuel.



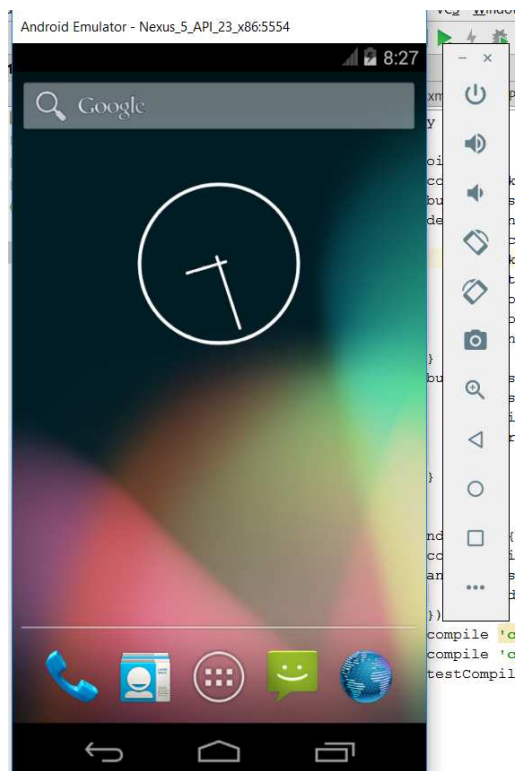
Si vous choisissez un terminal virtuel et qu'il n'est pas déjà lancé, il apparaît à l'écran et s'initialise.

Soyez patients

Le premier lancement du terminal virtuel peut être long

Le menu sur la droite permet de simuler les capteurs (mouvement, GPS ...) du terminal.

Pour que votre application y démarre il suffit de déverrouiller l'écran. Une fois lancé il est inutile de fermer ce terminal. Si vous modifiez votre application vous pouvez en exécuter la nouvelle version sur ce même terminal.



Si vous choisissez un terminal physique, votre application y est téléchargée et lancée.

ATTENTION

Pour pouvoir utiliser un terminal physique il faut avoir installé le bon driver USB ce qui ne vous sera pas possible en salles de TP ou avoir la chance que le SDK le reconnaisse.

Dans les deux cas l'onglet "**Logcat**" de l'onglet "**Android monitor**" en bas de la fenêtre d'Android Studio permet de voir les traces faites par l'application.

SOMMAIRE

Créer un projet avec Android Studio.....	1
Quelques commandes utiles	6
Compiler un projet Android	7
Exécuter un projet Android	7