

POEC Android



Pierre Duchemin ([@pierreduchemin](#))

Histoire

- Nokia est leader sur le marché
- 1999 : Andy Rubin crée Dangers Inc
- DangerOS : OS mobile basé sur Java



Histoire

- Concurrence : Palm OS, Symbian, Windows Mobile, Maemo...
- 2003 : Création de la société "Android"
- 2005 : Rachat par Google
- 2007 : Open Handset Alliance
- 2007 : Google dévoile Android quelques mois après l'iPhone
- Principalement sous licence Apache 2.0 : permet l'apparition de forks : <https://source.android.com>



Aujourd'hui



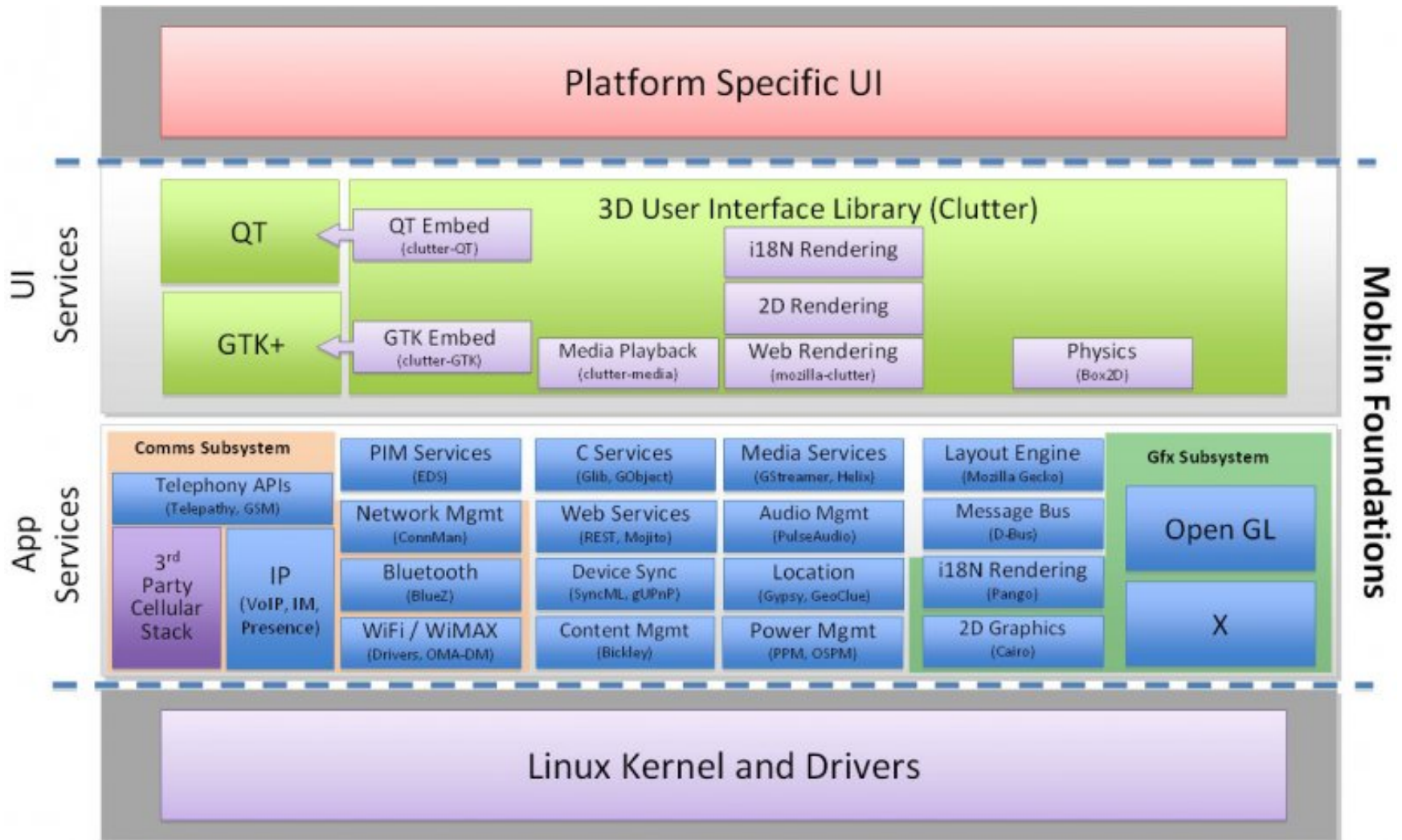
- 87,5 % de parts de marché
- Nov. 2015 - Nov. 2016 : + 10,3 %
- 97,4 % des appareils en API \geq 15 (sortie en 2012)

Histoire technique

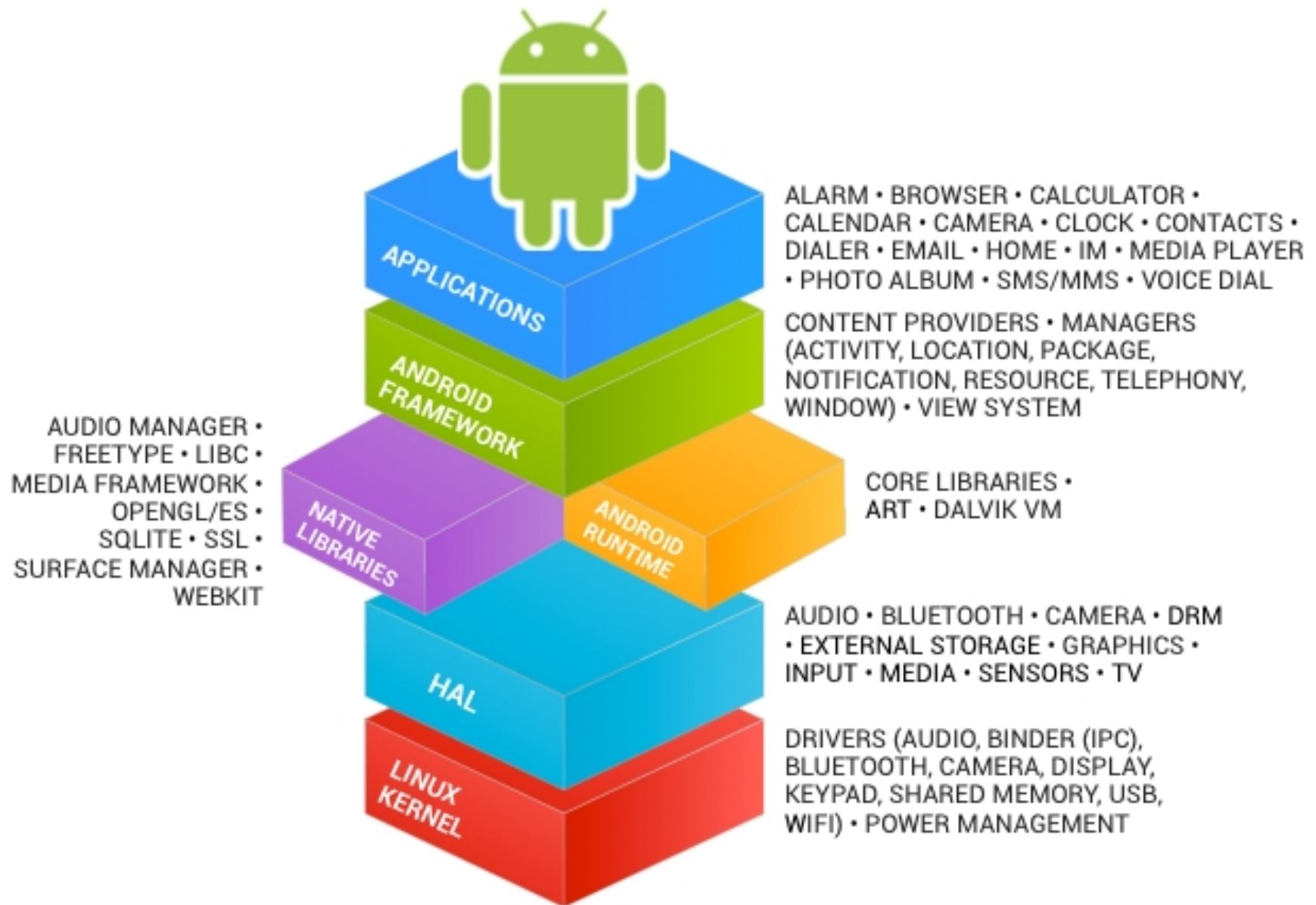
- Ancien IDE : Eclipse + ADT
- Nouvel IDE : Android Studio (IntelliJ)
- Utilisation d'un système de version pour gérer les évolutions
- Création backports (support library)
- Utilise Linux (mais pas GNU)



Architecture d'une distribution Linux



Architecture d'Android



Contraintes spécifiques à l'embarqué

Ressources limitées

- L'occupation mémoire vive et / ou morte
- La consommation d'énergie
- Bande passante réduite / appareil déconnecté du réseau

Une fragmentation importante

- La fragmentation des systèmes d'exploitation
- La fragmentation matérielle

Une ergonomie spécifique

- Utilisation d'écrans tactiles

Notions basiques

- Android != Java
- ...mais reste très proche et généralement compatible
- Compatible C++ via le NDK

Différences avec Java

- ~~JVM Oracle~~ → JVM Dalvik (2007) → JVM ART (2014)
- API spécifiques (accès au matériel, IHM...)
- Le format APK remplace le JAR
- ADB permet la communication avec les appareils / émulateurs
- Plusieurs nouvelles briques logicielles (SDK Manager, lint, ddms...)
- IHM en XML

Quiz

- Les applications Android sont principalement écrites en :

1. JavaScript
2. Java
3. Python
4. Swift

- Les fichiers de Layout sont faits en :

1. Java
2. XML

- Les applications Android ne peuvent être faites qu'à l'aide d'Android Studio.

1. Vrai
2. Faux

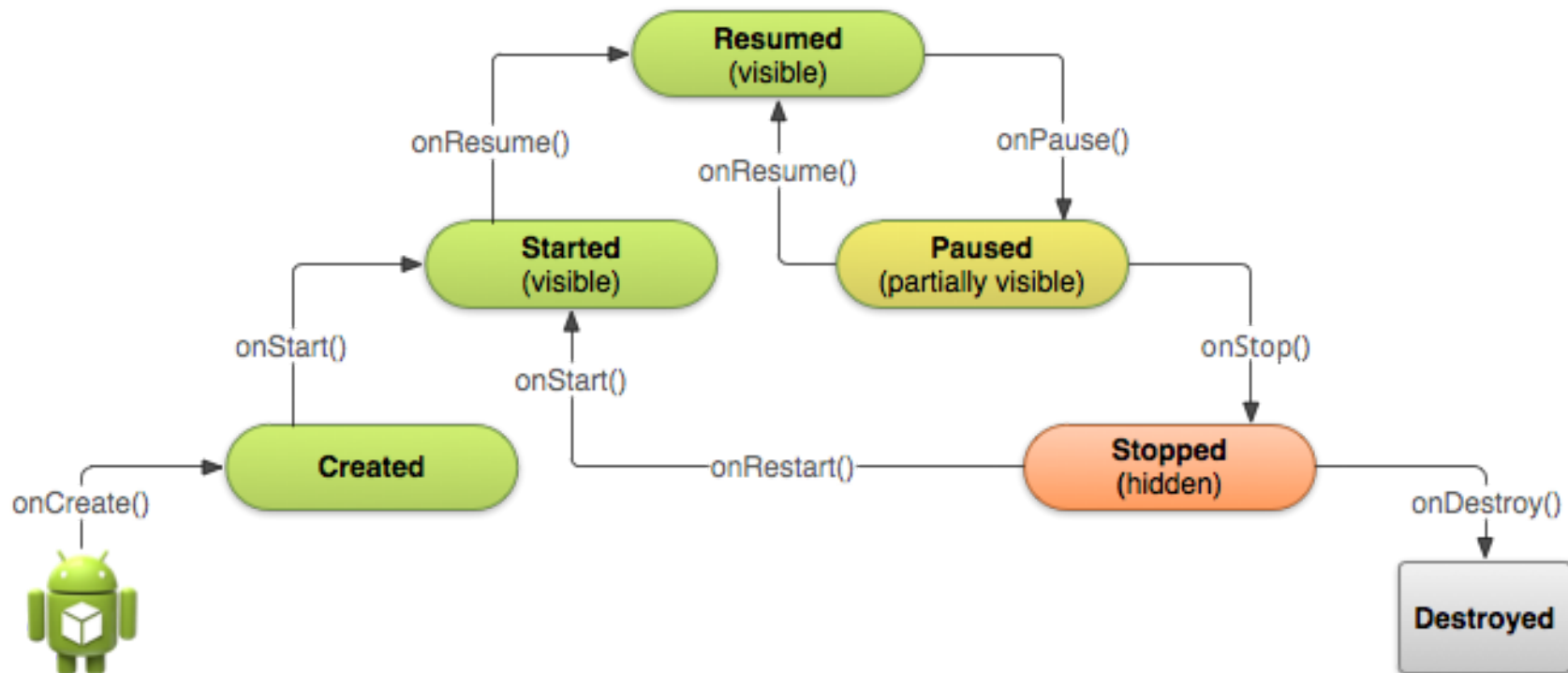
Le manifest

- **Défini le point d'entrée de l'application**
- Liste tous les éléments dont le système a besoin pour faire fonctionner l'application
- Permet de définir quelques options comme le thème de l'application
- Liste les permissions demandées par l'application

Activity

Les **Activity** sont des classes qui représentent un écran d'une application. Elles permettent d'afficher une vue basique, mais aussi des menus, des notifications, des dialogs. Une activity **peut être fermée** ou **ouvrir une autre activity**.

Les Activity doivent être déclarées dans le fichier manifest. Elles ont un cycle de vie.



Activity

- **setContentView** permet d'associer un fichier XML à une Activity
- **findViewById** permet de récupérer une View depuis le fichier XML

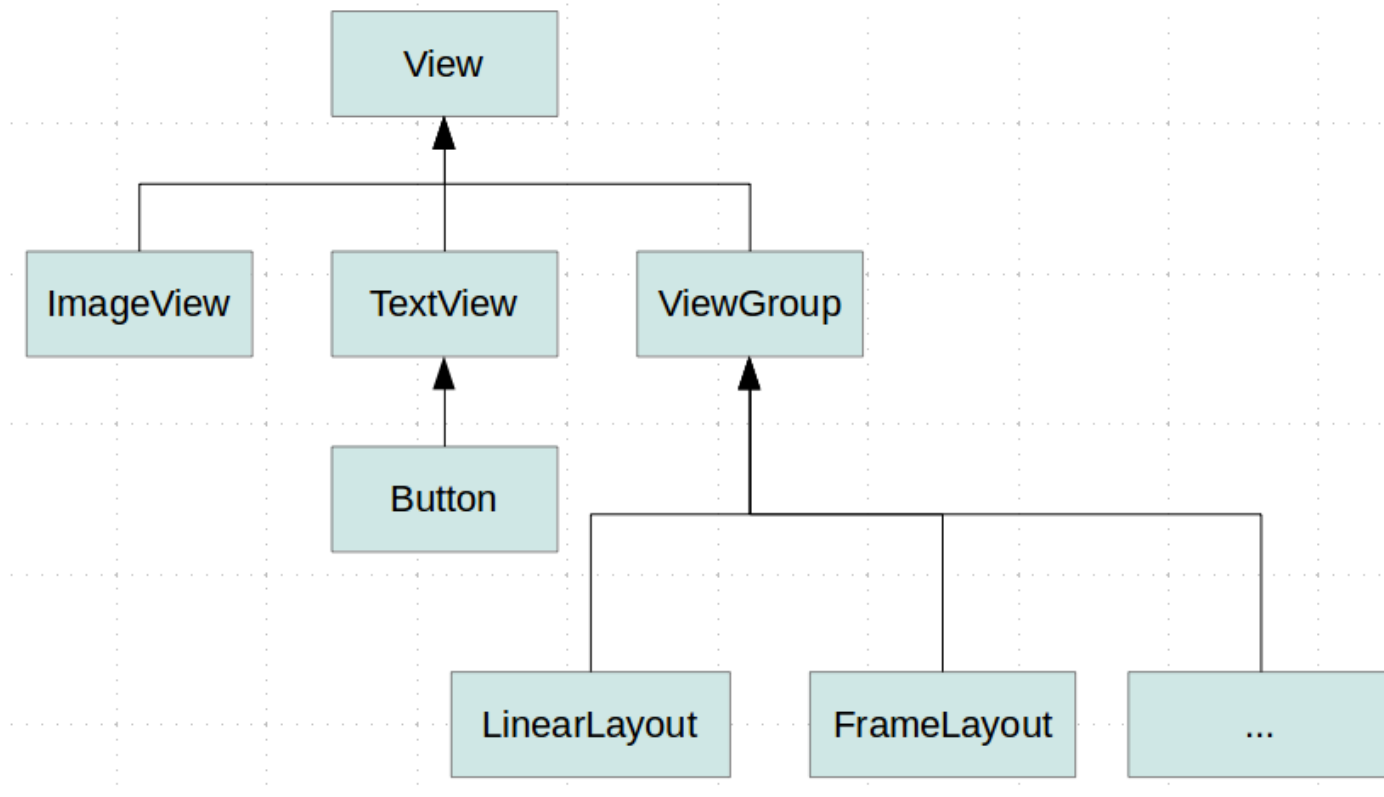
```
<Button  
    android:id="@+id/btnValidate"  
    android:text="OK"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

```
Button btnValidate = (Button) findViewById(R.id.btnValidate)
```

View

Une **View** est un élément de l'IHM. Les View peuvent être imbriquées et former des ViewGroups.

Android fourni par défaut un certain nombre de view basiques : Button, ImageView... Mais il est aussi possible de créer des view customisées.



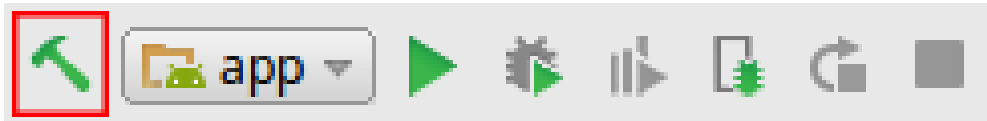
Gestion des ressources

Toutes les ressources (images, fichier binaire, strings...) ont **une clé qui les identifie** dans un fichier généré appelé "R".

Ce fichier est généré lors du build du projet.

Il faut utiliser ces clés pour utiliser les ressources.

Pour générer le fichier R, il suffit de cliquer sur

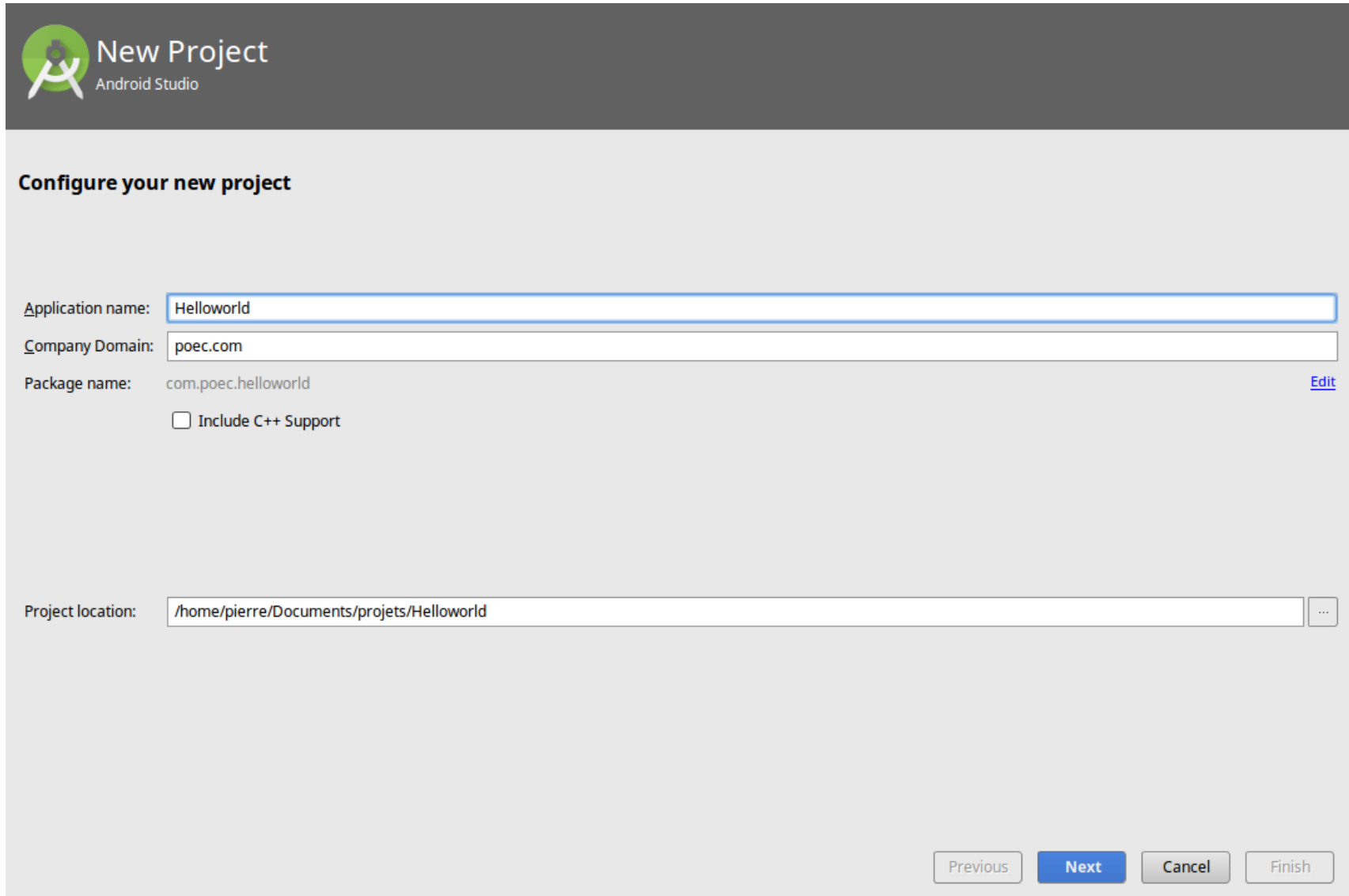


Permet de résoudre l'erreur :

Ressource (R) cannot be resolved

Un peu de code !

- Ouvrir Android Studio
- Créer un nouveau projet nommé "HelloWorld"



The screenshot shows the 'New Project' dialog in Android Studio. The title bar at the top says 'New Project' with the Android Studio logo. Below the title bar, the text 'Configure your new project' is displayed. The dialog contains several input fields: 'Application name' with the value 'Helloworld', 'Company Domain' with 'poec.com', and 'Package name' with 'com.poec.helloworld'. There is an 'Edit' link next to the package name. A checkbox for 'Include C++ Support' is present and unchecked. At the bottom, the 'Project location' field shows the path '/home/pierre/Documents/projets/Helloworld' with a browse button (three dots) to its right. At the very bottom of the dialog are four buttons: 'Previous', 'Next' (highlighted in blue), 'Cancel', and 'Finish'.

New Project
Android Studio

Configure your new project

Application name:

Company Domain:

Package name: [Edit](#)

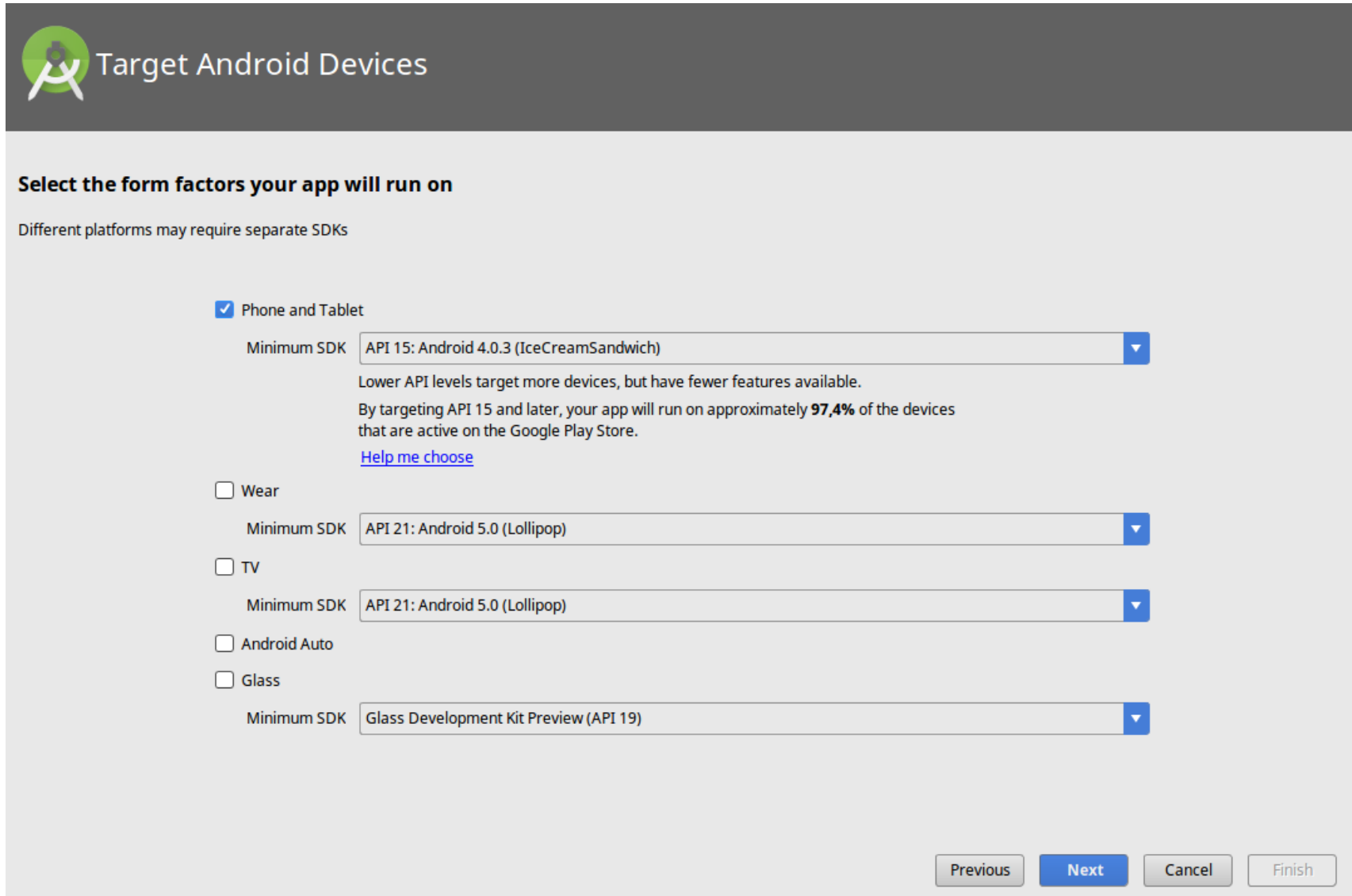
☐ Include C++ Support


Project location: ...

[Previous](#) [Next](#) [Cancel](#) [Finish](#)

Un peu de code !

- Garder la configuration par défaut



 Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

☒ Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately **97,4%** of the devices that are active on the Google Play Store.
[Help me choose](#)

☐ Wear

Minimum SDK

☐ TV

Minimum SDK

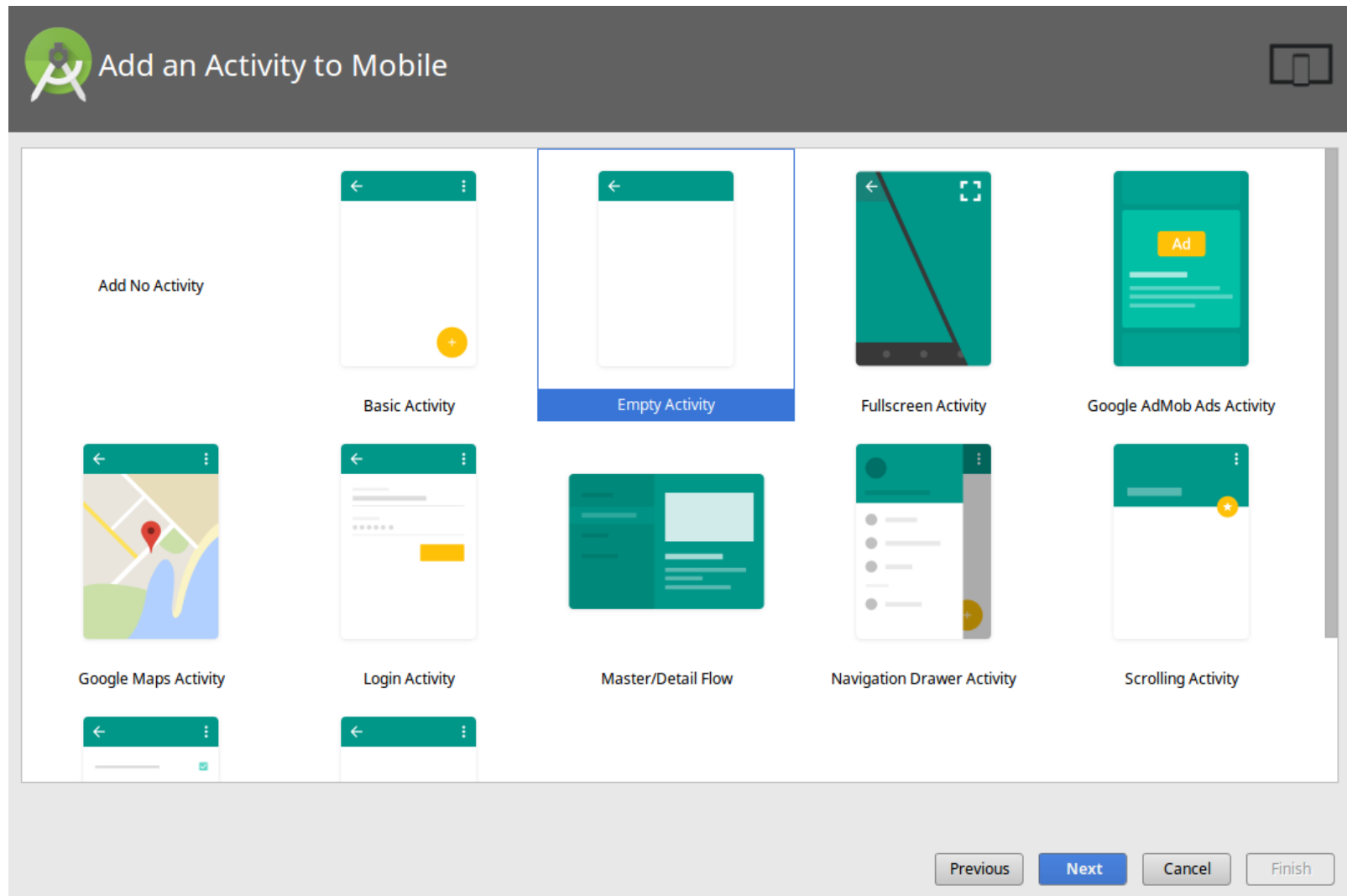
☐ Android Auto

☐ Glass

Minimum SDK

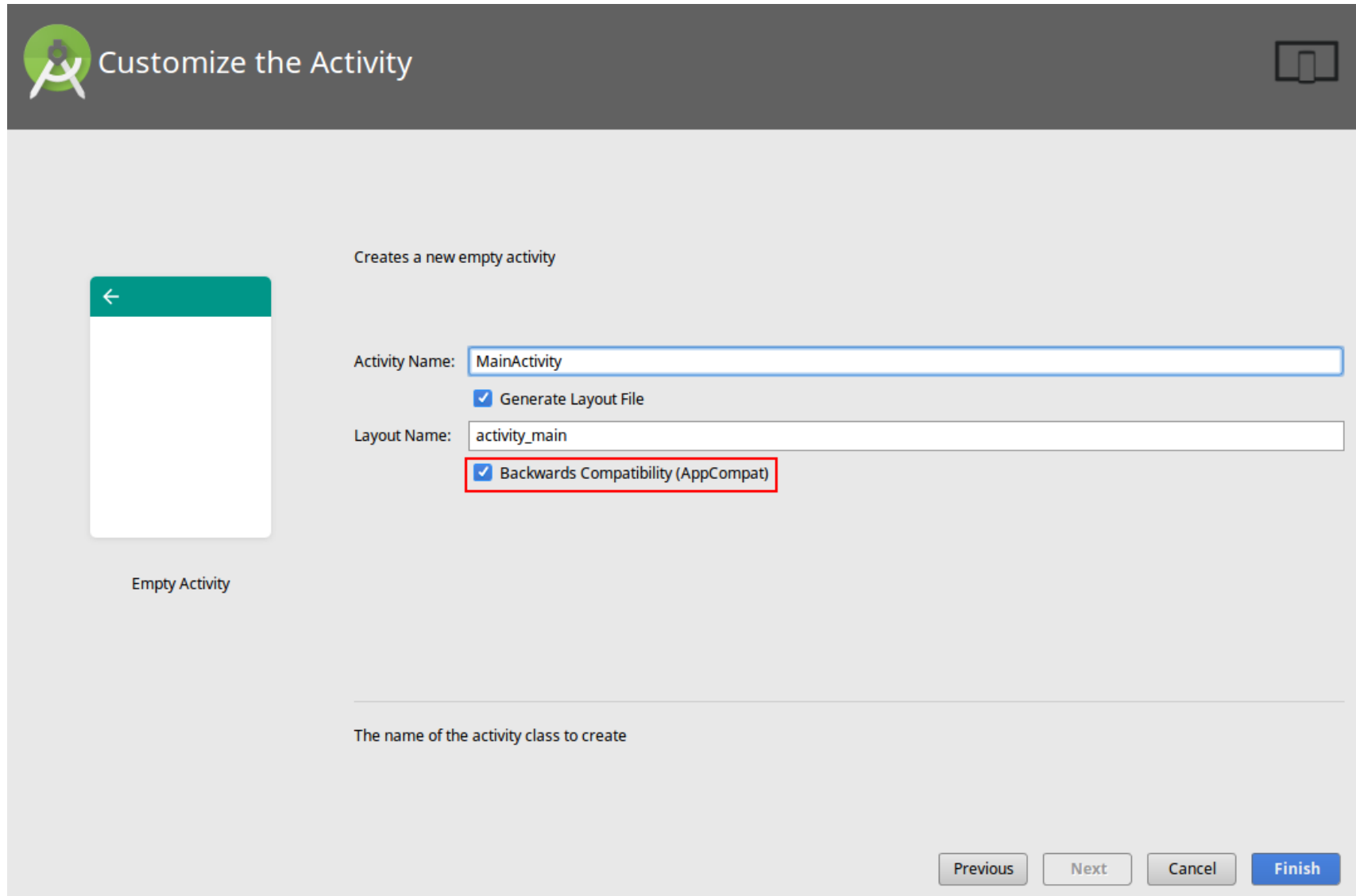
Un peu de code !

- Utiliser le template "Empty Activity"
- Les template de projets sont bien faits et mis à jour



Un peu de code !

- Utiliser AppCompatActivity



Customize the Activity

Creates a new empty activity

Activity Name: MainActivity

☒ Generate Layout File

Layout Name: activity_main

☒ Backwards Compatibility (AppCompat)

Empty Activity

The name of the activity class to create

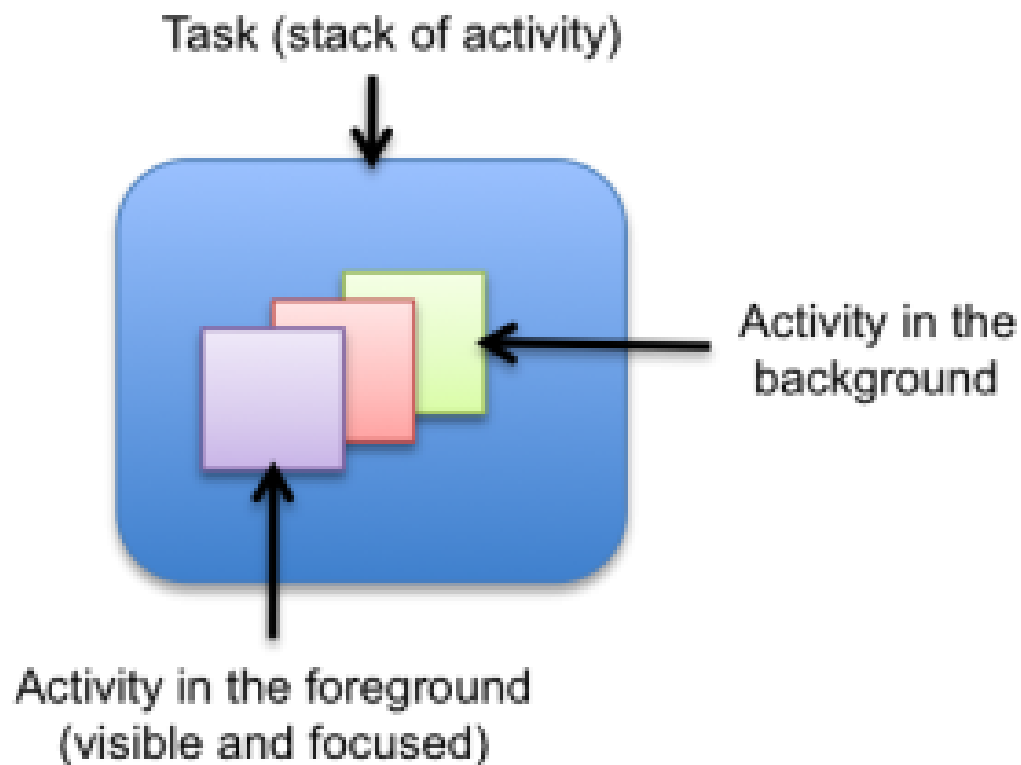
Previous Next Cancel Finish

Un peu de code !

- Ajouter un identifiant à la TextView (id)
- Changer le message "Hello World!" en Java

Le backstack

Par défaut, lors de la navigation dans une application, les activités précédentes ne sont pas fermées, mais simplement mises en arrière plan.

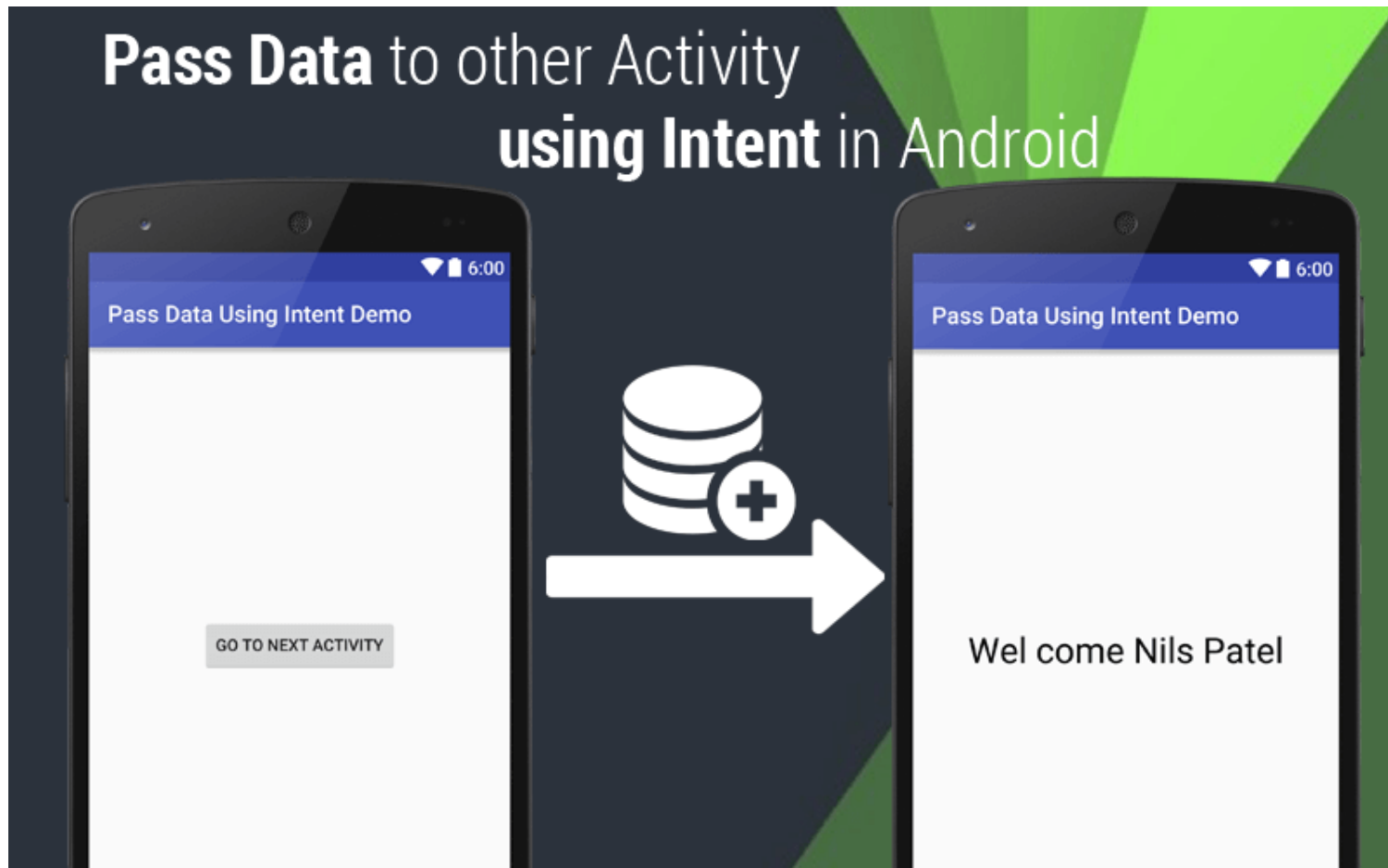


Elles s'emplissent donc et la touche "retour" permet de dépiler les activités.

Intent

Les **Intent** des relais entre des activity ou des applications. Ils permettent de passer des données d'une entité à l'autre.

Ils peuvent également être porteurs de "flags", des marqueurs indiquant une action particulière à réaliser.



Service

Les **Services** sont des **tâches de fond** similaires aux crons Unix.

Ils ne disposent pas d'une IHM.

Les Services doivent être déclarés dans le fichier manifest.

Versions d'Android

- Il existe des **numéro de version** à visée commerciale et des **niveau d'API**
- 25 niveau d'API



Cupcake
Android 1.5



Donut
Android 1.6



Eclair
Android 2.0



Froyo
android 2.2.x



Gingerbread
Android 2.3.x



Honeycomb
3.0



4.0
Ice Cream Sandwich



Jelly Bean
Android 4.1.x



KitKat
Android 4.4.x



Lollipop
Android 5.0



Marshmallow
android 6.0



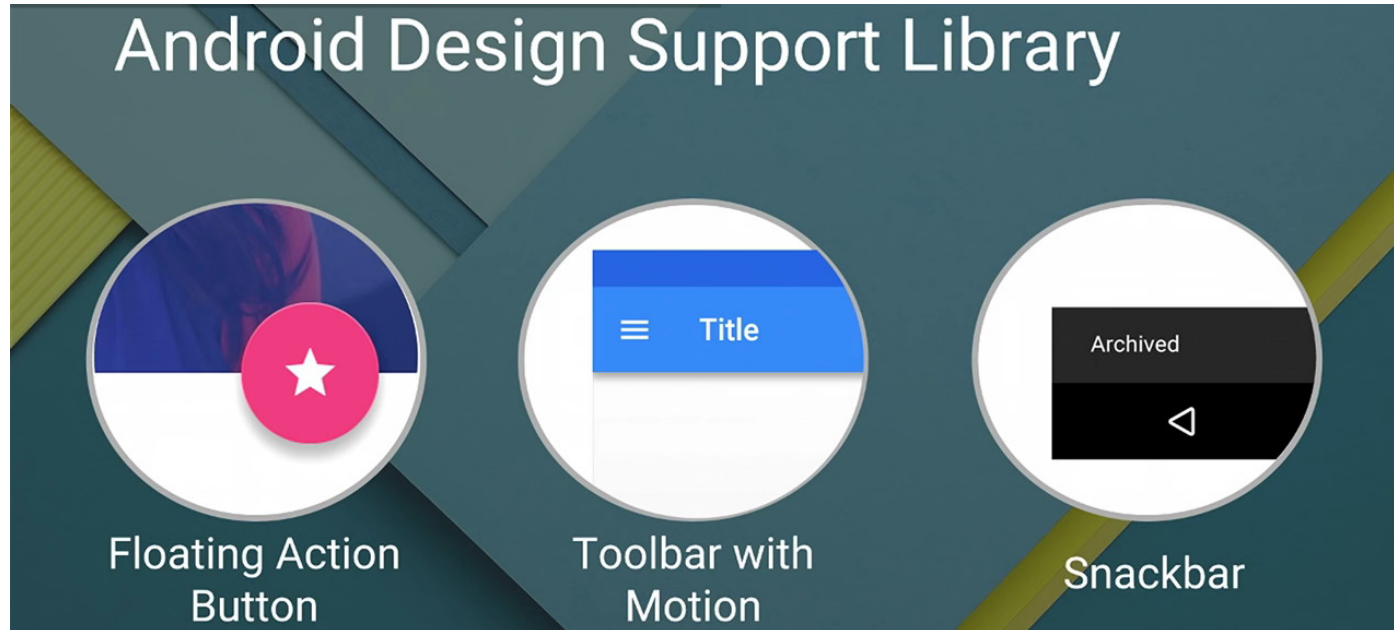
Nougat
android 7.0

Versions d'Android

Version ⇅	Code name ⇅	Release date ⇅	API level ⇅	DVM/ART ⇅	Distribution ⇅
7.1	Nougat	October 4, 2016	25	ART	0.3%
7.0		August 22, 2016	24	ART	0.9%
6.0	Marshmallow	October 5, 2015	23	ART	30.7
5.1	Lollipop	March 9, 2015	22	ART	23.1%
5.0		November 3, 2014	21	ART 2.1.0	9.8%
4.4	KitKat	October 31, 2013	19	DVM (and ART 1.6.0)	21.9%
4.3	Jelly Bean	July 24, 2013	18	DVM	1.6%
4.2		November 13, 2012	17	DVM	5.7%
4.1		July 9, 2012	16	DVM	4.0%
4.0	Ice Cream Sandwich	December 16, 2011	15	DVM	1.0%
2.3	Gingerbread	February 9, 2011	10	DVM 1.4.0	1.0%

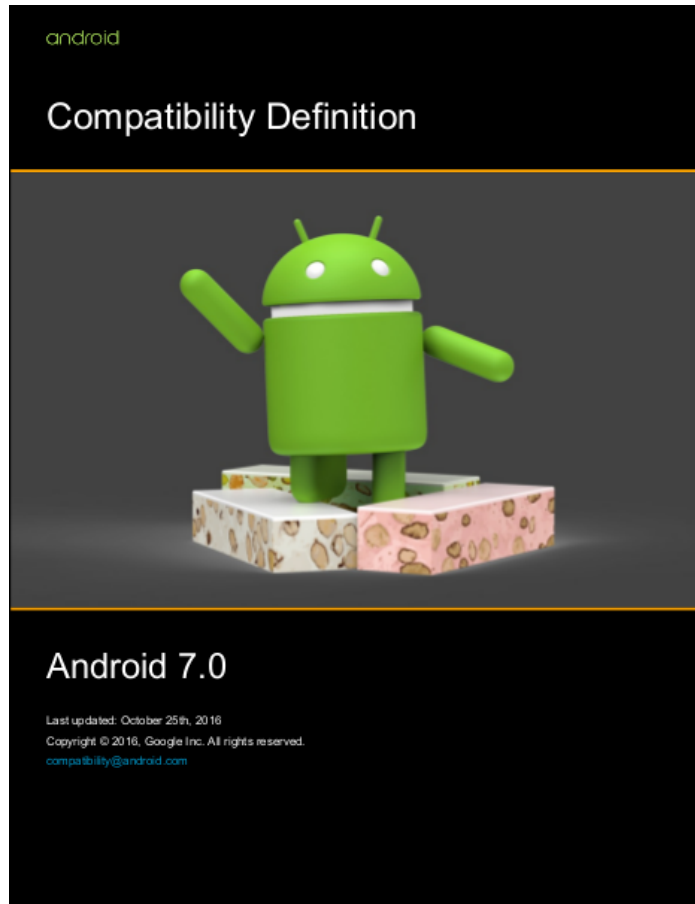
Les support libraries

- Backport des dernières fonctionnalités d'Android sur les anciennes versions d'Android
- Sauf cas particulier, il faut toujours utiliser les support libraries
- Il faut dériver "AppCompatActivity" et non "Activity" et toujours utiliser la version support des Objets Android



Implémentations différentes

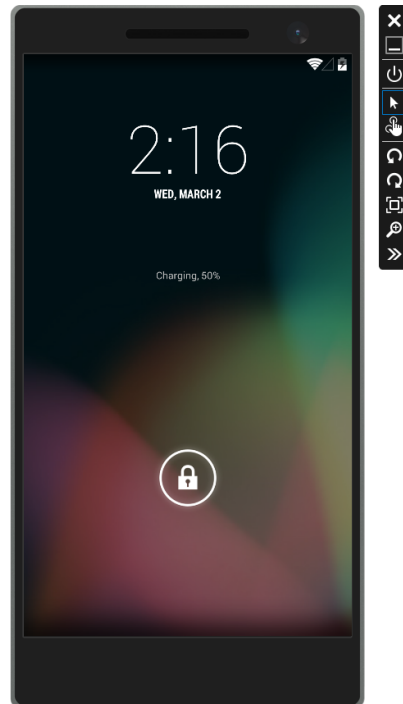
- Sur le web, tous les navigateurs ne réagissent pas de la même manière, **sur Android, tout dépend des constructeurs**
- Document de référence : Compatibility Definition



Nos outils

L'émulateur

- Permet de **simuler le fonctionnement d'un appareil Android.**
- Pas de Google Play à cause de restrictions imposées par Google.



Gradle

- Outil de build concurrent de Maven, Ant, ou Ivy.
- Configurable de la même manière au niveau du projet ou du module (sous élément d'un projet, peu utilisé).
- Pour ajouter une dépendance, il **faut** utiliser la gestion des dépendances de Gradle.

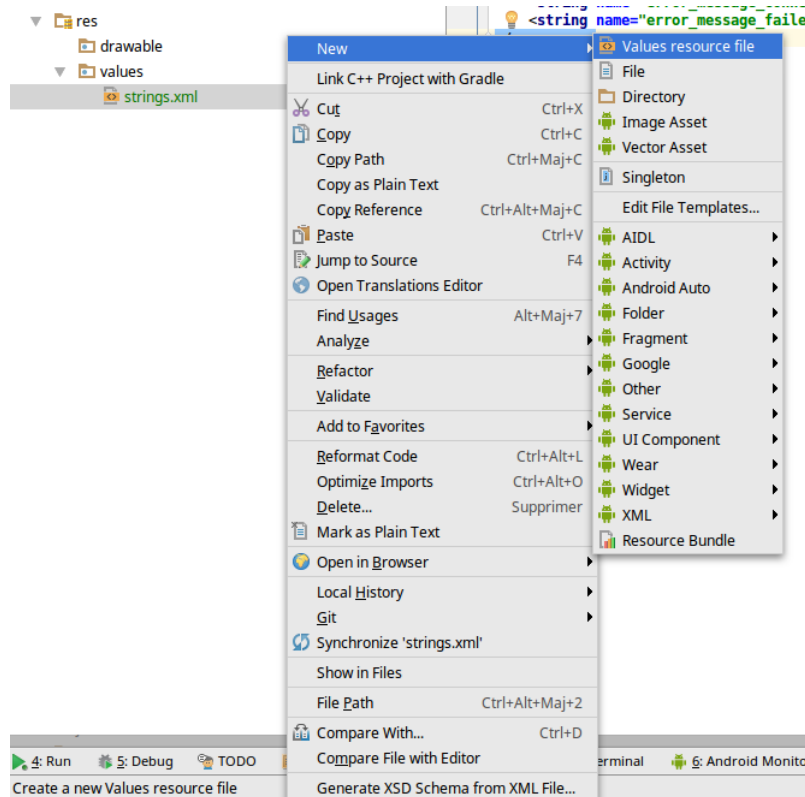


À propos des versions

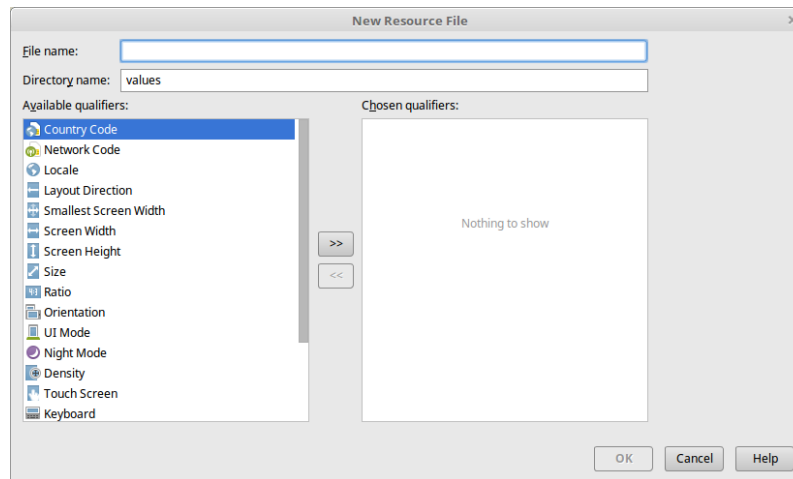
- **compileSdkVersion** : toujours utiliser la dernière version
- **minSdkVersion** : la version du SDK minimum requise pour pouvoir installer l'application
- **targetSdkVersion** : version cible pour la compatibilité du code. Lors de l'installation, toujours choisir la dernière version. Lors d'une mise à jour, tester le comportement pour valider qu'il n'y a pas de regression
- **versionName** : le numéro de version qui sera visible par les clients
- **versionCode** : un identifiant permettant à Android de déterminer qu'une version de l'application est plus récente qu'une autre

Système de gestion des cas particuliers

Android intègre un système permettant de gérer les traductions, mais aussi de nombreux cas spécifiques



Systeme de gestion des cas particuliers



Les noms de dossiers sont ensuite utilisés pour différencier les spécificités

Exercices

- Reprendre le projet HelloWorld
- Ajouter des traductions
- Ajouter un layout spécifique au mode paysage