Université
Internationale
de Casablanca

Introduction à la programmation mobile Android

Cours 3

Programmation des systèmes embarqués Android :
L'univers Android, configuration de l'IDE, Structure d'un projet Android,
les conteneurs et composants graphiques

Wilfried Y. H. Adoni

Université Internationale de Casablanca

Programme de la séance

- L'univers Android
- Installation et configuration des outils
- 1ere application Android
- Structure d'un projet Android
- Activités
- Les conteneurs graphiques
- Les composants graphiques

Université Internationale de Casablanca

Bref historique

En 2010, Android devient le système d'exploitation mobile le plus utilisé au Monde.

2003

Naissance de la startup *Android inc.*

2005

Rachat de la startup par Google inc.

2007

Annonce d'Android +SDK Open Handset Alliance

2008

Premier mobile sous Android : le T-Mobile G1

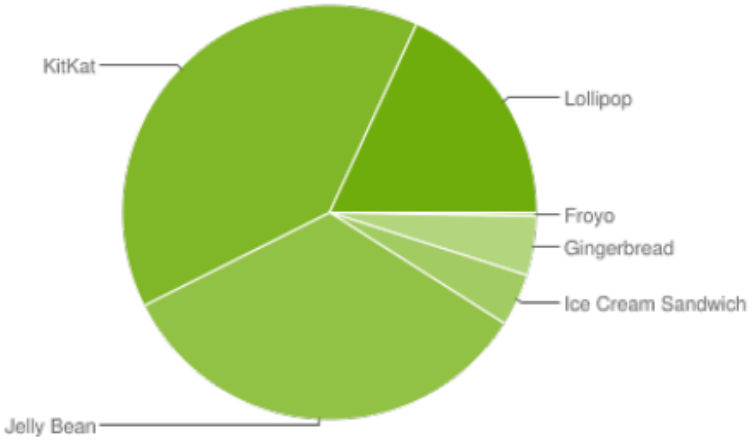
Voir annonce par Sergey Brin sur YouTube :

http://goo.gl/LejV9

Université Internationale de Casablanca

Versions d'Android

A ce jour, Google recommande de développer des programmes en choisissant l'API 14 : Android 4.0 (IceCreamSandwich), l'application sera en mesure de fonctionner sur 100% des appareils.



August 3, 2015 – Google Inc

CodeName	Platform	API Level
Cupcake	Android 1.5	3
Donut	Android 1.6	4
Eclair	Android 2.1	7
Froyo	Android 2.2	8
Gingerbread	Android 2.3	9
Honeycomb	Android 3.0	11
Ice Cream Sandwich	Android 4.0	14
Jelly Bean	Android 4.1	16
KitKat	Android 4.4	19
Lollipop	Android 5.0	20
Marshmallow	Android 6.0	23
Nougat	Android 7.0	24

Université Internationale de Casablanca

La philosophie et les avantages d'Android

- **Open Source et gratuit**
  - ❑ Le contrat de licence de Android respecte le Open Source
  - ❑ Android est gratuit, autant pour les développeurs que pour les constructeurs.
- **Facile à développer**
  - ❑ Toutes les API mises à disposition facilitent et accélèrent grandement le travail.
  - ❑ Ces APIs sont très complètes et très faciles d'accès
- **Facile à vendre**
  - ❑ Le Play Store est une plateforme immense et très visitée
  - ❑ L'achat d'un espace sur le Play Store coutera la modique somme de 25$
  - ❑ Publier des applications de façon illimitées
- **Flexible**
  - ❑ Le Système est extrêmement portable
  - ❑ Il s'adapte à beaucoup de structures différentes: smartphones, tablettes, Smart TV, Montre connectées

Université Internationale de Casablanca

Prérequis et outils nécessaires

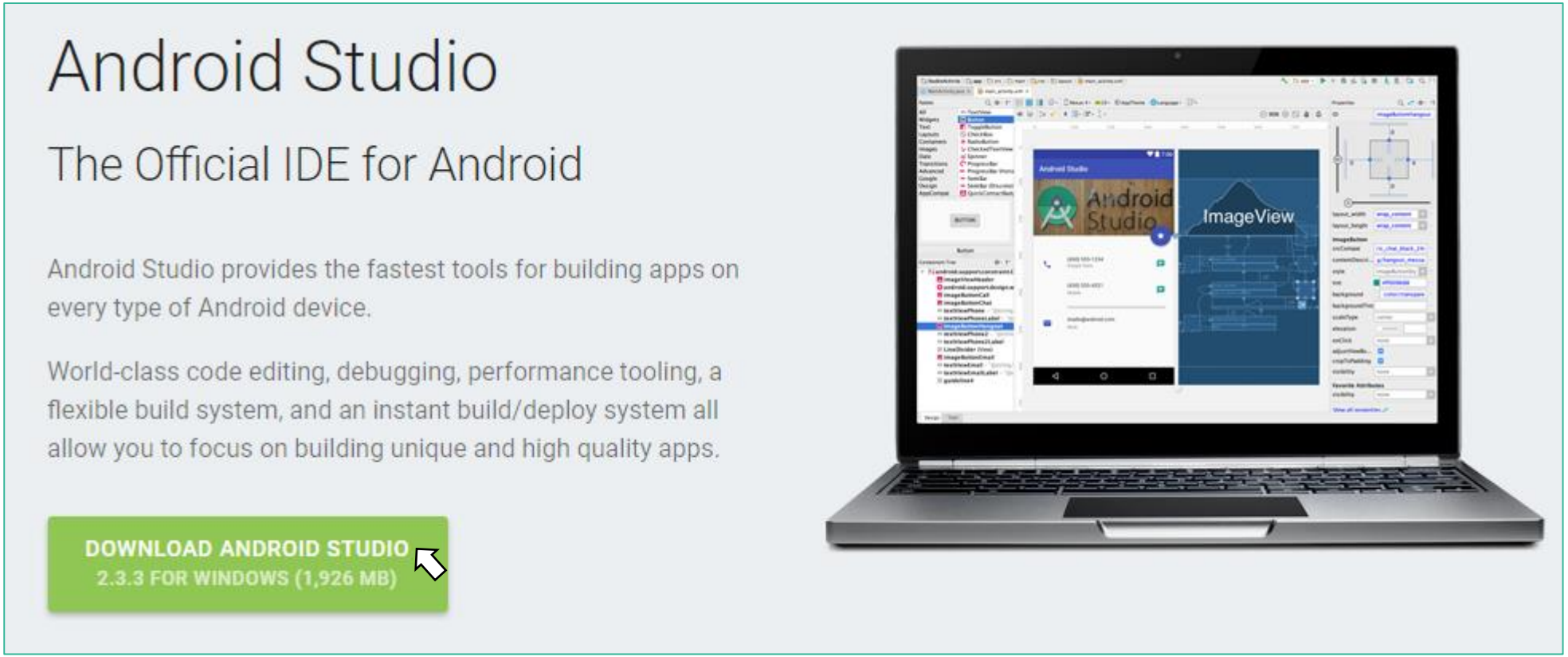
- **Configuration matérielle**
  - ❑ 3 GB RAM minimum, 8 GB RAM recommandée; plus 1 GB pour l'émulateur Android
  - ❑ 2 GB d'espace disque minimum pour tous installer,
  - ❑ Niveau processeur, l'émulation ne peut se faire que sur 1 cœur de votre processeur
- **JDK: Java Development Kit**
  - ❑ Outil de développement principal des applications Android est le **Java**
- **Android Studio et le SDK**
  - ❑ Un ensemble d'outils indispensables pour développer nos applications Android.
  - ❑ Android Studio, est l'IDE Android officiel supporté par Google.
  - ❑ SDK contient l'ensemble du Kit de développement des applications Android

- **Android Studio n'est pas le seul environnement de développement Android, Eclipse propose également un environnement de développement d'applications Android.**

Université Internationale de Casablanca

Téléchargement d'Android Studio

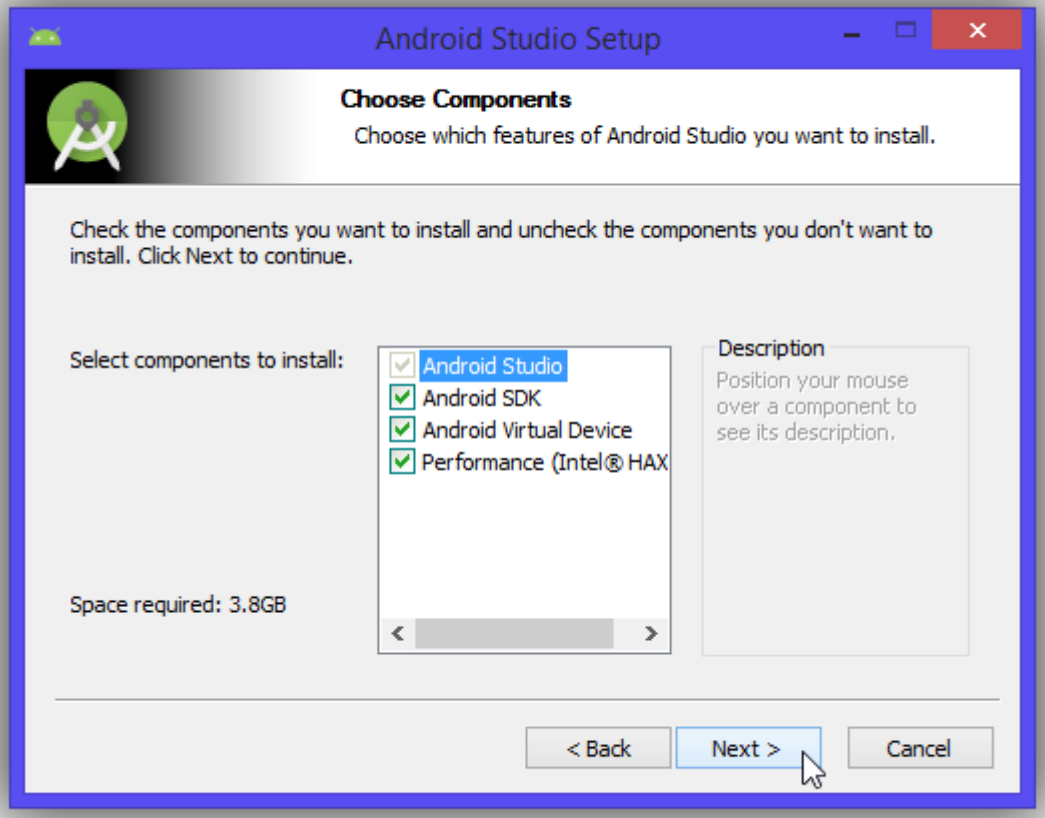
- Pour télécharger, rendez-vous sur la page officiel d'Android Studio: https://developer.android.com/studio/index.html
- La page détecte automatiquement votre système d'exploitation et vous propose de télécharger l'archive compatible (pour Mac, Windows ou Linux)



Université Internationale de Casablanca

Installation d'Android Studio (1/3)

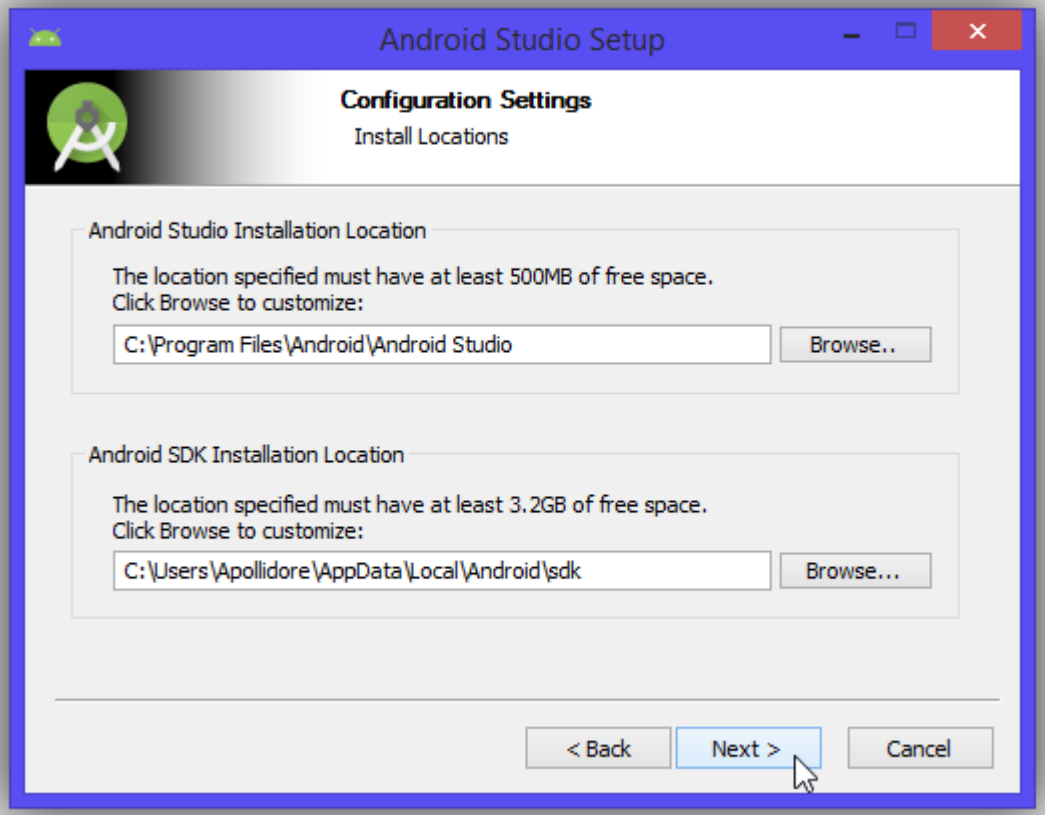
- Conservez les options Android SDK et Android Virtual Device.
- Sélectionnez Performance (Intel® HAXM) que si vous avez un processeur Intel capable d'émulation



Université Internationale de Casablanca

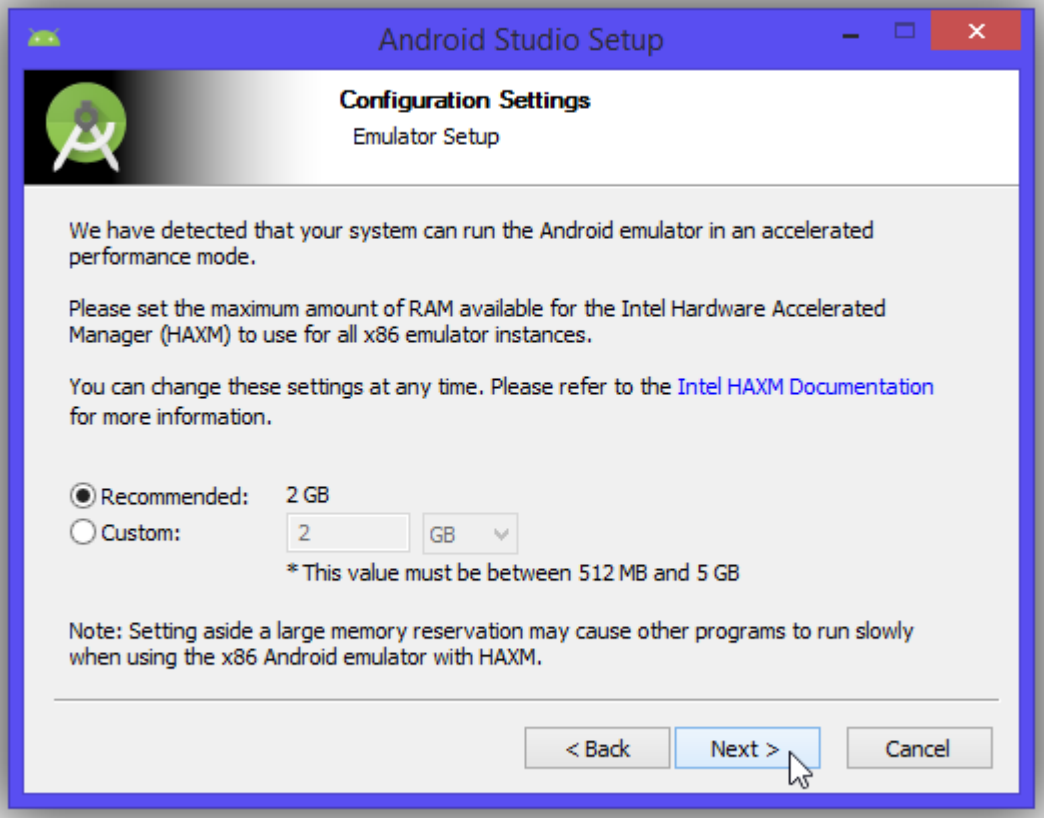
Installation d'Android Studio (2/3)

- Indiquez le répertoire d'installation de Android Studio et le SDK.
- Préférable de laisser le répertoire d'installation par défaut.



Installation d'Android Studio (3/3)

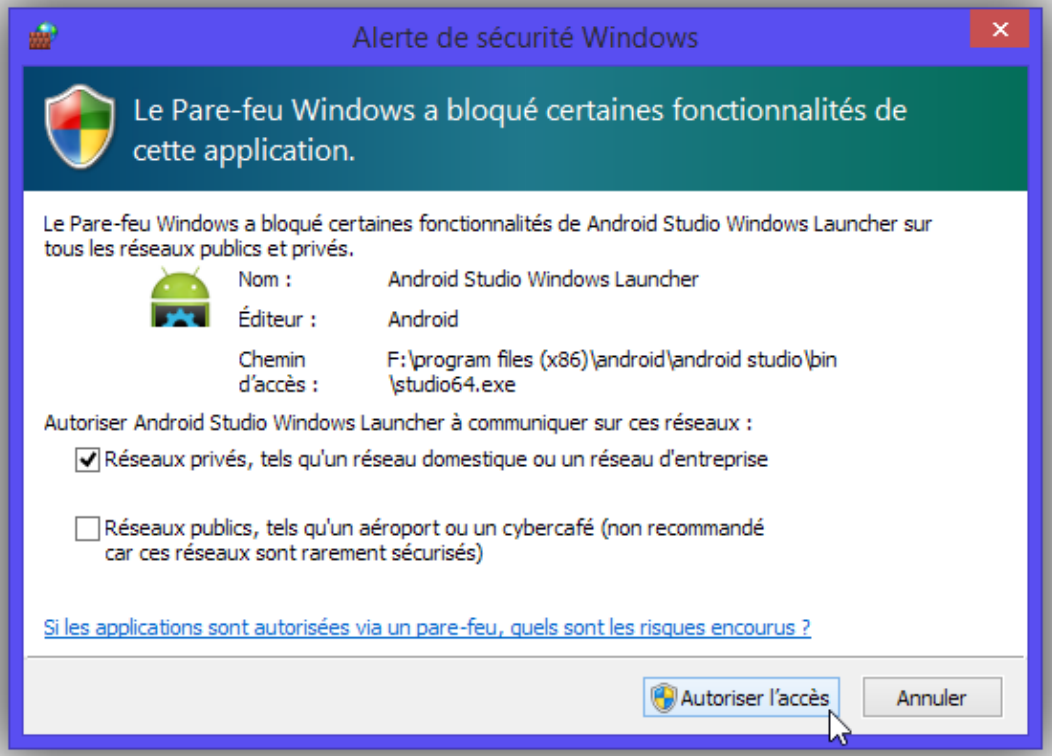
- Si votre ordinateur est puissant, cet écran vous propose d'indiquer quelle quantité de RAM vous souhaitez accorder à l'émulateur Android.
- La valeur par défaut est 2 Go mais vous pouvez indiquer une valeur en choisissant Custom.



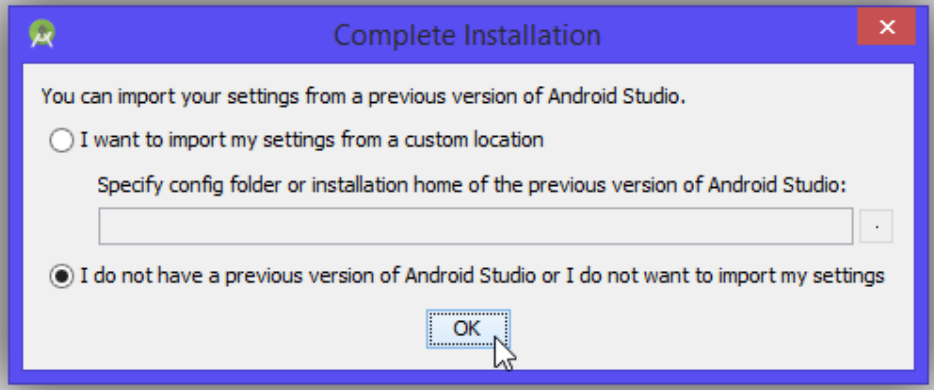
Université Internationale de Casablanca

Démarrage d'Android Studio (1/2)

- Au premier démarrage, le Pare-feu demande si Android Studio a l'autorisation d'accéder à internet.

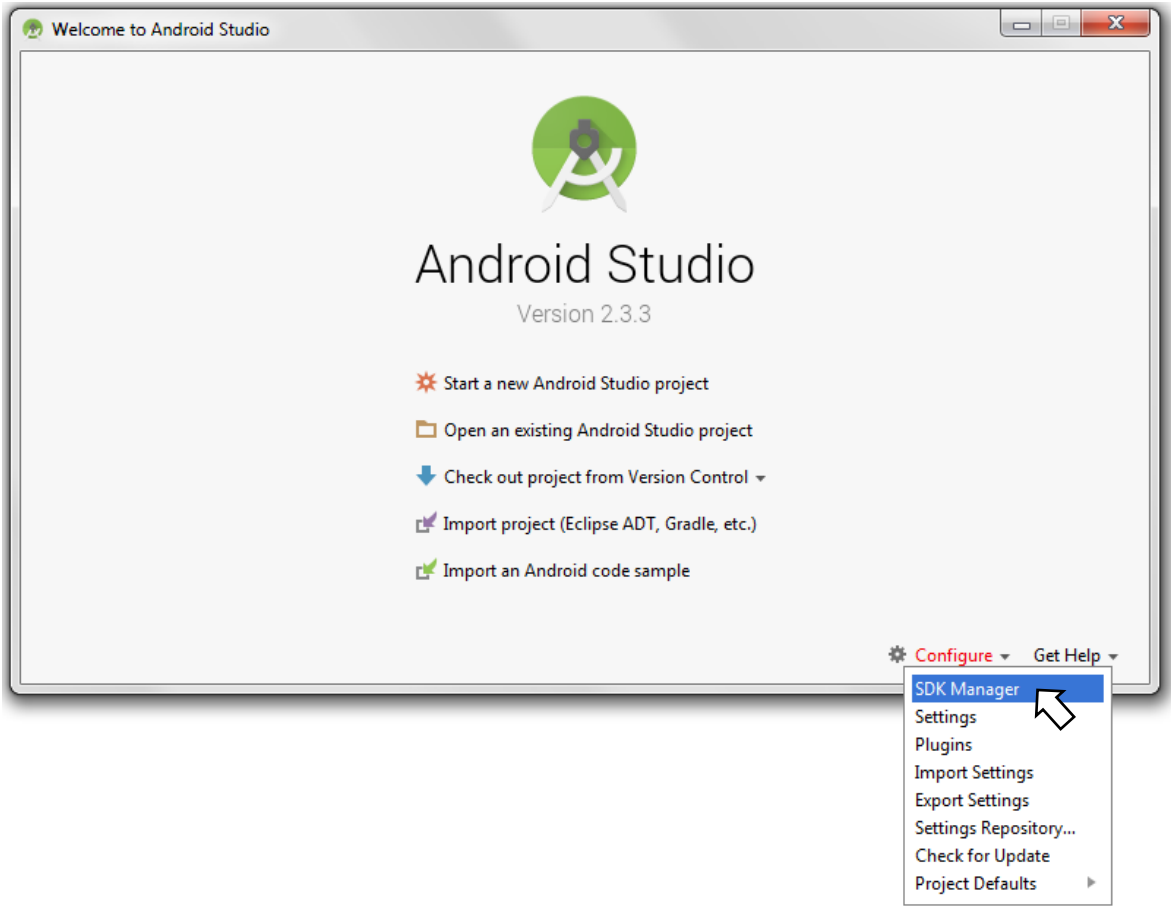


- Une boîte de dialogue s'affiche demander si vous aviez déjà une version d'Android Studio installée précédemment. Si ce n'est pas le cas, sélectionnez la seconde option



Démarrage d'Android Studio (2/2)

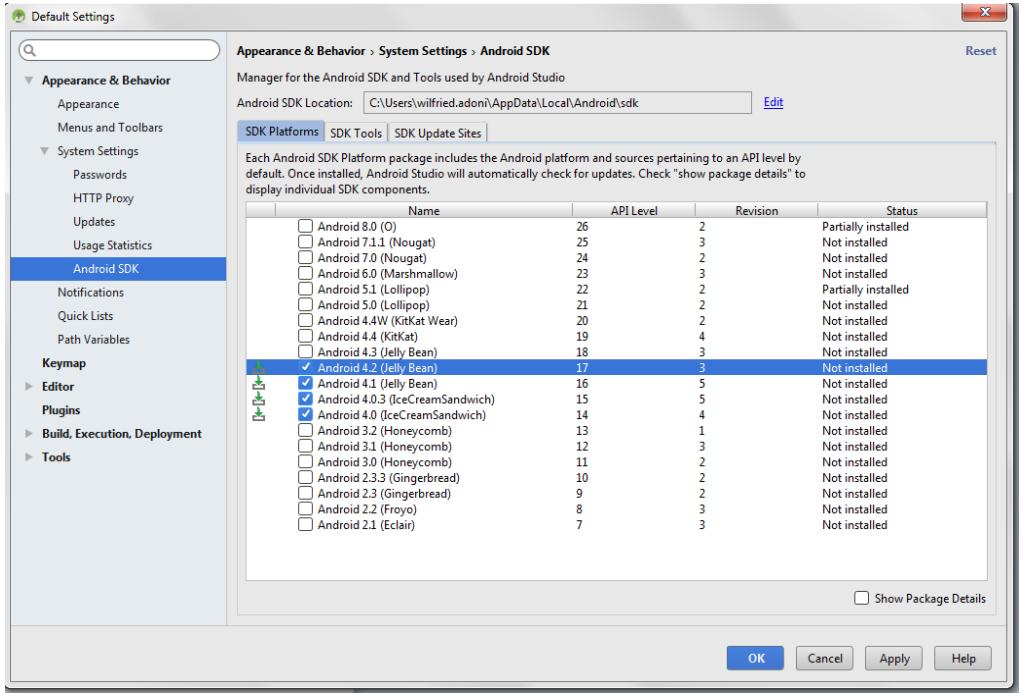
- Fenêtre principal d'Android Studio permet:
  - ❑ La création et l'importation projet
  - ❑ La configuration du Kit de developpement
- Pour configurer le SDK Android: sélectionner SDK Manager dans l'onglet déroulant « Configure ».

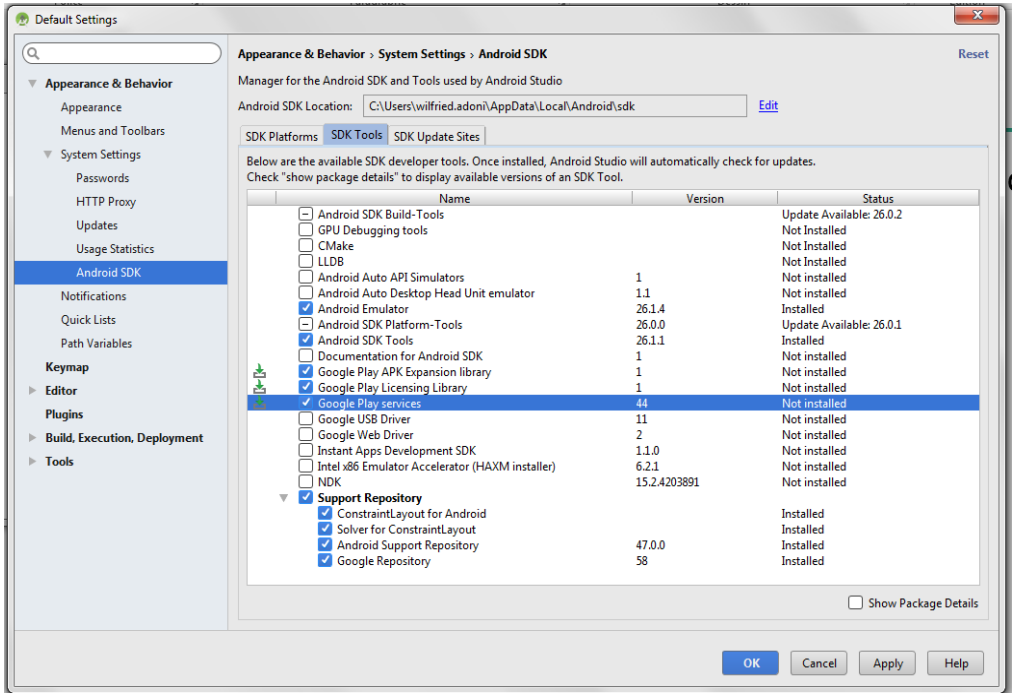


Université Internationale de Casablanca

Configuration du Kit de développement

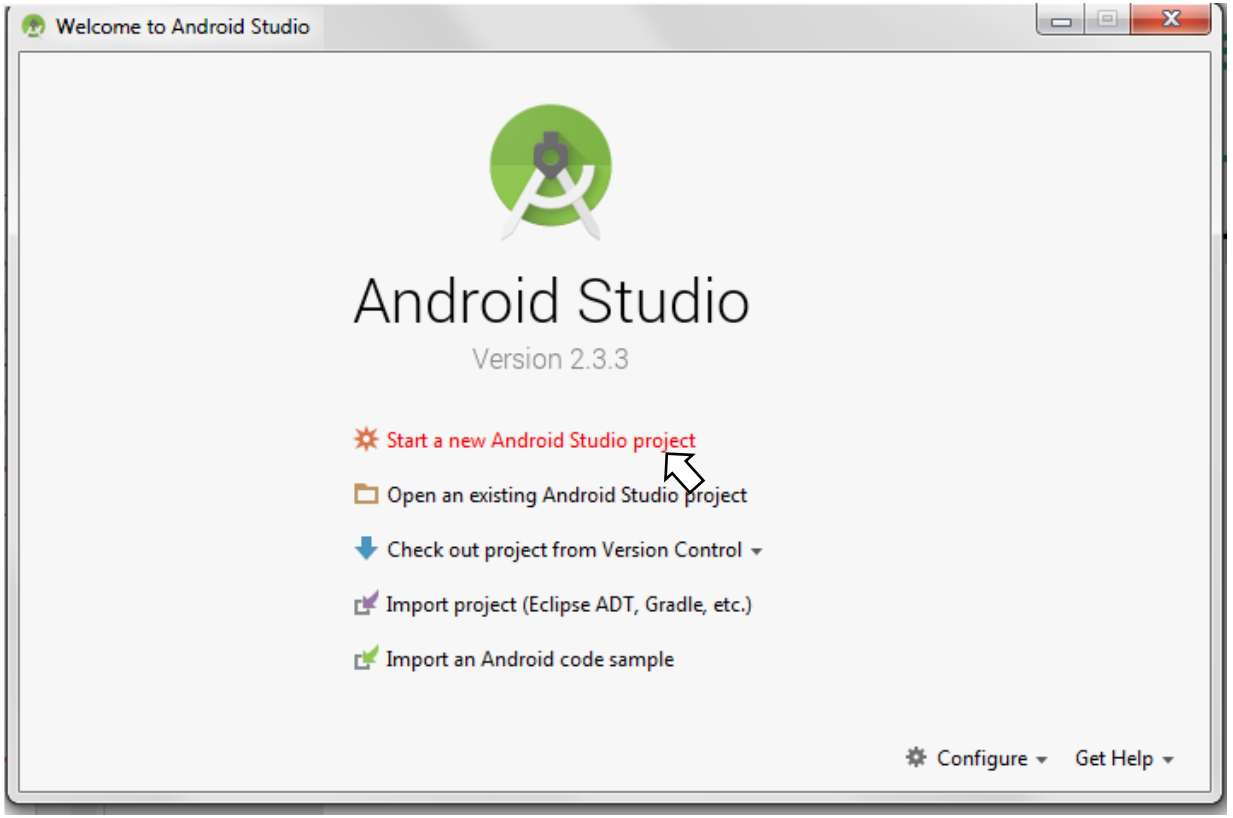
- Sélectionner et télécharger les outils nécessaires en fonction de la version Android cible.
- L'installation des packages se fait de façon automatique.
- Si vous installez tous ces paquets, vous aurez besoin de 1 Go sur le disque de destination.





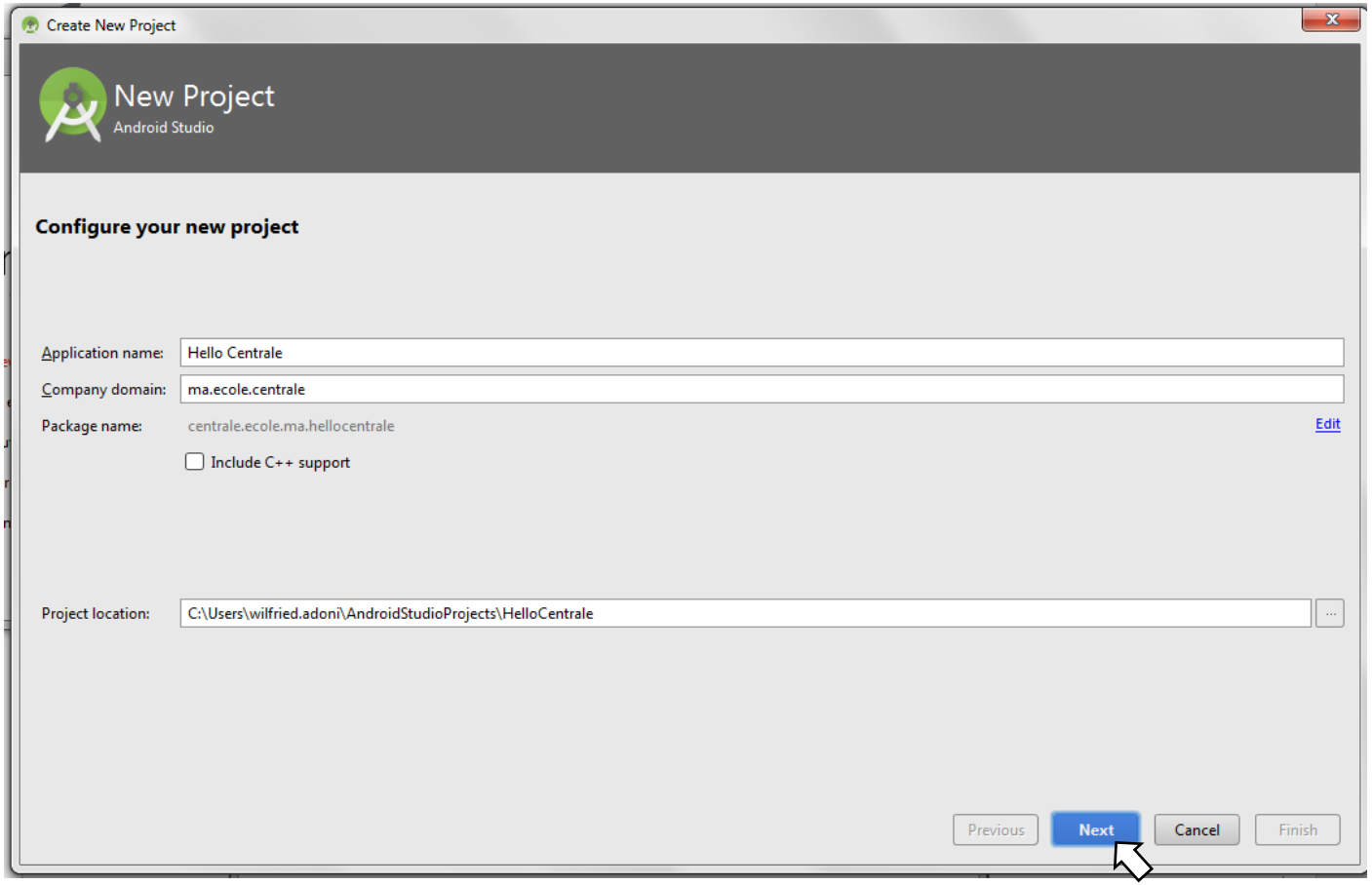
Création d'un projet Android(1/6)

- Pour créer un nouveau projet, cliquer sur « *Start a new Android Studio project* » pour ouvrir l'assistance de projet.



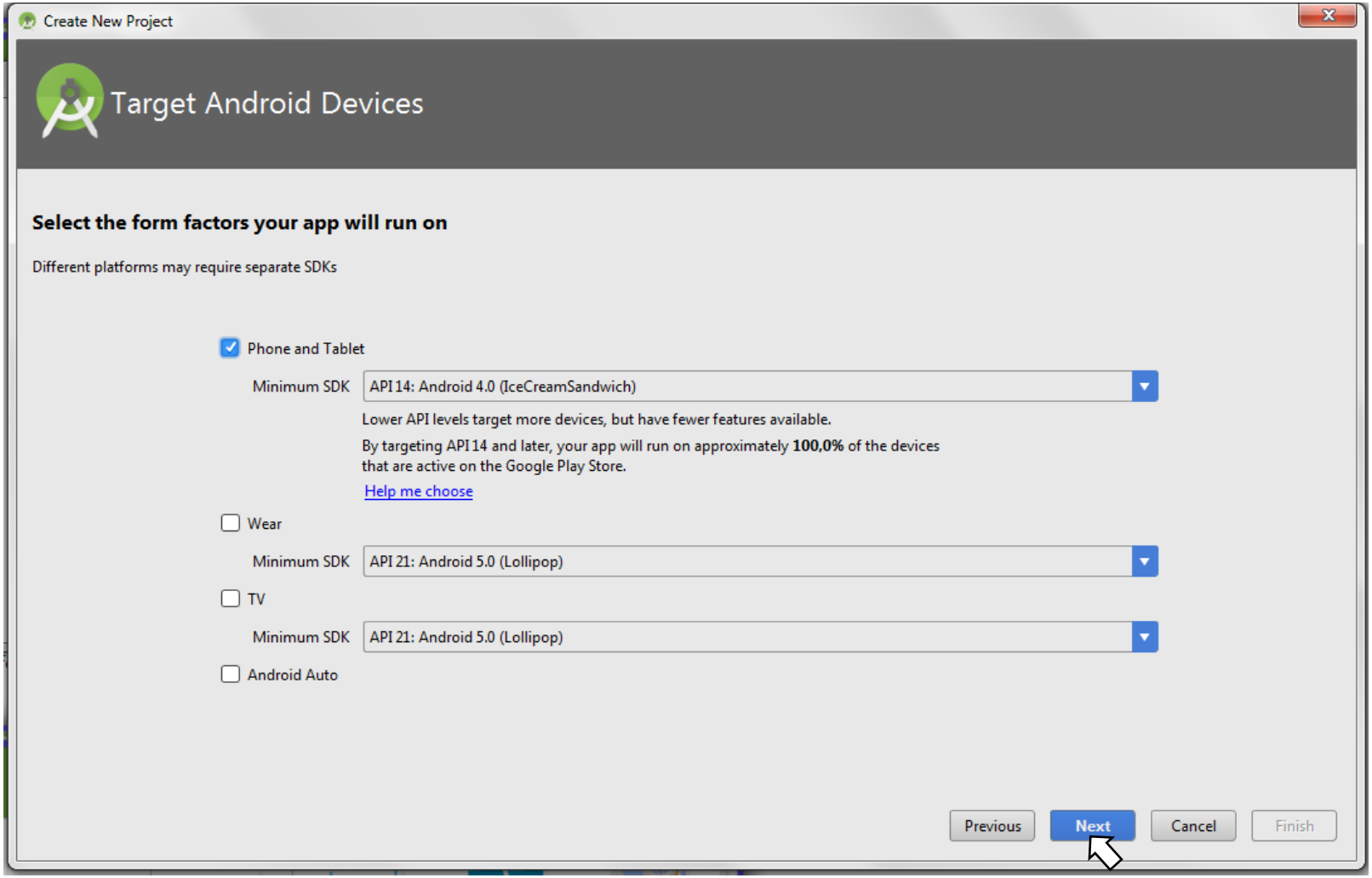
Création d'un projet Android (2/6)

- Une nouvelle fenêtre s'ouvrira. Elle contient trois champs:
  - **Application name** : il s'agit du nom qui apparaîtra sur l'appareil et sur Google Play
  - **Company domain** : nom de domaine de son entreprise permettant de déduire un package name.
  - **Project location** : indique l'emplacement où les fichiers de votre projet seront créé.



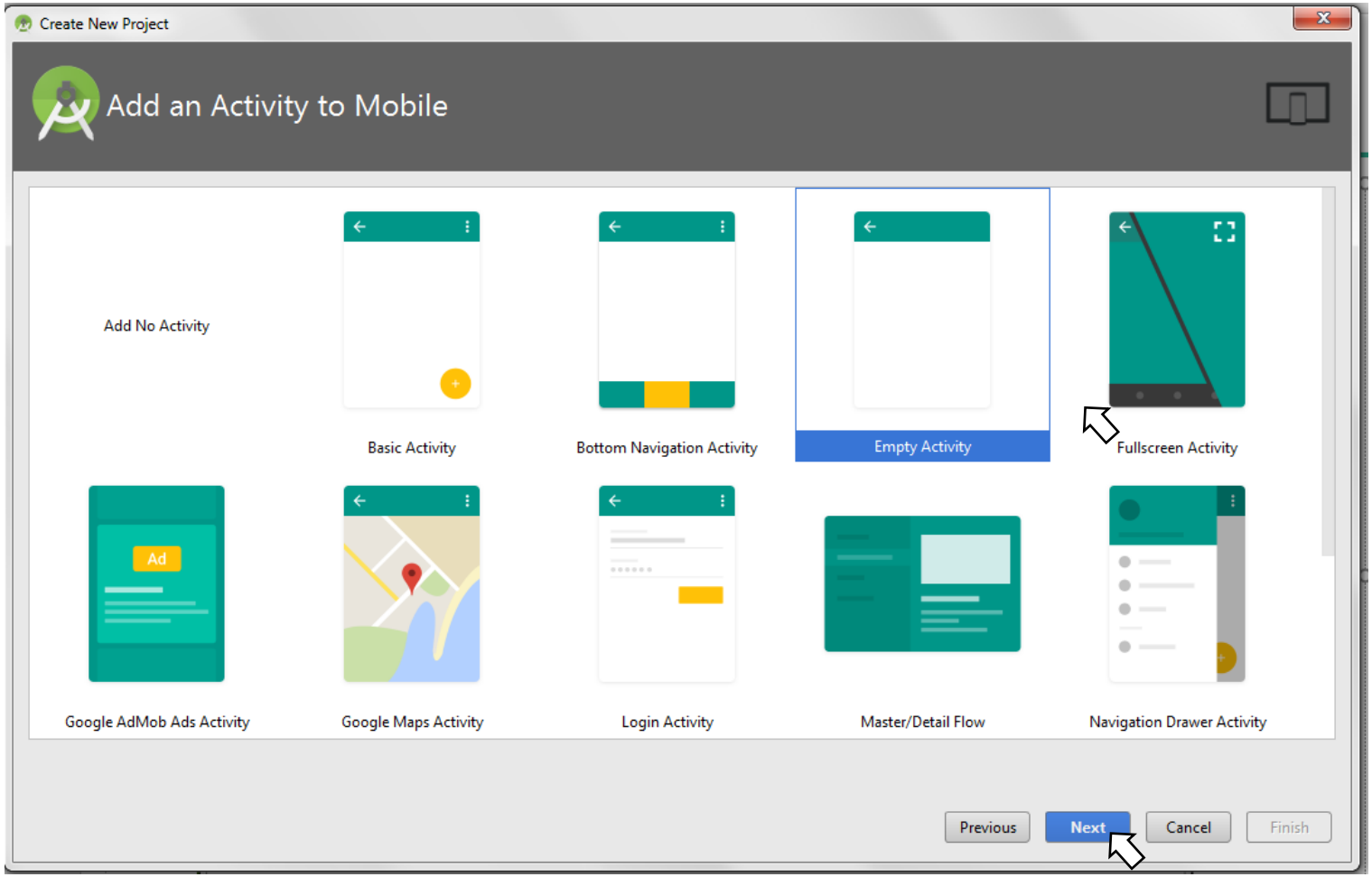
Création d'un projet Android (3/6)

- Cet écran permet de sélectionner le matériel cible de l'application, ainsi que la version d'Android minimum que doit utiliser ce matériel.



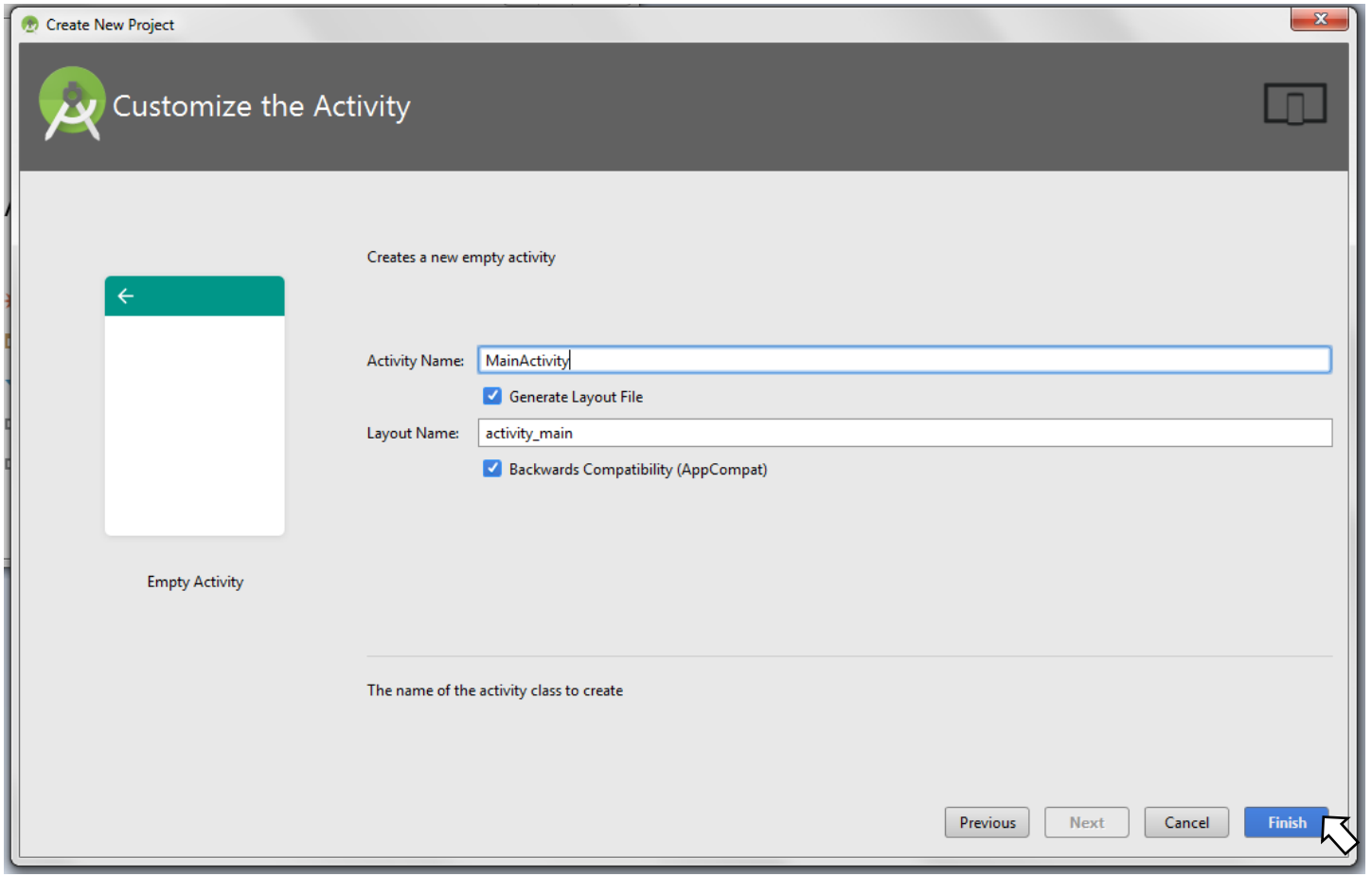
Création d'un projet Android (4/6)

- Choisir l'activité principale qui va contenir les éléments graphiques.
- Une activité est une brique fondamentale dans l'interaction avec l'utilisateur.



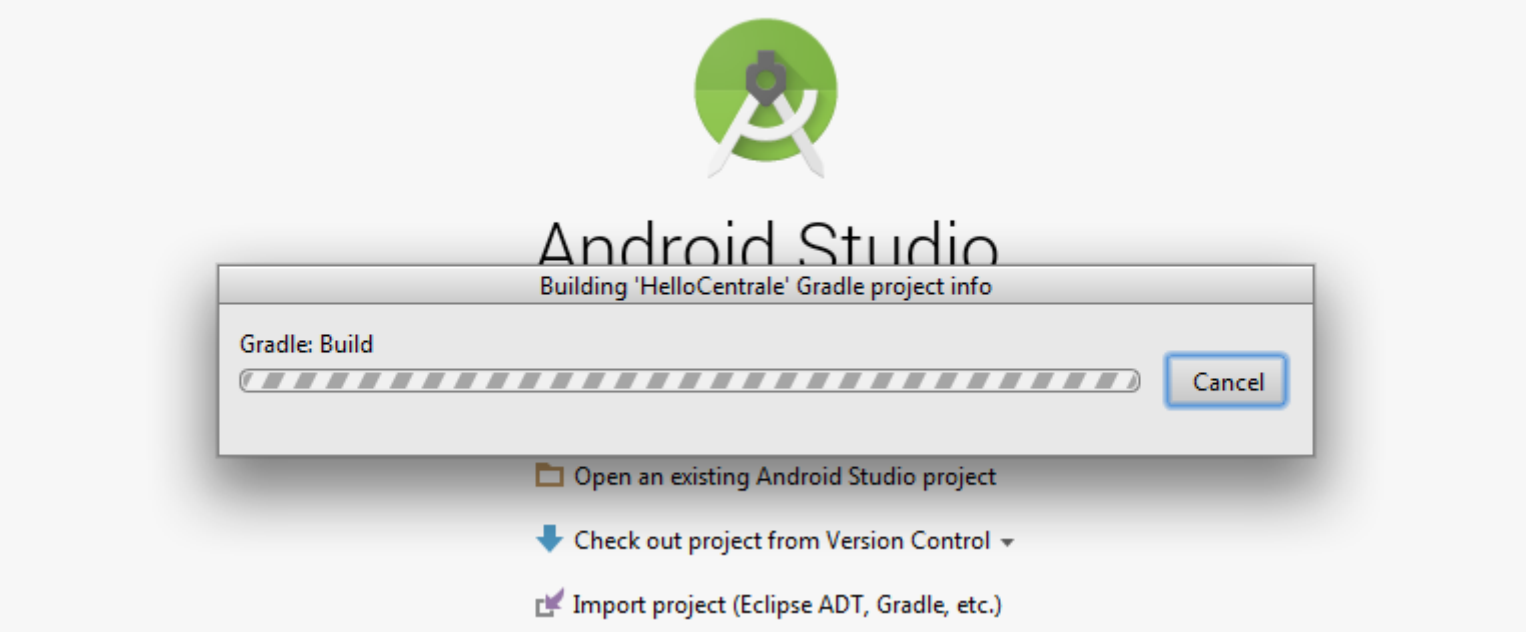
Création d'un projet Android (5/6)

- Cette fenêtre permet la configuration de l'activité principale. Elle contient deux champs:
  - **Activity Name** : nom de l'activité, par défaut le nom proposé est « *MainActivity* »
  - **Layout Name** : nom du fichier layout contenant les éléments graphiques associés à l'activité.



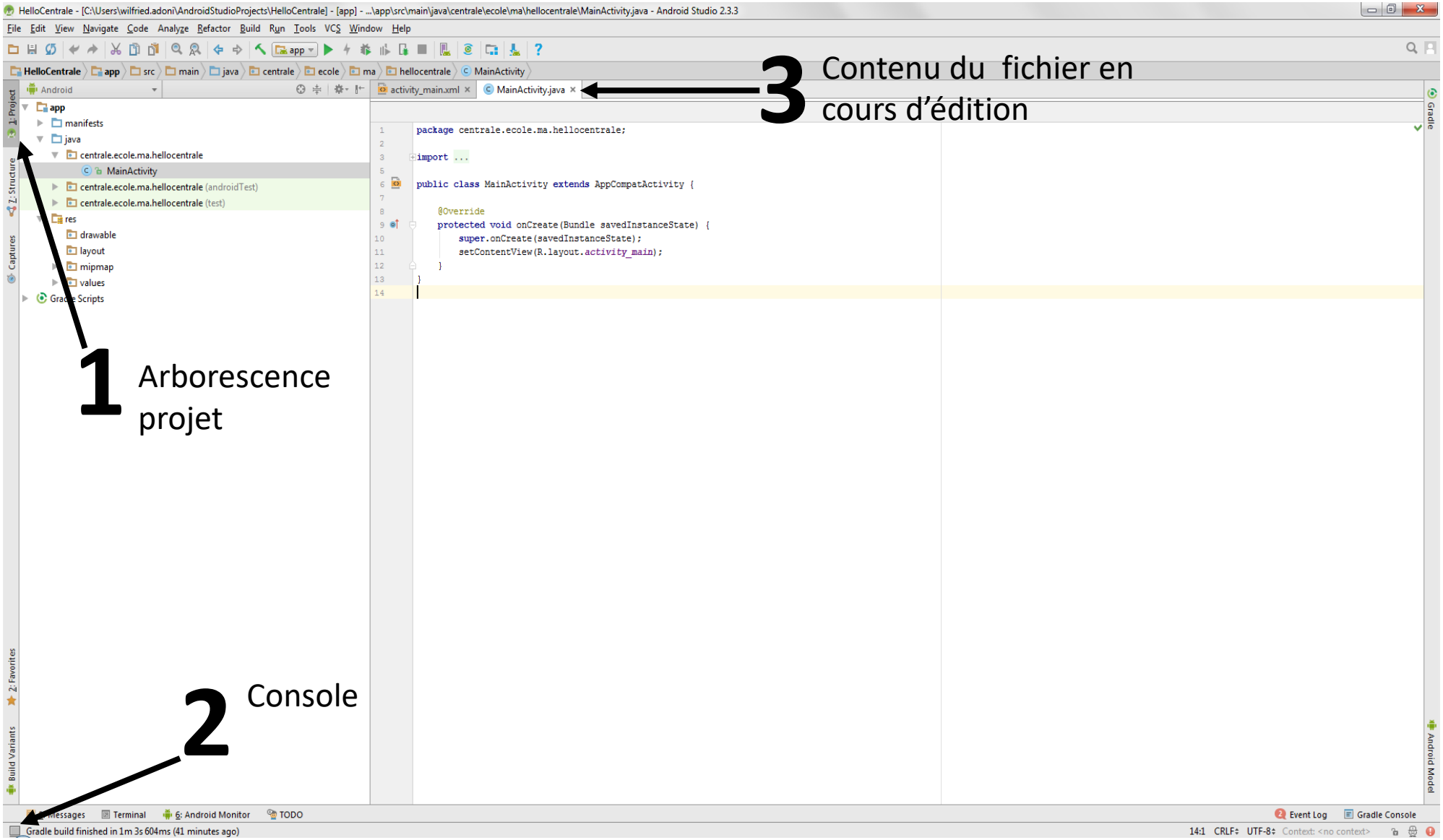
Création d'un projet Android (6/6)

- La fenêtre intitulée Building « *HelloCentrale' Gradle project info* » s'affiche, avec une barre de chargement.
- Patientez, le temps que les fichiers nécessaires au bon fonctionnement de votre projet soient téléchargés.
- L''écran principal d'Android Studio apparait après téléchargement des fichiers.



Université Internationale de Casablanca

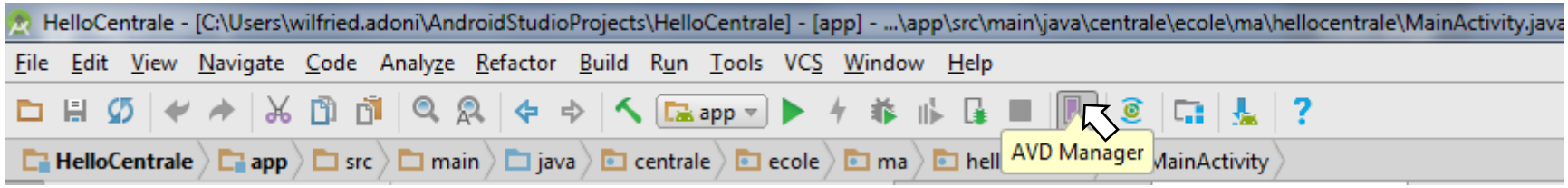
Découverte d'Android Studio



Université Internationale de Casablanca

Création d'un émulateur (1/5)

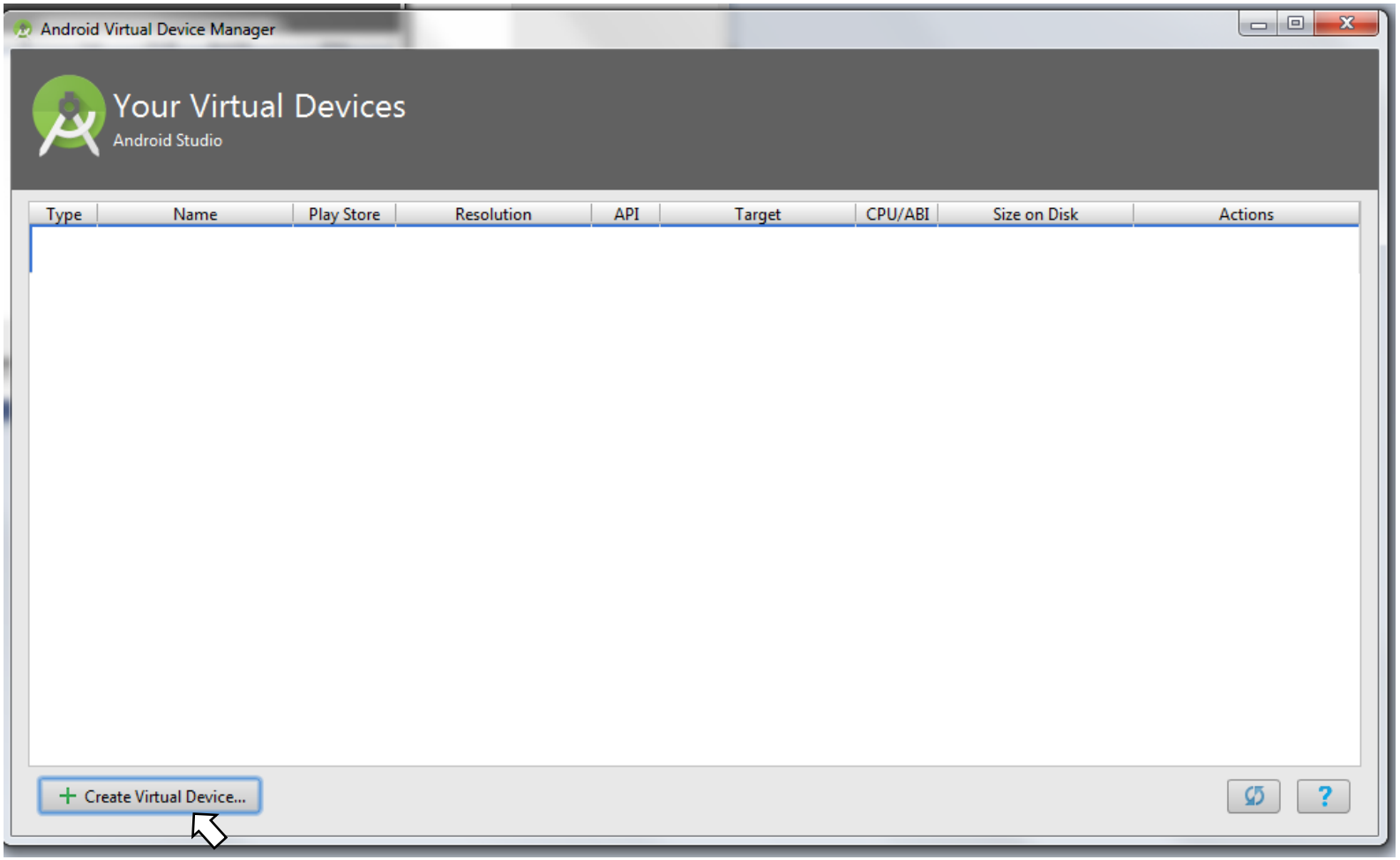
- Un émulateur est un téléphone virtuel permettant de tester l'exécution de l'application.
- Le téléphone virtuel fonctionne de la même manière qu'un téléphone physique à l'exception d'effectuer des appels.
- Dans l'onglet de gestion du SDK, cliquer sur l'icone « AVD Manager », comme le montre la figure ci-dessous:



- **AVD Manager**: **A**ndroid **V**irtual **D**evice, permet créer, gérer et supprimer un émulateur.

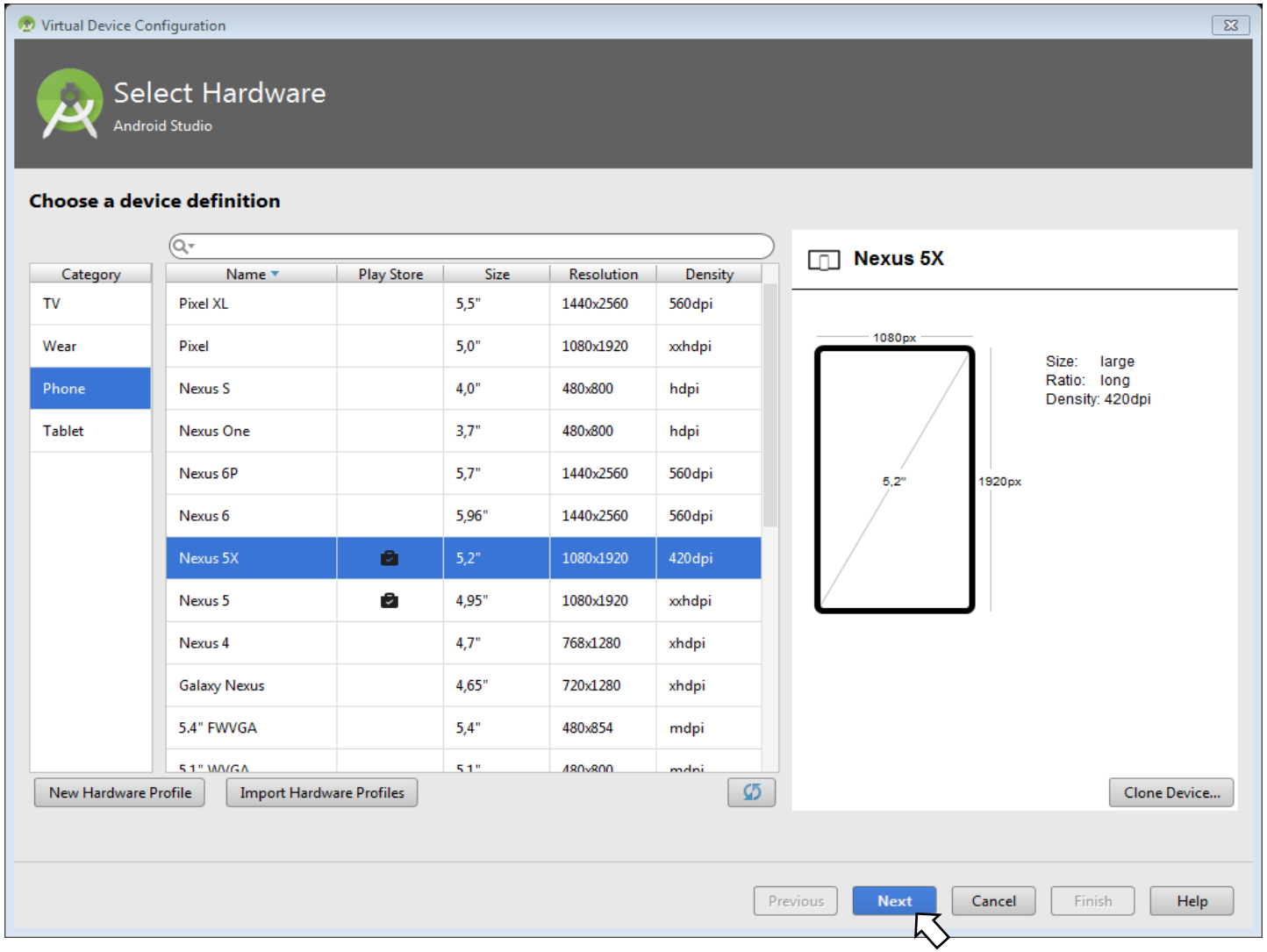
Création d'un émulateur (2/5)

- Cette fenêtre s'affiche, cliquer sur le bouton « *Create Virtual Device* »



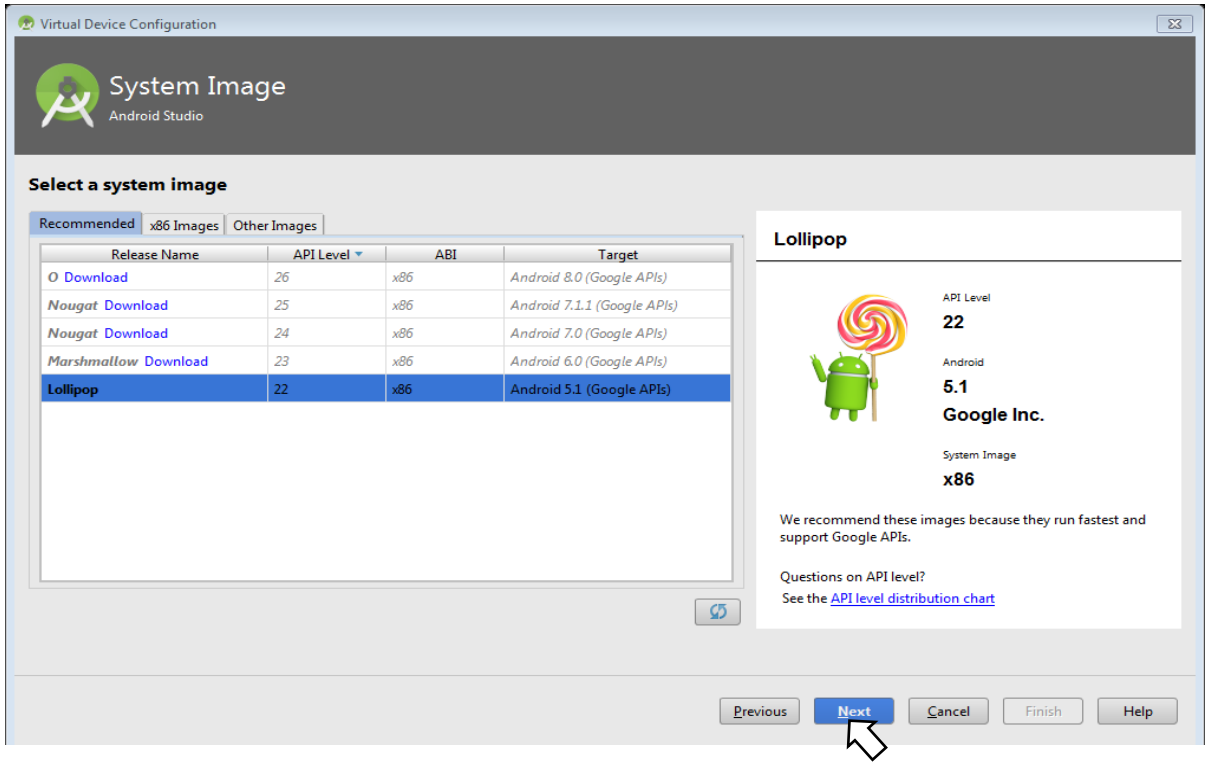
Création d'un émulateur (3/5)

- Sélectionner « Phone », puis choisir un téléphone suivant le model ou la taille de l'écran.



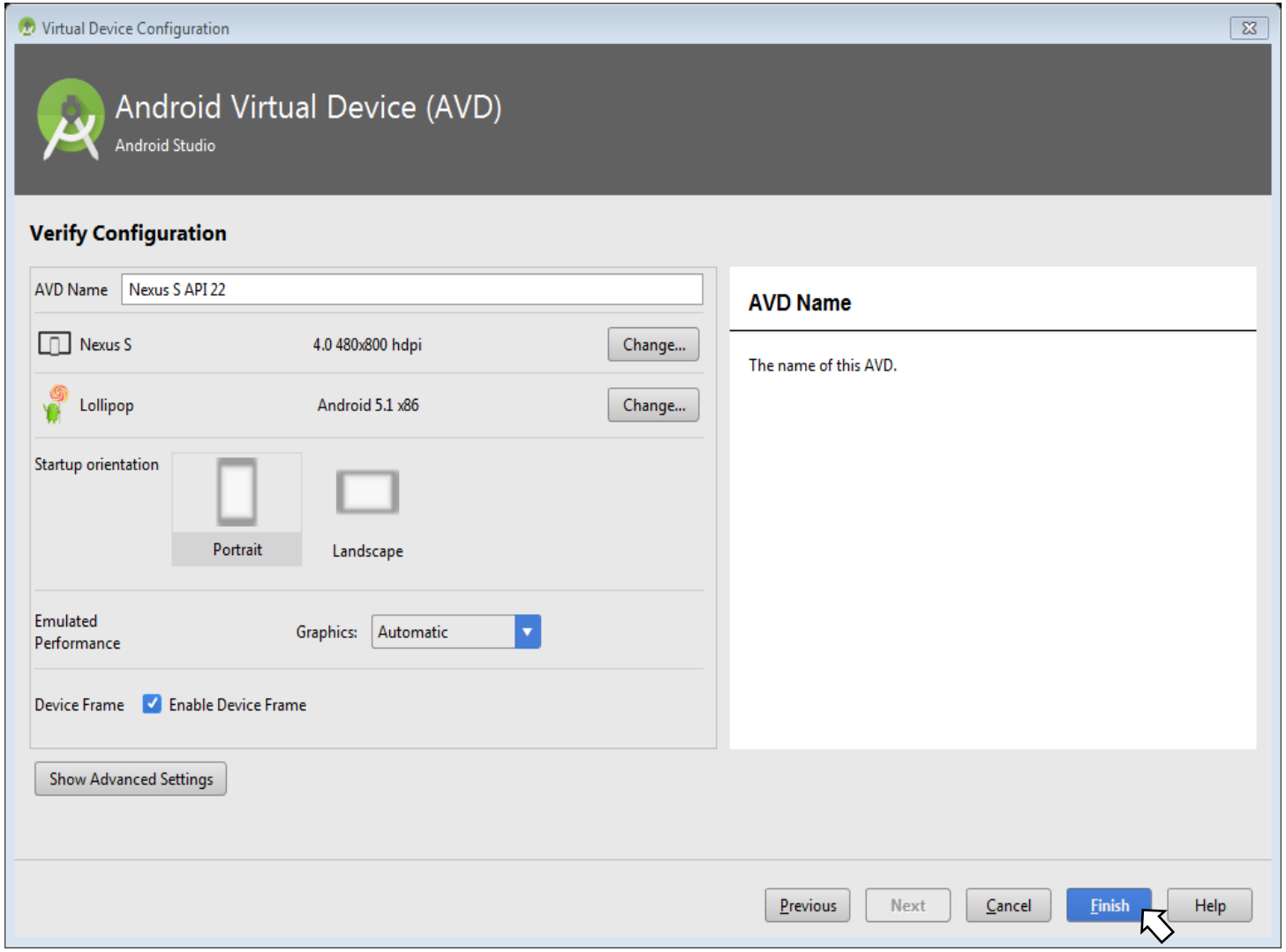
Création d'un émulateur (4/5)

- L'image système correspond à la version d'Android à installer sur l'équipement virtuel. Il est préférable rester dans l'onglet « Recommended ».
- Il est possible que vous deviez installer une image système pour pouvoir aller plus loin : cliquez pour cela sur le lien Download à côté de l'image à installer.



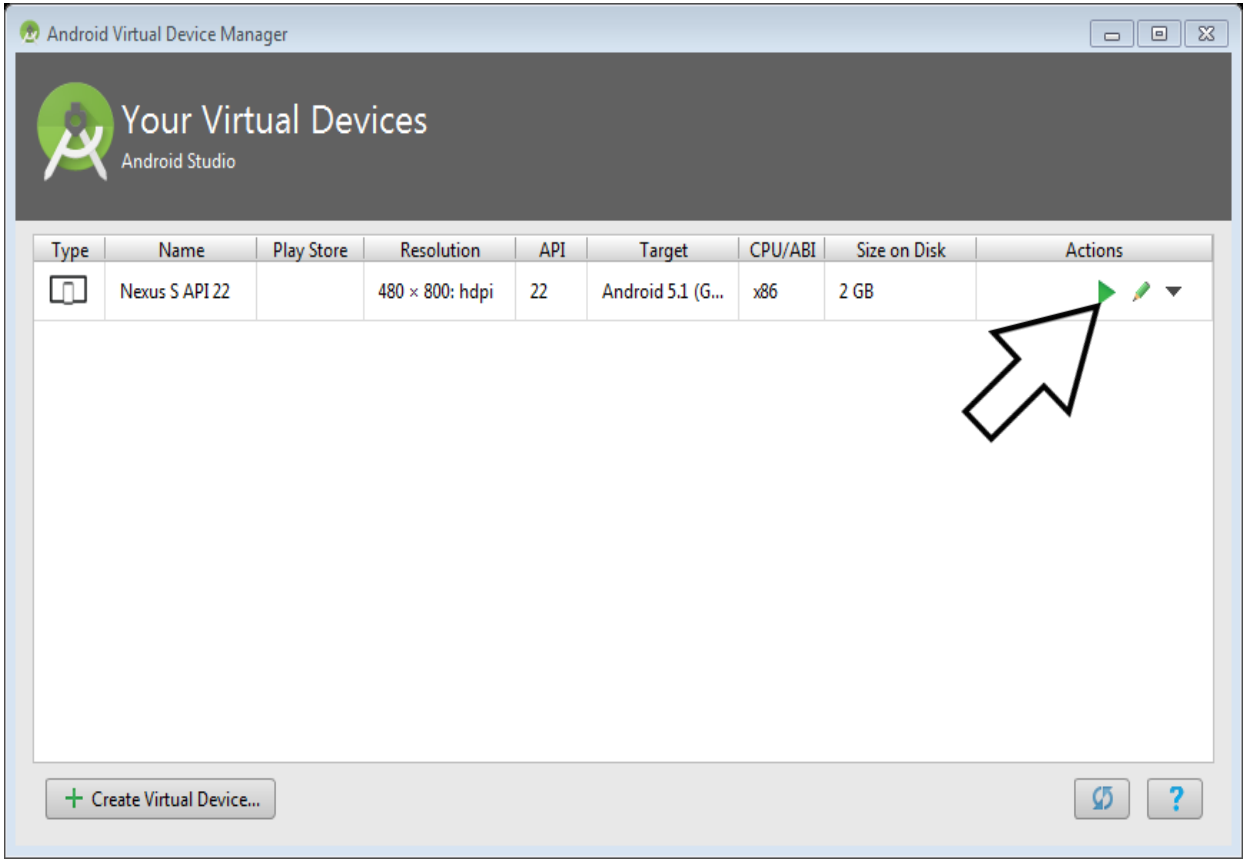
Création d'un émulateur (5/5)

- Dans le champ « *AVD Name* » , saisir le nom à attribuer au téléphone.



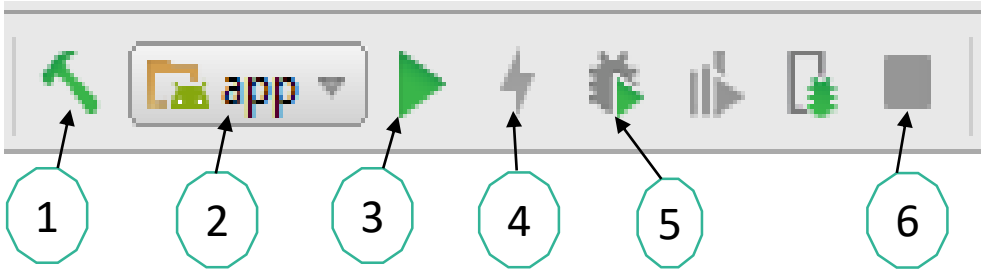
Lancement de l'émulateur

- Le nouvel équipement virtuel apparait dans la fenêtre.
- Sélectionnez l'équipement puis cliquez sur le bouton « play ». L'émulateur devrait se lancer, et après le chargement d'Android au démarrage.

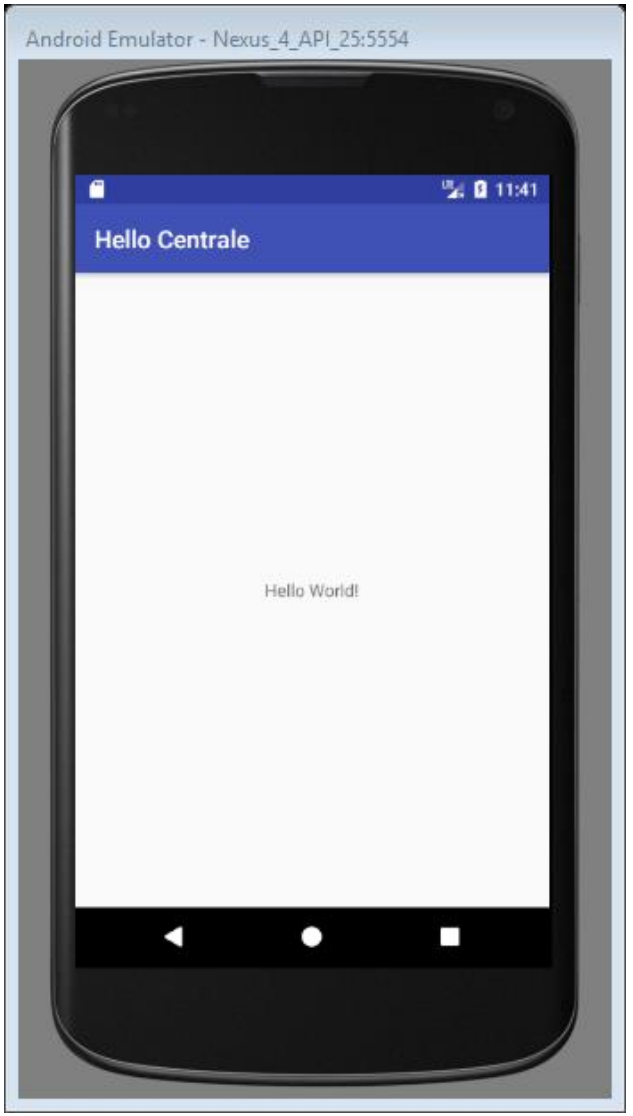


Exécution de l'application

- Cliquer sur le bouton « *play* » en vert pour exécuter l'application.



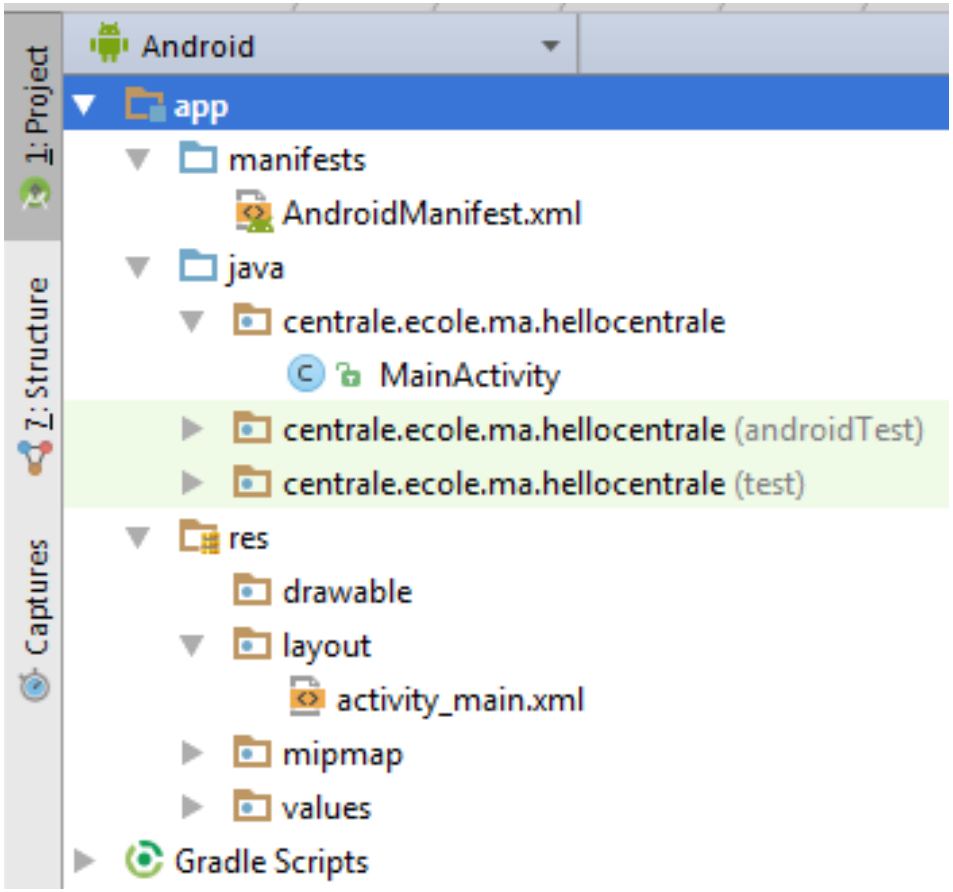
1. Compile le code source du projet
2. Choisir une configuration pour l'exécution du projet
3. Exécute normalement l'application
3. Applique les modifications effectuées sur le code sans réexécuter tous le projet
5. Debug l'application, avec des breakpoints
6. Stop l'exécution de l'application



Structure d'un projet Android

- Les trois répertoires principaux d'Android sont **manifests**, **java** et **res**.

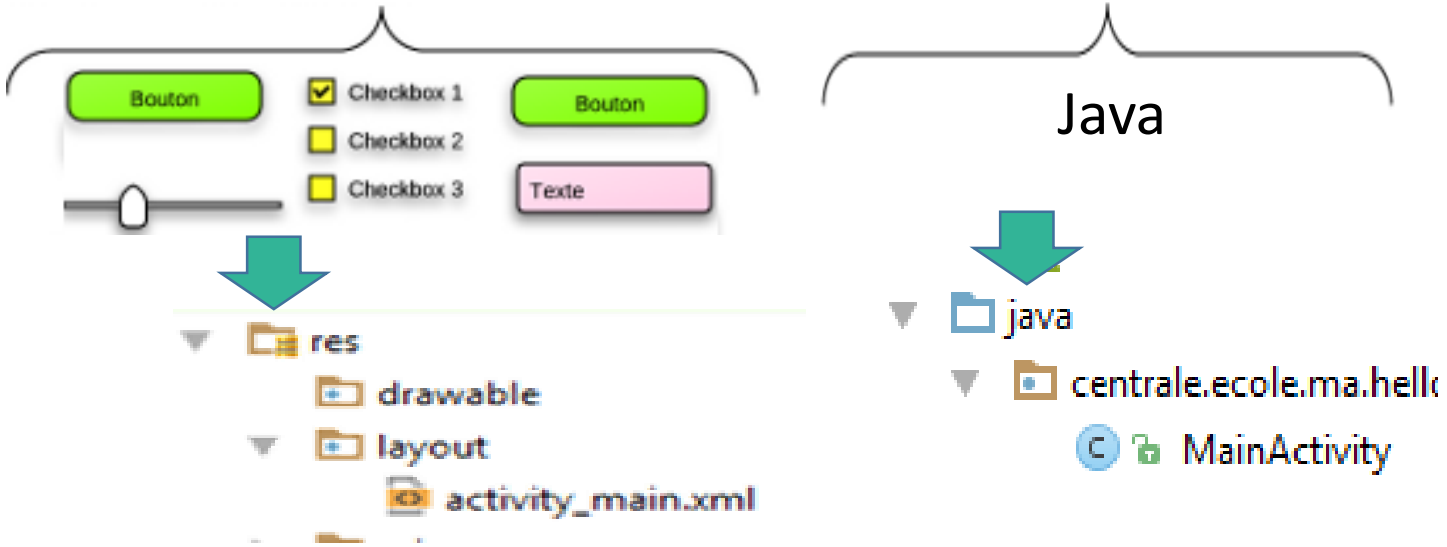
  - Le répertoire **manifests**: contient un seul fichier : le fichier ***AndroidManifest.xml***. Ce fichier contient l'ensemble des informations de l'application.
  - Le répertoire **res** : contient toutes les ressources de l'application.
  - Le répertoire **java** contient l'ensemble du code source Java de l'application.



Activités

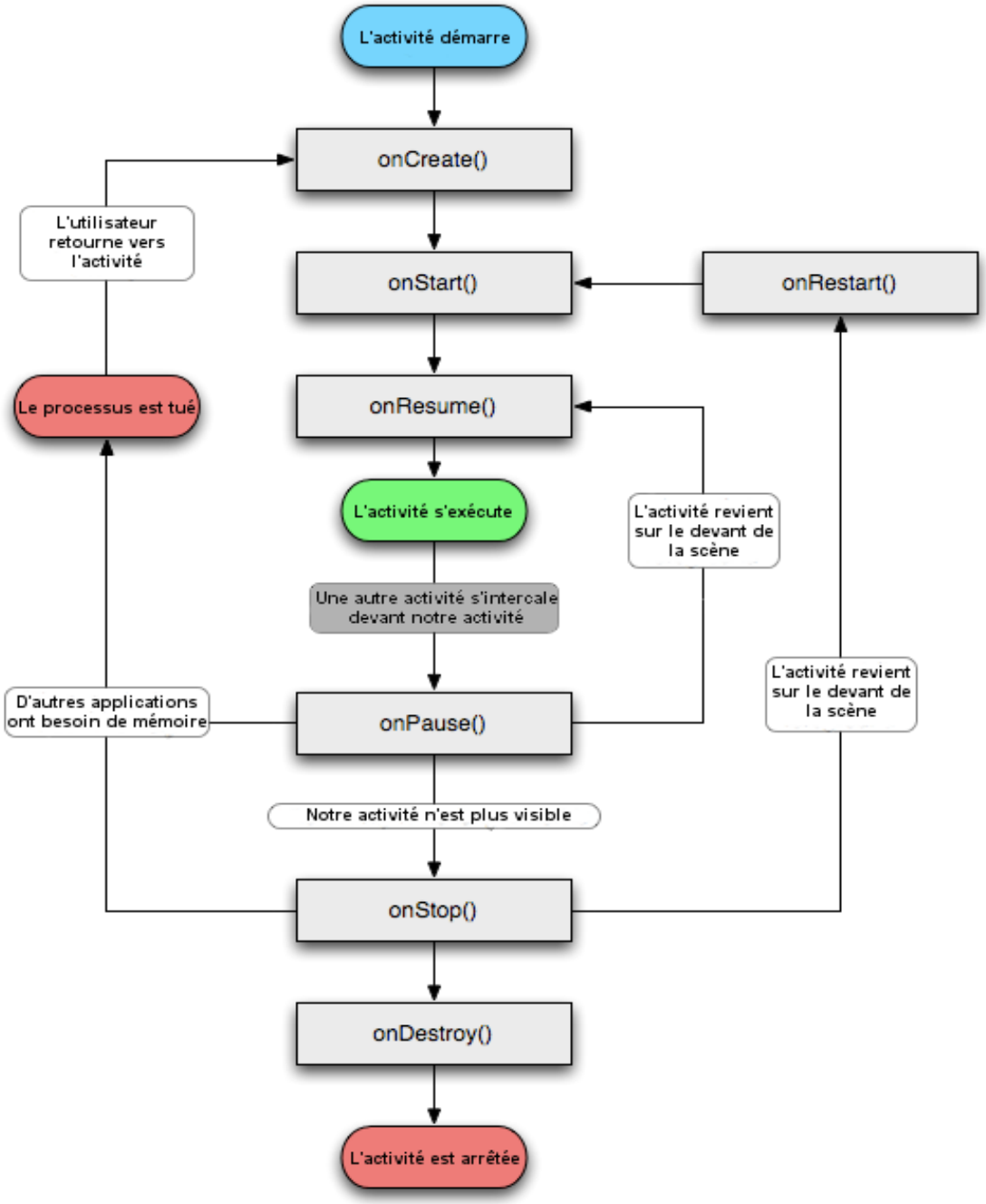
- Une **activité**, ou **Activity** en anglais, est une brique fondamentale d'Android. C'est le point d'entrée d'une application Android.
- Une activité a pour rôle principal d'interagir avec l'utilisateur.
- C'est une classe Java, qui hérite obligatoirement de la classe Android **Activity** ou **AppCompatActivity**.
- Un activité présente différents états.

Activity = Interface graphique + Traitement



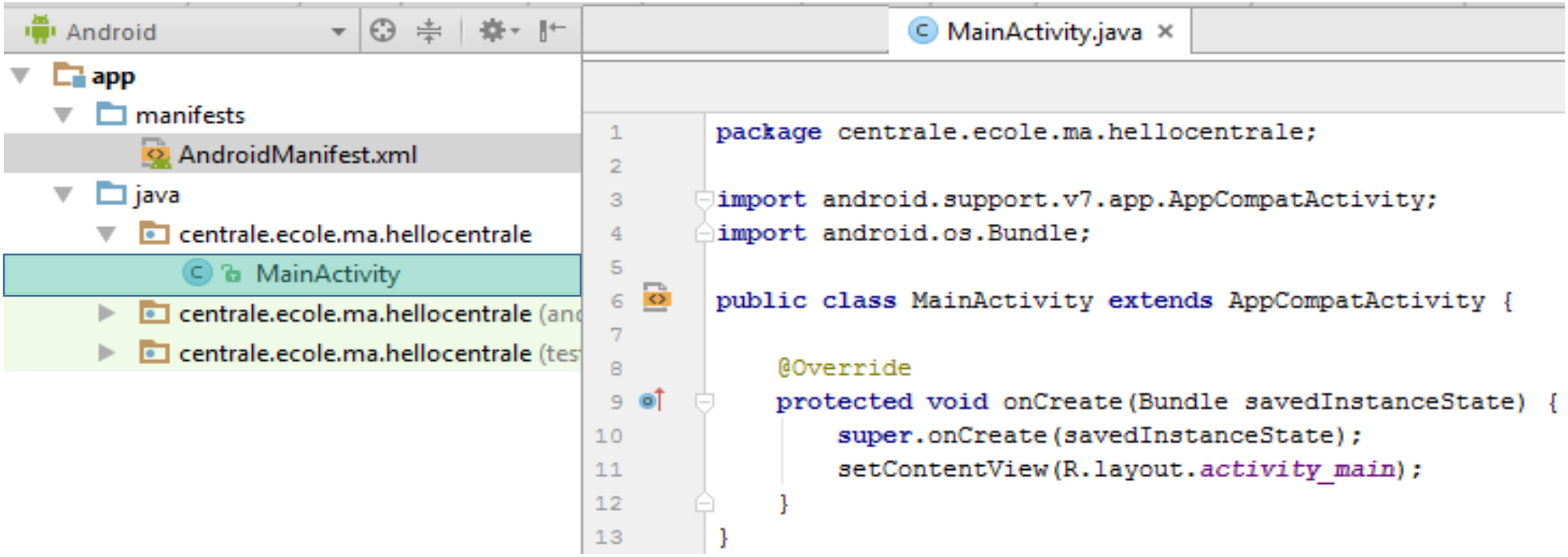
Université Internationale de Casablanca

Cycle de vie d'une activité



Université Internationale de Casablanca

La classe Activity: MainActicity



- La méthode **void onCreate(Bundle saveInstanceState)** est la première qui est lancée au démarrage de l'activité.
- Le paramètre de type **Bundle** permet de sauvegarder l'état de l'activité.
- La méthode **setContentView(...)** permet d'indiquer l'interface graphique de l'activité. Elle prend en paramètre le fichier layout de l'interface graphique associée à l'activité.

Université Internationale de Casablanca

Le fichier layout

- Le fichier layout est un fichier **XML** utilisé pour l'édition des éléments graphiques à afficher à l'écran.
- Ce fichier XML est toujours stocké dans le répertoire **res/layout** du projet.
- Par convention, s'il est lié à une activité, il est toujours préfixé par activity, suivi du nom de l'activité, le tout en minuscule et séparé par un underscore (_).
- Ci-dessous le fichier ***activity_main.xml*** :

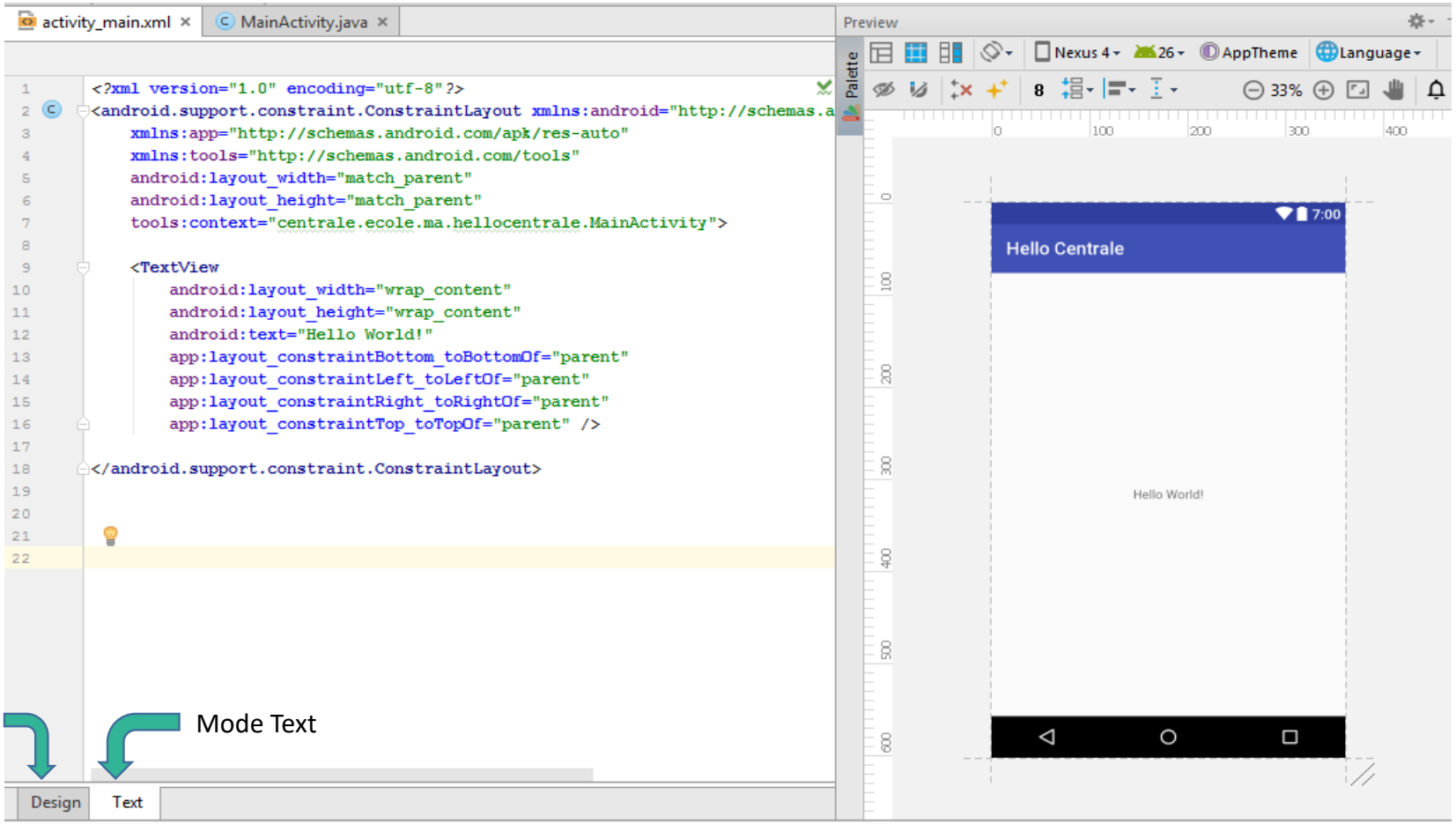
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.a
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="centrale.ecole.ma.hellocentrale.MainActivity">

    <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>
```

L'éditeur graphique

- Par défaut, Android Studio ouvre l'éditeur suivant deux modes:
  - **Mode Design** : permet de placer et configurer les différents éléments graphiques avec votre souris et générer automatiquement le contenu XML.
  - **Mode Text** : écriture manuelle du code source permettant d'avoir une meilleure maîtrise de l'ensemble.



Université Internationale de Casablanca

Structure du fichier layout XML

Conteneur

Composant

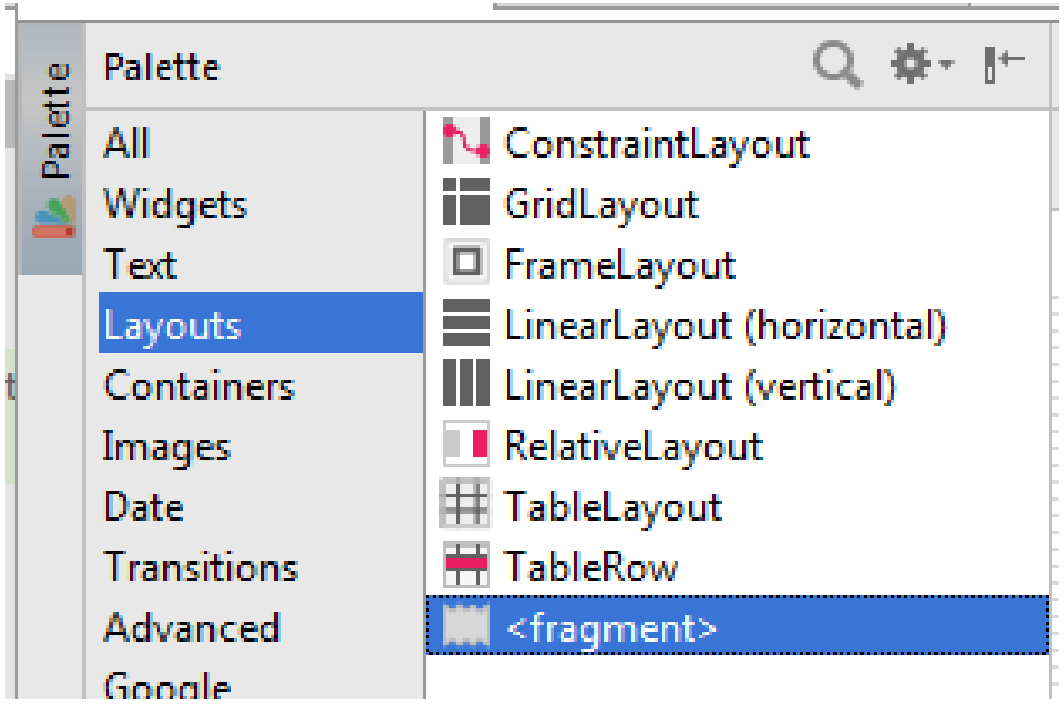
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.a
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="centrale.ecole.ma.hellocentrale.MainActivity">

    <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>
```

Les conteneurs

- Parmi les conteneurs proposés par Android, nous pouvons noter:
  - **FrameLayout** : permet de positionner les éléments les uns au dessus des autres.
  - **LinearLayout** : permet de positionner les éléments les uns à la suite des autres, dans le sens horizontal ou vertical.
  - **RelativeLayout** : permet de positionner les éléments les uns par rapport aux autres.
  - **ConstraintLayout** : comme le RelativeLayout, mais avec des règles de positionnement beaucoup plus puissantes.



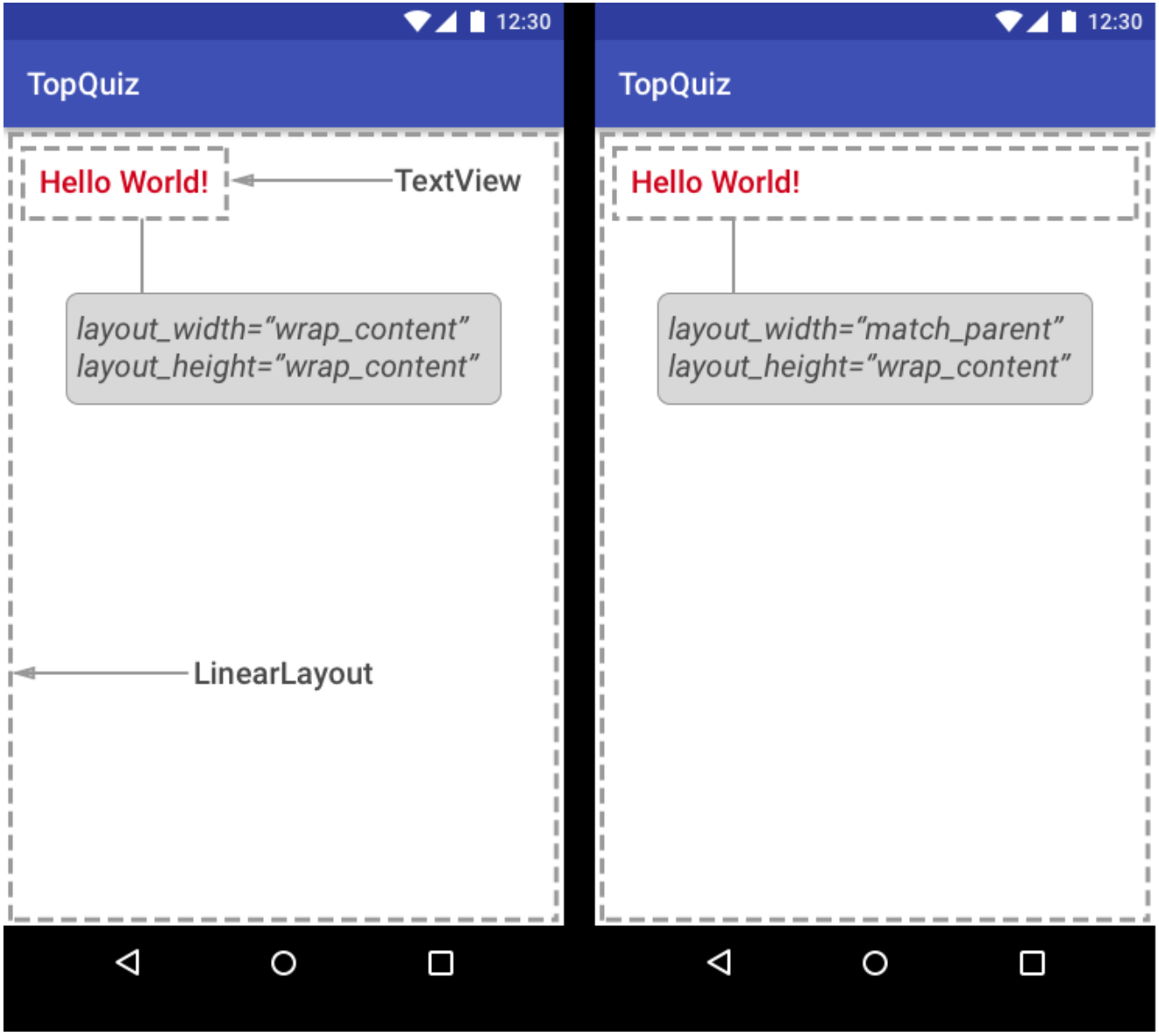
Les composants graphiques

- Parmi les composants proposés par Android, nous pouvons noter:
  - **TextView** : texte.
  - **EditText** : champ de saisie.
  - **Button** : bouton.
  - **CheckBox**: case à cocher.
  - **RadioButton**: bouton radio.
  - **Spinner**: liste déroulante.
  - **ListView**: liste de defilement
- Chaque composant possède plusieurs attributs. A minima, les deux attributs fondamentaux sont *layout_width* et *layout_height*. Ils permettent de déterminer comment afficher un élément au sein de son conteneur.

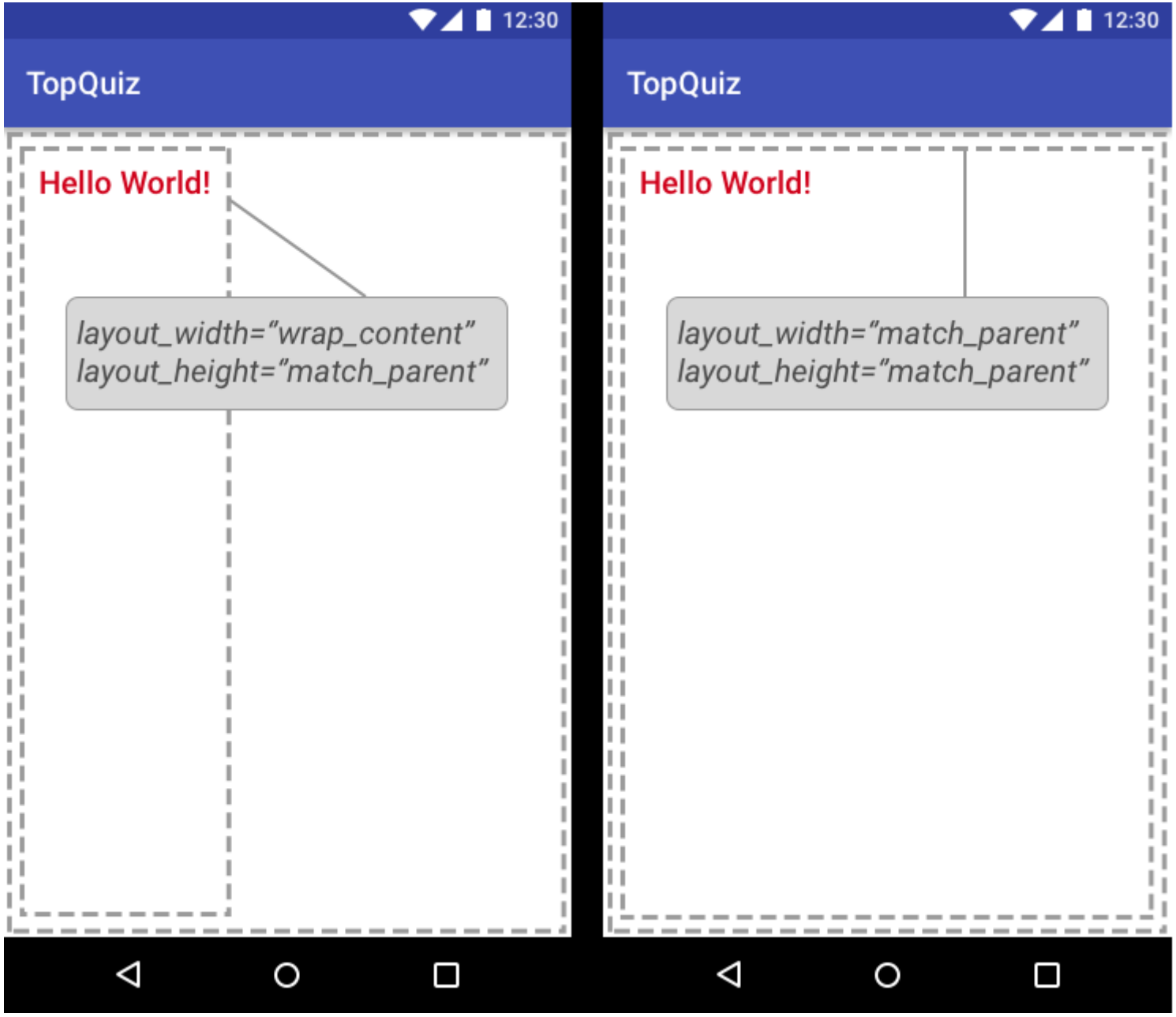
Université Internationale de Casablanca

Les attributs: layout_width et layout_height

- Utiliser pour gérer l'occupation de l'espace dans un conteneur



Les attributs: layout_width et layout_height



Les attributs: layout_gravity et gravity

layout_gravity

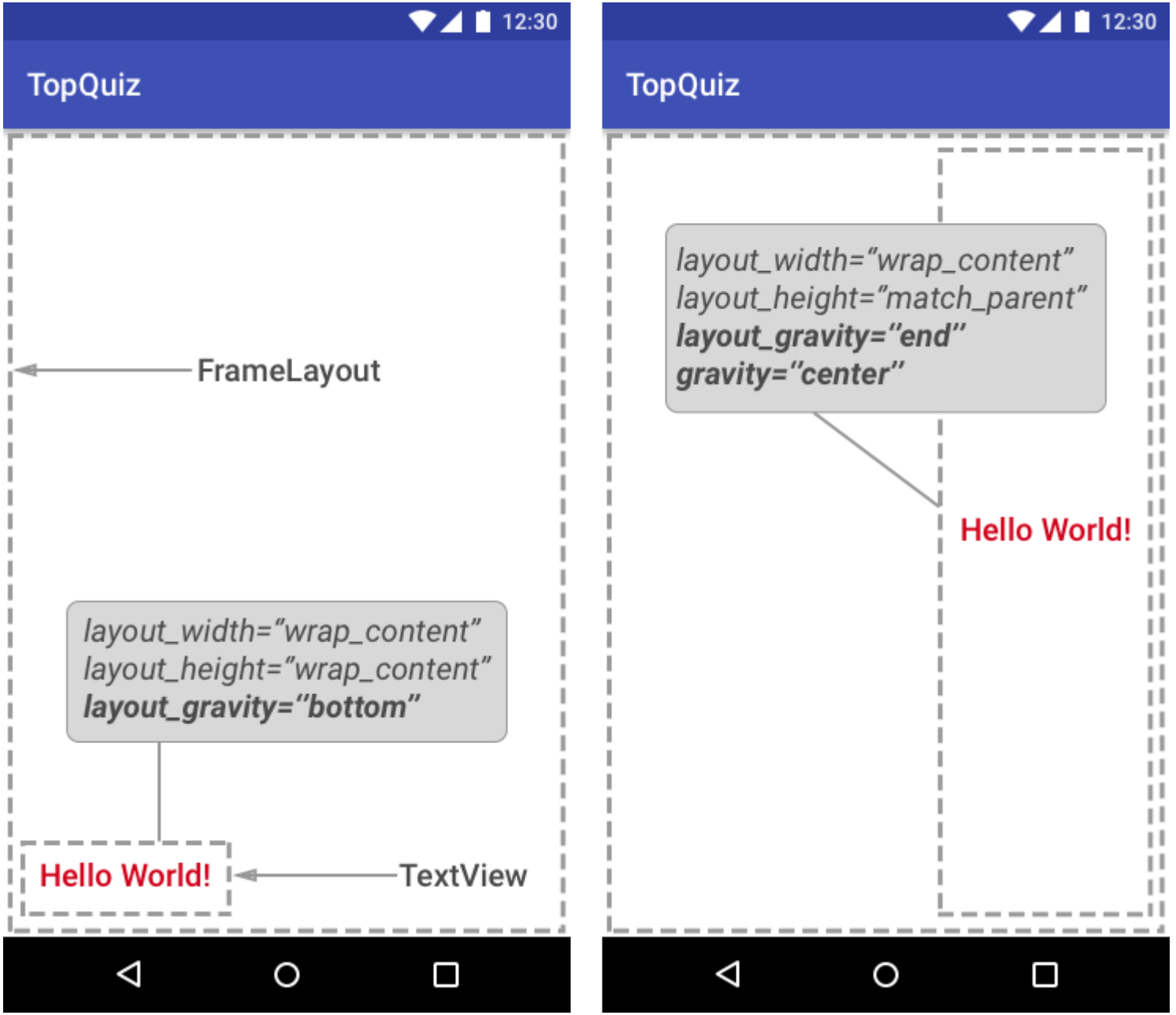
- Permet de gérer la gravitation, ou la position d'un élément par rapport à son conteneur.
- Les valeurs possibles sont nombreuses :
  - left, right
  - center, center_vertical, center_horizontal, etc.

gravity

- Permet de définir le positionnement d'un titre au sein d'un composant (bouton ou d'un champ texte par exemple).
- Les valeurs possibles sont nombreuses :
  - bottom, top, left, right, end
  - center, center_vertical, center_horizontal, etc.

Université Internationale de Casablanca

Les attributs: layout_gravity et gravity



Les attributs: layout_margin et padding

layout_margin

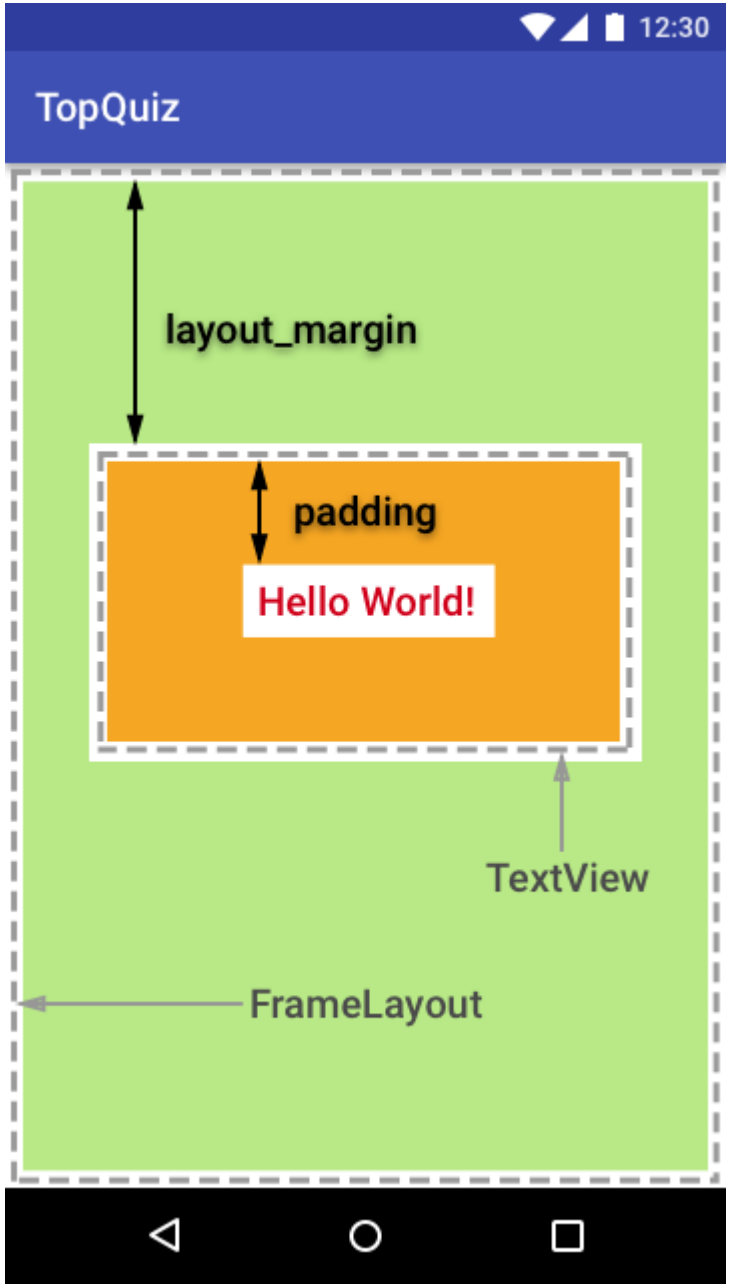
- Permet d'ajouter de l'espace entre un élément et son conteneur.
- Sa valeur se mesure en **dp**, pour Density-independent pixels. Par exemple, en précisant une valeur de 10dp, l'élément s'éloigne des bords de son conteneur.
- Si on souhaite modifier qu'une seule marge, on peut utiliser les versions suffixées suivantes :
  - Layout_marginTop, Layout_marginBottom, Layout_marginStart ou Layout_marginEnd

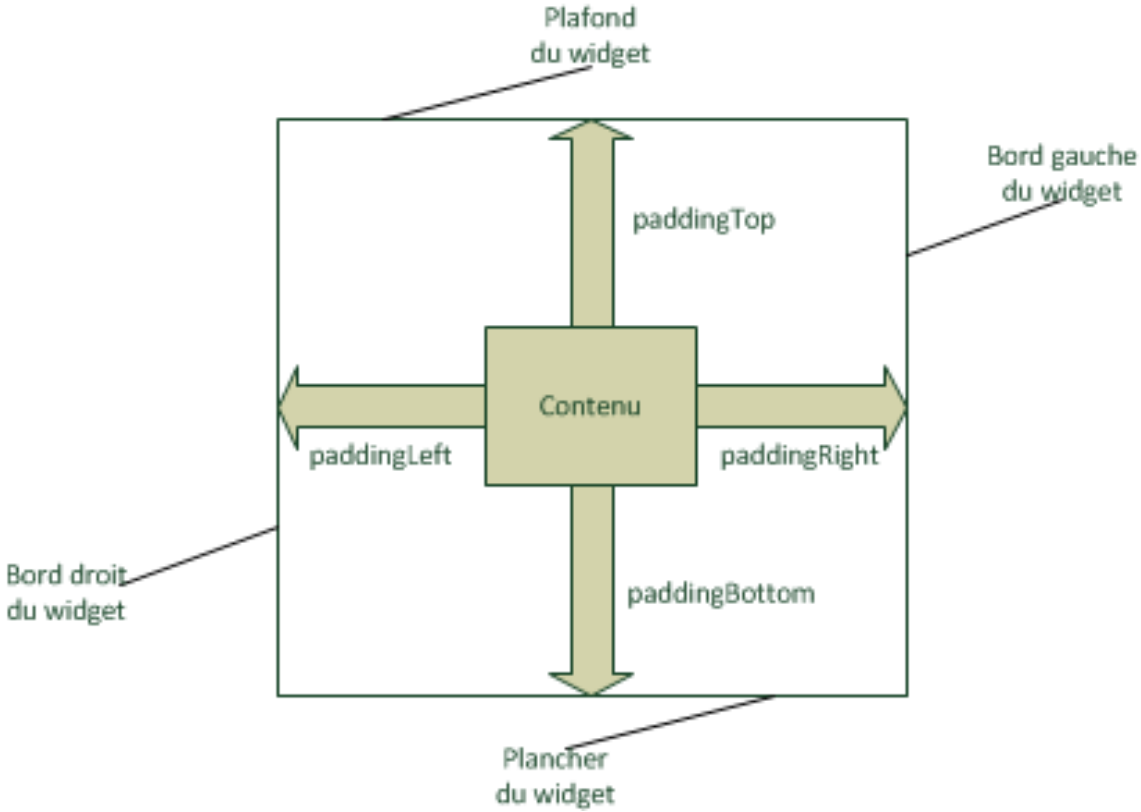
padding

- Permet d'ajouter de l'espace entre le contenu d'un élément et les bords de cet élément.

Université Internationale de Casablanca

Les attributs: layout_margin et padding





Université Internationale de Casablanca

L’attribut: id

- Permet d’identifier une ressource.
- On accède à une ressource à partir de son identifiant à l'aide.
- La valeur de l’attribut est :*"@+id/votre_identifiant"*
- Exemple:

```
<TextView
    android:id="@+id/text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
```

Université Internationale de Casablanca

Récupérer une vue

- La récupération d'une vue consiste à faire appel aux éléments graphiques dans votre activité pour des traitements.

- La méthode à utiliser pour cela est ***public View findViewById()***. Elle prend en paramètre l'identifiant de la vue qui nous intéresse, et renvoie la vue correspondante.

- Vous pourrez déclarer que votre activité utilise comme interface graphique la vue que vous désirez à l'aide de la méthode ***void setContentView (View view)***

Université Internationale de Casablanca

Ressources

- Documentation officielle :
  - https://developer.android.com/index.html
- Tutoriel Android :
  - https://developer.android.com/studio/index.html