

DEPARTEMENT INFORMATIQUE

**COURS DÉVELOPPEMENT
MOBILE & MULTIMEDIA**

Dr. KOPOIN N'Diffon Charlemagne

0749950512

Charlemagnekopoin@gmail.com

Plan du cours

- ❑ Chapitre 1: Introduction à la programmation mobile Android
- ❑ Chapitre 2: Interfaces utilisateurs et interactions
- ❑ Chapitre 3: Intégration multimédia
- ❑ Chapitre 4: Applications avancées et publications
- ❑ Projet

Objectifs du cours

❑ Objectif général

Ce cours vise à introduire les apprenants au développement d'applications mobiles, en mettant l'accent sur les aspects pratiques de la conception, du développement et du déploiement d'applications pour les plateformes mobiles (iOS, Android,...)

❑ Objectifs spécifiques

- Avoir une connaissance du développement mobile;
- Apprendre le langage Kotlin;
- Créer des applications multimedia;
- Manipuler des applis avec base de données

❑ Prérequis

- Connaissance de la programmation java
- Outils nécessaires : Ordinateur, internet, Android Studio, jdk

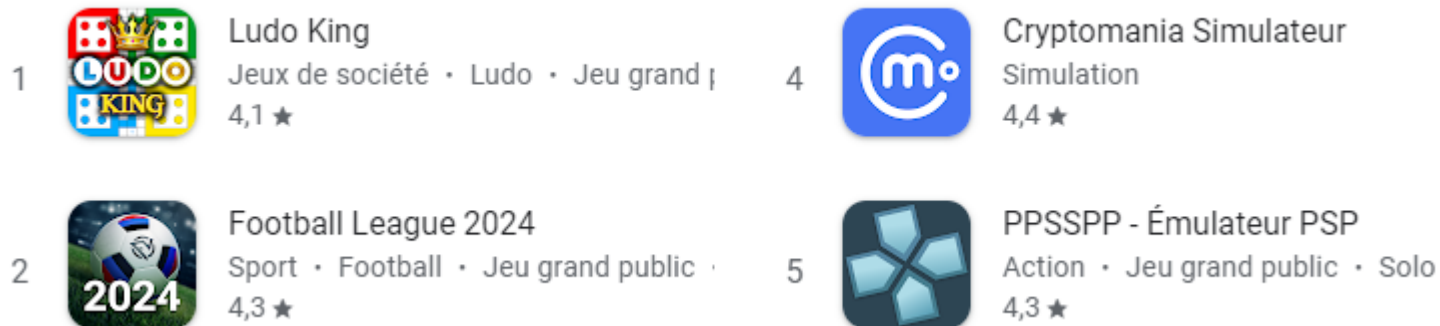
Chapitre 1: Introduction à la programmation mobile Android

- | 1. Applis mobiles
- | 2. Introduction à Android et installation
- | 3. Les bases de la programmation Android

Définition

❑ Application mobile

- Une application mobile est un logiciel applicatif transportable et autonome, développé pour être installé sur un appareil électronique mobile
- Elle est identifiée par un ou plusieurs programmes téléchargeables de façon gratuite ou payante depuis un magasin d'applications (Play store pour android et app store pour iOS par exemple), permettant d'accéder à un contenu homogène et exécutable à partir du système d'exploitation du Smartphone



Définition

□ Application mobile

- Une application mobile s'exécute sur un support matériel mobile:
 - ** Ressources limitées
 - Batterie (énergie), interface graphique, CPU, périphériques d'IO, ...
 - ** Supports physiques (matériels) très divers
 - De très élémentaire au très évolué
 - ** Utilisation ubiquitaire
 - Ubiquité géographique
 - Ubiquité des utilisateurs
 - Ubiquité des supports, ...

Historique et évolution

❑ Premières applications mobiles

- **1990:** Introduction des premiers téléphones mobiles dotés de capacités limitées.
- Les premières applications étaient principalement des jeux simples comme le célèbre Snake de Nokia.
- Les applications étaient préinstallées par les fabricants et ne pouvaient pas être téléchargées ou mises à jour par les utilisateurs.



Historique et évolution

□ Emergence des smartphones

- Lancement de l'iPhone et de l'App Store (2007-2008) :
 - ** 2007 : Lancement de l'iPhone par Apple, (interface tactile et son design innovant).
 - ** 2008 : Lancement de l'App Store d'Apple, permettant aux développeurs tiers de créer et de distribuer des applications.
- Introduction de l'Android Market (devenu Google Play Store) :
 - ** 2008 : Google lance Android Market, l'ancêtre du Google Play Store, avec un modèle similaire à l'App Store, offrant une plateforme pour les développeurs d'applications Android.

Historique et évolution

❑ Croissance et diversification

- Expansion des plateformes d'applications :
 - ** 2009-2010 : Croissance rapide du nombre d'applications disponibles sur l'App Store et Android Market.
 - ** Introduction de nouvelles plateformes : Windows Phone, BlackBerry App World, et autres tentent de rivaliser avec iOS et Android.
- Innovations technologiques :
 - ** Capacités matérielles améliorées : Introduction de caméras de haute qualité, de GPS, de capteurs de mouvement, etc., permettant des applications plus sophistiquées et variées.
 - ** Connexions internet plus rapides : L'essor de la 3G et de la 4G a permis des applications nécessitant une connectivité permanente, comme les réseaux sociaux, le streaming vidéo et les jeux en ligne.

Historique et évolution

❑ Évolution Récente et Tendances Actuelles

■ Applications multiplateformes :

** **Emergence de Frameworks de développement** :



** Progressive Web Apps (PWA), combinant les avantages des sites web et des applications mobiles.

■ Intégration des nouvelles technologies :

** **Intelligence artificielle et apprentissage automatique** : fonctionnalités basées sur l'IA, comme les assistants virtuels, la reconnaissance faciale, et les recommandations personnalisées.

** **Internet des objets (IoT)** : Les applications mobiles jouent un rôle crucial dans l'écosystème IoT, permettant le contrôle et la gestion des appareils connectés.

■ Économie des applications :

** **Modèles de monétisation** : Évolution des modèles économiques, incluant les achats intégrés, les abonnements, et la publicité.

Historique et évolution

❑ Discussion et questions? (10 mn)



Types d'applications mobiles

❑ Applications mobiles natives

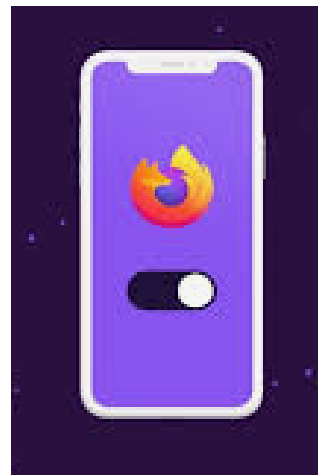
- Développées spécifiquement pour un système d'exploitation (OS) comme iOS ou Android.
- Utilisent les **langages de programmation natifs** de la plateforme (**Swift/Objective-C pour iOS, Kotlin/Java pour Android**).
- Accès complet aux fonctionnalités du matériel (caméra, GPS, etc.).



Types d'applications mobiles

❑ Applications mobiles web

- Accessibles via un navigateur web sur un dispositif mobile.
- Développées en utilisant des technologies web standard (HTML5, CSS, JavaScript)
- Ne nécessitent pas de téléchargement depuis un app store.



Types d'applications mobiles

❑ Applications mobiles hybrides

- Combinaison d'éléments natifs et web.
- Développées en utilisant des technologies web, encapsulées dans un conteneur natif.
- Utilisation de frameworks comme Cordova, Ionic, ou React Native.



Types d'applications mobiles

❑ Progressives Web Apps

- Applications web avec des fonctionnalités supplémentaires qui les rendent similaires aux applications natives.
- Fonctionnent hors ligne, envoient des notifications push, et peuvent être installées sur l'écran d'accueil.
- Développées en utilisant des technologies web modernes.



Types d'applications mobiles

❑ Avantages, inconvénients et cas d'utilisation

Types mobiles	Avantages	Inconvénients	Cas d'utilisation
Natives	<ul style="list-style-type: none">Performance optimale :Expérience utilisateur amélioréeAccès aux fonctionnalités du dispositif : (Utilisation complète des API natives)	<ul style="list-style-type: none">Coût de développement plus élevéTemps de développement plus long.	<ul style="list-style-type: none">jeux, applications graphiques intensivesApplications de réalité augmentée, outils de navigation).
WEB	<ul style="list-style-type: none">Compatibilité multiplateforme .Développement plus rapide et moins coûteuxMises à jour instantanées.	<ul style="list-style-type: none">Performance limitéeAccès limité aux fonctionnalités du dispositifExpérience utilisateur variable	<ul style="list-style-type: none">Applications d'information (blogs, actualités).nécessitant une compatibilité multiplateforme
Hybrides	<ul style="list-style-type: none">Compatibilité multiplateformeAccès aux fonctionnalités du dispositifDéveloppement plus rapide que les applications natives	<ul style="list-style-type: none">Performance inférieure aux applications nativesComplexité du développementMises à jour et maintenance	<ul style="list-style-type: none">Applications d'entreprise internes.Prototypes et applications de preuve de concept.Applications nécessitant un
PWA	<ul style="list-style-type: none">Compatibilité multiplateformeDéveloppement et déploiement simplifiésExpérience utilisateur améliorée	<ul style="list-style-type: none">Support matériel limité .Compatibilité variable	<ul style="list-style-type: none">Applications de commerce électronique.Services de médias en ligne.

Types d'applications mobiles

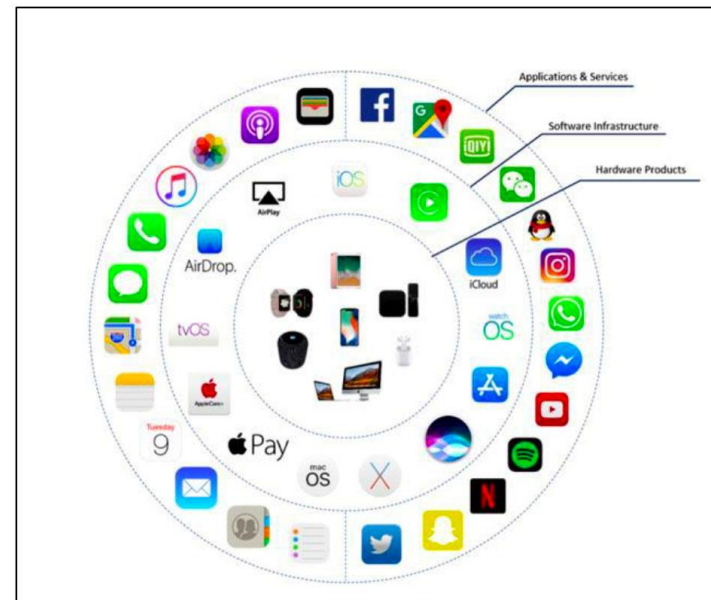
❑ Discussion et questions? (10 mn)



Ecosystèmes mobiles

❑ Définition

- Un écosystème mobile est un environnement interconnecté de dispositifs, applications, services, et plateformes.
- Comprend le système d'exploitation, le matériel, les app stores, et les services cloud.



Ecosystèmes mobiles

❑ Importance de l'écosystème

- Influencent le développement, la distribution, et la monétisation des applications.
- Offrent des fonctionnalités intégrées et des services aux utilisateurs.
- Ecosystème iOS & Ecosystème Android

Ecosystèmes mobiles

❑ Ecosystème iOS

🌈 Composants de l'écosystème iOS

➤ Système d'exploitation :

- ✓ iOS, connu pour sa stabilité, sécurité, et intuitive.

➤ Appareils :

- ✓ iPhone, iPad, Apple Watch, Apple TV.

➤ App Store :

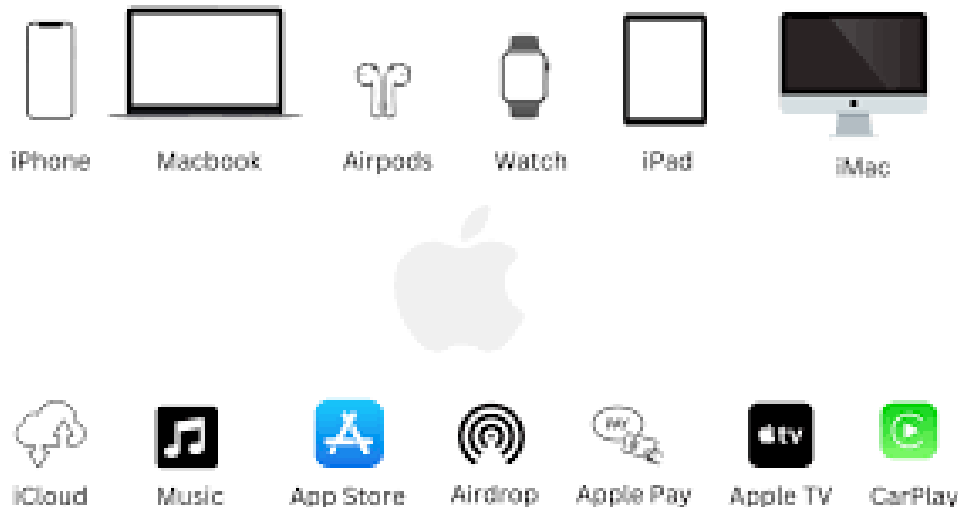
- ✓ Plateforme de distribution d'applications avec des règles strictes de validation.

➤ Services intégrés :

- ✓ iCloud, Apple Pay, Siri, Apple Music, HealthKit, HomeKit.

➤ Frameworks de développement : **

- ✓ Swift, Objective-C, Xcode, UIKit, SwiftUI, ARKit.



Ecosystèmes mobiles

❑ Ecosystème Android

🚩 Composants de l'écosystème iOS

➤ Système d'exploitation :

- ✓ Android, connu pour sa flexibilité, personnalisation, et large adoption.

➤ Appareils :

- ✓ Large gamme de dispositifs de différents fabricants

➤ Play Store :

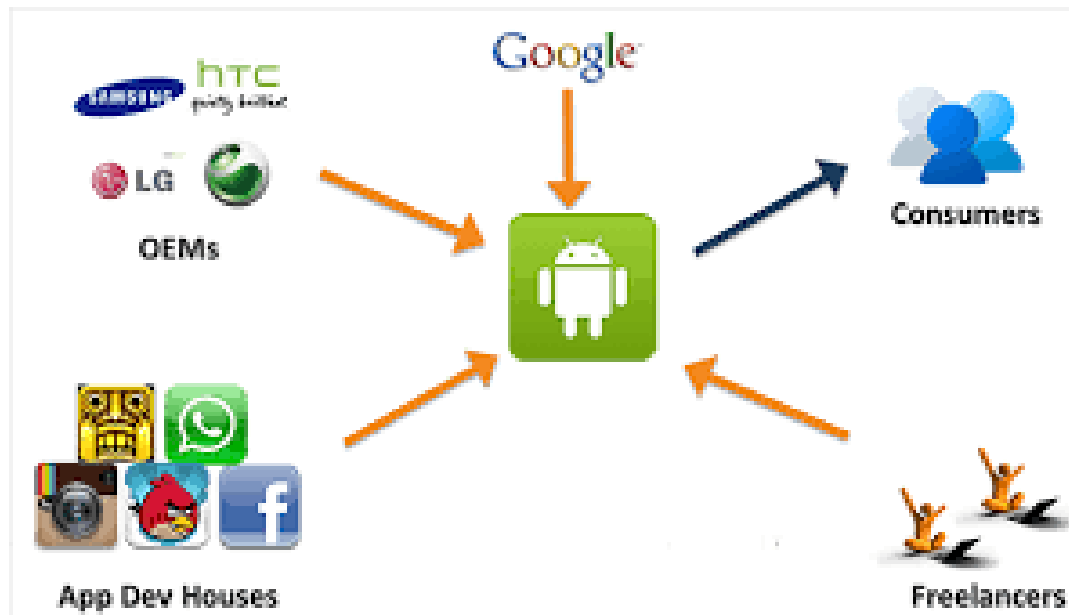
- ✓ Plateforme de distribution principale avec moins de restrictions que l'App Store d'Apple.

➤ Services intégrés :

- ✓ Google Play Services, Google Assistant, Google Drive, Google Pay, Google Fit..

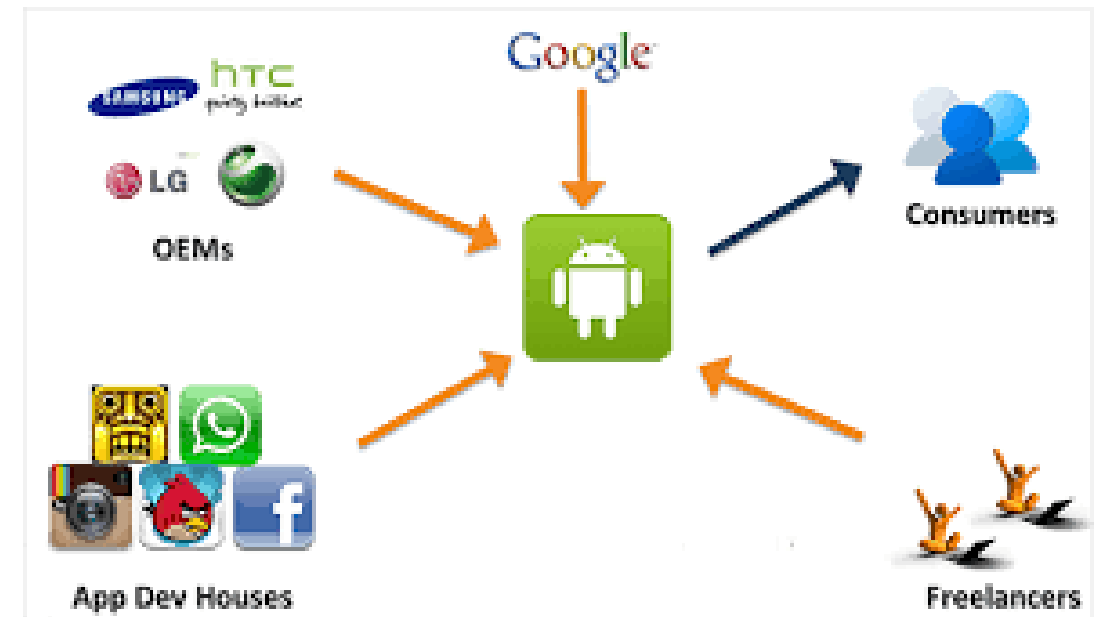
➤ Frameworks de développement : **

- ✓ Java, Kotlin, Android Studio, Jetpack, Firebase..



Ecosystèmes mobiles

❑ Discussion et questions? (10 mn)



Installation des Environnements de Développement

❑ Installation de l'environnement pour iOS

✓ Xcode pour iOS

- [Installation de Xcode](#)
- [YouTube: How to Install Xcode](#)

✓ Préréquis

- [Un Mac avec macOS.](#)
- [Un identifiant Apple](#)

Installation des Environnements de Développement

❑ Installation de l'environnement pour Android

✓ Android Studio pour Android

- [Installation d'Android Studio](#)
- [YouTube: How to Install Android Studio](#)

✓ Préréquis

- [Un ordinateur Marchant sous Windows ou Linux ou macOS.](#)
- [JDK \(Java Development Kit\) installé](#)

Chapitre 1: Introduction à la programmation mobile Android

- | 1. Applis mobiles
- | 2. Introduction à Android et installation
- | 3. Les bases de la programmation Android

Historique et évolution

❑ Android?

- Android, développé par Android Inc. En 2004 (startup américaine), racheté par Google en 2005.
- Première version publique d'Android (Android 1.0) lancée en 2008.
- De nombreuses versions depuis. On en est à la version 14 (juillet 2024) et l'API 34. La version 14 est le numéro pour le grand public, et les versions d'API sont pour les développeurs. Exemples :
 - Android 4.1 JellyBean = API 16,
 - Android 7.0 Nougat = API 24,
 - Android 13 Tiramisu = API 33
 - **Android 14 ? Nom de code ? API**

Architecture

❑ Android?

Architecture Android



SDK Android et Android studio

□ Android?

Le **Software Development Kit** (SDK) contient :

- Les librairies de classes et fonctions pour créer des logiciels
- les outils de fabrication des logiciels (compilateur. . .)
- **AVD** : un émulateur de tablettes pour tester les applications
- **ADB** : un outil de communication avec les vraies tablettes

Le logiciel Android Studio offre :

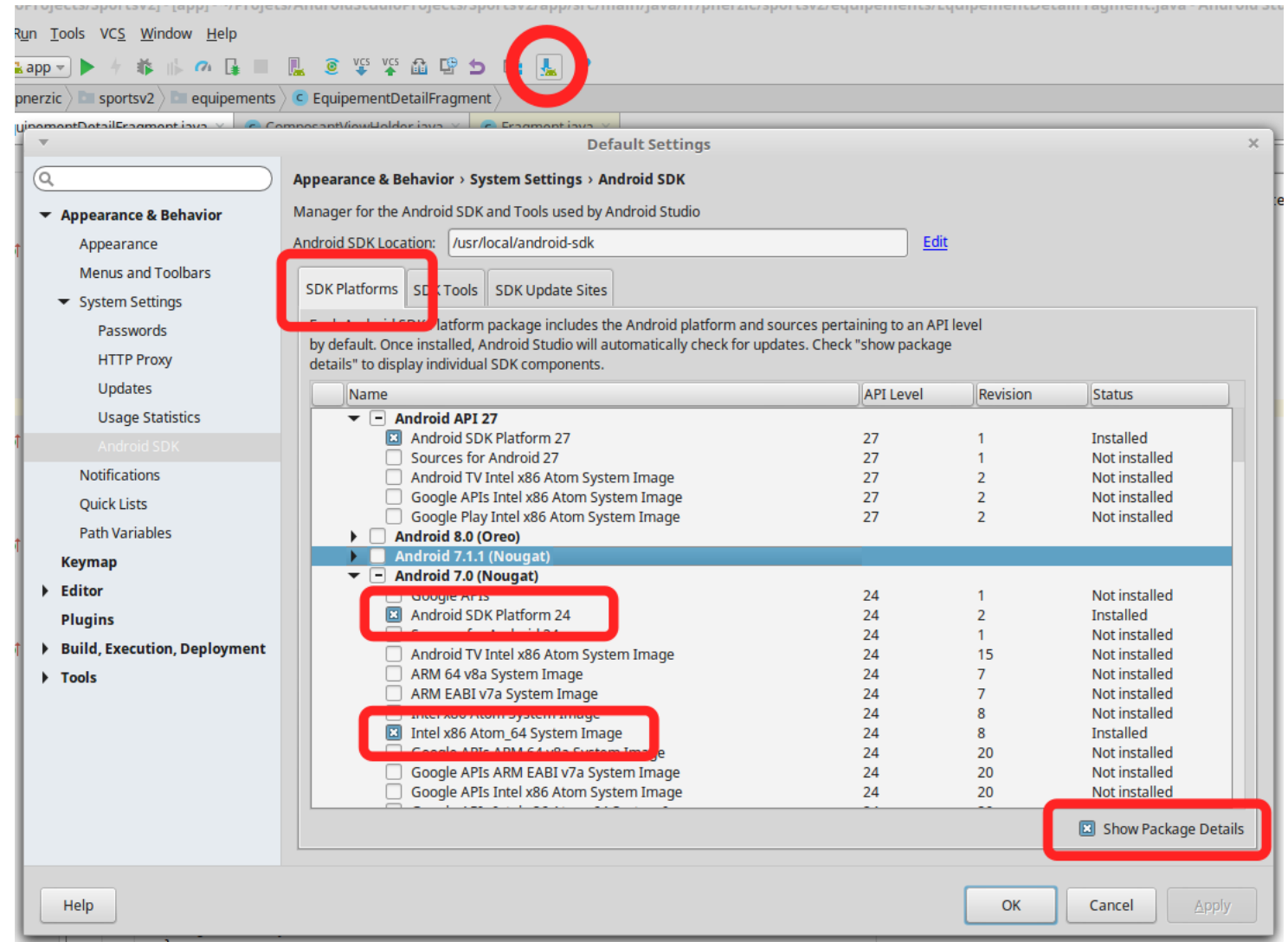
- un éditeur de sources et de ressources
- des outils de compilation : gradle
- des outils de test et de mise au point

SDK Android et Android studio

❑ Android? SDK manager

Éléments indispensables :

- Android SDK Platform
- Intel x86 Atom_64 System Image



Installation des Environnements de Développement

❑ Installation de l'environnement pour Android

. Étapes d'installation :

◦ Téléchargement d'Android Studio :

- . Aller sur le site officiel d'Android Studio :
developer.android.com/studio.
- . Télécharger la version correspondant au système d'exploitation.

◦ Installation :

- . Exécuter le fichier téléchargé et suivre les instructions de l'assistant d'installation.
- . Installer Android Studio et les composants recommandés (SDK, émulateur, etc.).

Windows
(64 bits)

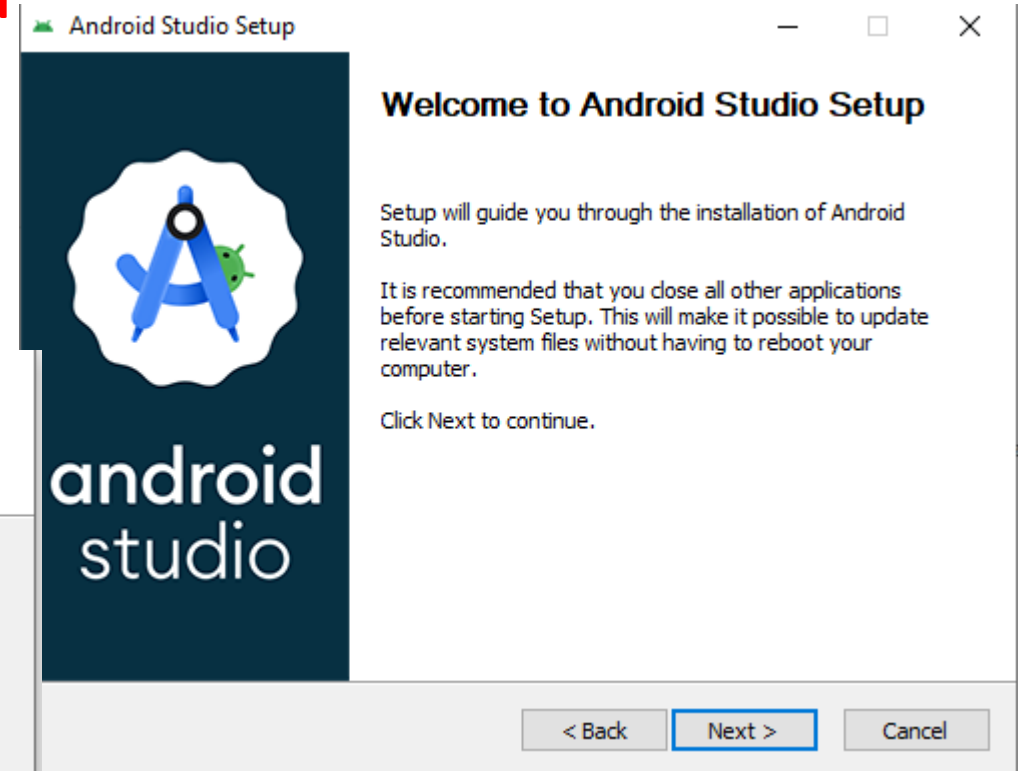
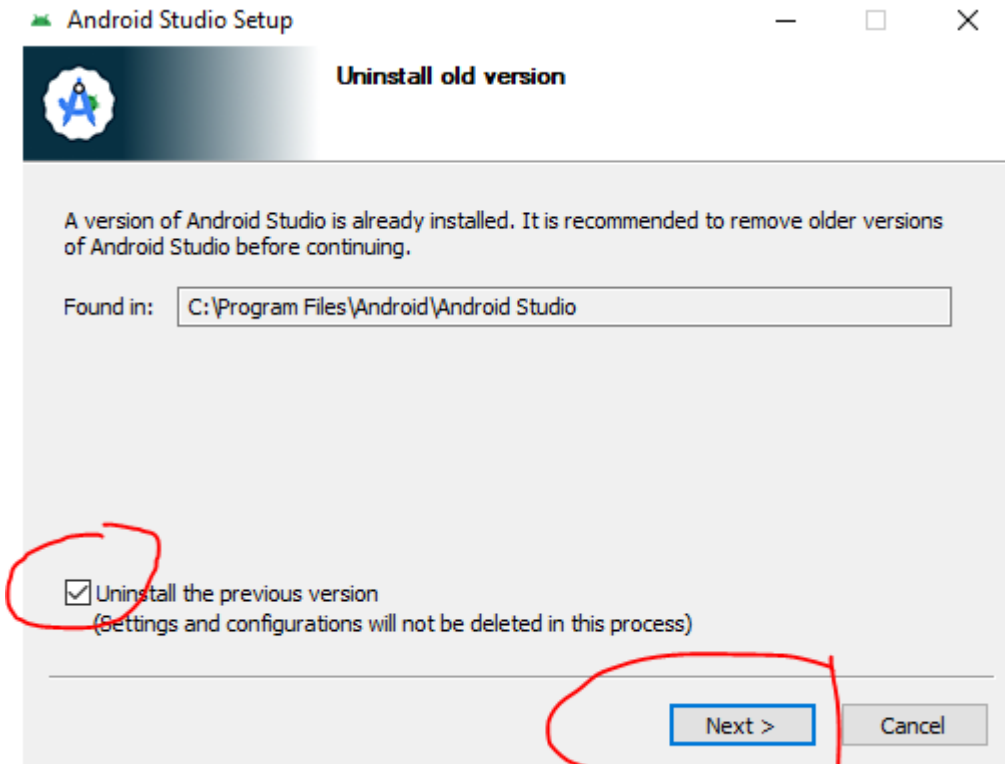
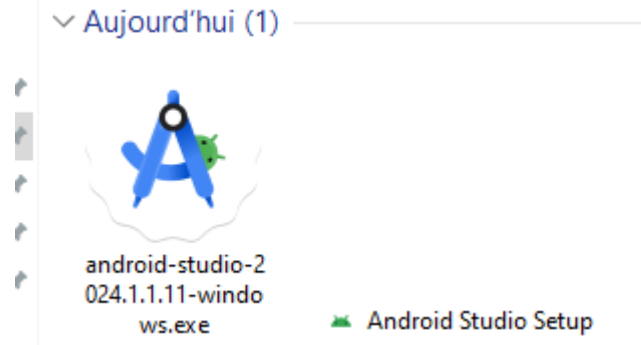
Mac
(64 bits)

Mac
(64 bits, ARM)

Linux
(64 bits)

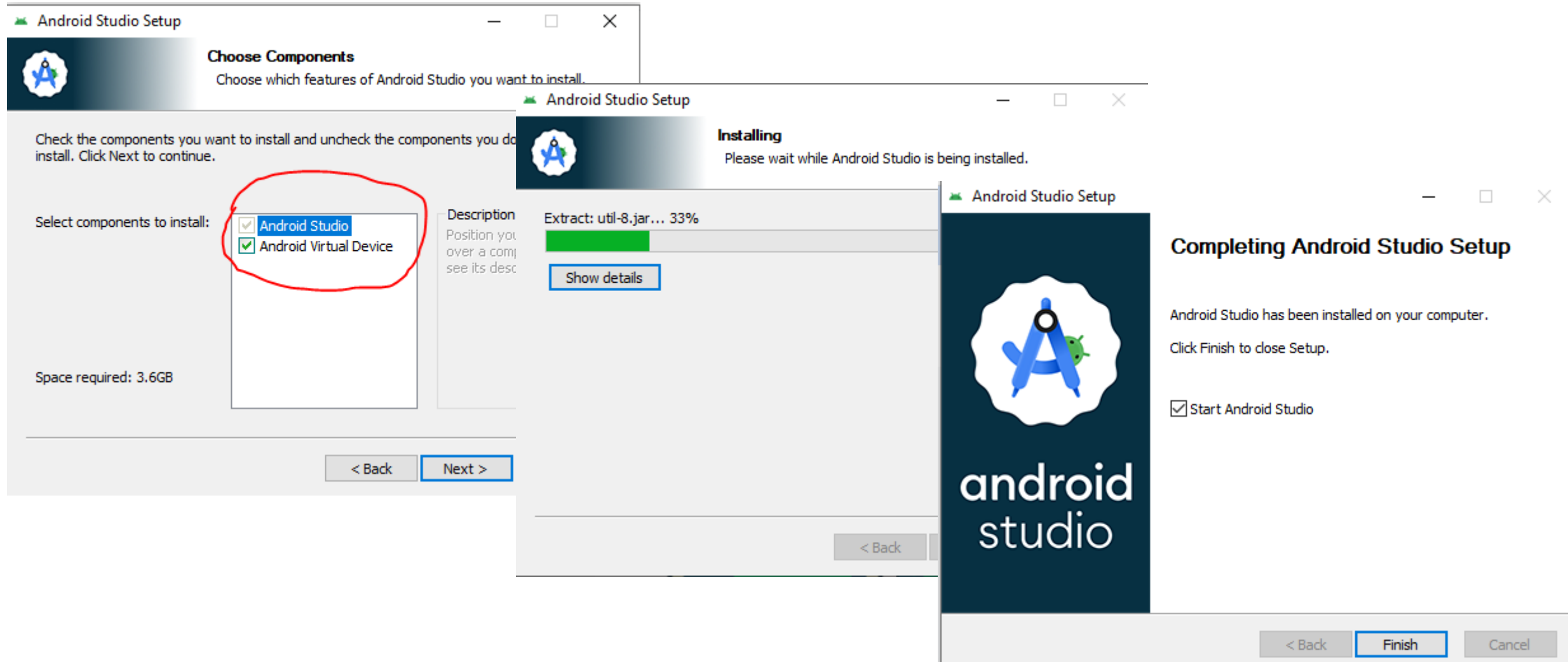
Installation des Environnements de Développement

❑ Installation de l'environnement pour Android



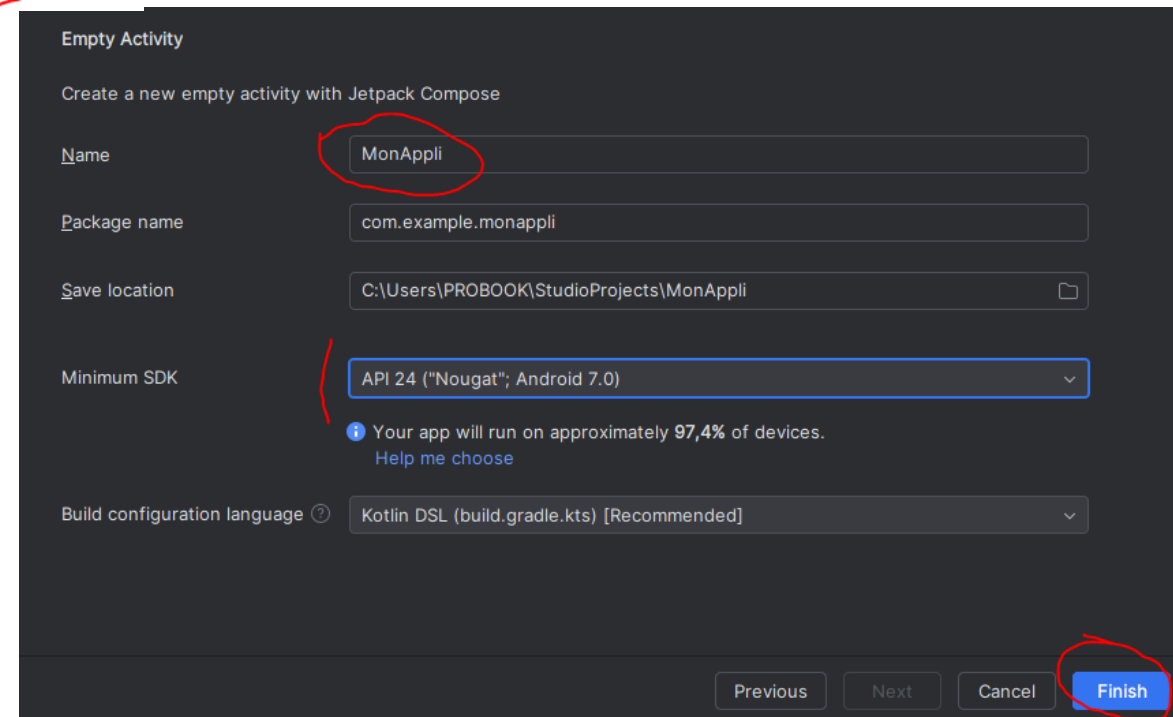
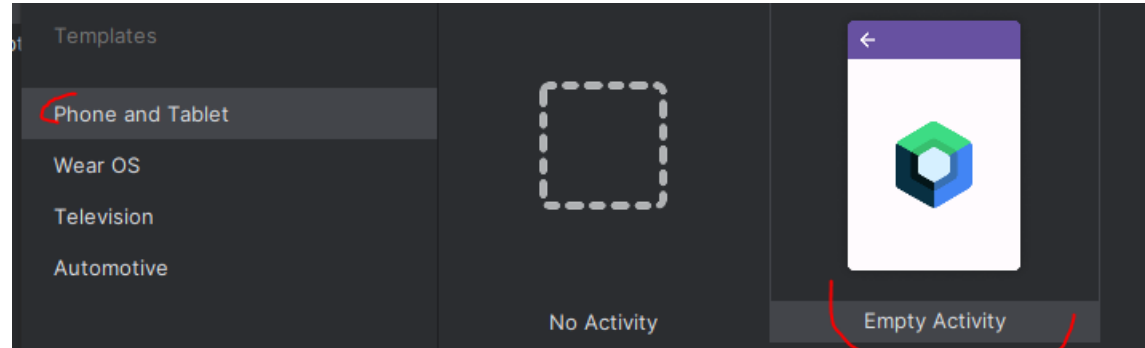
Installation des Environnements de Développement

❑ Installation de l'environnement pour Android



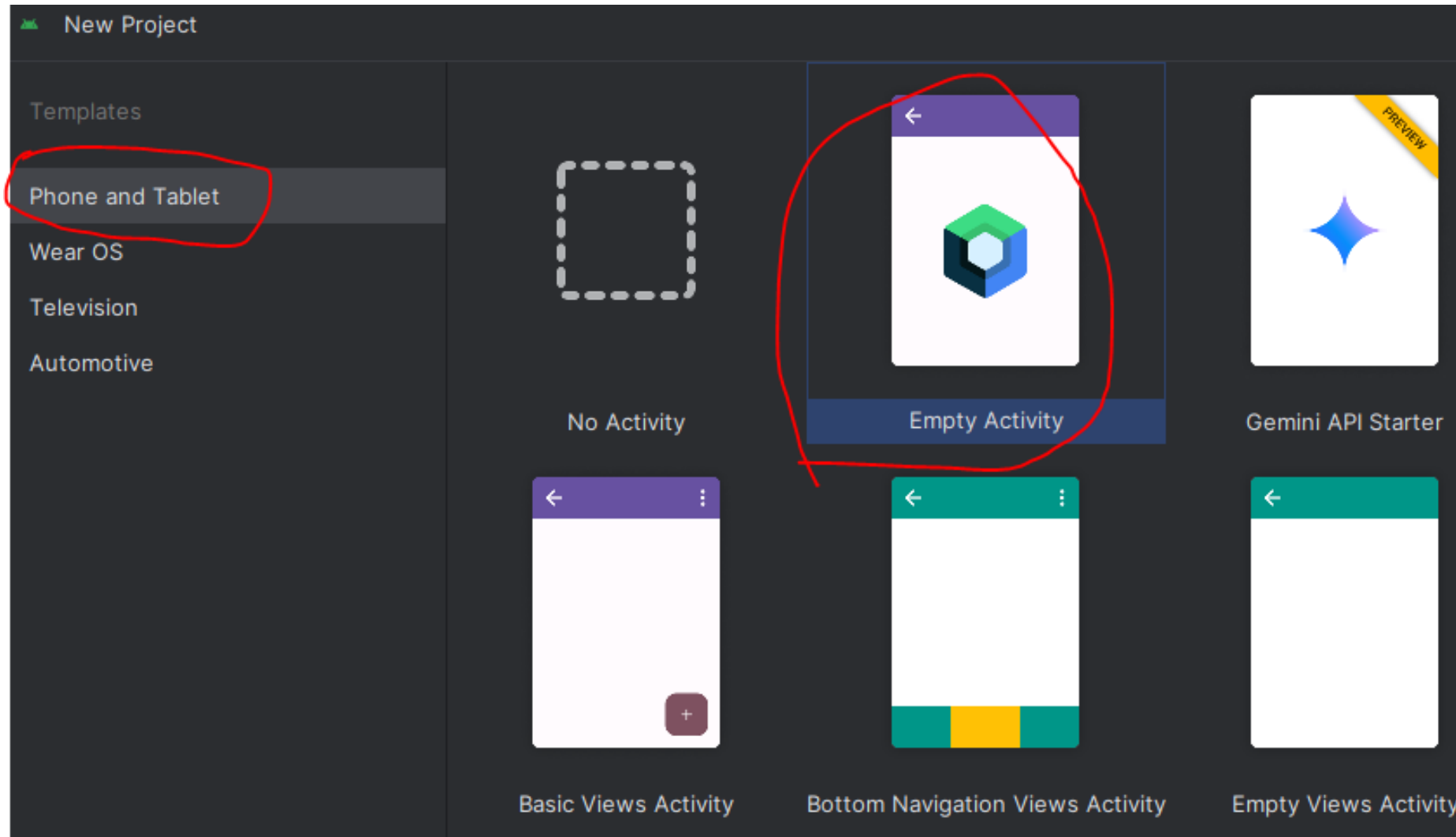
Installation des Environnements de Développement

❑ Installation de l'environnement pour Android



Création d'une première application

❑ Choix du matériel et du template



Création d'une première application

❑ Renseignement des informations de l'application

Empty Activity

Create a new empty activity with Jetpack Compose

Name

Package name

Save location

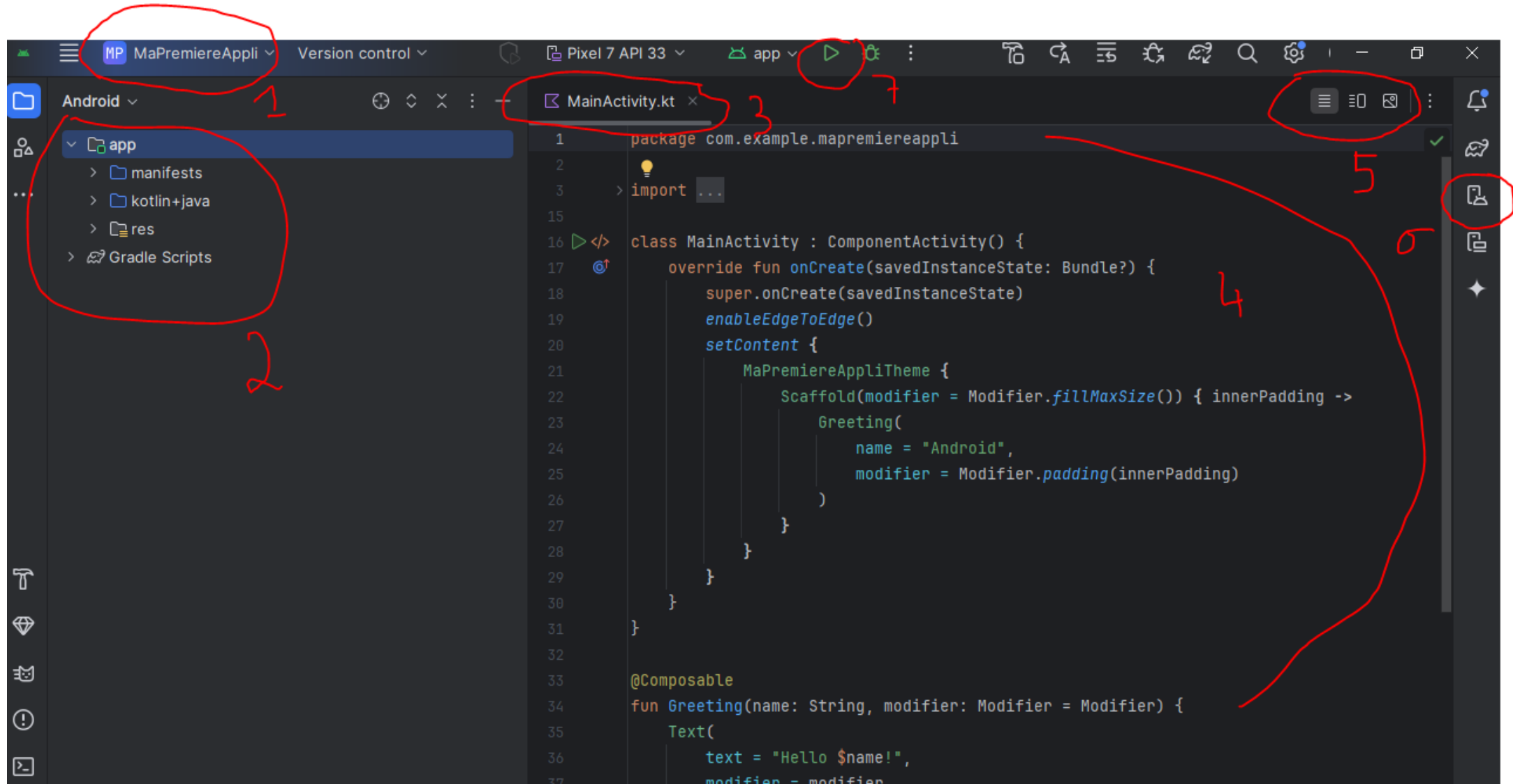
Minimum SDK

i Your app will run on approximately **97,4%** of devices.
[Help me choose](#)

Build configuration language

Création d'une première application

□ Première page de l'application créée



Apprentissage du langage Kotlin

❑ **Eléments de base:** <https://developer.android.com/courses?hl=fr>

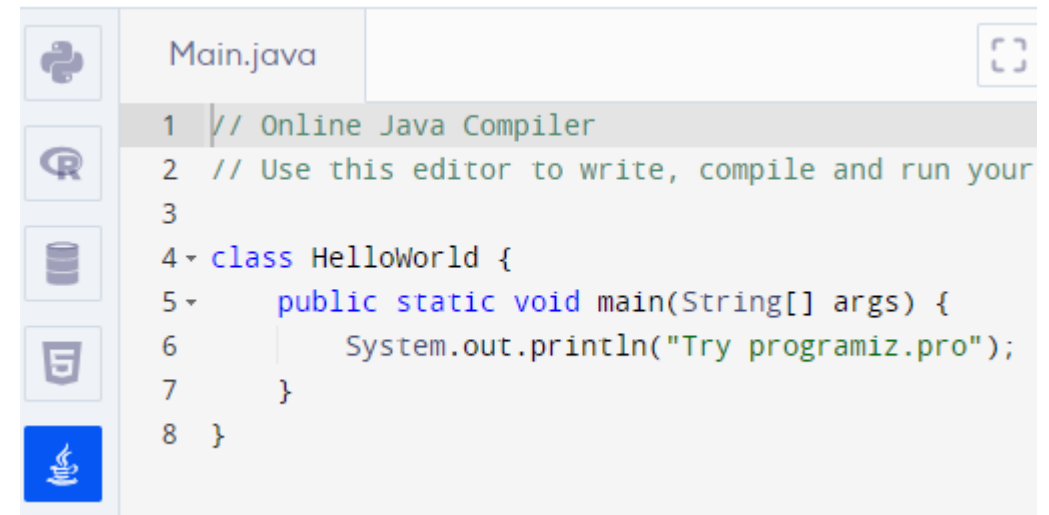
➤ Apprendre Kotlin en ligne

- <https://play.kotlinlang.org/>

```
/**
 * You can edit, run, and share this code.
 * play.kotlinlang.org
 */
fun main() {
    println("Bienvenue")
}
```

➤ Apprendre java en ligne

- <https://www.programiz.com/java-programming/online-compiler/>



```
Main.java
1 // Online Java Compiler
2 // Use this editor to write, compile and run your
3
4 class HelloWorld {
5     public static void main(String[] args) {
6         System.out.println("Try programiz.pro");
7     }
8 }
```

Apprentissage des langages natifs: Kotlin et java

❑ **Eléments de base:** <https://developer.android.com/courses?hl=fr>

➤ Apprendre Kotlin et java en ligne

Action	Kotlin	Java
1. Faire des commentaires	// commentaires sur une ligne /* commentaire sur plusieurs lignes */	
2. Déclarer une variable val : pour des constantes ou variables immuables var : pour des variables mutables ou changeantes	Avec val ou avec var val / var nom: type ou val / var nom : type = valeur ou val / var nom = valeur <pre>var a : String a = "Bonjour" val i: Int =5 val b= "Welcome"</pre>	type nom ; ou type nom = valeur ; <pre>String bonjour; int a = 10;</pre>

Apprentissage des langages natifs: Kotlin et java

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

➤ Apprendre Kotlin et java en ligne

Action	Kotlin	Java
3. Afficher à l'écran	<p>println(ce qu'on voudrait afficher)</p> <pre>val b= "Welcome" println(b) println("Bonjour")</pre>	<p>System.out.println(ce qu'on voudrait afficher)</p> <pre>System.out.println("Bonjour");</pre>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Éléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
4. Afficher à l'écran des chaînes de caractère, des variables et des chaînes de caractères et des variables et afficher des calculs	<p>println(le nom de la variable)</p> <p>println("la chaîne de caractère entre double cote")</p> <p>println("la chaîne de caractère entre double cote \$la variable")</p> <p>println("la chaîne de caractère entre double cote \$ {le calcul à effectuer}")</p>

```
val b= "Welcome"
println(b)
println("Bonjour")
println("Bonjour et $b")
println("la somme de 15 et 10 est ${15+10}")
```

```
Welcome
Bonjour
Bonjour et Welcome
la somme de 15 et 10 est 25
```


Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base:** <https://developer.android.com/courses?hl=fr>

Exercice 1: Manipulation de chaines de caractères

1. Créer une variable de type string qui reçoit la donnée **Bienvenue**
2. Créer une variable de type String qui reçoit votre **nom** comme donnée
3. Créer une variable qui reçoit la donnée **c'est la vie**
4. Créer une variable qui reçoit la donnée **"elle est belle"**
5. Créer une variable qui reçoit dans l'ordre les variables créées
6. Afficher toutes les variables dans une phrase
7. Afficher l'année de naissance de la personne dont l'âge est 20

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
<p>5. Manipuler des nombres Les différents types de nombre en kotlin</p> <ul style="list-style-type: none">• Long : 64 bits• Int : 32 bits• Short : 16 bits• Byte : 8 bits• Double : 64 bits• Float : 32 bits	<p>+ : addition - : soustraction • : multiplication / : division % modulo ou reste de la division</p>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Exercice 2: Manipulation des nombres

1. Créer deux variables de type Double
2. Créer une variable qui reçoit la multiplication des deux variables
3. Créer une variable qui reçoit la division de la première par la deuxième
4. Créer une variable qui reçoit le reste de la division
5. Créer une variable qui reçoit la multiplication des deux variables créées + 10
6. Que va retourner la syntaxe **uta +=7 ?** sachant que uta est un entier qui valait 6

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Éléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
6. Déclarer une variable booléenne	<code>var m: Boolean = true</code>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
<p>7. Déclarer une liste d'éléments, afficher la liste, afficher le premier élément, afficher le nième élément, supprimer un élément, ajouter un autre élément</p> <p>NB: si les éléments sont des chaines de caractères, il faut les mettre entre double cote</p> <p>La liste est pour des éléments de même type</p>	<pre>var nom_de_la_liste = listOf (élément1, élément2, élément3,...,élémentn) println(nom_de_la_liste) println(nom_de_la_liste[0]) println(nom_de_la_liste[n-1]) nom_de_la_liste = nom_de_la_liste-element nom_de_la_liste = nom_de_la_liste + element</pre>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Exercice 3: Listes

Déclarer la variable fruit qui est une liste des fruits suivants: **pomme, mangue, banane, raisin et ananas**

1. Afficher la liste
2. Afficher le 2^e élément de la liste
3. Ajouter goyave à la liste
4. Supprimer mangue de la liste
5. Peux ton ajouter un chiffre dans la liste?
6. Modifier la valeur du 1^{er} fruit en raisin

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Éléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
8. Les conditions	<ol style="list-style-type: none">1. Rappel sur quelques opérateurs<ul style="list-style-type: none">• égalité: <code>==</code>• différence: <code>!=</code>• Supérieur : <code>></code>• Supérieur ou égal : <code>>=</code>• inférieur : <code><</code>• inférieur ou égal : <code><=</code>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Éléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
8. Les conditions : si...alors...sinon si...alors... sinon...	<pre>if (a > b) { println("a is greater than b") } else if (a < b) { println("a is less than b") } else { println("a is equal to b") }</pre>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base:** <https://developer.android.com/courses?hl=fr>

Exercice 4: Structure conditionnelle SI

1. Désormais pour voter en côte d'ivoire, si tu es un homme et que tu as plus de 17 ans, tu es autorisé, sinon pas autorisé. Par contre si tu es une femme, c'est à partir de 18 ans que tu peux voter.
2. Que permet l'instruction suivante: `val max = if (a > b) a else b`

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
9. Les boucles : <ul style="list-style-type: none">- boucle for- Boucle while- Boucle do while- When Expression : semblable au switch	<pre>var i: Int ***** for (i in 1..10) { println(i) } ----- while (x > 0) { x-- } ----- do { val y = getY() } while (y != null) ----- when (x) { 1 -> println("x is 1") 2 -> println("x is 2") else -> println("x is neither 1 nor 2") }</pre>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Exercice 5: Boucles

1. Table de multiplication par 5 avec for
2. Même exercice avec while
3. Même exercice avec do while.
4. Faites des propositions de choix de couleur avec when expressions

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
10. Les fonctions : <ul style="list-style-type: none">- Fonction sans retour- Fonction avec retour- Fonctions avec paramètres et retour	<p>Une fonction permet de « ne pas réinventer la roue »</p> <p>Fun nom_fonction () {}</p> <p>Pour l'utiliser , il faut tout simplement appeler la fonction</p> <hr/> <pre>Fun salutation () { println ("a is less than b") }</pre> <hr/> <pre>fun sumsimple (): Int { return 15 + 10 }</pre> <hr/> <pre>fun sum(a: Int, b: Int): Int { return a + b }</pre>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Éléments de base: <https://developer.android.com/courses?hl=fr>**

Action	Kotlin
10. Les fonctions : <ul style="list-style-type: none">- Fonction anonymes (fonction lambda)- Fonction à Une Seule Expression	<p>Une fonction permet de « ne pas réinventer la roue »</p> <p>Fun nom_fonction () {}</p> <p>-----</p> <p>val sum = { x: Int, y: Int -> x + y }</p> <p>-----</p> <p>fun sum(a: Int, b: Int) = a + b</p> <p>-----</p>

Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **Eléments de base:** <https://developer.android.com/courses?hl=fr>

Exercice 6: Fonctions

1. Créez une fonction qui affiche « Bienvenue au cours à UTA »
2. Modifiez la fonction de sorte qu'elle prenne le nom et prénoms en paramètre puis affiche « bienvenue au cours à UTA M/Mme nom prénoms »
3. Créer une fonction qui retourne le périmètre d'un rectangle
4. Créer une fonction qui retourne l'aire d'un cercle

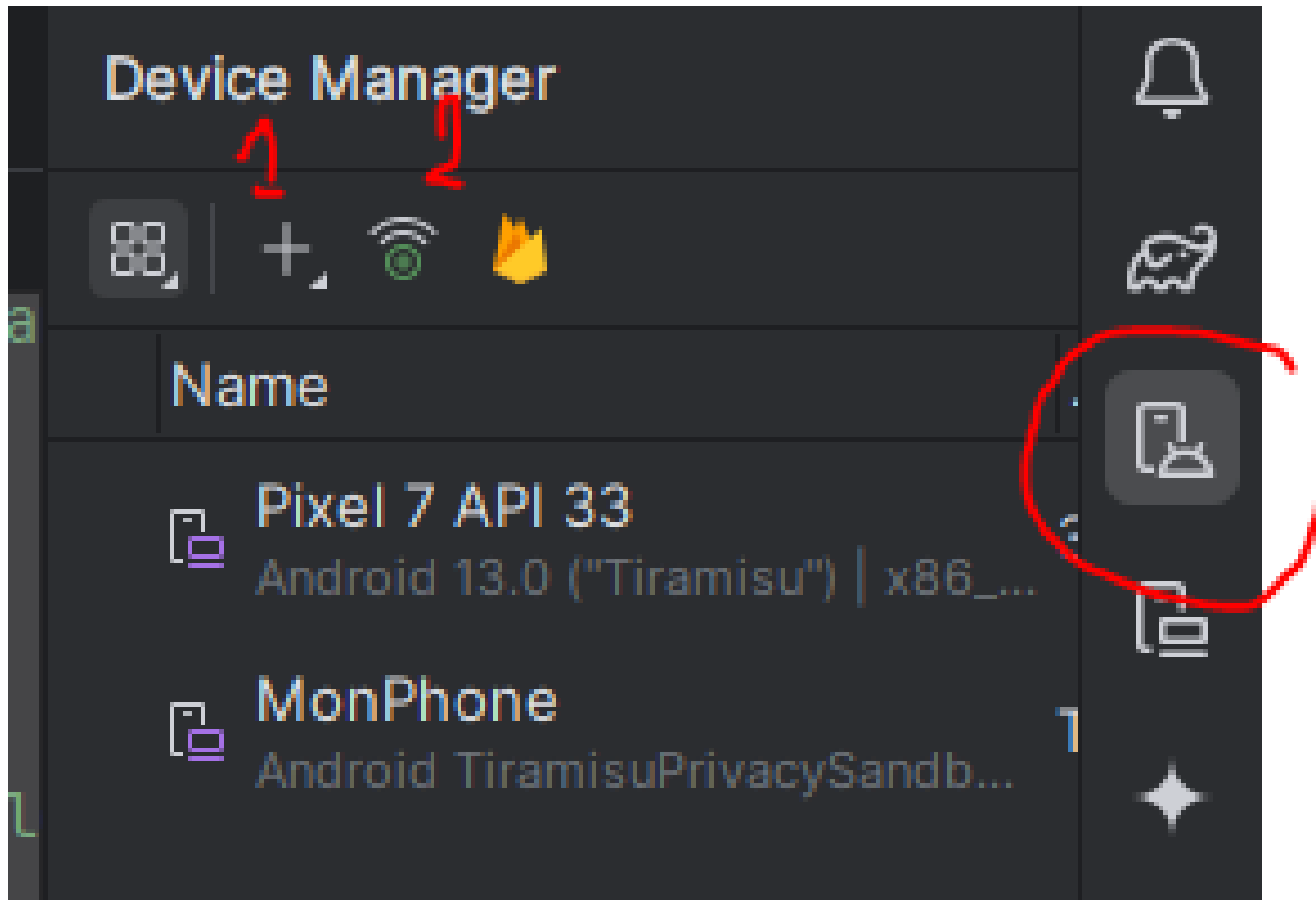
Apprentissage des langages natifs: Kotlin et java

➤ Apprendre Kotlin

❑ **POO: <https://developer.android.com/codelabs/basic-android-kotlin-compose-classes-and-objects?hl=fr#0>**

Ajout d'un AVD

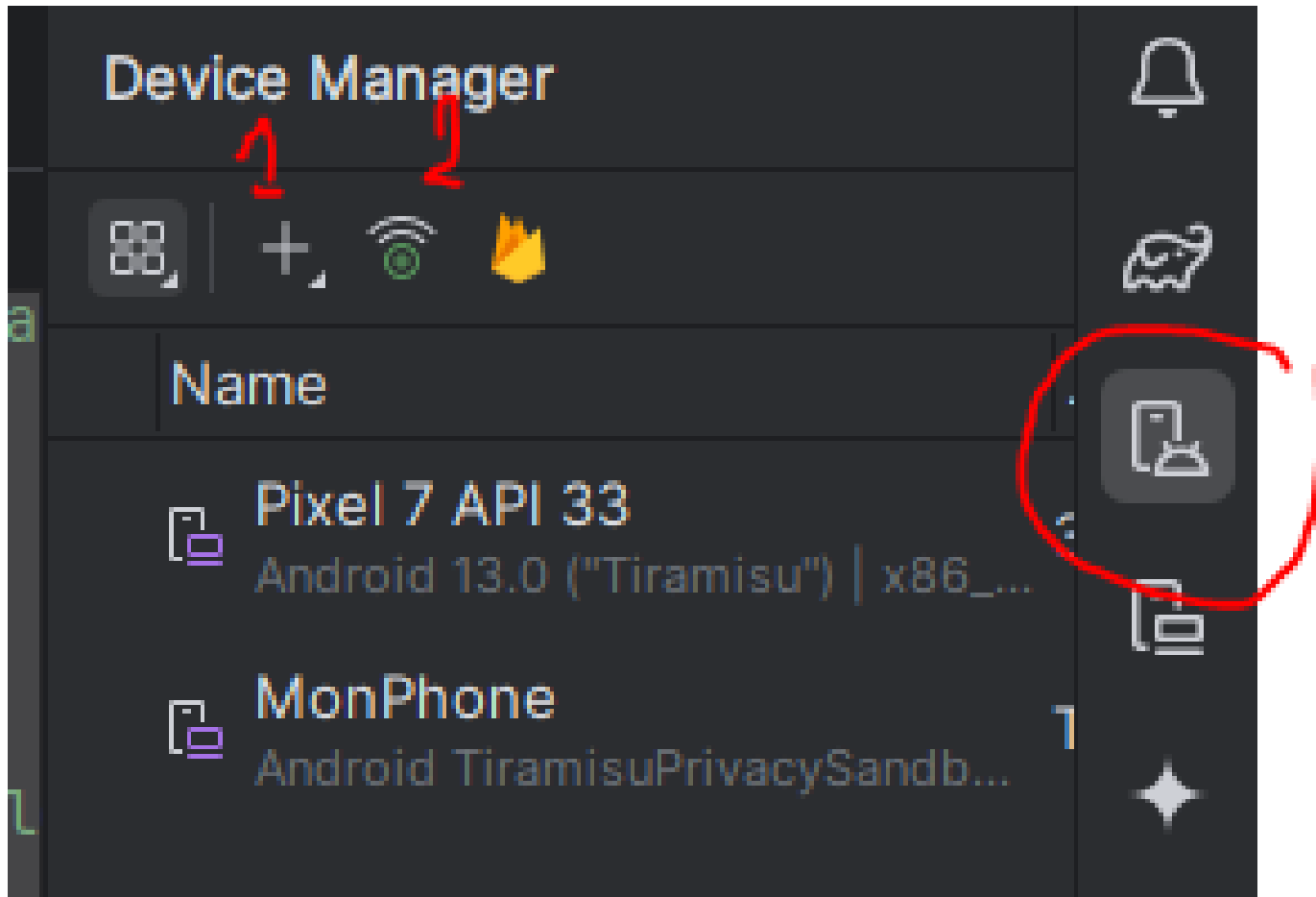
❑ Ajout d'un telephone virtuel



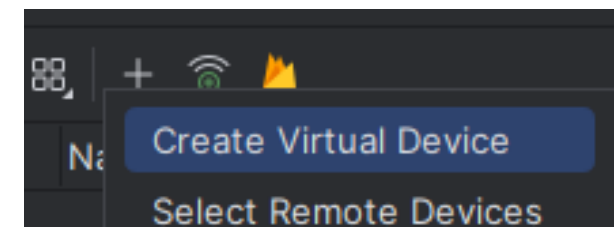
1. Cliquer sur l'icone du device manager
2. Cliquer sur le +

Ajout d'un AVD

□ Ajout d'un telephone virtuel



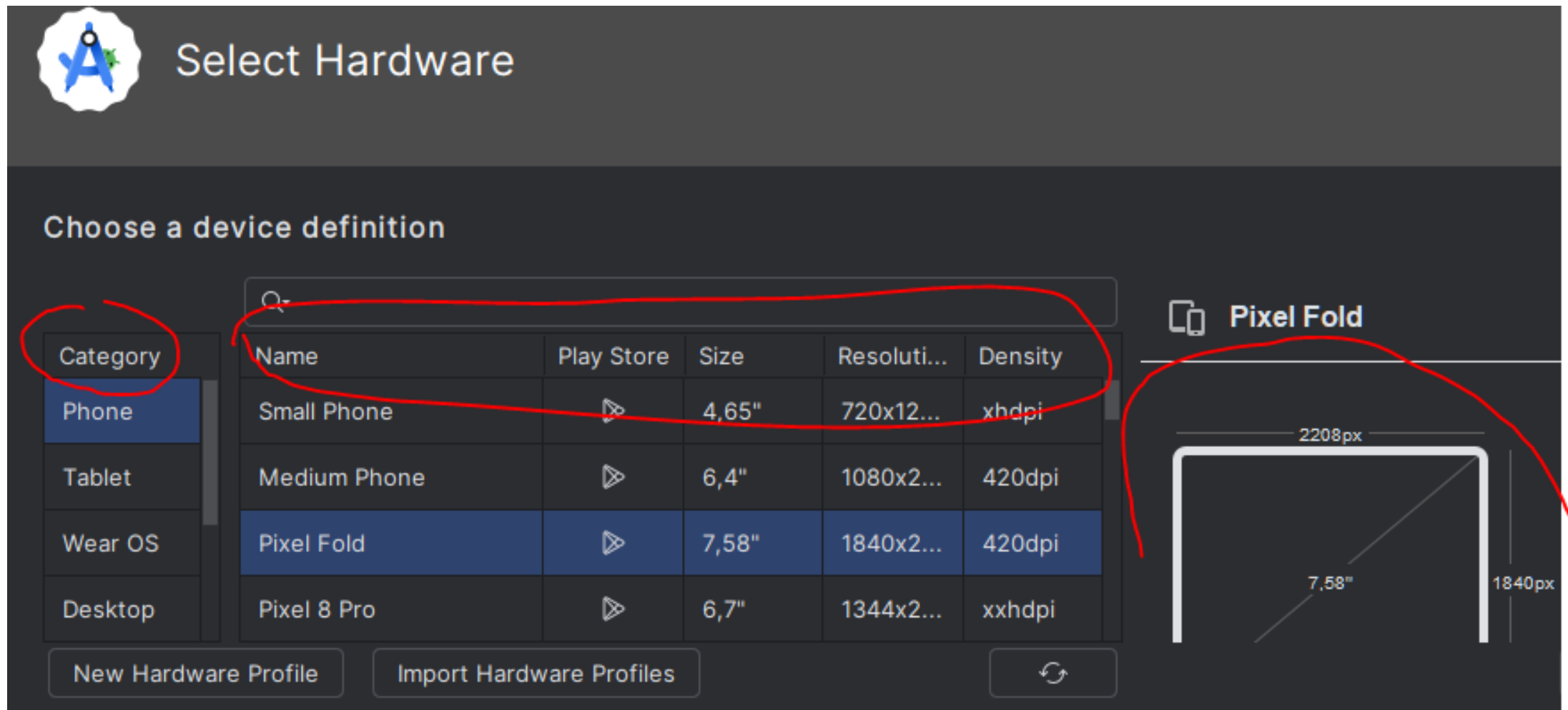
1. Cliquer sur l'icone du device manager
2. Cliquer sur le +
3. Cliquer sur



Ajout d'un AVD

❑ Ajout d'un telephone virtuel

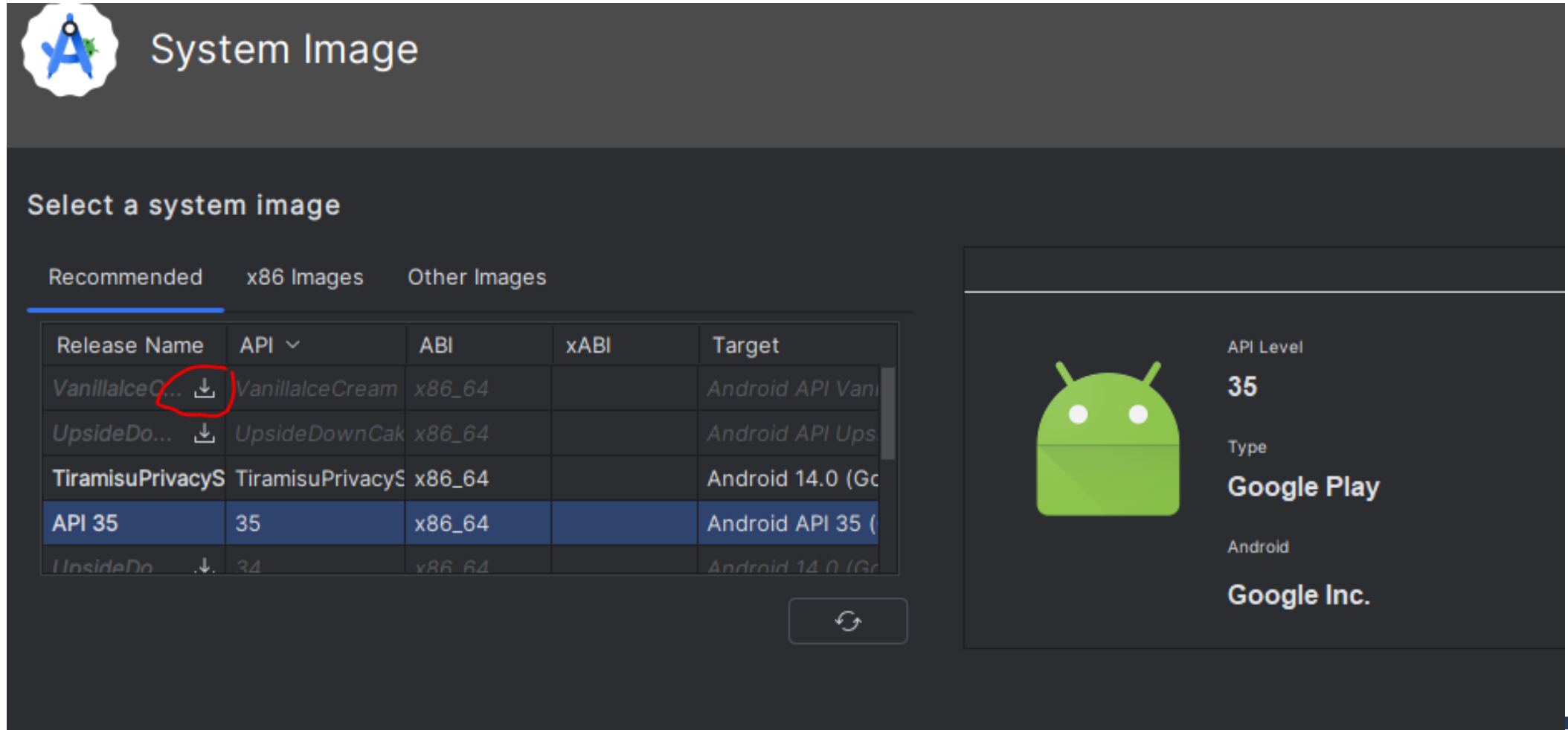
Choisir le format de l'appareil virtuel



Ajout d'un AVD

❑ Ajout d'un telephone virtuel




Cliquer sur le symbole de téléchargement pour avoir un système image




System Image

Select a system image

Recommended x86 Images Other Images

Release Name	API	ABI	xABI	Target
VanillalceC... 	VanillalceCream	x86_64		Android API Vani
UpsideDo... 	UpsideDownCak	x86_64		Android API Ups
TiramisuPrivacyS	TiramisuPrivacyS	x86_64		Android 14.0 (Gc
API 35	35	x86_64		Android API 35 (
UpsideDo... 	34	x86_64		Android 14.0 (Gc



API Level
35

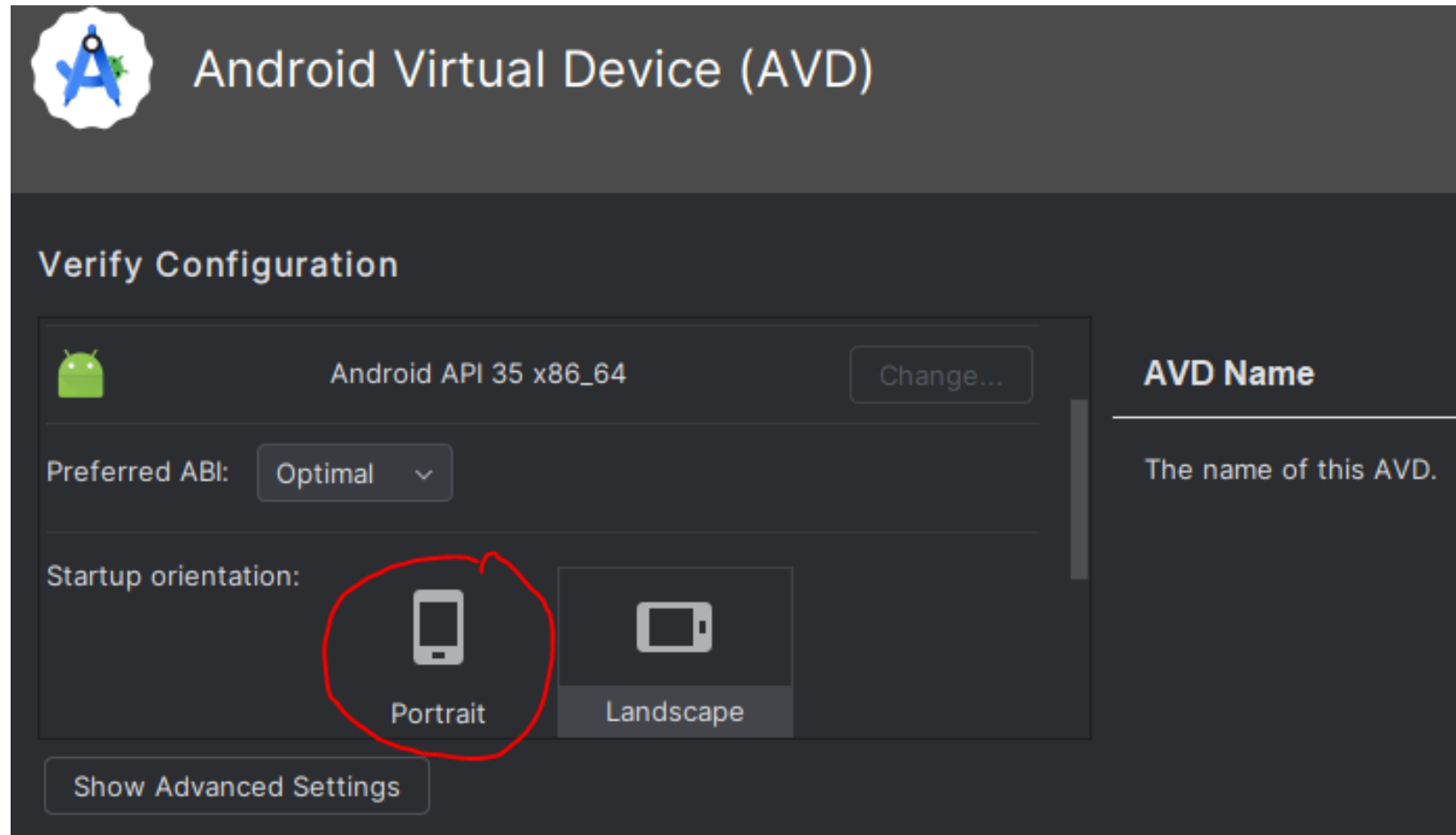
Type
Google Play

Android
Google Inc.

Ajout d'un AVD

❑ Ajout d'un telephone virtuel

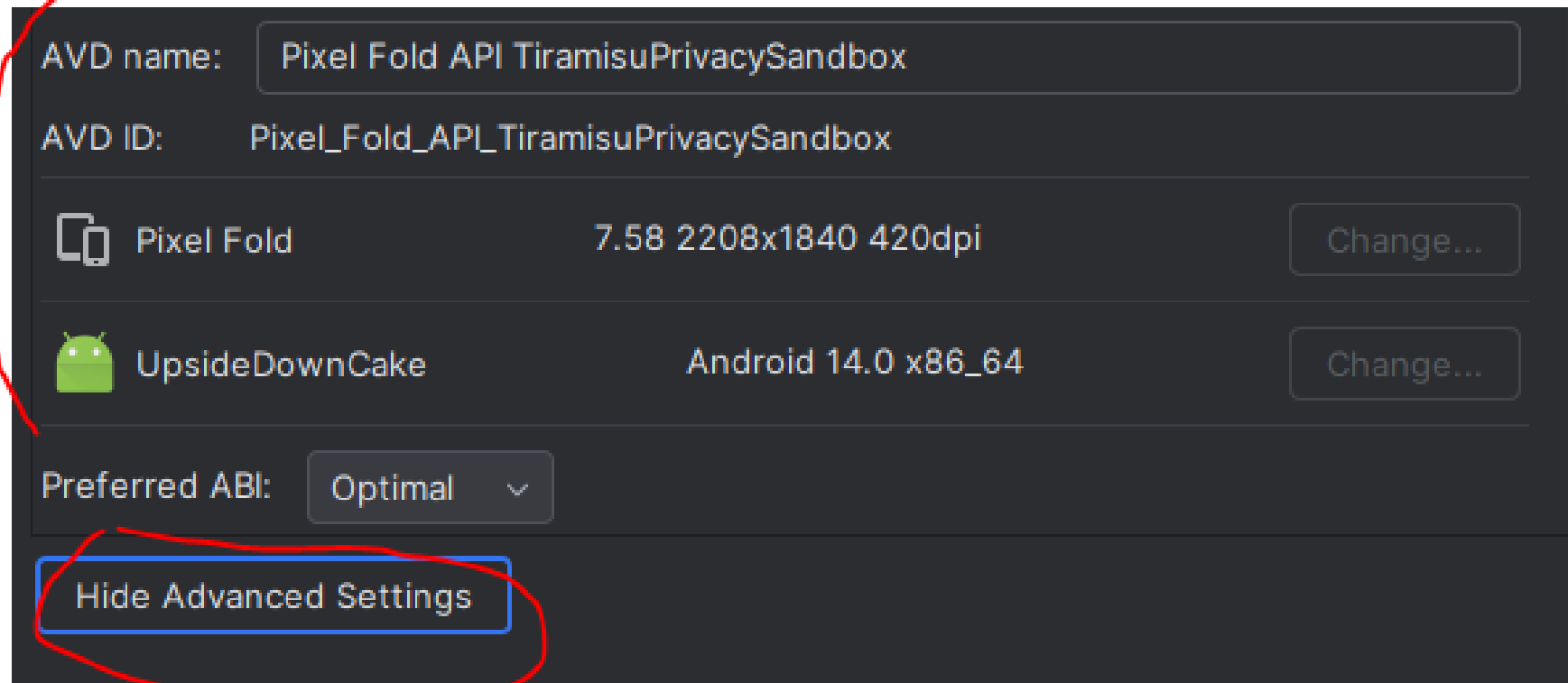
Choisir le mode d'affichage de l'application



Ajout d'un AVD

❑ Ajout d'un telephone virtuel

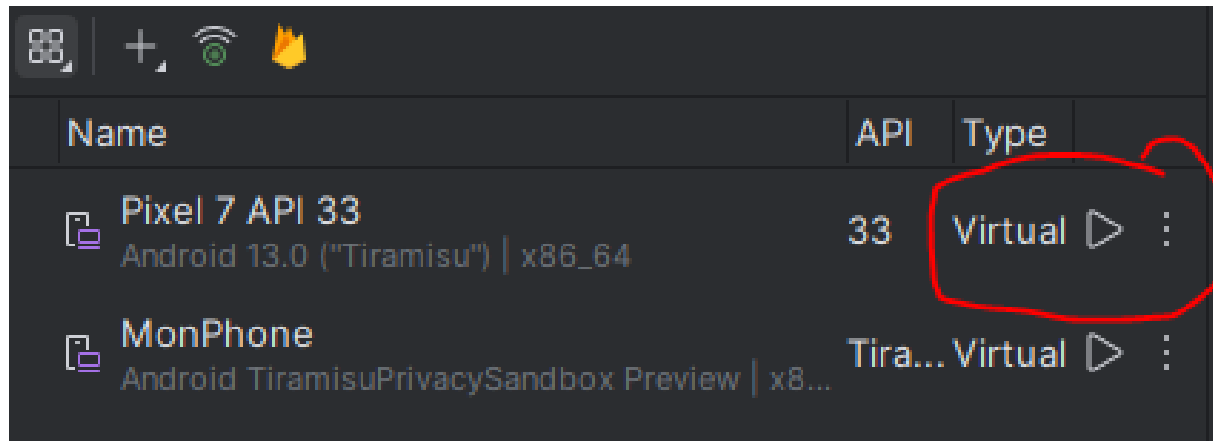
Cliquer sur Advancing Setting pour ajouter des infos supplémentaires



Ajout d'un AVD

❑ Ajout d'un telephone virtuel

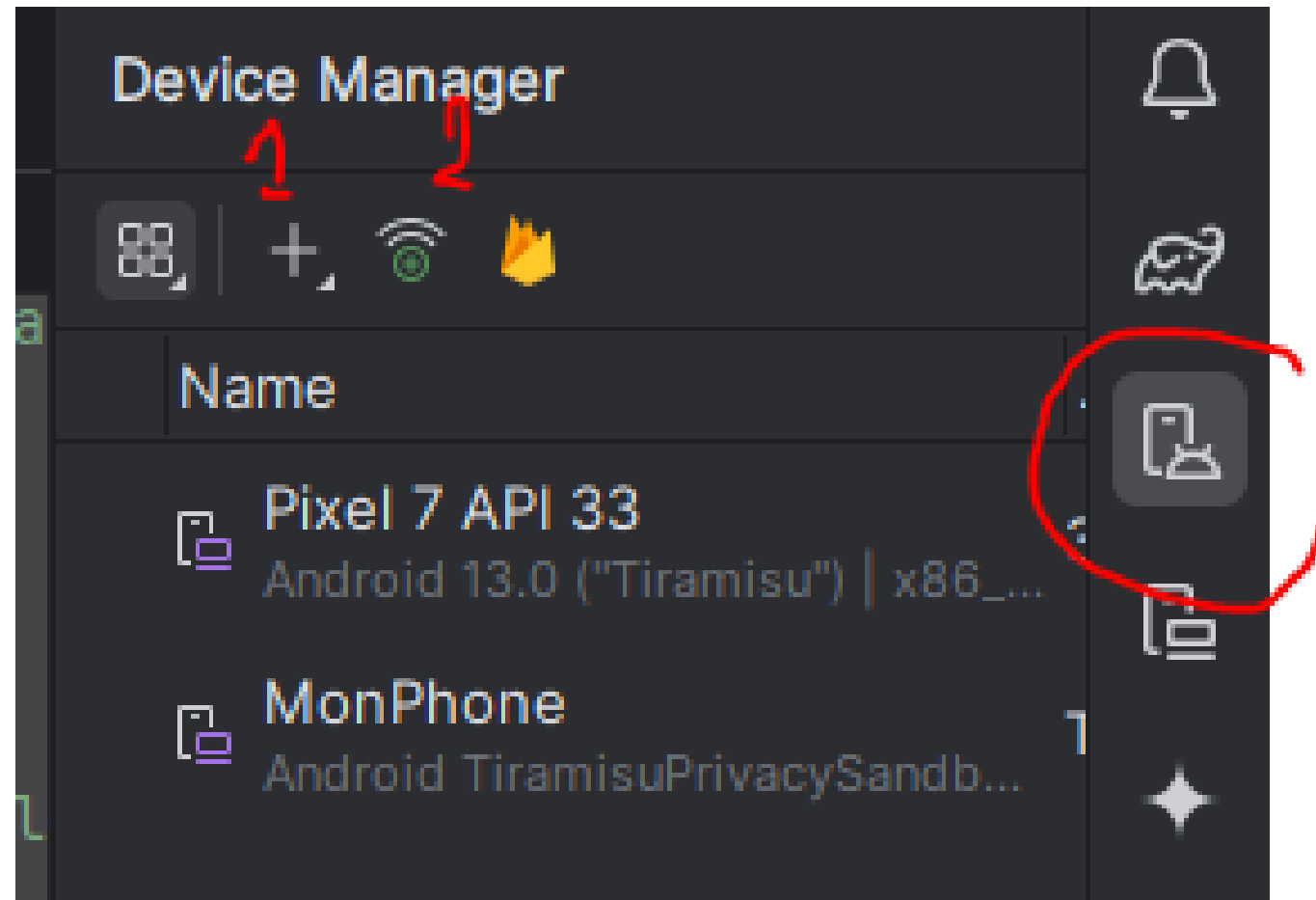
Cliquer sur le bouton de démarrage pour lancer le téléphone virtuel



Ajout d'un AVD

□ Ajout d'un telephone physique

Cliquer sur le 2



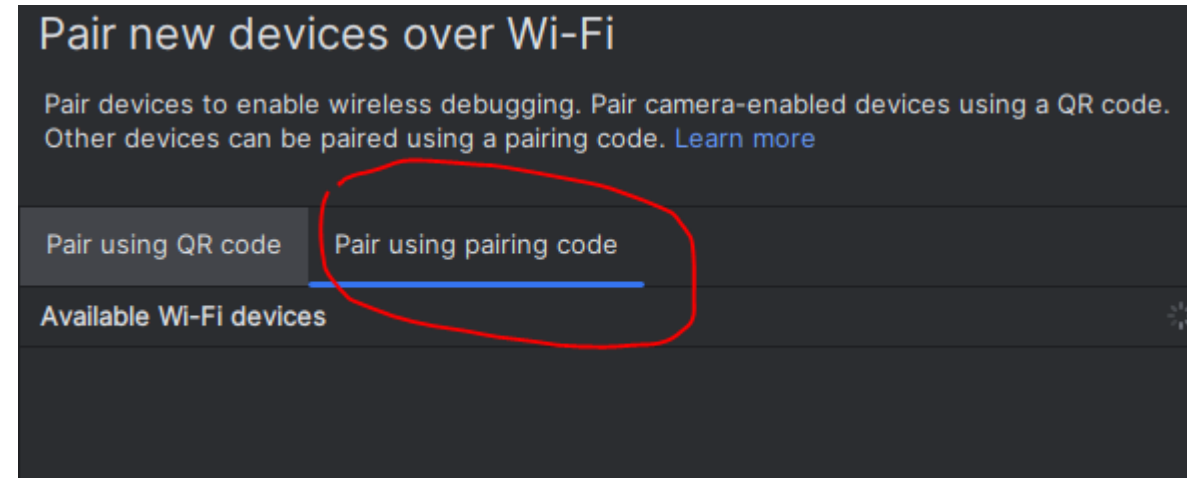
Ajout d'un AVD

❑ Ajout d'un telephone physique (3 modes)

Mode QR-code



Mode pairing code



Mode débogage USB



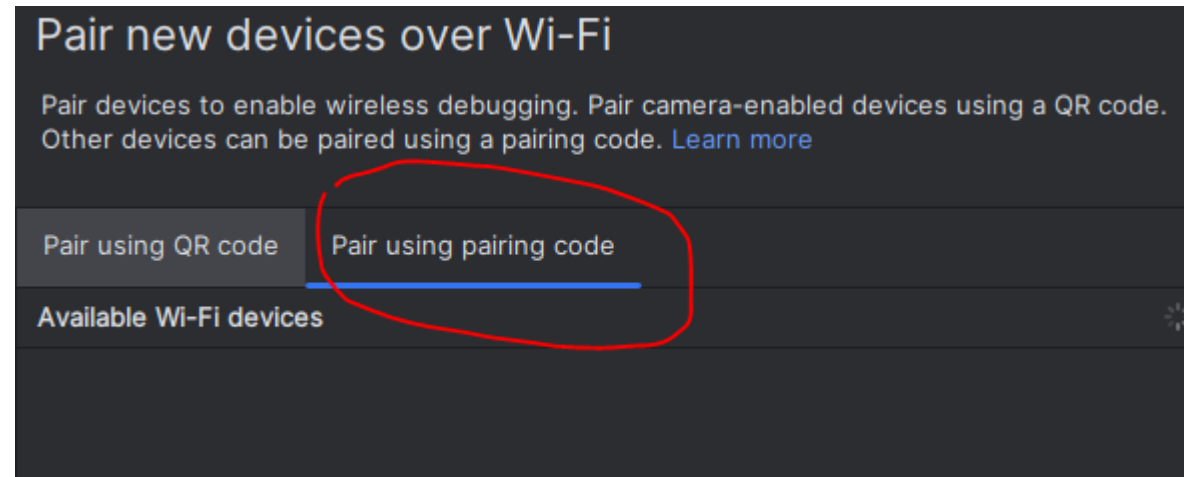
Ajout d'un AVD

❑ Ajout d'un telephone physique (3 modes)

Mode QR-code



Mode pairing code



Mode débogage USB



Eléments de bases de la programmation Androis

☐ Voir cours android