

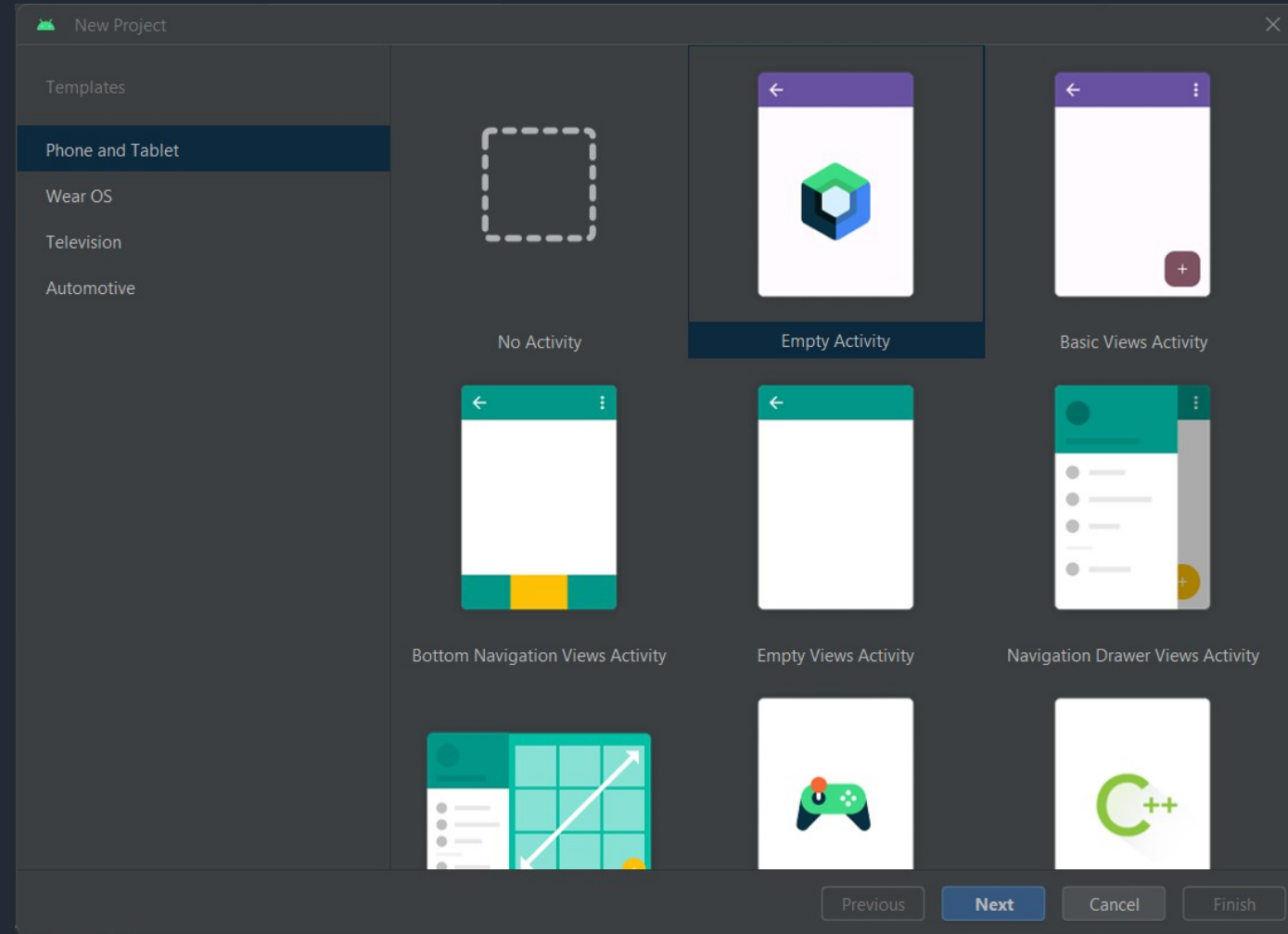
# Premières Bases



# Commencer une nouvelle application 1/2

Choisir une activité selon vos besoin:

- Dans ce cours nous ciblons les tablettes tactiles (smartphones)
- Commencer une empty view Activity



# Commencer une nouvelle application 2/2

- Save Location: par défaut dans le dossier User/name/AndroidStudioProjects
- Pour ce cour nous utiliserons le langage Kotlin
- Vous choisissez le SDK correspondant à la tablette Android que vous possédez

New Project

Empty Views Activity

Creates a new empty activity

Name: My Application

Package name: com.example.myapplication

Save location: C:\Users\Yassine\Desktop\Cours\DeveloppementMobile\_1\_Bases\Mini-Projets

Language: Kotlin

Minimum SDK: API 30 ("R"; Android 11.0)

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

ⓘ Your app will run on approximately 54,4% of devices.  
[Help me choose](#)

⚠️ 'Mini-Projets' already exists at the specified project location and it is not empty.

Previous Next Cancel Finish

# La conception d'une application sous Android Studio

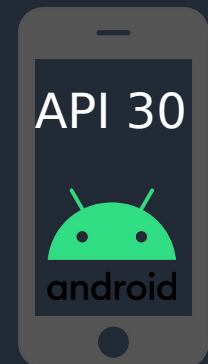
Les informations à connaître sur l'application que l'on développe:

- API (Application Programming Interface) -> « interface de programmation d'application » :
  - Connecte des applications entre elles
  - pour savoir sur qu'elles tablettes l'application peut être appliquée (niveau D'API)
  - Chaque nouvelle API implemente de nouvelles fonctionnalités -> (bibliothèques & améliorations)

1. Application Target API 2



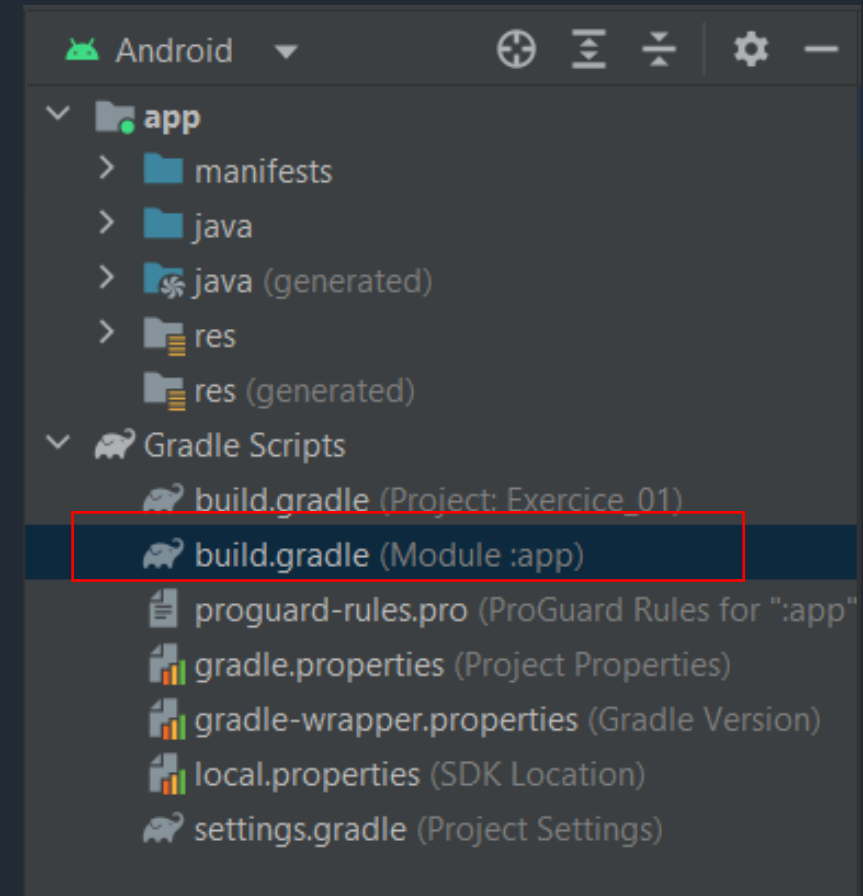
2. Application Target API 31









# La conception d'une application sous Android Studio

Certaines dépendances implémentent des versions trop récentes de certaines bibliothèques et leurs fonctionnalités : elles nécessitent un projet avec une API de niveau 33 minimum ce qui ne correspond pas à la majorité des tablettes à l'heure actuelle.

- Il ne faut donc pas hésiter à modifier les versions des bibliothèques importer dans les dépendances en choisissant des versions ultérieures.
- Ces changements se font au niveau du fichier Gradle où l'on trouve : les dépendances (importation des bibliothèques utilisées) et les plugins.
- Lorsque l'on change de version l'on appuie sur syn (synchroniser)



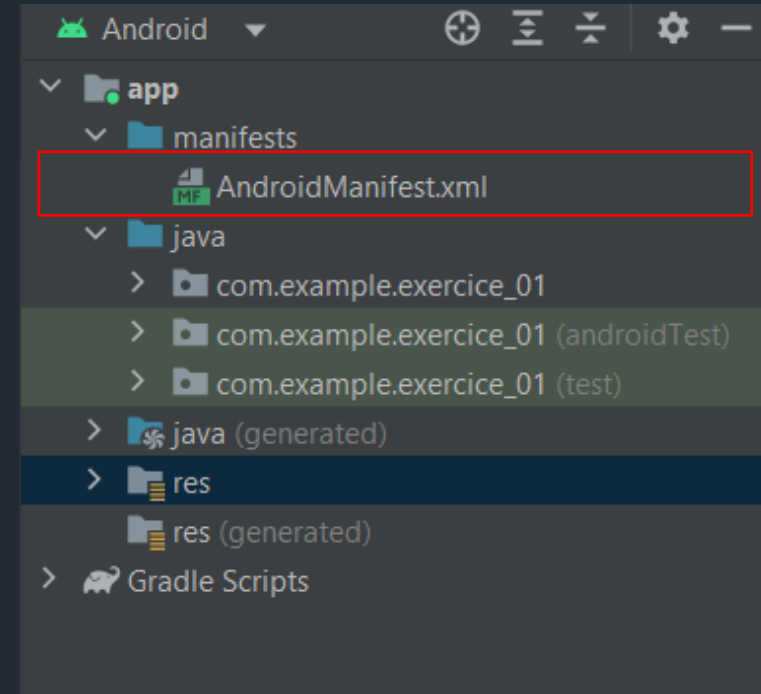
# UI (User Interface) – Interface Utilisateur

-  UI de l'SDK Android Studio
-  Arborescence -> gestion de l'emplacement de vos fichiers
-  Lignes de codes || Interface visuel -> Layouts & images ...
- 
-  Device Manager || Attributs -> Layouts ect...
-  Console: débbugger, erreurs, Logcat



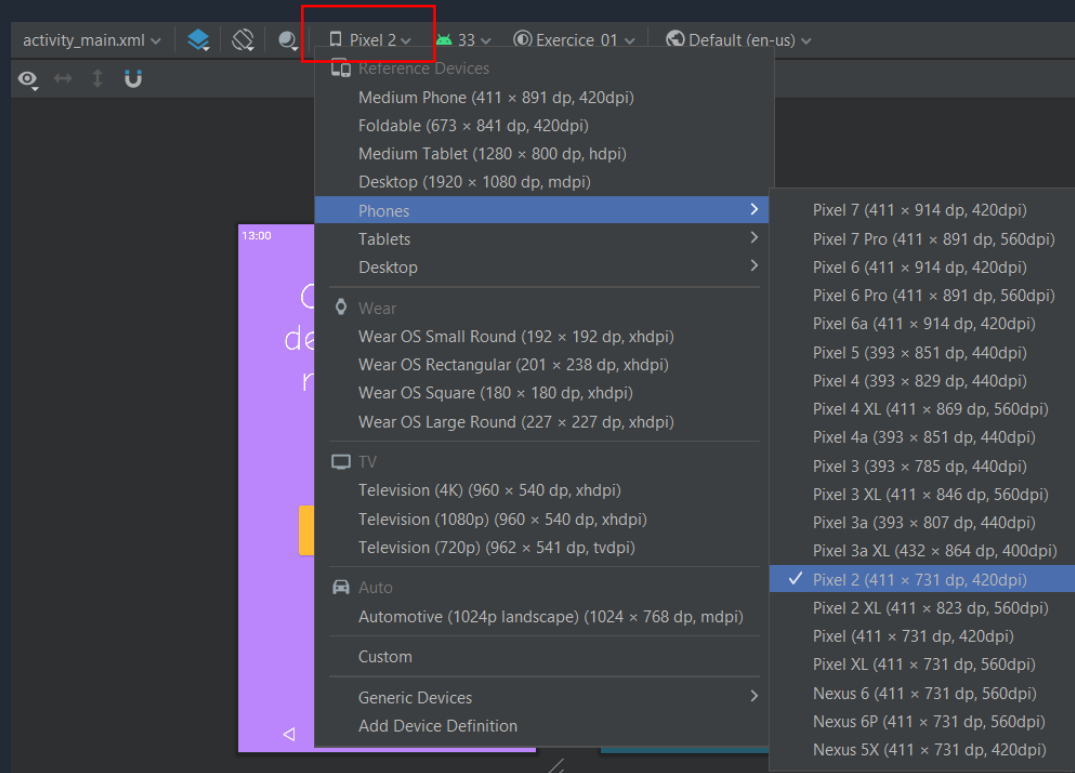
# Le fichier AndroidManifest.xml

- Il s'agit d'un fichier .xml
- Il déclare les caractéristiques de l'application:
  - Le titre
  - Le niveau d'API
  - Les activités
  - La version de OS mobile
  - Définir l'activité d'ouverture de l'application
  - Etc.
- Ce qui permet de vérifier l'éligibilité de votre smartphone avant le téléchargement de l'application



# Introduction au responsive Design – part 1/2

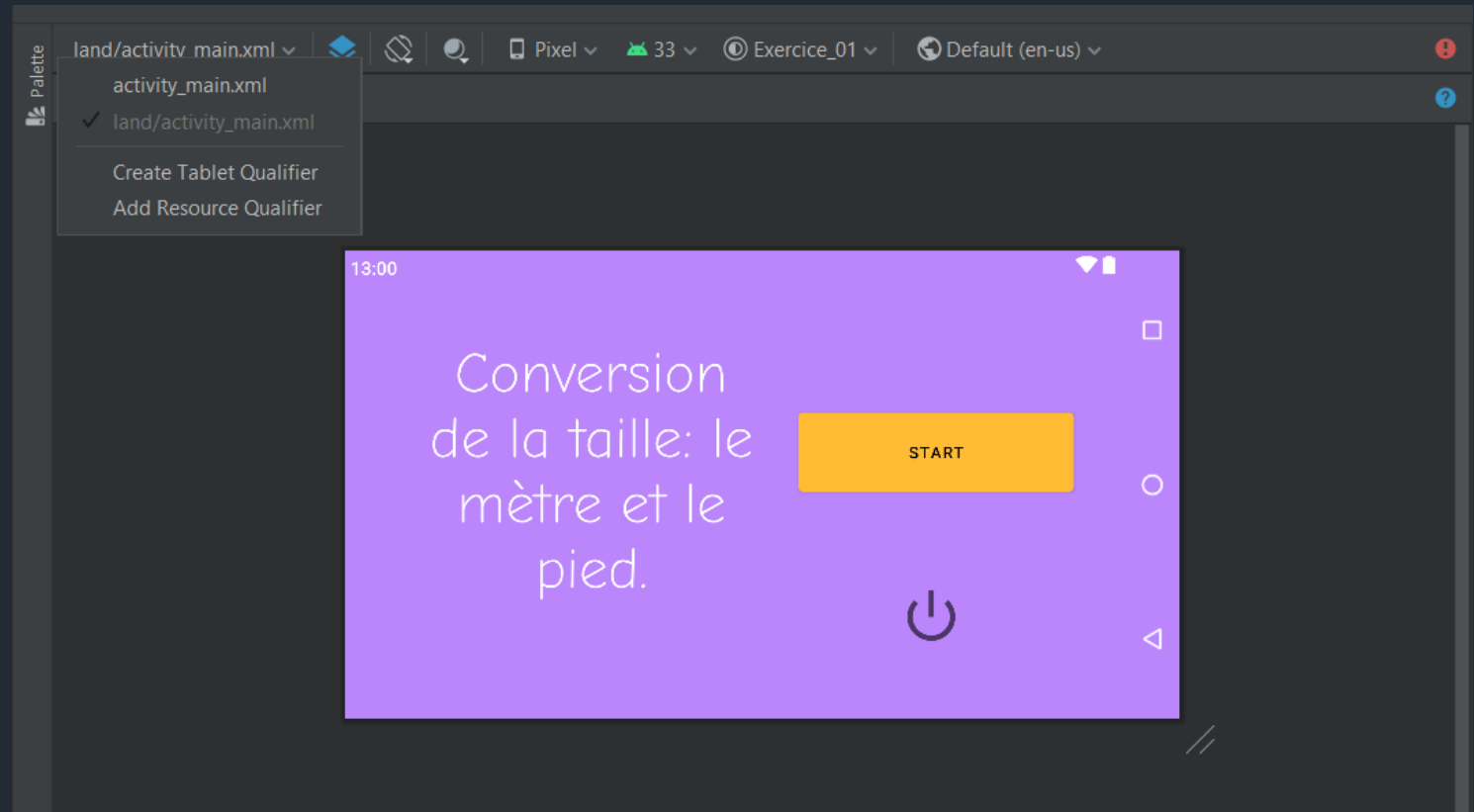
- Choisir une référence approximative de celle de votre smartphone





# Introduction au responsive Design – part 2/3

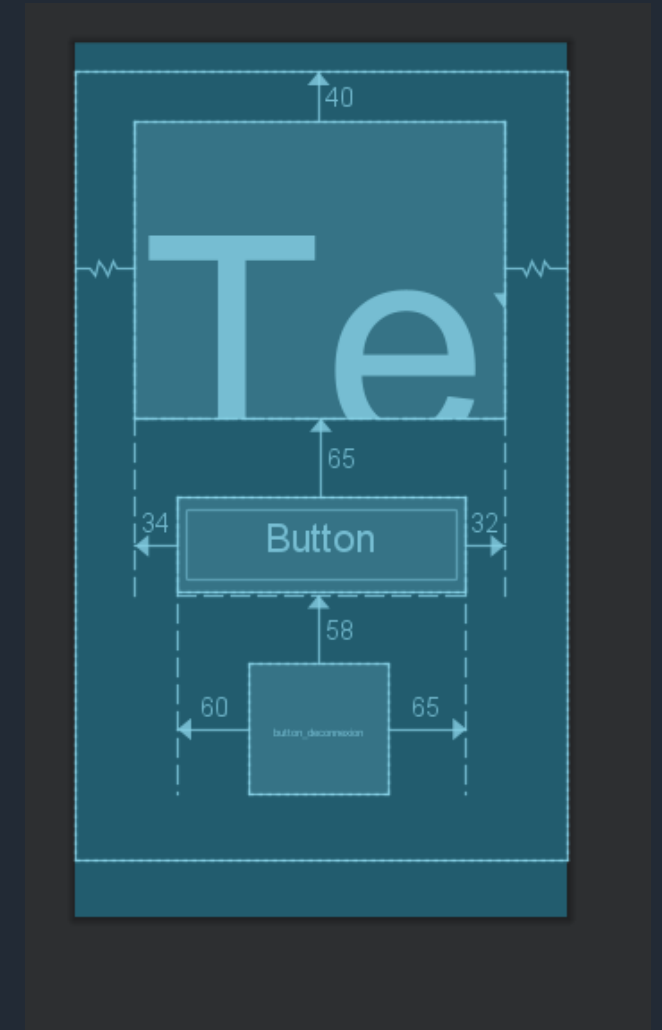
- Permettre un changement de la configuration 6> Mode rotation -> Deux layouts pour les modes portraits/paysages
- Réagencer les widgets selon votre IHM (Interface Humain-Machine).
- Pas besoin de toucher au programme (.kt) l'outil Android Studio se charge d'intégrer votre mode paysage (Landscape) dans votre application.



# Introduction au responsive Design – part 3/3

- Les premières bonnes habitudes:
  - Disposition des Widgets (composants -> boutons, textes etc...) les uns selon les autres -> qu'ils soient groupés
  - Pour les attributs des dimensions (Width et Height) -> par rapport à la taille de l'écran (« match parent » || « wrap-content »)

Pour agencer librement les widgets dans l'interface visuelle nous utiliserons `<RelativeLayout>` -> en tant que conteneur principal.



# Le cycle de vie d'une application

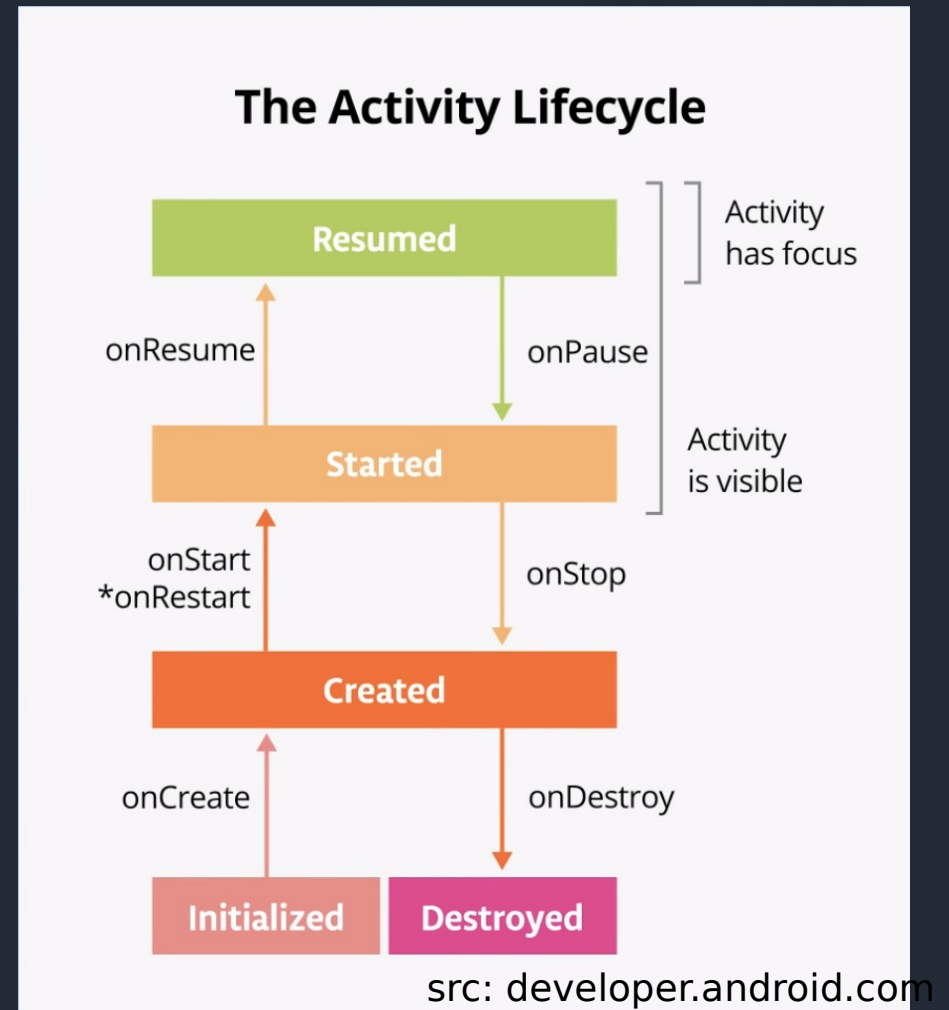
- Une application peut être fermée (notifications)
- Une application peut être ouverte (sur l'écran)
- Une application peut être en pause (pas sur l'écran)

1. Liens:

<https://developer.android.com/codelabs/basic-android-kotlin-compose-activity-lifecycle?hl=fr#0>

2. Quel sont les différents états d'une application?

3. Pourquoi devons nous les utiliser?



# Comment compiler les applications?

## **Solution 1:** Le téléversement

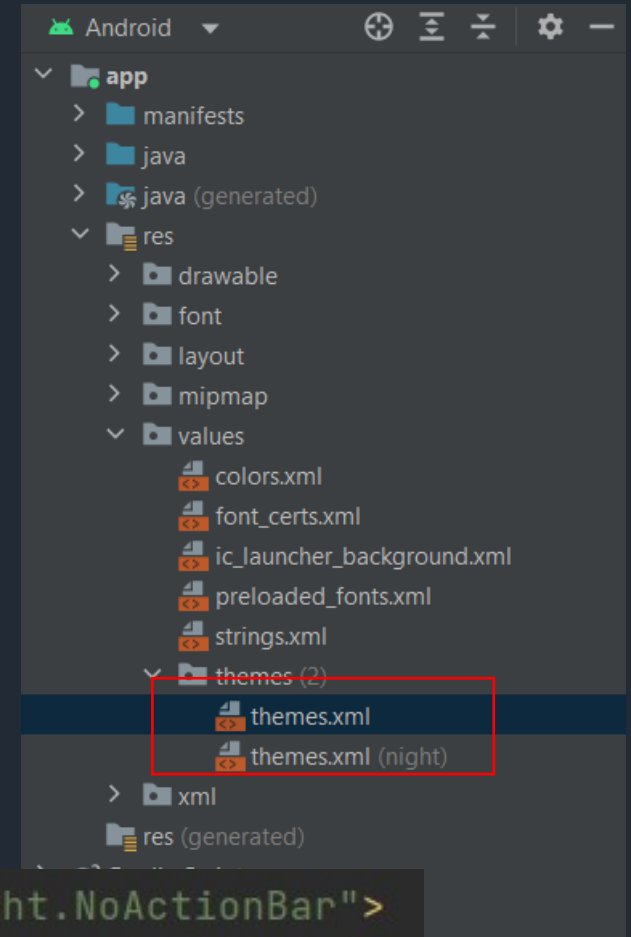
- Être en possession d'une tablette -> Android
- Un câble Usb de transfert de données
- Activer le mode développeur
  - Permet la simulation à taille réelle
  - Permet de tester/valider les interactions
  - **Faire attention aux applications dangereuses!**

## **Solution 2:** La machine virtuelle

- Installer un smartphone virtuel -
- Simuler l'application sur le smartphone virtuel
  - **Fatigue votre ordinateur!**
  - **N'est pas suffisante pour valider les interactions!**

# Les thèmes (le titre => actions bar)

- Pour supprimer la barre de titre de votre application :
  - On ouvre le fichier xml theme.xml
  - On met NoActionBar



```
<style name="Theme.Exercice_01" parent="Theme.MaterialComponents.DayNight.NoActionBar">
```

# Les thèmes

- Fichier androidManifest.xml, c'est ce fichier qui définit où se trouve les thèmes – par défaut déjà créé

```
<activity  
  android:name=".ListActivity"  
  android:exported="true"  
  android:theme="@style/Theme.MiniProjet_LaListe">
```

- Les deux lignes suivantes attribuent les thèmes (couleurs) des barres d'actions et des boutons du système

```
<style name="Theme.MiniProjet_LaListe" parent="Theme.MaterialComponents.DayNight.NoActionBar">
```

```
<item name="android:statusBarColor">?attr/colorPrimaryVariant</item>
```

# Modifier l'icone de l'application

1. Clique droit -> sur le dossier RES
2. Sélectionner: Nouveau/Image Asset
3. Sélectionnée l'image pour l'icone -> centrer l'image
4. Choisir l'option main pour appliquer l'image dans les fichiers dédiés à l'icone



## Configure Image Asset

Icon type: Launcher Icons (Adaptive and Legacy) ▼

Name: ic\_launcher

Foreground Layer Background Layer Options

Layer name: ic\_launcher\_foreground

Source Asset

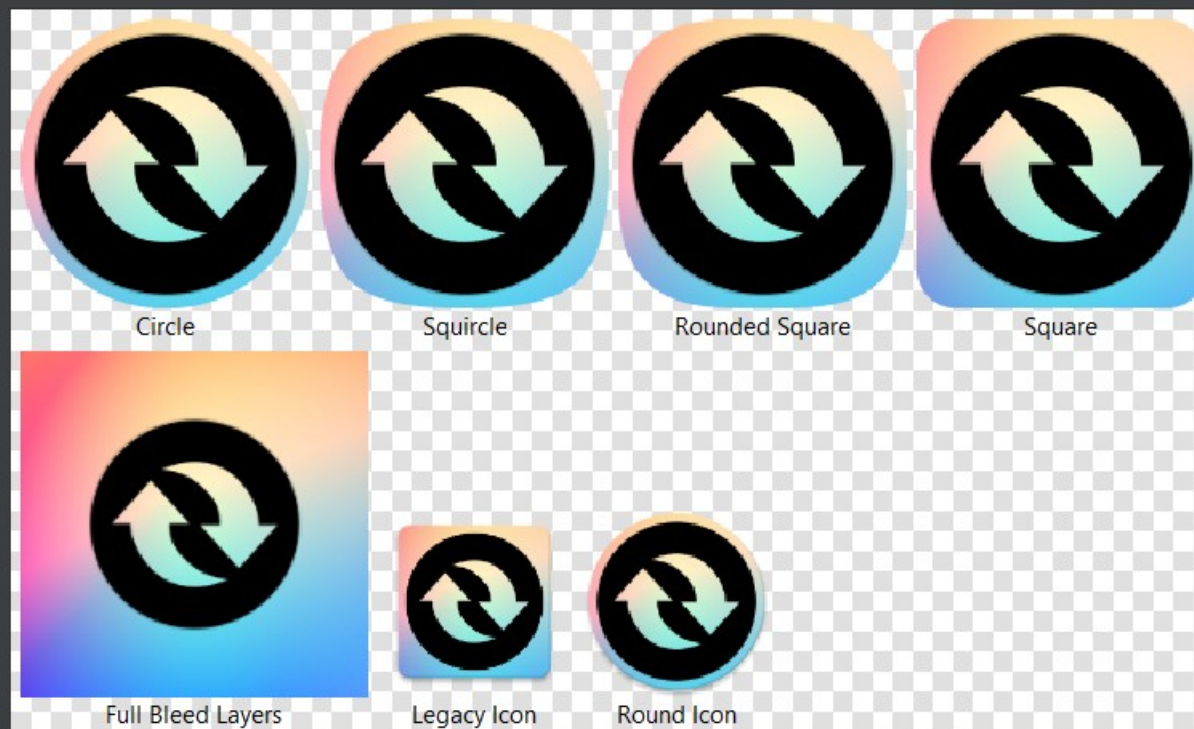
Asset type: ☒ Image ☐ Clip art ☐ Text

Path: C:\Users\Yassine\Desktop\Cours\Devel

Scaling

Trim: ☐ Yes ☒ No

Resize: 60 %

Preview xhdpi ☒ Show safe zone ☐ Show grid

⚠ An icon with the same name already exists and will be overwritten.



Previous

Next

Cancel

Finish





## Confirm Icon Path

Source set:

main *src/main/res*

Output directories:

main *src/main/res*debug *src/debug/res*release *src/release/res*androidTest *src/androidTest/res*androidTestDebug *src/androidTestDebug/res*

mipmap-xxhdpi



ic\_launcher.webp



ic\_launcher\_background.webp



ic\_launcher\_foreground.webp



ic\_launcher\_round.webp



mipmap-xhdpi



ic\_launcher.webp



ic\_launcher\_background.webp



ic\_launcher\_foreground.webp



ic\_launcher\_round.webp



mipmap-xhdpi



ic\_launcher.webp

Output file

File type:

PNG File

Density:

nodpi

Size (dp):

512x512

Size (px):

512x512



Some files (shown in red) will overwrite existing files.



Previous

Next

Cancel

Finish