TP°2 : Naviguer entre des écrans

Exercice 1 :
Création et lancement d'une Activity principale

1. Créer une Activity dans `java/com/ensiie/tp2`
2. Overrider la méthode `onCreate()` et affecter le layout `activity_exercice_01.xml`
3. Déclarer l'Activity dans `AndroidManifest.xml`
4. Lancer l'application et vérifier que l'Activity se lance.

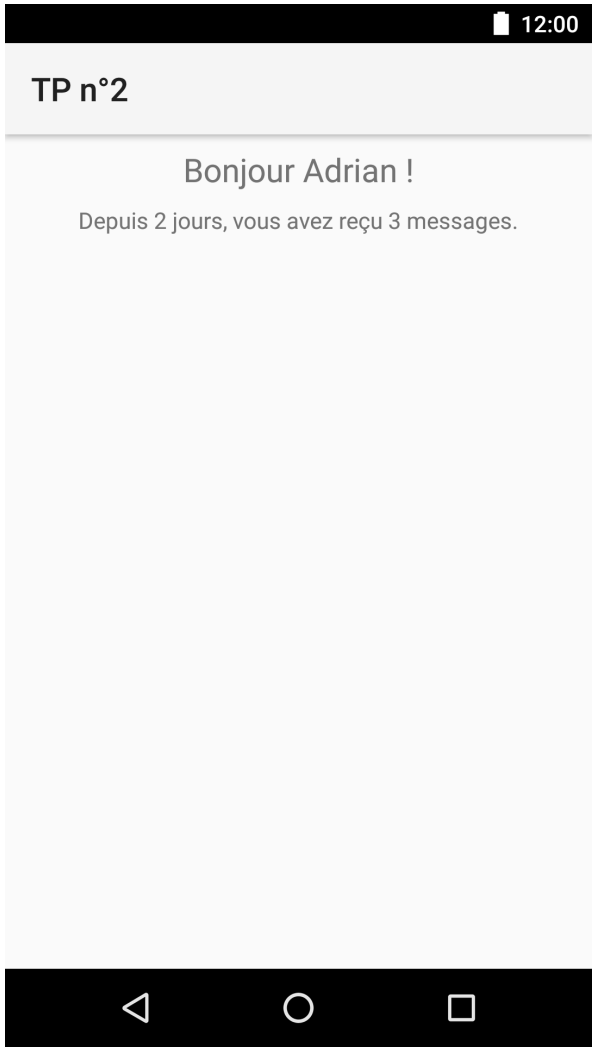
Exercice 2 : Changer l'icône et le nom de l'app

A partir de l'AndroidManifest :

- Mettre l'icône `ic_launcher_new`
- Mettre le nom `app_name_new`

Exercice 3 : Affecter un texte avec un placeholder

- Créer une Activity principale avec le layout `activity_exercice_03.xml`
- Remplacer le placeholder %s du texte `R.string.exercice_3_title` avec un texte et l'affecter à la TextView `R.id.title`
- Remplacer les placeholders %1$d et %2$d du texte `R.string.exercice_3_description` avec deux valeurs numériques et l'affecter à la TextView `R.id.description`



Info

Utiliser `String.format(textWithPlaceholders, valueToReplacePlaceholder1, valueToReplacePlaceholder2)` pour remplacer des placeholders dans un texte.

Exercice 4 :
Changer le style du texte dynamiquement

- Créer une Activity principale avec le layout `activity_exercice_04.xml`
- Affecter un listener `OnClickListener` au bouton `R.id.button`
- Au clic sur le bouton, appliquer la couleur `red` définie dans le fichier `colors.xml` à la `TextView`

Exercice 5 :
Utiliser le même listener pour plusieurs vues

- Créer une Activity principale avec le layout `activity_exercice_05.xml`
- Affecter le même `OnClickListener` aux trois boutons
- Au clic d'un bouton, affecter le texte du bouton cliqué à la `TextView`

Info

La méthode `OnClick(View v)` du listener `OnClickListener` donne en paramètre la view cliquée.

Exercice 6 : Passer des données à un listener utilisé pour plusieurs vues

- Créer une Activity principale avec le layout `activity_exercice_06.xml`
- Pour chaque `ImageButton`, affecter en tant qu'image le drawable correspondant (ex: `R.drawable.image_button_1` pour l'`ImageButton` `R.id.image_button_1`)
- Pour chaque `ImageButton` affecter également dans son tag le drawable affecté précedemment
- Affecter le même `OnClickListener` aux trois `ImageButtons`
- Au clic d'un `ImageButton`, récupérer le drawable affiché de l'`ImageButton` à partir de son tag et l'affecter à l'`ImageView` `R.id.image`

Info

Il n'existe pas de moyen simple de récupérer le drawable affiché dans un `ImageButton`.

Une des techniques est d'utiliser la méthode `setTag(Object o)`. Cette méthode permet d'affecter un object arbitraire à une view. La méthode `getTag()` permet de récupérer cette objet plus tard.

Exercice 7 :
Changer visibilité d'une vue dynamiquement

- Créer une Activity principale avec le layout `activity_exercice_07.xml`
- Affecter un listener `OnCheckedChange` à la `CheckBox`
- Quand `CheckBox` cochée :
  - Rendre visible la `TextView R.id.description`
  - Rendre invisible le `TextView R.id.title`
- Quand `CheckBox` pas cochée : remettre état initial.

Info

Une vue peut avoir 3 types de visibilité différentes :

- **`View.VISIBLE`** : la vue est affichée à l'écran
- **`View.INVISIBLE`** : la vue n'est pas affichée à l'écran mais prend la place nécessaire comme si elle était transparente
- **`View.GONE`** : la vue n'est pas affichée à l'écran

`setVisibility()` permet de changer la visibilité d'une view

Exercice 8-1 : Lancer une deuxième Activity

- Créer une Activity principale avec le layout `activity_exercice_08_1.xml` et une Activity avec le layout `activity_exercice_08_2.xml`
- Au clic sur le bouton « Lancer Activity sans donnée » de l'Activity principale, lancer la deuxième Activity.

Exercice 8-2 : Passage d'une donnée à la deuxième Activity

Au clic sur le bouton « Lancer Activity avec donnée », lancer la deuxième Activity en passant le texte saisie dans l'`EditText` :

- Récupérer le texte saisie dans l'`EditText`
- Placer le texte dans un Intent et le passer à la deuxième Activity
- Récupérer l'extra dans la deuxième Activity, si le texte n'est pas vide, l'affecter à la `TextView` `R.id.text_from_activity_08_1` et la rentre visible

Info

`myEditText.getText().toString()` permet de récupérer le texte d'un EditText.

`TextUtils.isEmpty()` permet de tester si un texte est vide ou non

Exercice 8-3 : Retourner une donnée à une Activity

Au clic sur le bouton « Lancer Activity et attente résultat », lancer la deuxième Activity et retourner le texte saisie dans l'`EditText` à la première Activity :

- Lancer la deuxième Activity en attendant un résultat

Au clic sur le bouton « Retour Activity » :

- Retourner le texte saisie dans l'`EditText` à la première Activity
- Récupérer le texte envoyé par la deuxième Activity et s'il n'est pas vide, rendre visible la `TextView` `R.id.text_from_activity_08_2` et l'affecter

Exercice 9 : Partager un texte

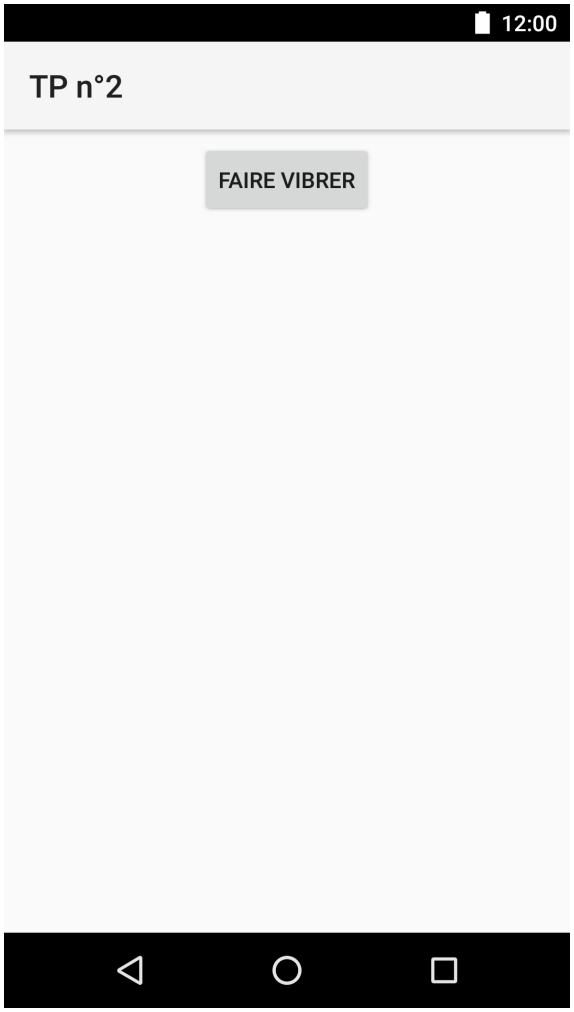
- Créer une Activity principale avec le layout `activity_exercice_09.xml`
- Au clic sur le bouton « Partager texte », partager le texte de l'`EditText` à une autre application à l'aide d'une implicit intent.

Exercice 10 : Lifecycle d'une Activity

- Lancer l'Activity `TimerActivity` comme Activity principale.
- Modifier `TimerActivity` pour que :
  - A l'appuie sur HOME le timer s'arrête (`pauseTimer()`)
  - Au retour sur l'Activity, le timer se relance (`playTimer()`)

Exercice 11 : Déclaration d'une permission normale

- Créer une Activity principale avec le layout `activity_exercice_11.xml`
- Au clic sur le bouton « Faire vibrer », appeler la méthode `Utils.vibratePhone()`:
  - Constater le crash dans les logs
  - Apporter un correctif en déclarant la bonne permission



Exercice 12 : Demande une permission dangereuse à partir du code

- Lancer l'Activity `DangerousPermissionActivity` comme Activity principale
- Entrer un numéro de téléphone et un message
- Cliquer sur envoyer et constater le crash
- Déclarer la bonne permission dans le manifest
- Au clic sur le bouton, vérifier si l'utilisateur a déjà accepté la permission à partir du code (`hasSendSmsPermission()`)
- Si oui, envoyer SMS. Sinon, demander la permission a l'utilisateur (`askSendSmsPermission()`)
- Si permission acceptée, envoyer SMS. Sinon, afficher un message d'erreur (`Utils.`*`showMessage()`*)

Exercice 13 : Reproduire un lifecycle

- Installer l'app lifecycle.apk sur le téléphone
  - utiliser adb (ANDROID_SDK/platform-tools/)
  - adb install lifecycle.apk
- Lancer l'app et ouvrir les logs dans Android Studio
- Filtrer les logs sur « LIFECYCLE: »
- Reproduire les logs affichés en utilisant l'application (Chaque méthode des lifecycles des Activities de l'application est loguée)

```
D/Activity1: LIFECYCLE: onCreate
D/Activity1: LIFECYCLE: onStart
D/Activity1: LIFECYCLE: onResume
D/Activity1: LIFECYCLE: onPause
D/Activity1: LIFECYCLE: onStop
D/Activity1: LIFECYCLE: onDestroy
D/Activity1: LIFECYCLE: onCreate
D/Activity1: LIFECYCLE: onStart
D/Activity1: LIFECYCLE: onResume
D/Activity1: LIFECYCLE: onPause
D/Activity1: LIFECYCLE: onResume
D/Activity1: LIFECYCLE: onPause
I/Activity2: LIFECYCLE: onCreate
I/Activity2: LIFECYCLE: onStart
I/Activity2: LIFECYCLE: onResume
D/Activity1: LIFECYCLE: onStop
I/Activity2: LIFECYCLE: onPause
I/Activity2: LIFECYCLE: onStop
I/Activity2: LIFECYCLE: onRestart
I/Activity2: LIFECYCLE: onStart
I/Activity2: LIFECYCLE: onResume
I/Activity2: LIFECYCLE: onPause
I/Activity2: LIFECYCLE: onResume
I/Activity2: LIFECYCLE: onPause
D/Activity1: LIFECYCLE: onRestart
D/Activity1: LIFECYCLE: onStart
D/Activity1: LIFECYCLE: onResume
I/Activity2: LIFECYCLE: onStop
I/Activity2: LIFECYCLE: onDestroy
D/Activity1: LIFECYCLE: onPause
D/Activity1: LIFECYCLE: onStop
```