



# ANDROID



# TD 3. Les fragments

# Premiers codes (codé généré pour « voir »)

- Nouveau projet par assistant avec 'empty activity'
  - puis ajout d'un fragment (Blank Fragment), sans le relier à l'activité
    - Pour voir le code généré (`newInstance`, `onCreate...`)
  - Puis ajout d'un ViewModel Fragment, sans le relier à l'activité
    - Pour voir le code généré, dont le `onActivityCreated` **deprecated** !
  - Puis ajout d'un List Fragment, sans le relier à l'activité
    - Pour voir le code généré (type `RecyclerView` qu'on verra + tard)

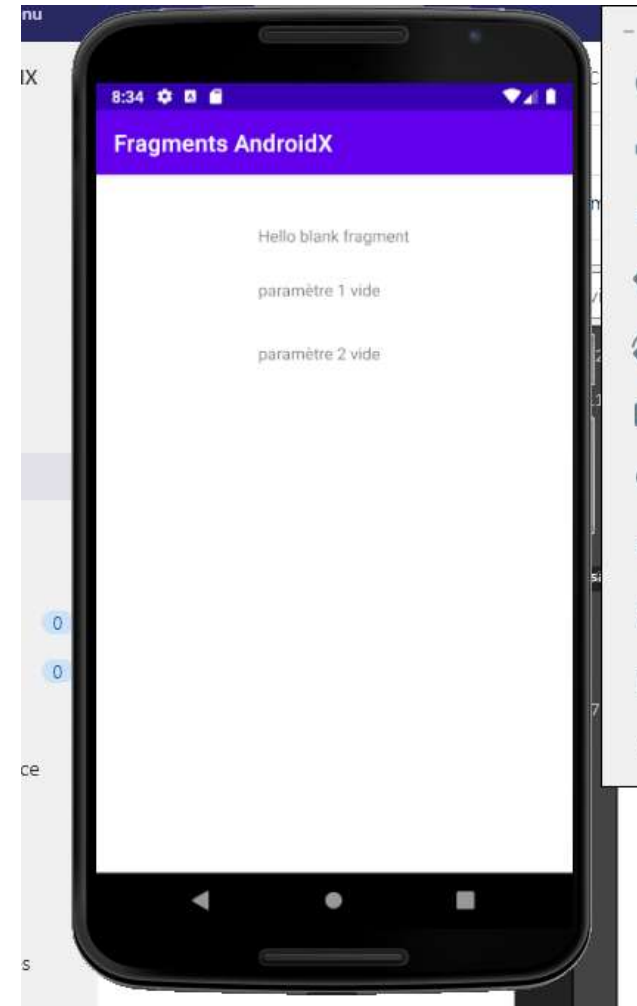
# Exemples vus en cours (rappel)

- Fragment et cycle de vie : <https://gitlab.com/m1-ue-tac/td3-fragments/cycle-de-vie-activite-et-fragment.git>
  - Cf. readme
  - Attention : ancien code (<fragment>)
- Activity et Fragments : <https://gitlab.com/m1-ue-tac/td3-fragments/activity-and-fragments.git>
  - Différents exemples d'affichage de fragments statiques (en dur, par bouton, par add/replace, recherche par tag pour éviter les doublons....)
  - Attention : ancien code (<fragment>)
- La doc officielle pour rappel : <https://developer.android.com/guide/fragments>

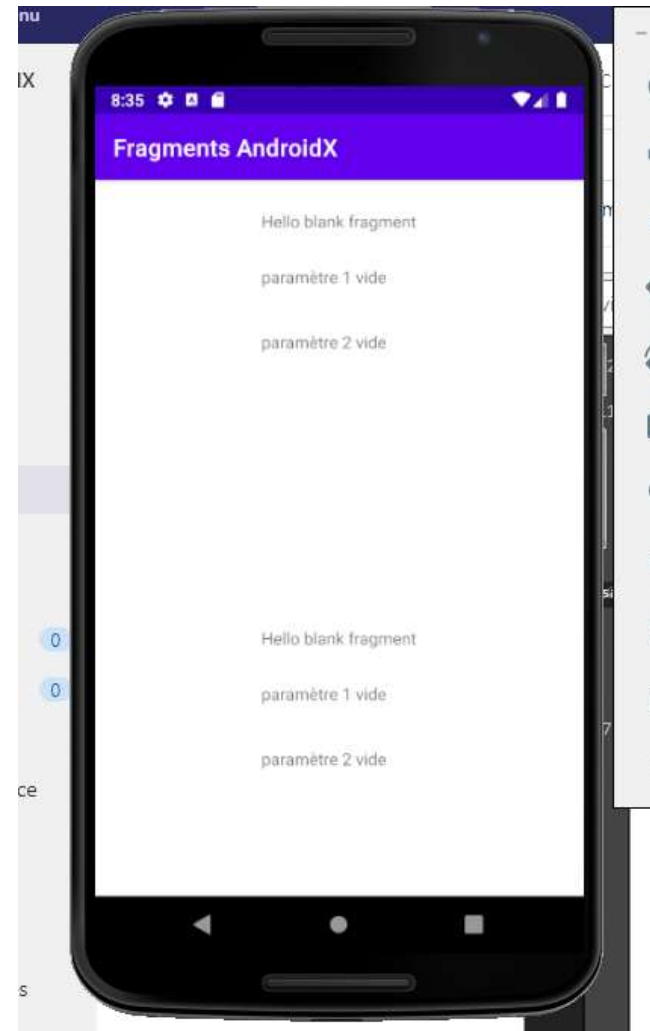
# Application(s) à réaliser (détails ci-après)

- Activité avec 1 fragment statique
  - Puis avec 2 fragments statiques (identiques ou non), affichés sur le même écran (50/50)
- Activité avec 1 fragment dynamique affiché au démarrage
- Activité avec 2 fragments dynamiques qui sont créés lorsqu'on clique sur le bouton associé
  - Puis qui reçoivent des données depuis l'activité
  - Puis qui renvoient des données à l'activité
  - Puis qui sont gérés par un ViewPager2
  - Puis avec de « jolis » boutons

# Activité avec 1 fragment statique

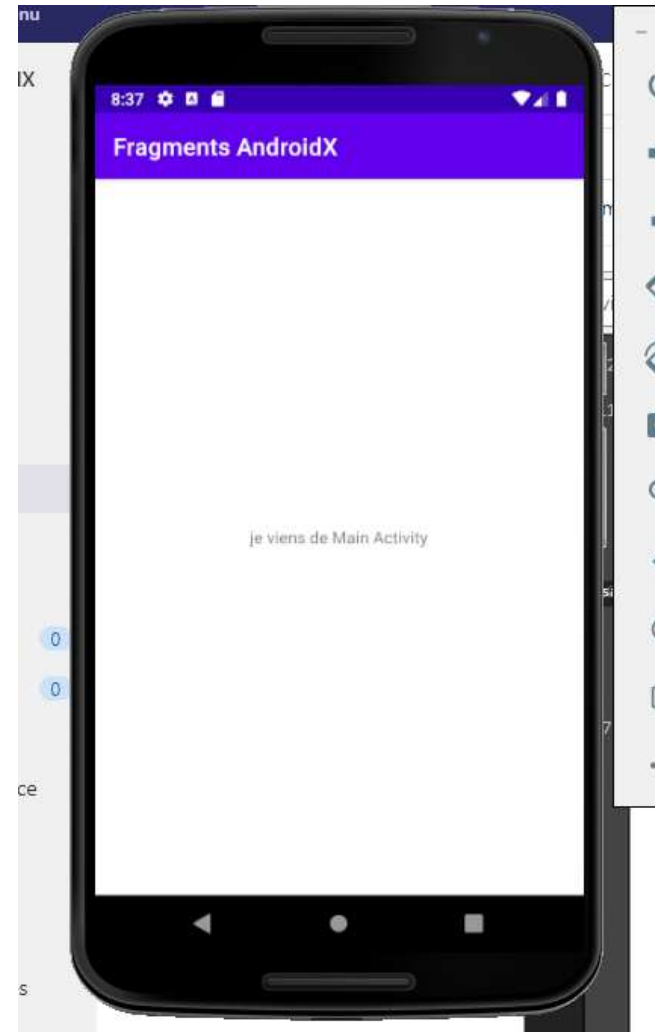


# Activité avec 2 fragments statiques (identiques)



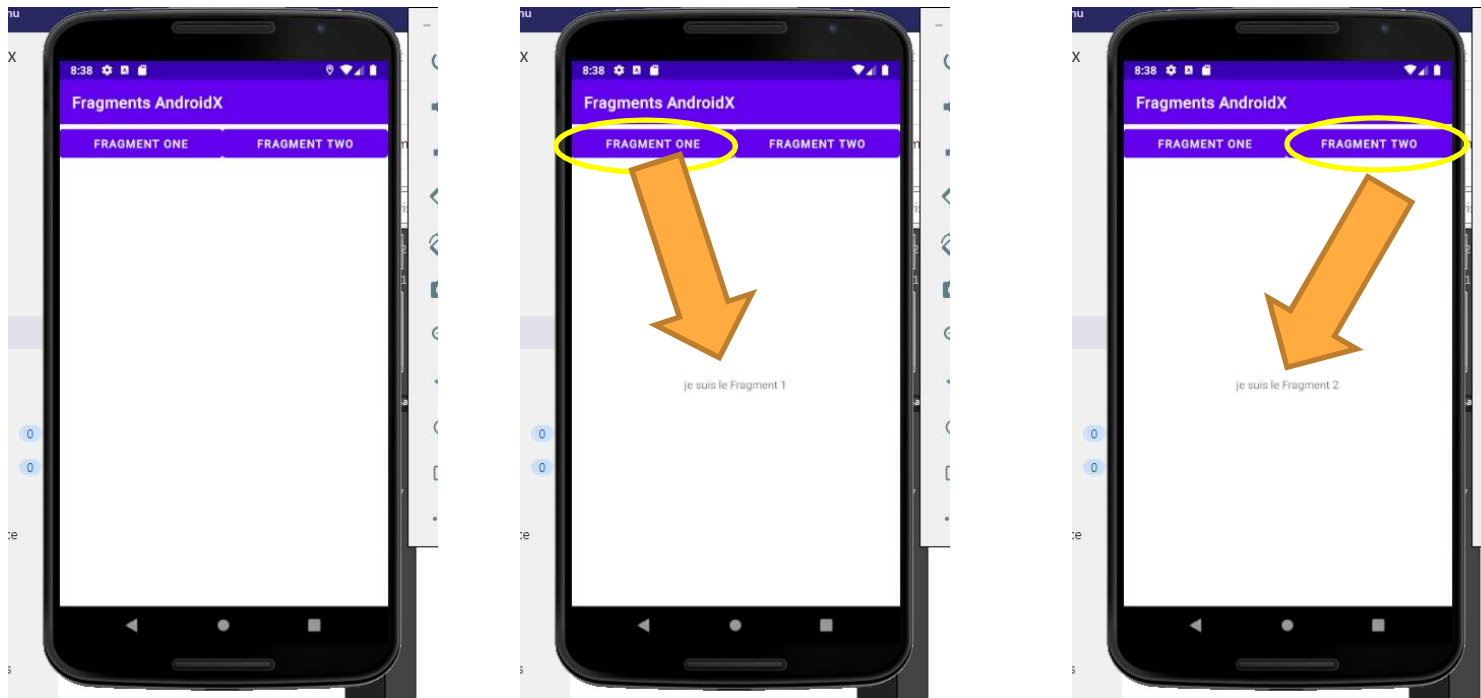
# Activité avec 1 fragment dynamique

- L'activité envoie un paramètre au fragment qui l'affiche

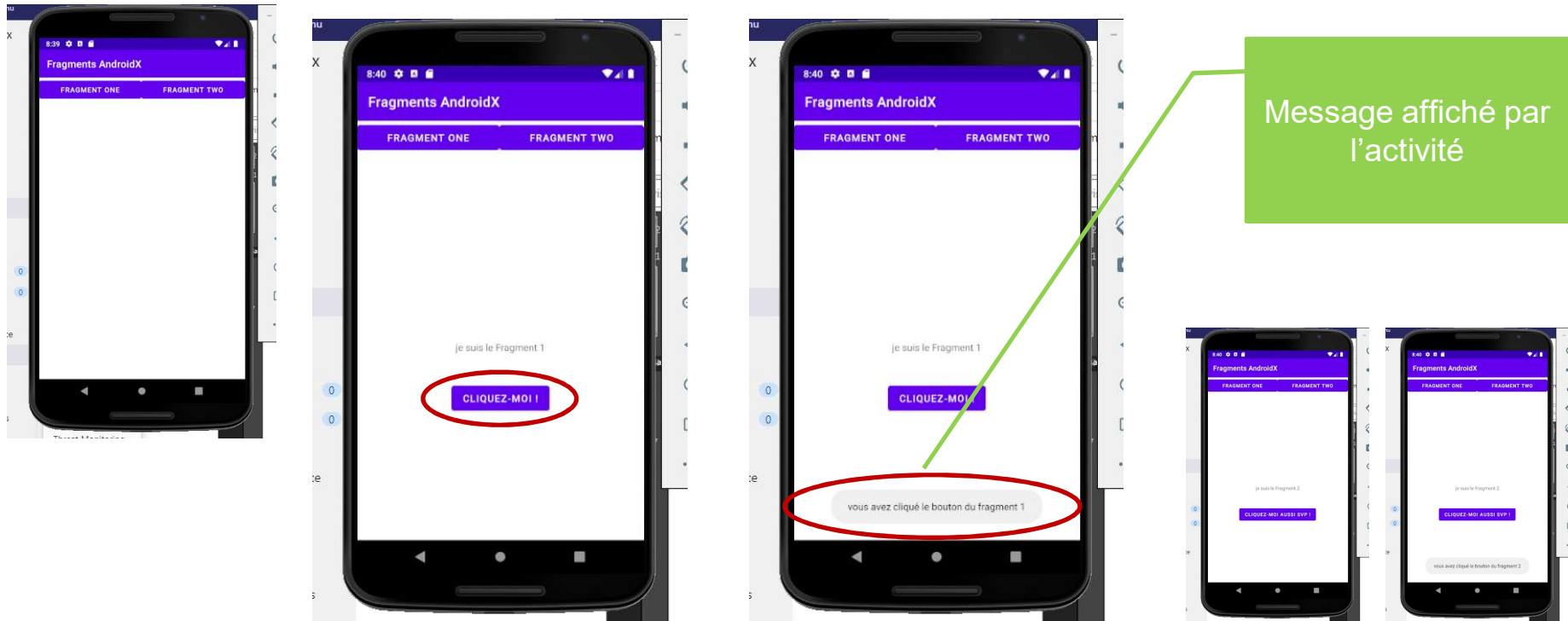




# Activité avec 2 fragments dynamiques qui sont créés lorsqu'on clique sur le bouton associé

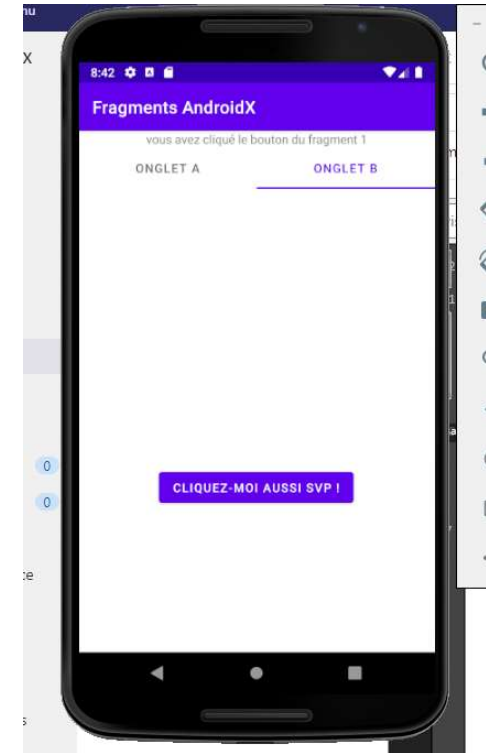
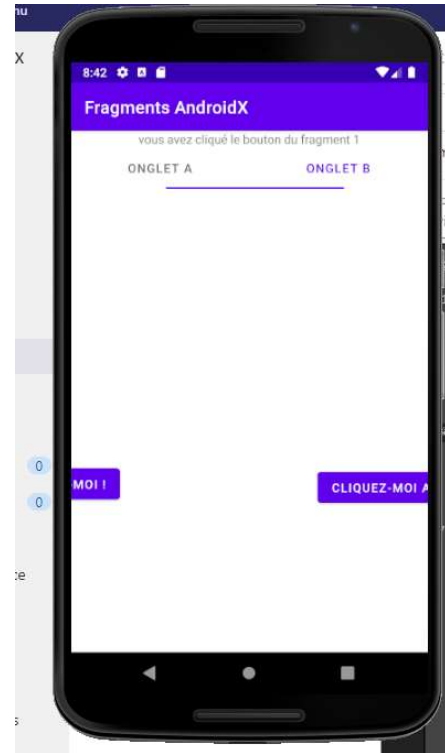
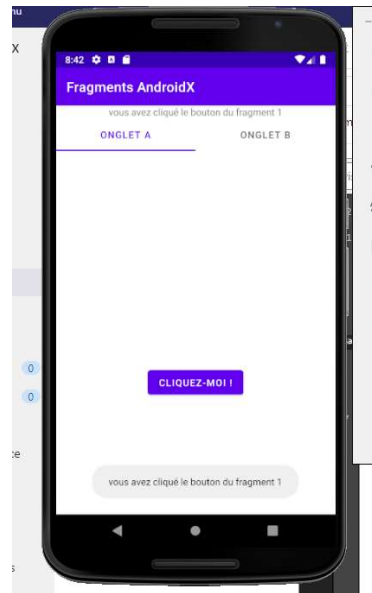
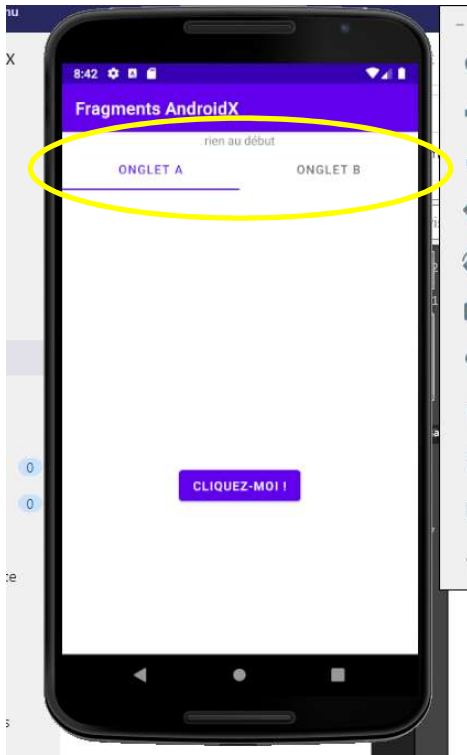


# Activité avec 2 fragments dynamiques [...] et qui envoient des données à l'activité



Le 2 est important !!!

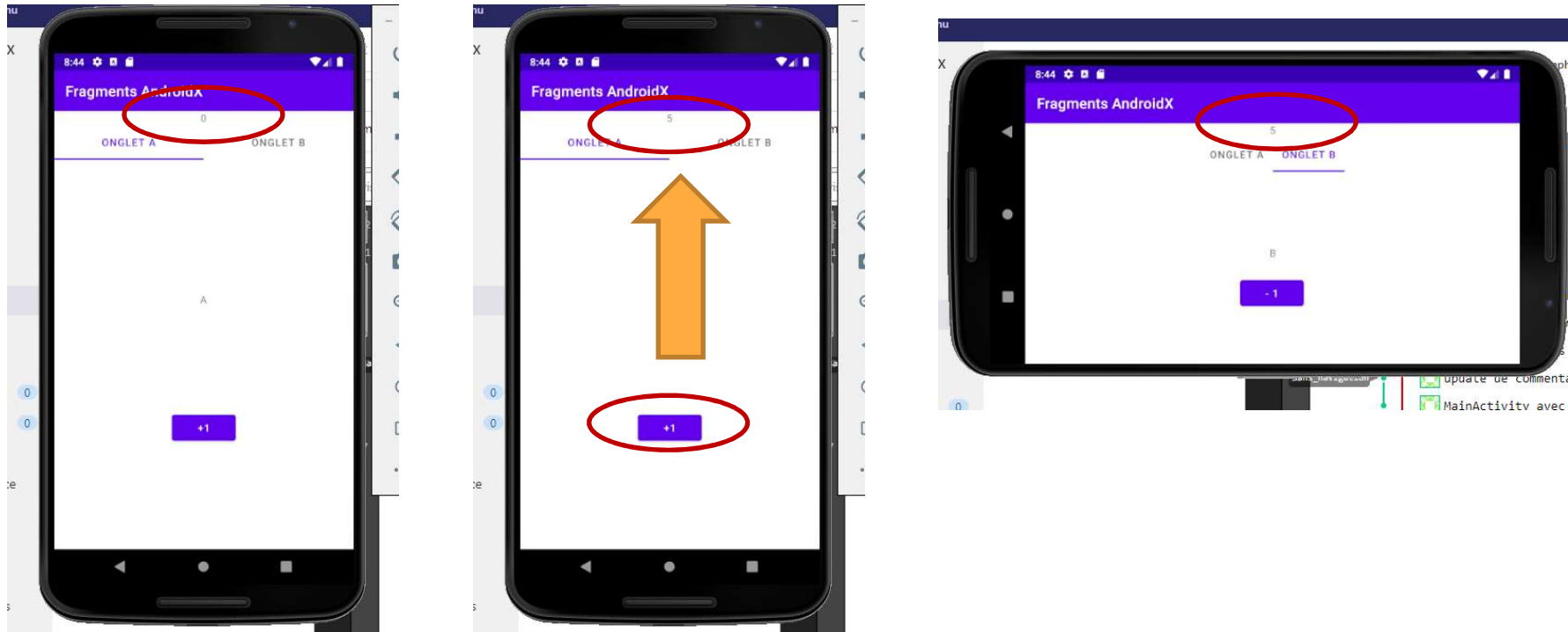
[...] avec un ViewPager2 pour naviguer entre les fragments



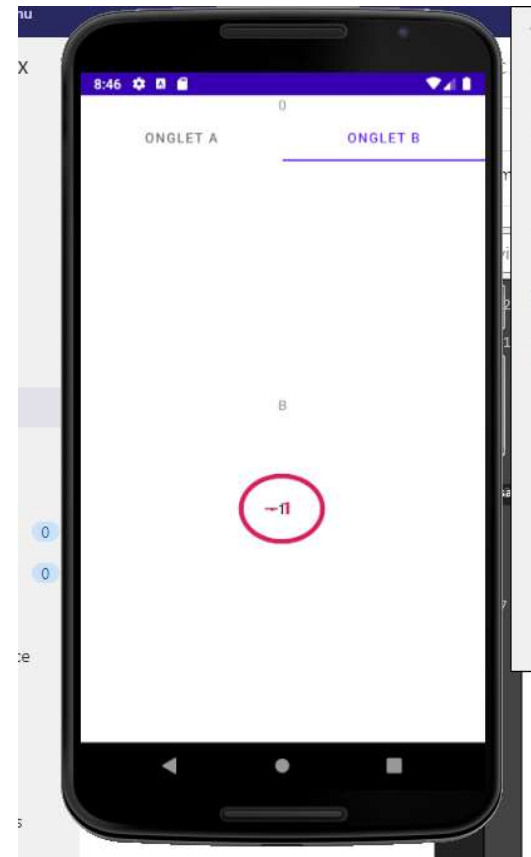
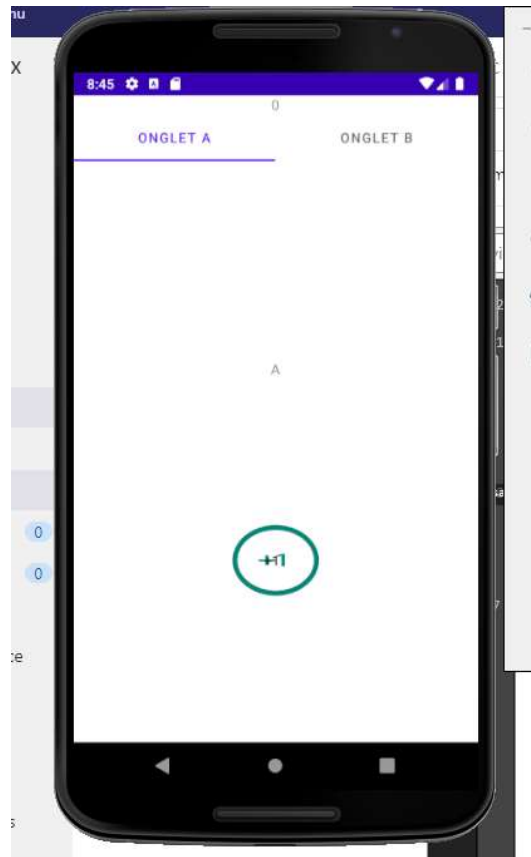
# ViewPager2 et TabLayout

- Un point de départ en cas de blocage à <https://gitlab.com/m1-ue-tac/td3-fragments/viewpager2-et-tablayout>

[...] et les fragments manipulent le compteur de l'activité (qui est maintenu en cas de rotation écran)



[...] avec de « jolis » boutons



# Complément pour le « jolis » boutons

- StateListDrawable
  - LayerDrawable
  - InsetDrawable
  - Selector
  - Animation... cf. details ci-après si besoin
- 
- Boutons disponibles à <https://gitlab.com/m1-ue-tac/td3-fragments/les-jolis-boutons.git>

## Bouton



- A **StateListDrawable** is a drawable object defined in XML that uses a several different images to represent the same graphic, depending on the state of the object.
  - <https://developer.android.com/guide/topics/resources/drawable-resource#StateList>



# Bouton



```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android"
  android:constantSize=["true" | "false"]
  android:dither=["true" | "false"]
  android:variablePadding=["true" | "false"] >
  <item
    android:drawable="@[package:]drawable/drawable_resource"
    android:state_pressed=["true" | "false"]
    android:state_focused=["true" | "false"]
    android:state_hovered=["true" | "false"]
    android:state_selected=["true" | "false"]
    android:state_checkable=["true" | "false"]
    android:state_checked=["true" | "false"]
    android:state_enabled=["true" | "false"]
    android:state_activated=["true" | "false"]
    android:state_window_focused=["true" | "false"] />
</selector>
```

# Bouton



- A **LayerDrawable** is a drawable object that manages an array of other drawables. Each drawable in the list is drawn in the order of the list—the last drawable in the list is drawn on top.
  - <https://developer.android.com/guide/topics/resources/drawable-resource#LayerList>
- A **InsetDrawable** is a drawable defined in XML that insets another drawable by a specified distance. This is useful when a View needs a background that is smaller than the View's actual bounds.
  - <https://developer.android.com/guide/topics/resources/drawable-resource#Inset>

# Bouton



```
<layer-list xmlns:android="http://schemas.android.com/apk/res/android" >
  <item>
    <selector android:exitFadeDuration="@android:integer/config_mediumAnimTime">
      <item android:state_pressed="true">
        <shape
          android:shape="oval" >
          <stroke android:width="5dp" android:color="@color/colorPrimaryDark" />
          <solid android:color="@android:color/transparent" />
          <padding android:bottom="24dp" android:top="24dp" android:left="20dp" android:right="28dp"/>
        </shape>
      </item>
      <item>...      </item>
    </selector>
  </item>
</layer-list>
```

# Bouton



```
<item>... </item>
  </selector>
</item>
<item>
  <inset>
    <selector android:exitFadeDuration="@android:integer/config_mediumAnimTime">
      <item android:drawable="@drawable/ic_plus_one_pressed" android:state_pressed="true" />
      <item android:drawable="@drawable/ic_plus_one" />
    </selector>
  </inset>
</item>
</layer-list>
```

# TD

- Solution à <https://gitlab.com/m1-ue-tac/td3-fragments/fragments-androidx>