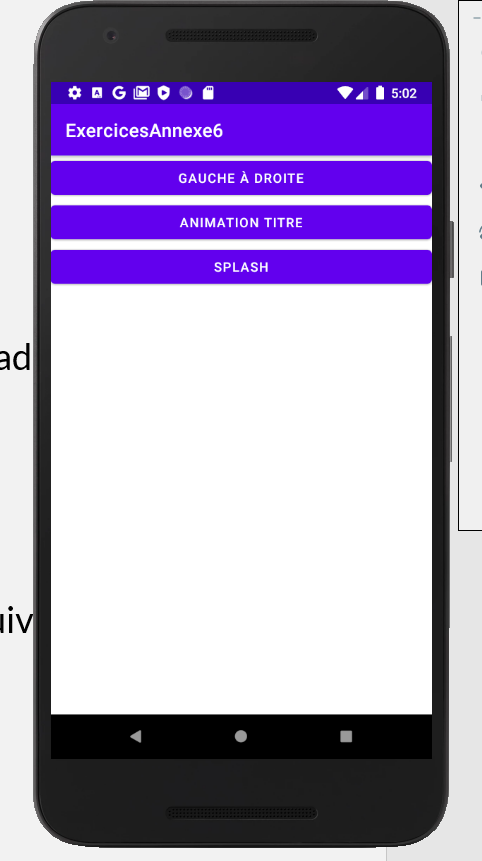
# Annexe 8B– Animations ( suite )

Lors du dernier épisode…

* On a vu comment utiliser le MotionLayout, pas senti beaucoup d'amour envers les outils d'Android Studio….( du groupe intensif h2022 du moins ☺ )
* On peut réaliser des animations avec l'objet ObjectAnimator

Exercice :

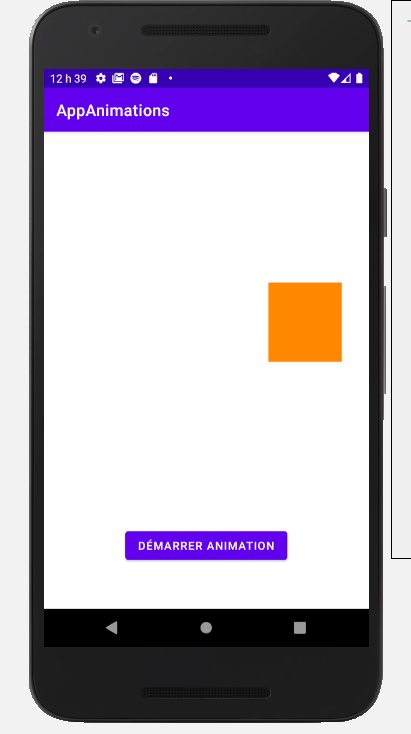
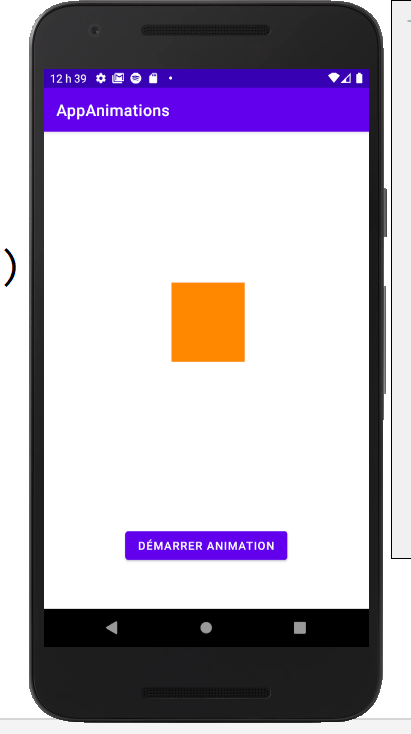
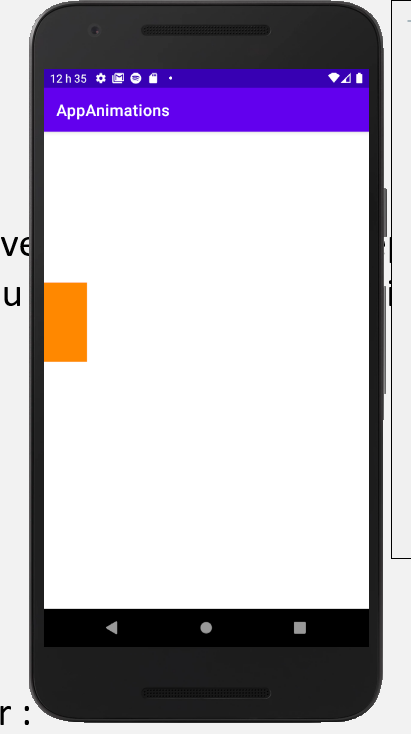
Concevez une application dont l’activité de départ contiendra les boutons suivants :



* Chaque bouton amène vers une Activity; faites les intents en conséquence

**1ère animation : Gauche à droite**

On veut voir un carré ( un Objet View ) arriver de la gauche, se déplacer de gauche à droite de l'activité, en oscillant / hésitant au centre de l'activité suite au clic du bouton.



* Dessinez l'interface avec un bouton et un objet View que vous attachez au ConstraintLayout
* Donnez dans le xml de positionnement une translationX à l'objet View pour ne pas le voir au démarrage ( qu'il soit à l'extrême gauche )
* Création de l'objet [ObjectAnimator](https://developer.android.com/reference/android/animation/ObjectAnimator.html)
  + Entre autres :

[ofFloat](https://developer.android.com/reference/android/animation/ObjectAnimator#ofFloat(T,%20android.util.Property%3CT,%20java.lang.Float%3E,%20android.util.Property%3CT,%20java.lang.Float%3E,%20android.graphics.Path))(T target, [Property](https://developer.android.com/reference/android/util/Property)<T, [Float](https://developer.android.com/reference/java/lang/Float)> xProperty, [Property](https://developer.android.com/reference/android/util/Property)<T, [Float](https://developer.android.com/reference/java/lang/Float)> yProperty, [Path](https://developer.android.com/reference/android/graphics/Path) path)

* On peut donc utiliser un Path ( voir cours C44 ) pour créer l'itinéraire que suivra notre objet lors de l'animation et faire start () pour démarrer l'animation

Ca fonctionne ? Explorer ces améliorations :

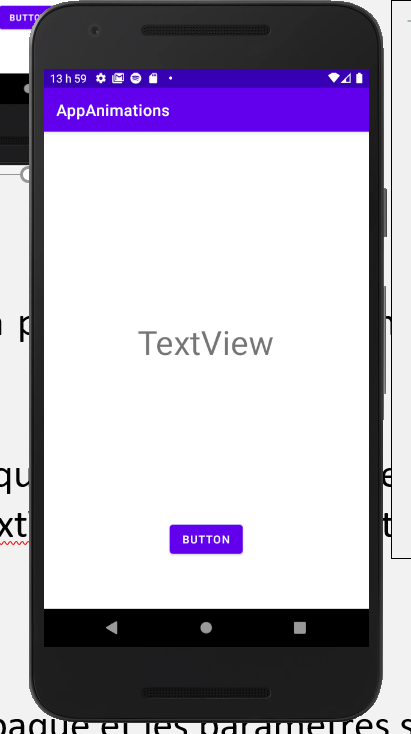
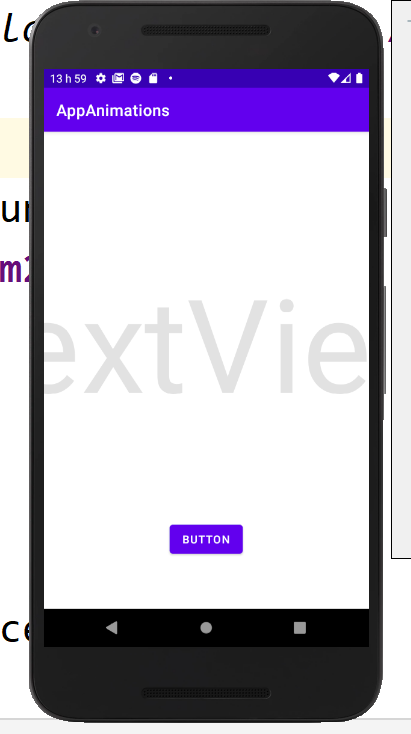
* Durée de l'animation : vous pouvez la changer avec setDuration. Quelle est la durée par défaut ? 300 milisecond
* Interpolateurs de l'animation : plusieurs interpolateurs existent et permettent de change le rythme de l'animation, essayez-en certains :

<https://medium.com/mobile-app-development-publication/illustrating-android-interpolator-31ea09051d78>

<http://cogitolearning.co.uk/2013/10/android-animations-tutorial-5-more-on-interpolators/>

\*\*\* pour arrêter une animation en cours : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2e animation : le titre qui apparaît**



* État de départ : changer le "scale" du TextView pour qu'il soit très grand et changer le "alpha" du TextView pour qu'il soit invisible
* Utiliser des ObjectAnimator ( 3 ) pour rendre à la fin le scale du TextView plus petit et le alpha complétement opaque.
* On peut jumeler différents ObjectAnimators ensemble en utilisant un AnimatorSet
* 2 utilisations possibles : playTogether et playSequentially
* Utiliser un interpolateur pour avoir un effet de rebond à la fin de l'animation

***Gestion des événements***

*On peut inscrire les ObjectAnimators/AnimatorSet à un écouteur de type AnimatorListener; peut permettre de faire des actions au début de l'animation ou à la fin.*

3e Animation : Splash

* Créer un View qui a l’apparence d’un cercle : utiliser un fichier xml ( resource file ) dont le « root element » est Shape
* Regarder le vidéo sur LEA