

IX. La Splash Screen

Au démarrage d'une application, il est souvent agréable ou plus visuel d'avoir une animation.



Création du layout

Vous pouvez utiliser l'image suivante [logo](#).

Pour plus de simplicité dans l'import d'images, vous pouvez installer le plugin **Android Drawable Importer** et importez cette image dans votre projet.

Elle se logera dans le dossier res/drawable/gestclilogo/ dans différents formats pour un meilleur affichage.

Créez un layout pour notre vue de démarrage.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="#A8E956"
    >

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="40dp"
        android:layout_marginTop="100dp"
        android:layout_marginEnd="40dp"
        android:src="@drawable/gestclilogo"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="59dp"
        android:fontFamily="@font/acme"
        android:text="Application de gestion de Clients"
        android:textColor="#FFFFFF"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

ôtons l'ActionBar en modifiant le thème dans style.xml

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

De même dans le MainActivity enlevons l'affichage de la bar d'information:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // enlevons la bare en haut

    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager
    .LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_main);
    ...
}
```

Animons le splash screen

Avec un clique droit sur le dossier res puis new, ajoutez le directory anim puis ajoutez dans anim une **animation resource file** que nous allons nommer top_animation_layout.xml.

Nous allons faire une translation et un fondu pour faire apparaître le logo

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:fromXDelta="0%"
        android:fromYDelta="-100%"
        android:duration="2000"/>
    <alpha
        android:fromAlpha="0.1"
        android:toAlpha="1.0"
        android:duration="1500"/>
</set>
```

La méthode translate prend deux paramètres de départ de l'animation en X et en Y, et une durée. Ici l'élément garde le même X (horizontal) mais va partir à -100% à partir du haut. Cette animation durera 2 secondes.

L'alpha contrôle la transparence. Ici nous partons avec un coefficient de 0,1 pour arriver à 1 (visible) en 1,5 secondes.

Pour le texte faites un deuxième fichier d'animation bottom_animation_layout.xml où le texte montera et apparaîtra en 1,5 secondes.

Mise en place dans le main

Il nous reste à déclarer l'animation dans le MainActivity.

Tout d'abord, déclarons les animations dans la méthode onCreate():

```
//Animations
topAnim = AnimationUtils.loadAnimation(this, R.anim.top_animation);
bottomAnim =
AnimationUtils.loadAnimation(this, R.anim.bottom_animation);
```

ensuite les éléments à animer:

```
//Hooks
    logo = findViewById(R.id.imageView);
    desc = findViewById(R.id.textView);
```

enfin affectons leur les animations:

```
//Assignment
    logo.setAnimation(topAnim);
    desc.setAnimation(bottomAnim);
```

Une dernière touche de code pour passer à l'application après 5 secondes

Après cette splendide animation, nous voulons que l'application s'ouvre enfin. Nous utilisons un mécanisme de **Handler** avec un **delay**. * Il nous faut déclarer la durée avec

```
private static int SPLASH_SCREEN = 5000;
```

- mettre en place le Handler

```
new Handler().postDelayed(new Runnable() {
    @Override
    public void run() {
        //Déclaration d'une nouvelle intention
        Intent intent = new
Intent(MainActivity.this,Dashboard.class);
        //Démarrage de L'intention
        startActivity(intent);
        //Cloture du Splash Screen
        finish();
    }
}, SPLASH_SCREEN);
```

Sans oublier, bien sûr, de créer l'activité correspondante.

Pour les sources: [github](#)