

Travail dirigé - Exercice

Classe : Groupe 9432

Cours : Développement d'application orientée mobile (420-AM1-TT)

Enseignant : Rodrigue TONGA

Vous avez droit à

Il vous est demandé

Vous avez l'obligation de fournir

OBJECTIF : À la découverte de l'objet `ImageView` - Dans ce document, nous découvrirons dans un premier temps avec quelle facilité il est possible d'animer les attributs de nos vues .

Condition : **Individuel**

LES IMAGES

Comme toute plateforme qui se respecte, Android nous permet à travers les éléments de sa boîte à outils graphiques d'afficher et de manipuler des images. Dans la section suivante, nous allons vous montrer comment utiliser le contrôle **ImageView** pour ajouter des images à votre application.

AJOUT D'UNE IMAGE À UN PROJET

En vous basant sur les étapes de créations d'un projet décrites dans le document théorique **Cours 1**, créez un nouveau projet Android et sauvegardez-le dans un dossier nommé **DemolImageView**.

Une fois votre projet ouvert, cliquez dans votre arborescence de projet, sur le sous-dossier **RES**; vous verrez alors apparaître de sous-dossier **drawable**. C'est dans ce dossier que vous devez déposer les images que vous désirez installer localement avec votre application sur le système de vos futurs utilisateurs.

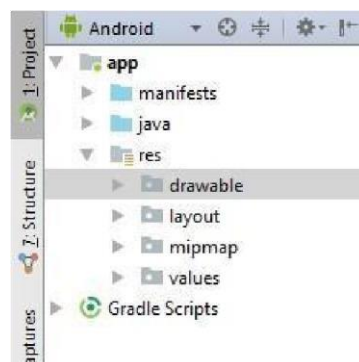


Figure 1: Dossier drawable

À l'aide du browser web de votre choix, téléchargez sur le bureau de votre ordinateur portable les images pointées par les URL suivant :



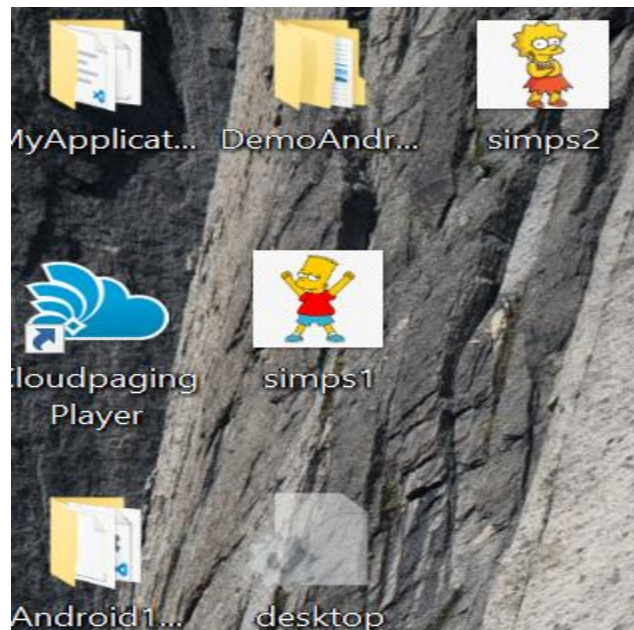


Figure 2: Images pour les animations

[https://www.google.ca/search?q=simpsons.png&tbm=isch&ved=2ahUKEwjx_J_ztuqBAXxCmlAHbvVDSAQ2-cCegQIABAA&oq=Simpsons.png&gs_lcp=CgNpbWcQARgAMgUIABCABDIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjoKCAAQigUUqsQMQQZoHCAAQigUQQ1DIEFjpF2C2K2gAcAB4AlABA4gBi_gOSAQM0LjGYAQCgAQGgAQtnD3Mtd2l6LWIz8ABAQ&sclient=img&ei=SLYkZfG4D_GViLMPu6u3gAI&bih=581&biw=1280#imgrc=P-gwnmwA8QyTOM&imgdii=_IBGZRGEpy6LM](https://www.google.ca/search?q=simpsons.png&tbm=isch&ved=2ahUKEwjx_J_ztuqBAXxCmlAHbvVDSAQ2-cCegQIABAA&oq=Simpsons.png&gs_lcp=CgNpbWcQARgAMgUIABCABDIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjIECAAQHjoKCAAQigUUqsQMQQZoHCAAQigUQQ1DIEFjpF2C2K2gAcAB4AlABA4gBi_gOSAQM0LjGYAQCgAQGgAQtnD3Mtd2l6LWIz8ABAQ&sclient=img&ei=SLYkZfG4D_GViLMPu6u3gAI&bih=581&biw=1280#imgrc=P-gwnmwA8QyTOM&imgdii=_IBGZRGEpy6LM)

Il faudra par contre renommer les images, car Android n'accepte pas les images débutant par un chiffre et contenant des caractères spéciaux ou majuscules.

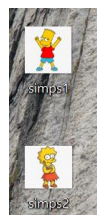


Figure 3: Images avec noms valides

Maintenant que nous avons des images, il faut les ajouter à notre sous-dossier **drawable**. En utilisant votre souris, sélectionnez simultanément les deux images et tapez la combinaison des touches **CTRL + C** pour copier les fichiers. Sélectionnez en suite le sous-dossier **drawable** et tapez la combinaison des touches **CTRL + V**, pour y transférer les images.

La fenêtre suivante devrait apparaître, vous demandant de confirmer le transfert des images. Faites **OK**.

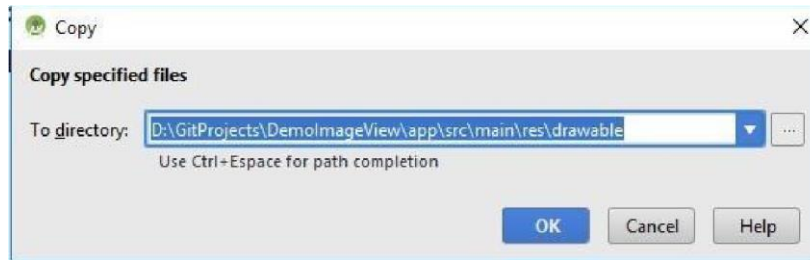


Figure 4: première fenêtre de transfert d'image

Et voilà les images ont été ajoutées à notre projet. Si vous double cliquez sur une des images dans le dossier **drawable**, vous devriez voir cette image s'afficher dans l'espace de travail de AndroidStudio.

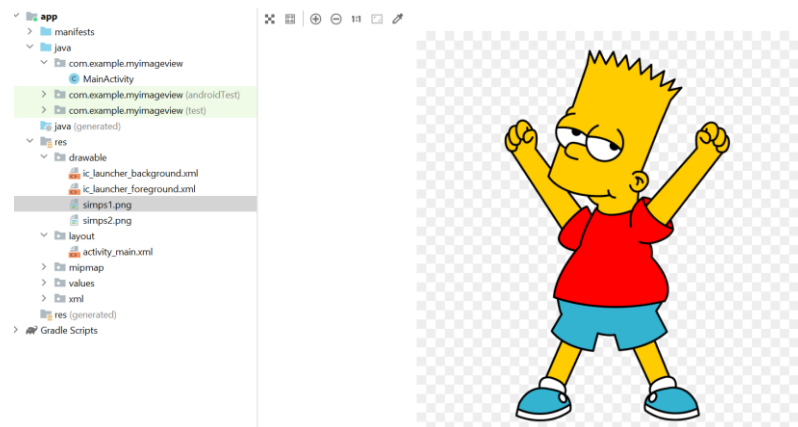


Figure 5: Visualisation d'images dans AdroidStudio

AJOUT D'UN ImageView

Dans le sous-dossier **layout**, double cliquez maintenant sur le fichier **activity_main.xml** et affichez-le en mode **code**.

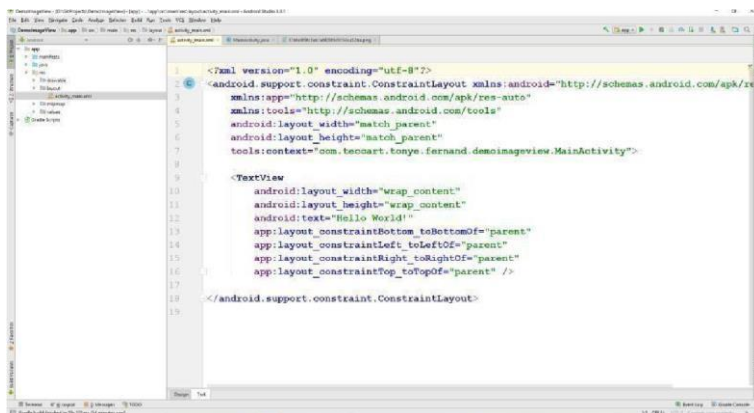


Figure 6: Layout en mode XML



Retirez la balise **TextView** et remplacez-la par l'extrait de code suivant :

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:id="@+id/img"  
    android:src="@drawable/simps1" />
```

Figure 7: Ajout du **ImageView** avec une image de départ

Lancez maintenant votre application en sélectionnant un **AVD** ou un téléphone/tablette physique. Vous devriez observer le résultat suivant.

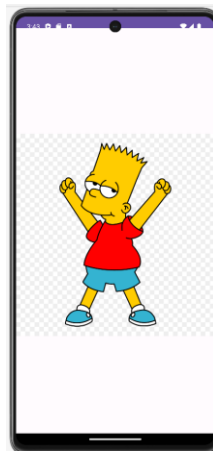


Figure 8: Application avec **ImageView**

LES ANIMATIONS

Modifier le **ConstraintLayout** fourni par défaut et remplacez-le par un **RelativeLayout** et ajoutez-y quatre Buttons encapsulés dans un **LinearLayout**.

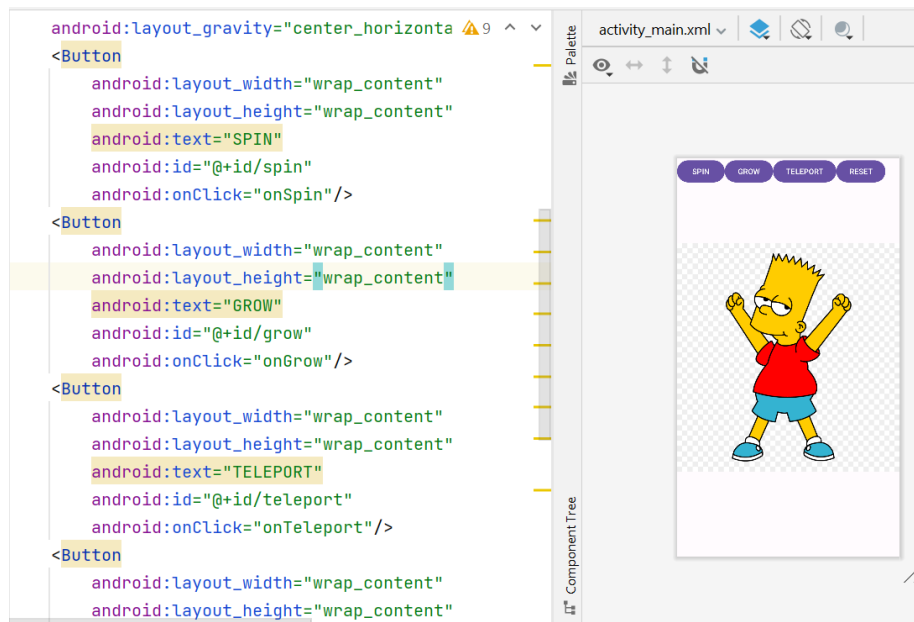


Figure 9: Ajout de contrôles Buttons pour l'interaction

AJOUT DU CODE À L'ACTIVITY

Double cliquez maintenant sur le fichier **MainActivity.java**, se trouvant dans le dossier **java** de votre arborescence de projet.

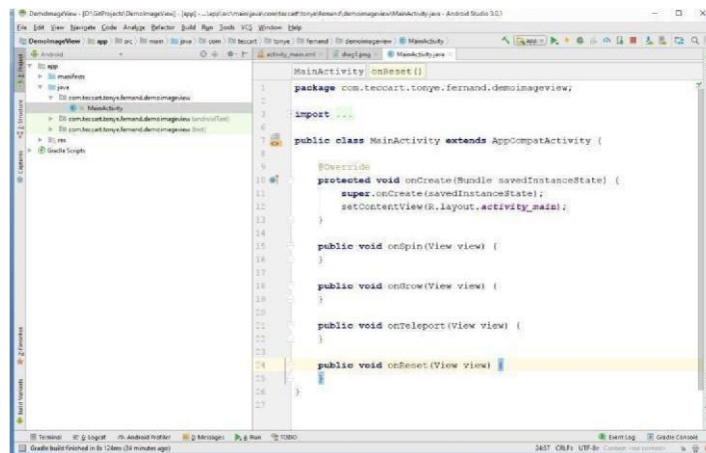


Figure 10: MainActivity avec les EventHandler non implémenté

Afin d'être en mesure d'animer notre **ImageView**, à l'aide des quatre boutons, nous devons obtenir une référence vers notre **ImageView**. Ajoutez le code suivant à votre classe.



```

8 public class MainActivity extends AppCompatActivity {
9
10     private ImageView img;
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         img = (ImageView) findViewById(R.id.img);
18     }

```

Figure 12:Reference vers le ImageView

Maintenant que nous avons une référence vers notre **ImageView**, nous allons pouvoir utiliser la méthode **animate()**, pour animer les différents attributs de cette vue.

LA MÉTHODE ANIMATE

- 1- Quand l'utilisateur cliquera sur le bouton **SPIN**, nous voudrions voir notre personnage effectuer une rotation de 360°

```

public void onSpin(View view) {
    img.animate().rotation(360f).setDuration(2000);
}

```

Figure 13:Animate() to spin ImageView

- 2- Quand l'utilisateur cliquera sur le bouton **GROW**, nous voudrions voir notre personnage Grandir en x et en y

```

public void onGrow(View view) {
    img.animate()
        .scaleXBy(1.2f)
        .scaleYBy(1.2f)
        .setDuration(2000);
}

```

Figure 14:Animate() pour scale x et Y

- 3- Quand l'utilisateur cliquera sur le bouton **TELEPORT**, nous voudrions voir notre personnage graduellement disparaître.

```

public void onTeleport(View view) {
    img.animate().alpha(0).setDuration(2000);
}

```

Figure 15:Animate() pour changer Alpha



- 4- Quand l'utilisateur cliquera sur le bouton **Reset**, nous voudrions voir notre personnage reprendre son allure initiale.

```
public void onReset(View view) {  
  
    img.setImageResource(R.drawable.drag1);  
    img.animate().scaleY(1f).scaleX(1f).alpha(1f);  
}
```

