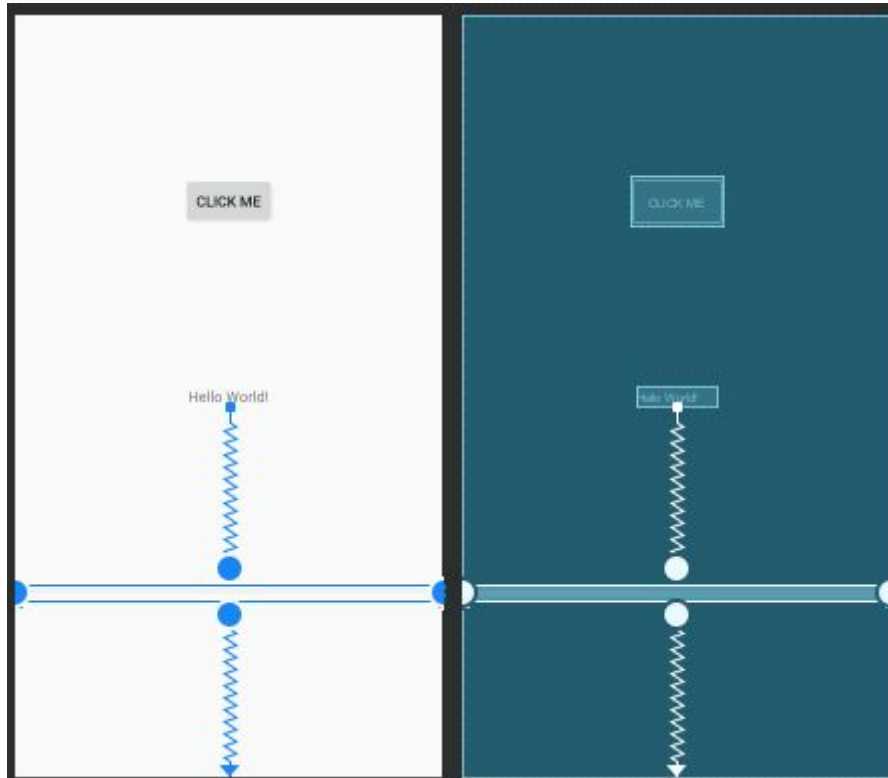


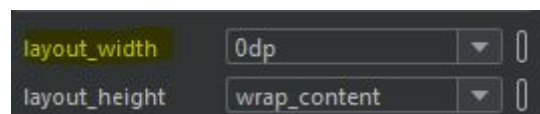
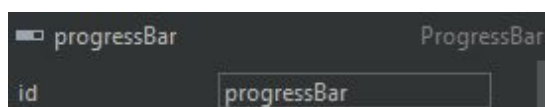
Pour aller plus loin:

Ajout de conditions:

Dans un premier temps nous allons ajouter une barre de progression de la même manière que l'ajout du bouton (onglet Widgets), avec les contraintes ci dessous:



Ensuite, nous allons ,ci cela n'est pas déjà fait, lui changer son id via le panel attribut pour **"progressBar"**, ainsi que la largeur du composant pour qu'il prenne la totalité de la place horizontalement parlant:



Cette fois ci dans le code de l'activité du fichier **MainActivity.java** vous allez ajouter dans un premier temps la déclaration d'une variable du type de votre composant:

```
public class MainActivity extends AppCompatActivity {  
  
    private TextView text;  
    private ProgressBar pgBar;  
    private int compteur = 0;  
}
```

Dans le **onCreate** vous lierez la variable avec le composant via son id, changerez quelques unes de ses propriétés, et ajouterez l'appelle d'une fonction que l'on va créer ensuite, ce faisant il y aura une erreur temporairement:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    text = (TextView) findViewById(R.id.textView1);  
    pgBar = (ProgressBar) findViewById(R.id.progressBar);  
    pgBar.setMax(10);  
    pgBar.setProgress(0);  
    UpdateBar(compteur);  
}
```

Ajouter de nouveau dans la fonction **IncrementNbr** l'appel de la fonction qui n'est toujours pas créée:

```
public void IncrementNbr(View view) {  
    compteur++;  
    text.setText(String.valueOf(compteur));  
    UpdateBar(compteur);  
}
```

Enfin écrivez en dessous de la fonction **IncrementNbr**, la fonction suivante de A à Z:

```
@SuppressWarnings("SetTextI18n")  
public void UpdateBar(int cpt){  
    if (pgBar.getProgress() != pgBar.getMax() || pgBar.getProgress() < pgBar.getMax()){ pgBar.setProgress(cpt);}  
}
```

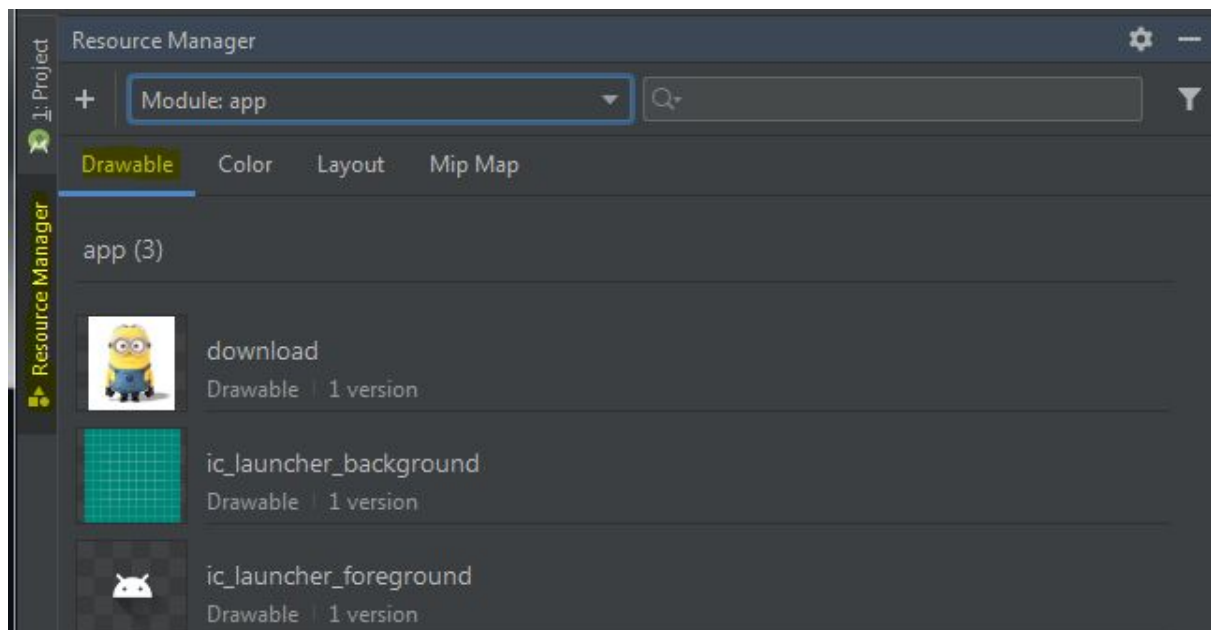
Désormais vous pouvez tester l'ajout de ce code en compilant le code (maj + F10):



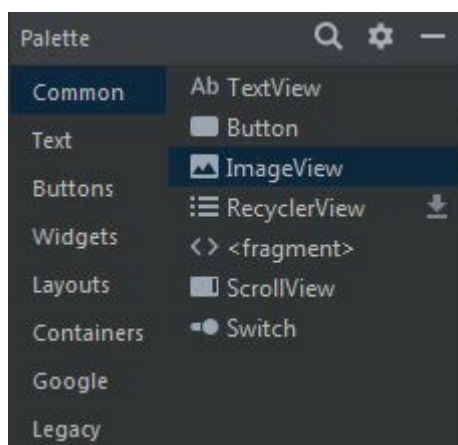
Implémentation d'une image et interaction:

Dans un premier temps allez dans le resources manager, accessibles via l'onglet du même nom se trouvant à gauche du panel hiérarchie, ou bien en cliquant sur l'onglet **View -> Tool Windows -> Resource Manager**.

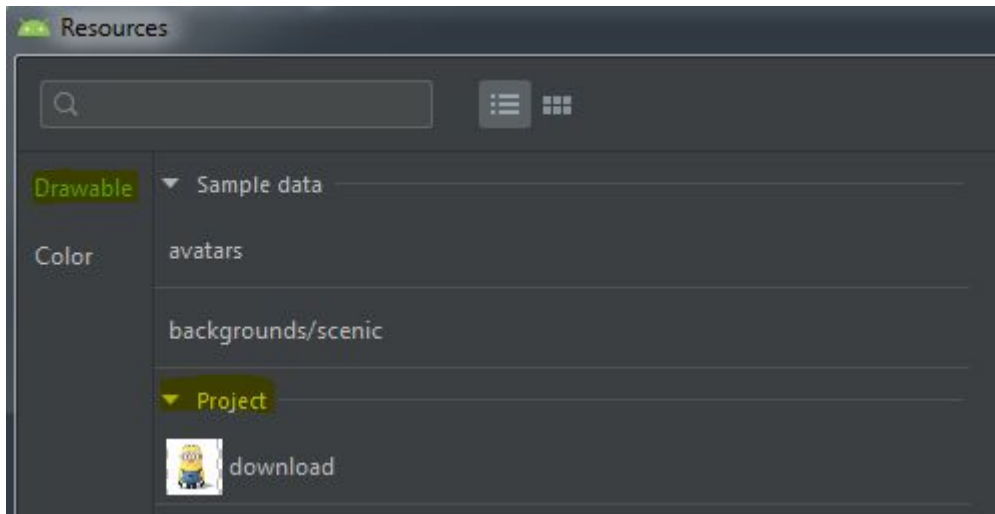
Une fois dans l'onglet, allez chercher une image et faites un drag and drop de celle-ci dans l'onglet en vérifiant bien d'être dans la partie Drawable qui s'affiche par défaut normalement:



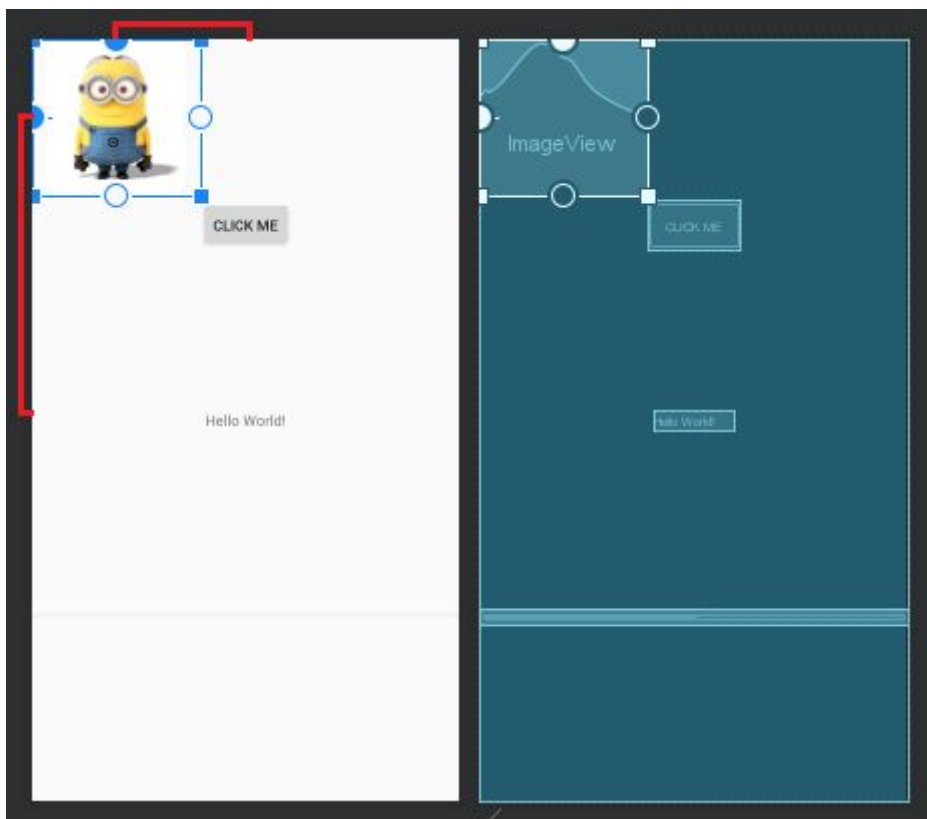
On peut dès maintenant ajouter à notre activité le composant **ImageView** via le panel **palette** accessible dans la partie design de notre **activity_main.xml**, pour cela faites un drag and drop du composant sur la preview comme on l'a fait pour le bouton.



Normalement un pop up va apparaître vous demandant de sélectionner le contenu de l'image, déroulez l'onglet **Project** se trouvant dans **Drawable** et choisissez l'image que vous avez importé:



Au niveau des contraintes, contentez vous de relier l'image au haut du preview et à gauche de celui-ci, redimensionnez là si nécessaire:



*Si vous avez recréer l'image veuillez changer l'id de celle-ci avec l'onglet attribut, vous lui mettez **ImageView**.

Au niveau du code dans notre **MainActivity.java**, ajoutez une variable du type de notre composant que vous nommerez image:

```
private TextView text;
private ProgressBar pgBar;
private ImageView image;
private int compteur = 0;
```

Dans le **onCreate**, liez la variable avec notre composant et modifiez la propriété indiquée à l'aide du code ci dessous:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    text = (TextView) findViewById(R.id.textView1);
    pgBar = (ProgressBar) findViewById(R.id.progressBar);
    pgBar.setMax(10);
    pgBar.setProgress(0);
    image = (ImageView) findViewById(R.id.imageView);
    image.setVisibility(View.INVISIBLE);
    UpdateBar(compteur);
}
```

Cette fois-ci vous aurez à modifier **UpdateBar** en ajoutant le code indiqué comme ci dessous:

```
@SuppressWarnings("SetTextI18n")
public void UpdateBar(int cpt) {
    if (pgBar.getProgress() != pgBar.getMax() || pgBar.getProgress() < pgBar.getMax()) { pgBar.setProgress(cpt); }
    if (pgBar.getProgress() == pgBar.getMax()) { image.setVisibility(View.VISIBLE); }
}
```

Si aucune erreur a été faite alors vous pouvez désormais compiler le code et ainsi tester le changement de la propriété de la visibilité de l'image lorsque la barre de progression sera totalement remplie..