Fondamentaux d'Android

3. Editeur de mise en page

Hatem Aziza - September 2024

Developers

Introduction

Cet atelier pratique présente également deux autres sous-classes ViewGroup:

- LinearLayout: Un groupe qui aligne les éléments enfants horizontalement ou verticalement.
- RelativeLayout: Un groupe d'éléments enfants dans lequel chaque élément View est positionné et aligné par rapport à un autre élément View du ViewGroup. Les positions des éléments enfants sont décrites les uns par rapport aux autres ou par rapport au ViewGroup parent.

Ce que vous apprendrez

- Comment positionner les vues dans un **LinearLayout**.
- Comment positionner les vues dans un **RelativeLayout**.

Ce que tu feras

- Utiliser LinearLayout.
- Utiliser RelativeLayout.

Tâche 1 : Modifier la mise en page en LinearLayout

LinearLayout est un ViewGroup qui organise sa collection de vues dans une rangée horizontale ou verticale. LinearLayout est l'une des layouts les plus courantes car elle est simple et rapide. Il est souvent utilisé dans un autre groupe de vues pour organiser les éléments de l'interface utilisateur horizontalement ou verticalement.
LinearLayout doit avoir ces attributs :

- layout_width
- layout_height
- orientation

L'orientation peut être :

- horizontal: Les vues sont disposées de gauche à droite.
- vertical: Les vues sont disposées de haut en bas.

Dans cette tâche, vous allez modifier ConstraintLayout Hello Toast par LinearLayout afin de pouvoir vous entraîner à utiliser LinearLayout.

Changez le groupe de vues racine en LinearLayout

1. Ouvrez l'application Hello Toast de l'atelier précédent.
2. Ouvrez le fichier de layout activity_main.xml et cliquez sur l'onglet Code pour voir le code XML. Tout en haut du code XML se trouve le slogan suivant :
   <android.support.constraint.ConstraintLayout xmlns:android="http:…
3. Remplacez la balise **<android.support.constraint.ConstraintLayout** par **<LinearLayout** pour que le code ressemble à ceci :
   <LinearLayout xmlns:android="http:...
4. Assurez-vous que la balise de fermeture à la fin du code est devenue **</LinearLayout>** (Android Studio modifie automatiquement la balise de fermeture si vous modifiez la balise d'ouverture). S'il n'a pas changé automatiquement, modifiez-le manuellement.
5. Sous **<LinearLayout**, ajoutez l'attribut suivant après l'attribut **android:layout_height** :
   android:orientation="vertical"

Après avoir effectué ces modifications, certains attributs XML d'autres éléments sont soulignés en rouge car ils sont utilisés avec ConstraintLayout et ne sont pas pertinents pour LinearLayout.

Modifier les attributs des éléments pour LinearLayout

Suivez ces étapes pour modifier les attributs des éléments de l'interface utilisateur afin qu'ils fonctionnent avec LinearLayout :

1. Ouvrez le fichier activity_main.xml et cliquez sur l'onglet Code .
2. Recherchez le bouton **button_toast** et mettez l'attribut suivant :
   android:layout_width="match_parent"

3. Supprimez les attributs suivants de l'élément **button_toast** :
   app:layout_constraintEnd_toEndOf
   app:layout_constraintStart_toStartOf
   app:layout_constraintTop_toTopOf
   app:layout_constraintBottom_toTopOf

4. Recherchez le bouton **button_count** et mettez l'attribut suivant :
   android:layout_width="match_parent"

5. Supprimez les attributs suivants de l'élément **button_count** :
   app:layout_constraintBottom_toBottomOf
   app:layout_constraintEnd_toEndOf
   app:layout_constraintStart_toStartOf
   app:layout_constraintTop_toBottomOf

6. Recherchez l'élément TextView show_count et modifiez les attributs suivants :
   android:layout_width="match_parent"
   android:layout_height="wrap_content"

7. Supprimez les attributs suivants de l'élément show_count :
   app:layout_constraintBottom_toTopOf="@+id/button_count"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/button_toast"

Changer les positions des éléments dans LinearLayout

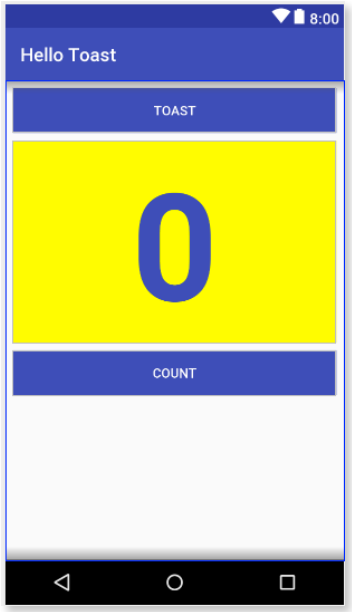
LinearLayout dispose ses éléments dans une rangée horizontale ou verticale. Vous avez déjà ajouté l'attribut **android:orientation="vertical"** pour LinearLayout, de sorte que les éléments sont empilés les uns sur les autres verticalement.
Pour modifier leurs positions afin que le bouton Count soit en bas, procédez comme suit :

```
<Button
    android:id="@+id/button_toast"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="@color/colorPrimary"
    android:onClick="showToast"
    android:text="@string/button_label_toast"
    android:textColor="@android:color/white" />
```

```
<TextView
    android:id="@+id/show_count"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:background="#FFFF00"
    android:gravity="center_vertical"
    android:text="@string/count_initial_value"
    android:textAlignment="center"
    android:textColor="@color/colorPrimary"
    android:textSize="160sp"
    android:textStyle="bold" />

<Button
    android:id="@+id/button_count"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:background="@color/colorPrimary"
    android:onClick="countUp"
    android:text="@string/button_label_count"
    android:textColor="@android:color/white" />
```

L'aperçu de la mise en page ressemble désormais à ce qui suit :



Ajouter du poids à l'élément TextView

Les attributs **gravity** et **weight** vous donne un contrôle supplémentaire sur l'organisation des vues et du contenu dans un fichier LinearLayout.
L'attribut **android:gravity** spécifie l'alignement du contenu de l'élément View dans l'élément lui-même.
Définissez cet attribut pour le TextView afin de centrer le contenu (le chiffre 0) au milieu du Texte View:

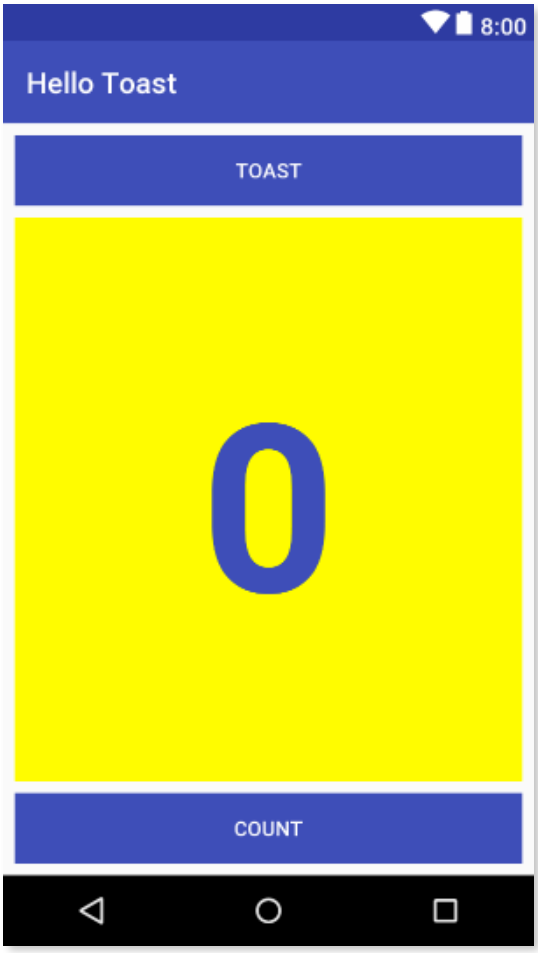
android:gravity="center_vertical"

L'attribut **android:layout_weight** indique quelle quantité d'espace supplémentaire dans **LinearLayout** sera allouée au View. Si un seul View possède cet attribut, il obtient tout l'espace supplémentaire sur l'écran. Pour plusieurs éléments, l'espace est calculé au prorata.
Par exemple, si les boutons ont chacun un poids de 1 et le TextView 2, totalisant 4, les éléments Button obtiennent chacun ¼ de l'espace et la moitié pour TextView.

Recherchez l'élément TextView et ajoutez l'attribut suivant :

android:layout_weight="1"

L'aperçu ressemble maintenant à la figure suivante.



Tâche 2 : Modifier la mise en page en RelativeLayout

RelativeLayout est un regroupement de vues dans lequel chaque vue est positionnée et alignée par rapport aux autres vues du groupe. Dans cette tâche, vous apprendrez à créer une mise en page avec RelativeLayout.

Remplacez LinearLayout par RelativeLayout

Un moyen simple de remplacer LinearLayout par RelativeLayout consiste à ajouter des attributs XML dans l' onglet Code.

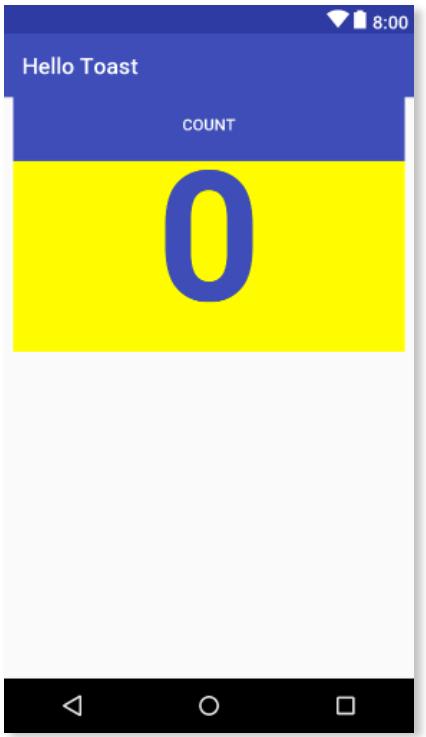
1. Ouvrez le fichier Activity_main.xml et cliquez sur l'onglet Code pour voir le code XML.
2. Remplacez le <LinearLayout en haut par <RelativeLayout pour que l'instruction ressemble à ceci :

   <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3. Faites défiler vers le bas pour vous assurer que la balise de fin </LinearLayout>a également changé en</RelativeLayout> ; si ce n'est pas le cas, modifiez-le manuellement.

Réorganiser les vues dans un RelativeLayout

Un moyen simple de réorganiser et de positionner les vues dans un RelativeLayout consiste à ajouter des attributs XML.

1. Cliquez sur l'onglet Design pour voir l'aperçu de la mise en page, qui ressemble maintenant à la figure ci-dessous.



Avec la modification apportée à RelativeLayout, l'éditeur de layout a également modifié certains attributs de vue.
Par exemple:

- Le bouton Count recouvre le bouton Toast, c'est pourquoi vous ne pouvez pas voir le bouton Toast.
- La partie supérieure du TextView recouvre les éléments Button.

2. Ajoutez l'attribut **android:layout_below** au bouton **button_count** pour le positionner directement sous le TextView **show_count**. Cet attribut est l'un des nombreux attributs permettant de positionner des vues dans un RelativeLayout: vous placez des vues par rapport à d'autres vues.
   android:layout_below="@+id/show_count"

3. Ajoutez l'attribut **android:layout_centerHorizontal** au bouton **button_count** pour centrer la vue horizontalement au sein de son parent, qui dans ce cas est le RelativeLayout de vues.
   android:layout_centerHorizontal="true"

4. Le code XML complet du bouton **button_count** est le suivant :

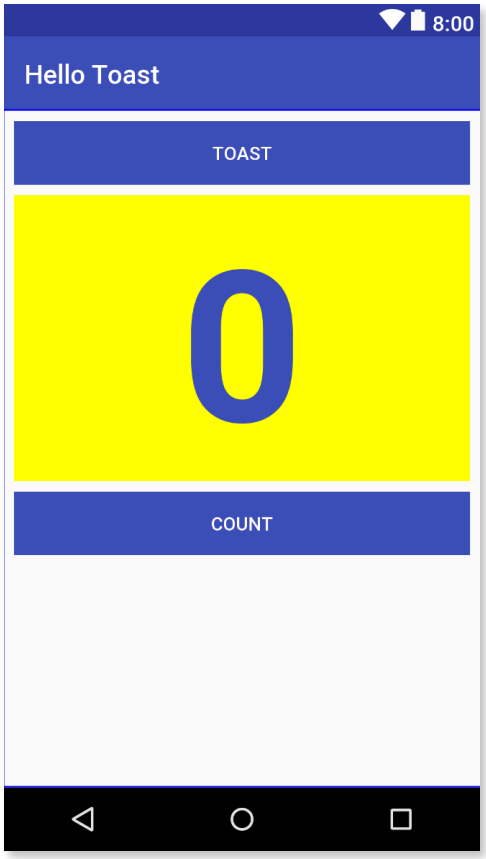
```
<Button
    android:id="@+id/button_count"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginStart="8dp"
    android:background="@color/colorPrimary"
    android:onClick="countUp"
    android:text="@string/button_label_count"
    android:textColor="@android:color/white"
    android:layout_below="@+id/show_count"
    android:layout_centerHorizontal="true"/>
```

5. Ajoutez les attributs suivants au TextView show_count :
   android:layout_below="@+id/button_toast"
   android:layout_alignParentLeft="true"
   android:layout_alignParentStart="true"

L'attribut **android:layout_alignParentLeft** aligne la vue sur le côté gauche du RelativeLayout parent. Bien que cet attribut soit suffisant à lui seul pour aligner la vue sur le côté gauche, vous souhaiterez peut-être que la vue s'aligne sur le côté droit si l'application s'exécute sur un appareil qui utilise une langue de droite à gauche. Ainsi, l'attribut **android:layout_alignParentStart** fait correspondre le bord "de départ" de cette vue au bord de départ du parent. Le début est le bord gauche de l'écran si

la préférence est de gauche à droite, ou le bord droit de l'écran si la préférence est de droite à gauche.

6. Supprimez l'attribut android:layout_weight="1" du TextView show_count , qui n'est pas pertinent pour un RelativeLayout. L'aperçu de la mise en page ressemble désormais à la figure suivante :



Le code XML dans **activity_main.xml** :

```
<RelativeLayout 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"
    android:orientation="vertical"
    tools:context="com.example.android.hellotoast.MainActivity">

    <Button
        android:id="@+id/button_toast"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
```

```
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@color/colorPrimary"
        android:onClick="showToast"
        android:text="@string/button_label_toast"
        android:textColor="@android:color/white" />

    <TextView
        android:id="@+id/show_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="#FFFF00"
        android:text="@string/count_initial_value"
        android:textAlignment="center"
        android:textColor="@color/colorPrimary"
        android:textSize="160sp"
        android:textStyle="bold"
        android:layout_below="@+id/button_toast"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:id="@+id/button_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:background="@color/colorPrimary"
        android:onClick="countUp"
        android:text="@string/button_label_count"
        android:textColor="@android:color/white"
        android:layout_below="@+id/show_count"
        android:layout_centerHorizontal="true"/>
</RelativeLayout>
```