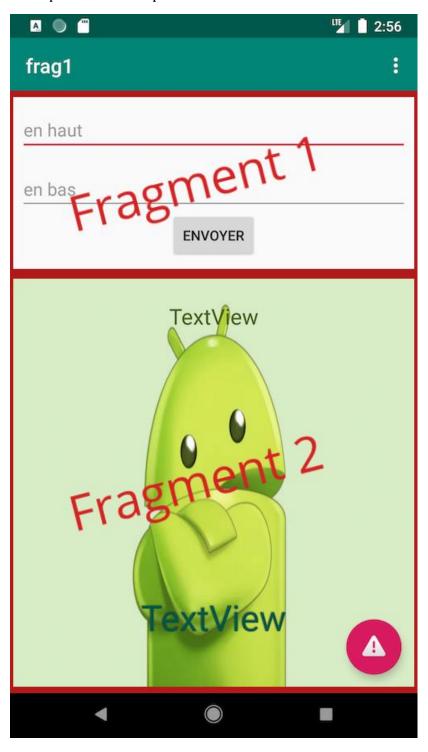
IX. Les Fragments

Nous pouvons découper notre interface en morceaux avec les Fragments



ici notre vue est divisée en deux fragments, ce qui permet de répartir notre code sur plusieurs classes et plusieurs layout.

Création des Fragments

Le premier fragment

Créez un layout que vous nommerez activity_top.xml afin de réaliser le Fragment 1 sur l'image ci dessus.

notre layout est constitué de 2 TextView et d'un bouton. Nous devons les contraindre.

Vous devriez avoir le xml suivant:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    android:layout width="match parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/textView3"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:layout_marginTop="15dp"
        android:layout alignParentTop="true"
        android:gravity="center"
        android:text="Messages" />
    <EditText
        android:layout width="match parent"
        android:layout height="wrap content"
        android:id="@+id/input top image text"
        android:layout below="@id/textView3"
        android:layout marginTop="15dp"
        android:layout centerHorizontal="true"
        android:layout marginLeft="10dp"
        android:layout marginRight="10dp" />
    <EditText
        android:layout_width="match_parent"
        android:layout height="wrap content"
        android:id="@+id/input bottom image text"
        android:layout below="@id/input top image text"
        android:layout marginTop="15dp"
        android:layout centerHorizontal="true"
        android:layout marginLeft="10dp"
        android:layout marginRight="10dp"
        />
    <Button
```

```
android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Envoyer"
    android:id="@+id/button_send"
    android:layout_below="@+id/input_bottom_image_text"
    android:layout_centerHorizontal="true" />
    </RelativeLayout>
```

Le second fragment

Créez un second nommé activity_bottom.xml pour le Fragment 2. Celui-ci est un background avec 2 TextView.

Ajoutez l'image (lien) dans le dossier mipmap

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    android:layout width="match parent"
    android:layout_height="match_parent"
    android:background="@mipmap/androidfun"
    android:orientation="vertical">
    <TextView
        android:id="@+id/top image text"
        android:layout width="match parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout alignParentTop="true"
        android:layout_marginStart="0dp"
        android:layout marginTop="30dp"
        android:gravity="center"
        android:text="TextView"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#4B6105" />
    <TextView
        android:id="@+id/bottom_image_text"
        android:layout width="match parent"
        android:layout_height="wrap_content"
        android:layout alignParentBottom="true"
        android:layout marginStart="0dp"
        android:layout marginBottom="50dp"
        android:gravity="center"
        android:textColor="@color/colorPrimaryDark"
        android:shadowColor="@color/text shadow"
        android:shadowDx="-1"
        android:shadowDy="-1"
        android:shadowRadius="4"
```

```
android:text="TextView"
android:textSize="36sp" />
</RelativeLayout>
```

Les contrôleurs

Chaque fragment doit avoir un contrôleur. Construisons donc le premier TopFragment.java. Il doit hériter de la classe Fragment et implémenter ses méthodes.

De même pour BottomFragment.java.