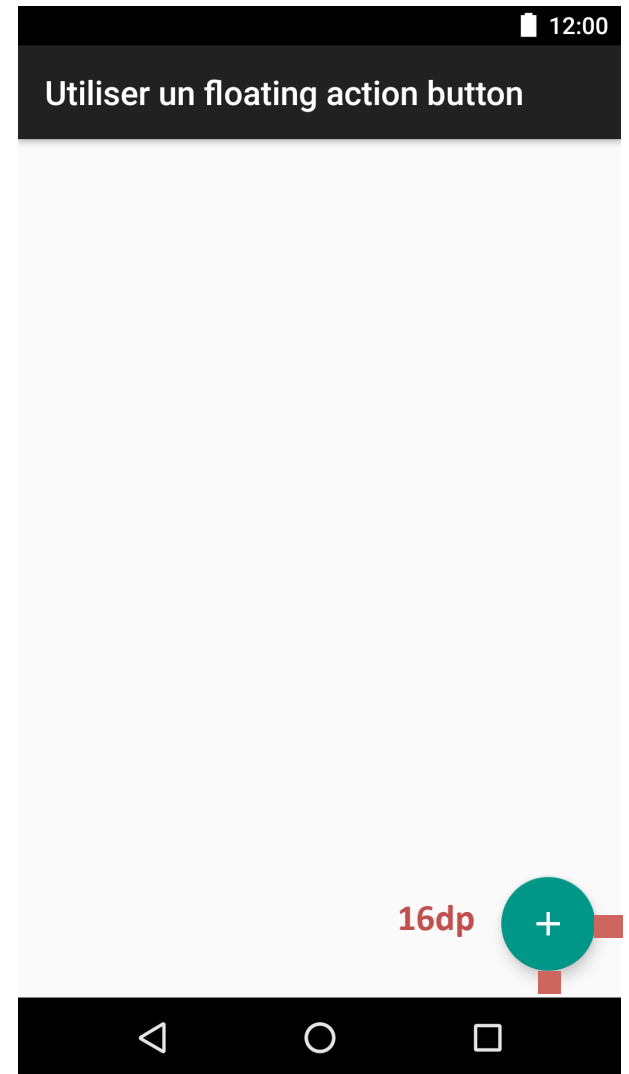


# TP°5 : Design et ergonomie

# Exercice 1 : Utiliser un floating action button

- Dans l'Activity FabActivity, afficher un floating action button avec l'icône `R.drawable.ic_add_white_24dp`
- Au clic sur le button, afficher le texte `R.string.fab_toast_action` à l'aide de la fonction `Utils.showToast()`



# Exercice 2 : Utiliser une CardView

Créer le layout suivant pour l'Activity CardViewActivity.

**Titre :** Texte : `R.string.cardview_title`; Taille : 24sp

**Description :** Texte : `R.string.cardview_text`

**Bouton ACTION 2 :**

- Texte : `R.string.cardview_action2`
- Style : `R.style.Widget.AppCompat.Button.Borderless`

**Bouton ACTION 1 :**

- Texte : `R.string.cardview_action1`
- Style : `R.style.Widget.AppCompat.Button.Borderless.Colored`

## Info

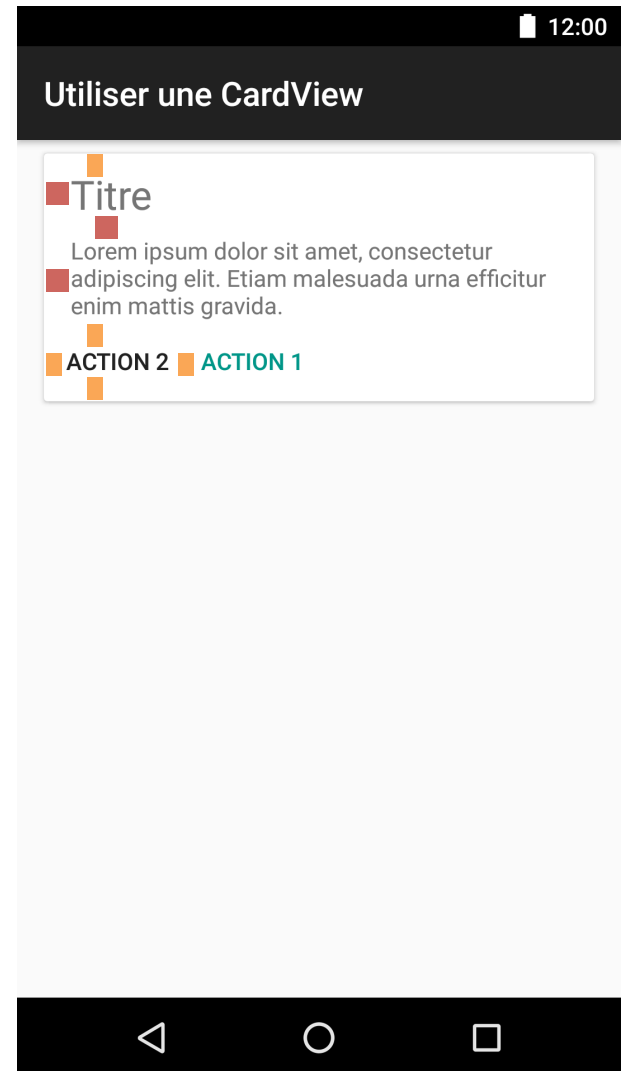
Les boutons ont un padding de 8dp ajouté automatiquement à cause du style utilisé.

Il est possible d'ajouter un espace négatif pour annuler un padding indésirable :

`<dimen name="negative_spacing">-8dp</dimen>`

16dp

8dp



# Exercice 3 : Utiliser un thème custom

Customiser le thème par défaut de l'application pour que les écrans précédent s'affichent avec les couleurs suivantes :



#691A99



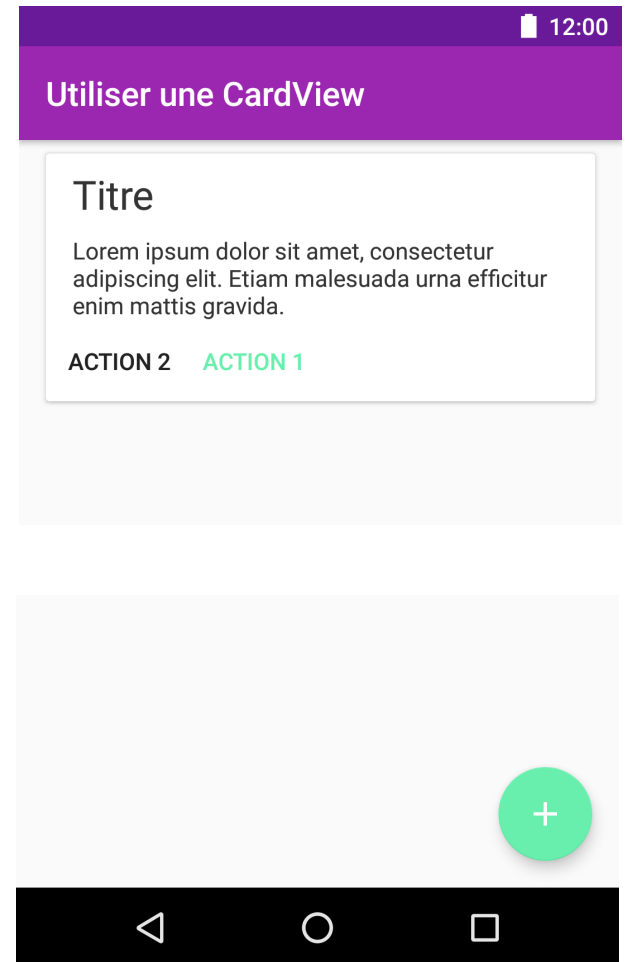
#9B26AF



#68EFAD



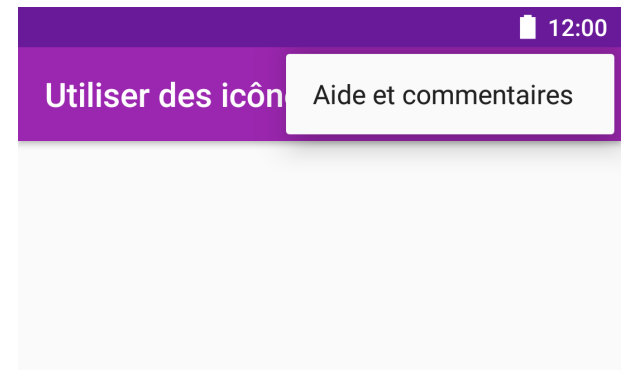
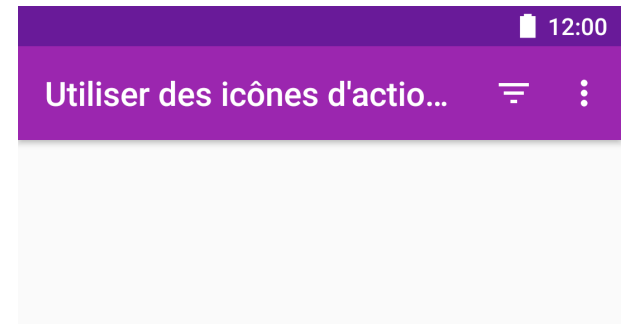
#333333 (couleur du texte)



# Exercice 4 : Utiliser des icônes d'action menu

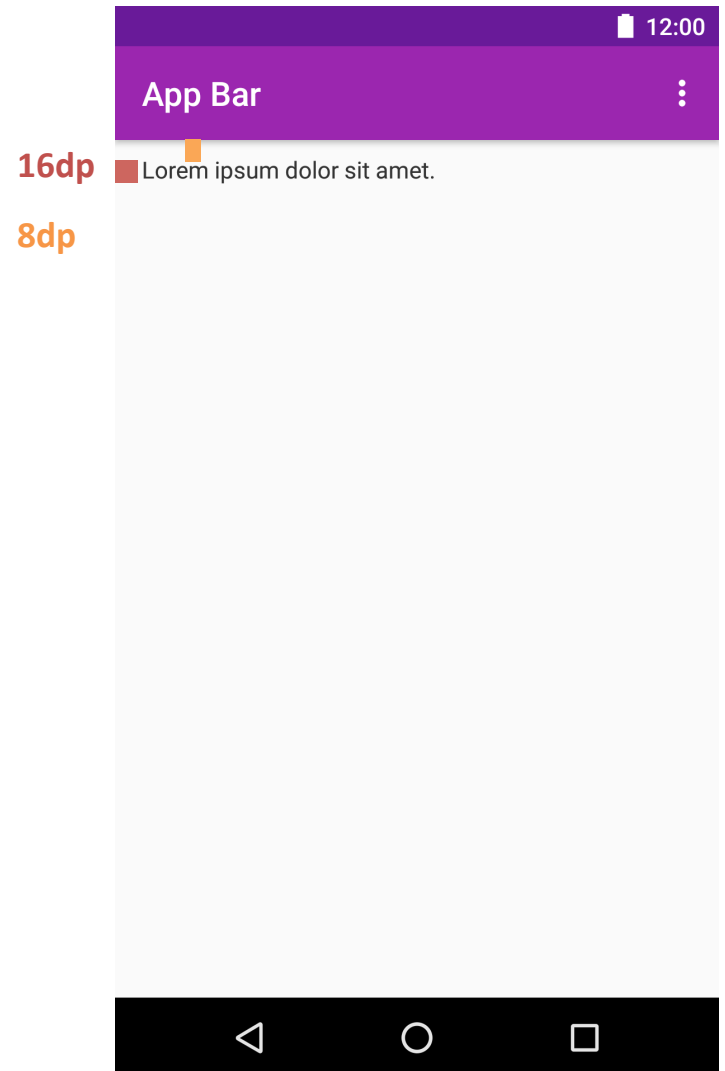
Ajouter à l'Activity IconsActivity, deux icônes d'action :

- Filtre : Action toujours visible. Au clic, afficher le texte `R.string.menu_toast_filter` à l'aide de la fonction `Utils.showToast()`
- Aide et commentaires : Action jamais visible. Au clic, afficher le texte `R.string.menu_toast_help_comment` à l'aide de la fonction `Utils.showToast()`



# Exercice 5 : Utiliser une Toolbar comme App bar

- Utiliser une Toolbar comme App Bar dans l'Activity AppCompatActivity et affecter le texte R.string.app\_bar\_title
- Afficher le texte R.string.text dans le layout
- Ajouter le menu R.menu.menu\_default



# Exercice 6 : Utiliser un spinner dans l'App bar

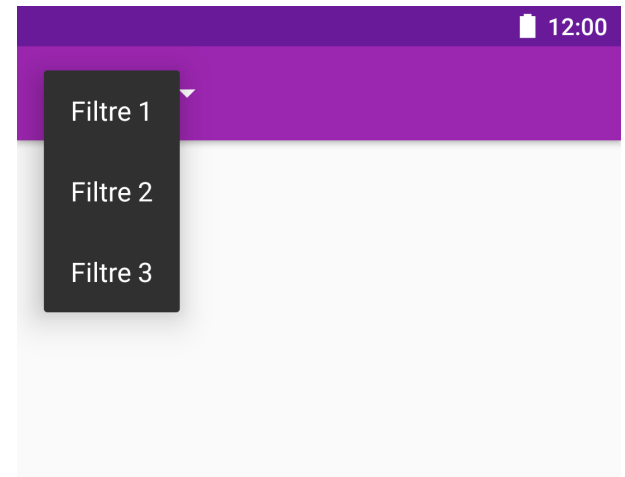
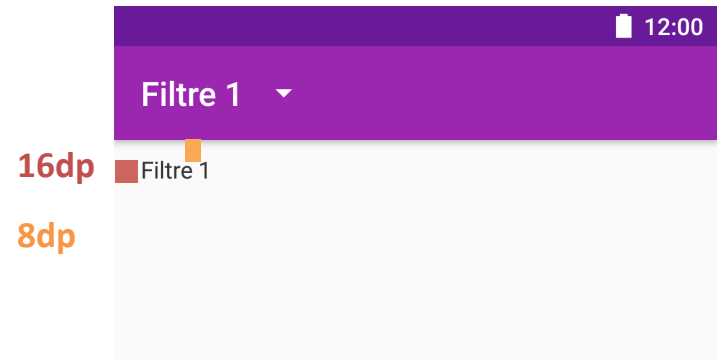
- Dans l'Activity AppBarSpinnerActivity, utiliser une Toolbar comme App Bar et ajouter un Spinner.
- Le spinner contient trois items. Il utilise le layout `android.R.layout.simple_list_item_1` pour l'affichage des items dans la popup.
- Au clic sur un item, le texte de la page se met à jour pour refléter le filtre sélectionné

Style du texte du spinner dans l'App bar :

fontFamily : sans-serif-medium

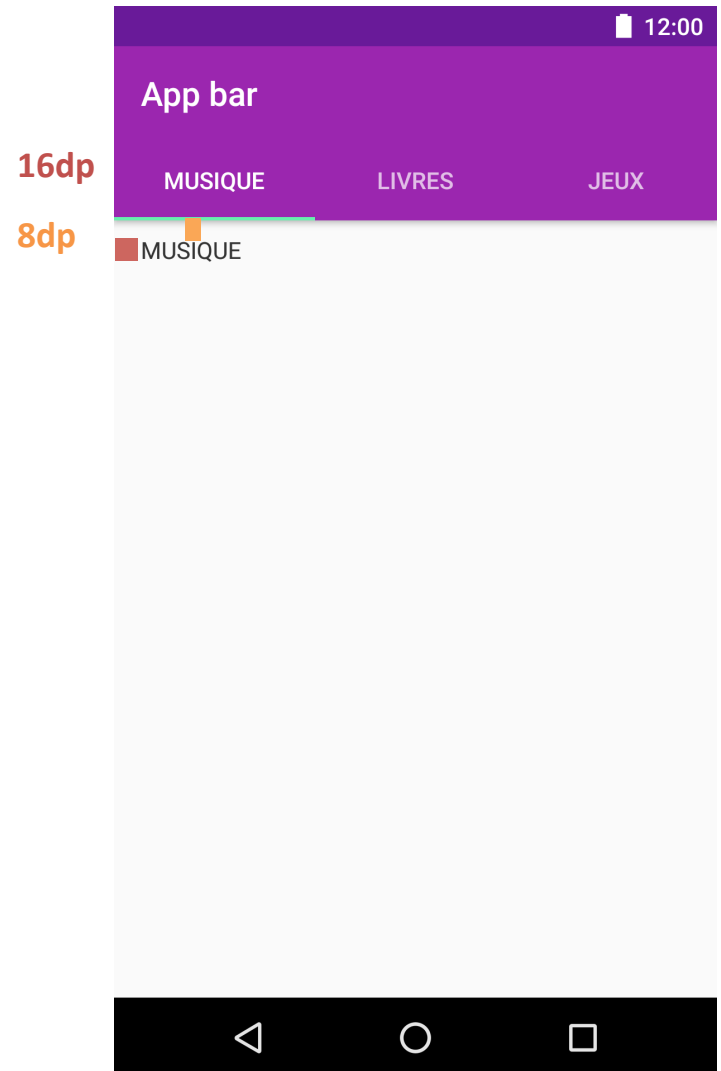
textColor : #ffffff

textSize : 20sp



# Exercice 7 : Utiliser des tabs dans l'App bar

- Utiliser une Toolbar comme App Bar dans l'Activity AppBarTabsActivity et affecter le texte R.string.app\_bar\_title
- Ajouter trois tabs dans l'App Bar avec les textes R.string.fragment\_music, R.string.fragment\_books, R.string.fragment\_games
- Utiliser le fragment MainFragment pour afficher le contenu de chaque tab.
- La TextView de MainFragment, affiche le texte de la tab sélectionnée



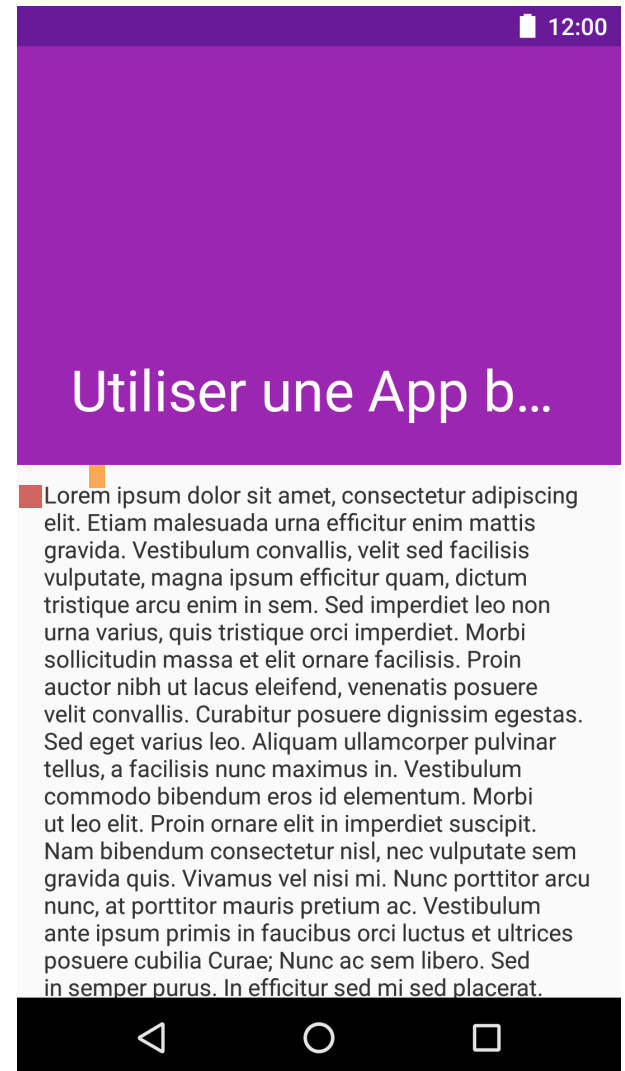


# Exercice 8 : Utiliser une App bar collapsible

- Utiliser une Toolbar comme App bar dans l'Activity AppBarCollapsibleActivity et affecter le texte `R.string.exercise_collapsible`
- Permettre à l'App bar de s'étendre en hauteur sur 250dp
- En plus de l'App bar, l'écran contient une TextView avec le texte `R.string.text_long`

16dp

8dp



## Exercice 8 : Utiliser une App bar collapsible



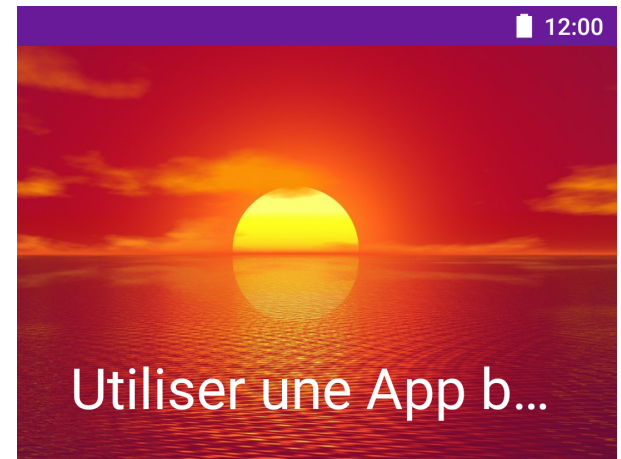
# Exercice 9 : Utiliser une App bar collapsible avec un effet parallaxe

- Utiliser une Toolbar comme App bar dans l'Activity `AppBarCollapsibleParallaxActivity` et affecter le texte `R.string.exercise_collapsible_parallax`
- Permettre à l'App bar de s'étendre en hauteur sur 250dp
- L'App bar contient une `ImageView` affichant l'image à cette url :  
<http://livehdwallpaper.com/wp-content/uploads/2016/10/Sunset-Photos.jpg>
- L'image est `centerCrop` dans l'`ImageView`
- En plus de l'App bar, l'écran contient une `TextView` avec le texte `R.string.text_long`

## Info

Pour charger une image à partir d'une url et la placer dans une `ImageView`, utiliser la librairie externe Picasso :

```
Picasso.with(context).load("my_url").into(imageView);
```



16dp

8dp

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam malesuada urna efficitur enim mattis gravida. Vestibulum convallis, velit sed facilisis vulputate, magna ipsum efficitur quam, dictum tristique arcu enim in sem. Sed imperdiet leo non urna varius, quis tristique orci imperdiet. Morbi sollicitudin massa et elit ornare facilisis. Proin auctor nibh ut lacus eleifend, venenatis posuere velit convallis. Curabitur posuere dignissim egestas. Sed eget varius leo. Aliquam ullamcorper pulvinar tellus, a facilisis nunc maximus in. Vestibulum commodo bibendum eros id elementum. Morbi ut leo elit. Proin ornare elit in imperdiet suscipit. Nam bibendum consectetur nisl, nec vulputate sem gravida quis. Vivamus vel nisi mi. Nunc porttitor arcu nunc, at porttitor mauris pretium ac. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc ac sem libero. Sed in semper purus. In efficitur sed mi sed placerat.

## Exercice 9 : Utiliser une App bar collapsible avec un effet parallaxe



# Exercice 10 : Utiliser une App bar collapsible avec un FAB

- Utiliser une Toolbar comme App bar dans l'Activity AppBarCollapsibleFabActivity et affecter le texte R.string.exercise\_collapsible\_fab
- Permettre à l'App bar de s'étendre en hauteur sur 250dp
- Ajouter un FAB au niveau de l'App bar étendue. Au clic, afficher le texte R.string.fab\_toast\_action à l'aide de la fonction Utils.showToast()
- En plus de l'App bar, l'écran contient une TextView avec le texte R.string.text

