



## TP N°4

## Les Menus

### 1. Menus

Il existe plusieurs types de menus comme partie la barre d'action, des menus contextuels, des popups menu, drawer

#### 1.1 Créer un menu d'option ou un ActionBar

Pour ce faire

- Créer un fichier xml dans le dossier menu de votre application
- Ecrire le code suivant dans le fichier xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:tools="http://schemas.android.com/tools"
      xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:ActionBar="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/menu_search"
    android:title="@string/menu_search"
    android:icon="@drawable/search"
    ActionBar:showAsAction="ifRoom|withText"/>
</menu>
```

- Dans votre fichier kotlin écrire le code suivant dans la méthode ***onCreateOptionsMenu*** pour lier la barre d'action à l'activité

```
override fun onCreateOptionsMenu(menu: Menu, inflater: MenuInflater) {
    inflater.inflate(R.menu.actionbar, menu)
}
```

#### d) Gestion du clic

Afin de gérer le clic sur les différents éléments surchargez la méthode ***onOptionsItemSelected*** et programmer l'action qui correspond à l'élément cliqué ***search*** et l'action ***exit***.

Il s'agit d'afficher une notification « **Menu Search selected** » pour le menu **search**

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.menu_search -> {
```

```

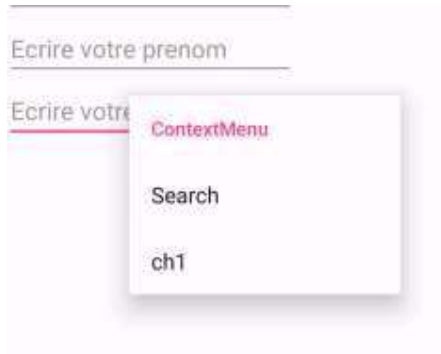
.....
// afficher un Toast avec le message "Menu search sélectionné"
true
}

.....
// compléter de meme pour menu_exit

else -> super.onOptionsItemSelected(item)
}
}

```

## 1.2 Menu contextuel



### a) Relier le menu à un objet

Dans la méthode onCreate, relier la vue email au menu par l'ajout de de l'istruction suivante :

```
registerForContextMenu(email);
```

### b) Implémenter OnCreateContextMenu

```

override fun onCreateContextMenu(menu: ContextMenu, v: View,
                                menuInfo: ContextMenu.ContextMenuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo)
    val inflater: MenuInflater = menuInflater
    inflater.inflate(R.menu.contextmenu, menu)
    // menu.setHeaderTitle("ContextMenu");
}

```



Remarque : objet qui Si notre activité a plusieurs vues ayant un menu contextuel différent, nous pouvons utiliser ContextMenu.ContextMenuInfo qui fournit des informations supplémentaires sur l'élément sélectionné.les paramètres pour déterminer le menu contextuel à gonfler.

### c) Implémenter OnContextItemSelected

```

override fun onContextItemSelected(item: MenuItem): Boolean {
    val info = item.menuInfo as AdapterView.AdapterContextMenuInfo
    return when (item.itemId) {
        .....
        else -> super.onContextItemSelected(item)
    }
}

```

## 2. Travail à faire

A. Ajouter un bouton avec popup menu :

Pour cela :

A.1 Instanciez un PopupMenu avec son constructeur, qui prend le contexte de l'application actuelle et la view à laquelle le menu doit être ancré.

A.2 Utilisez MenuInflater pour apparaître votre menu (xml) dans l'objet Menu renvoyé par PopupMenu getMenu().

A.3 Appelez PopupMenu.show()

B. Ajouter un menu avec un groupe de sous menu. Le menu du groupe est nommé « couleur » et les sous menu cocheable « checkable menu items » (avec boutons d'option), formé de deux éléments « rouge » et « bleu ».