

PROGRAMMATION DE COMPOSANTS MOBILES (ANDROID)

Wieslaw Zielonka

Menu

Menu

Créer un répertoire res/menu dans les ressources.
Y mettre le fichier layout de menu.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/sort"
        android:title="tri"
        app:showAsAction="ifRoom">
        <menu>
            <item
                android:id="@+id/byName"
                android:title="par nom" />
            <item
                android:id="@+id/byFirstname"
                android:title="par prénom" />
        </menu>
    </item>
    <item
        android:id="@+id/action_search"
        android:icon="@drawable/ic_action_search"
        android:title="@string/action_search"
        app:showAsAction="ifRoom"
        app:actionViewClass="androidx.appcompat.widget.SearchView" />

    <item
        android:id="@+id/quitter"
        android:title="quitter"
        app:showAsAction="never" />

</menu>
```

dans chaque item du menu :

`app:showAsAction="ifRoom"`

le item apparaît s'il y a de la place

`app:showAsAction="never"`

le item tjrs dans l'espace
de débordement

Menu

Créer un répertoire res/menu dans les ressources.
Y mettre le fichier layout de menu.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">

  <item
    android:id="@+id/sort"
    android:title="tri"
    app:showAsAction="ifRoom">
    <menu>
      <item
        android:id="@+id/byName"
        android:title="par nom" />
      <item
        android:id="@+id/byFirstname"
        android:title="par prénom" />
    </menu>
  </item>
  <item
    android:id="@+id/action_search"
    android:icon="@drawable/ic_action_search"
    android:title="@string/action_search"
    app:showAsAction="ifRoom"
    app:actionViewClass="androidx.appcompat.widget.SearchView" />

  <item
    android:id="@+id/quitter"
    android:title="quitter"
    app:showAsAction="never" />

</menu>
```

le menu peut avoir
des sous-menu

l'icône de recherche à mettre dans
res/drawable

Menu

Changer le style de l'application dans AndroidManifest.xml :

```
<application
```

```
.....
```

```
android:theme="@style/Theme.AppCompat.Light.NoActionBar">
```

Il faut choisir un thème qui ne possède pas d'ActionBar.

Google recommande d'implémenter ActionBar avec un Toolbar, donc il faut un thème qui n'a pas d'ActionBar.

Si dans une activité nous voulons un menu avec le bouton "back" qui permet de revenir à l'activité précédente alors, toujours dans AndroidManifest.xml, il faut ajouter un attribut android:ParentActivityName dans l'activité:

```
<activity android:name=".AddAuthorsActivity"  
    android:parentActivityName=".MainActivity"  
/>
```

Le bouton "back" dans le menu de AddAuthorsActivity permet de revenir vers MainActivity

Menu

Ajouter Toolbar dans le layout de l'activité (pour que le menu apparaisse en haut de l'écran):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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=".AddAuthorsActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/my_toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        android:elevation="4dp"
        android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
```

... et d'autres widgets du layout

Menu

Dans onCreate() de l'activité :

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    binding = ActivityAddAuthorsBinding.inflate(layoutInflater)  
    setContentView(binding.root)  
  
    /* indiquer que le Toolbar joue le rôle de ActionBar */  
    setSupportActionBar( binding.myToolbar )  
  
    /*Si le bouton "back" dans le menu alors il faut mettre en place.  
    * S'il n'y a pas de bouton "back" dans le menu alors  
    *la ligne suivante à supprimer */  
    supportActionBar?.setDisplayHomeAsUpEnabled(true)
```


Menu

Réécrire la méthode onCreateOptionsMenu() de l'activité :

```
/* la ligne suivante uniquement s'il y a la recherche dans le menu */
private var searchItem : MenuItem? = null

override fun onCreateOptionsMenu(menu: Menu): Boolean {
    val inflater: MenuInflater = menuInflater
    inflater.inflate(R.menu.author_menu_layout, menu)

    /* le code suivant uniquement si vous avez la recherche dans le menu */
    searchItem = menu.findItem(R.id.action_search)
    val searchView = searchItem?.actionView as SearchView
    searchView.setOnQueryTextListener(object :
        SearchView.OnQueryTextListener {
            override fun onQueryTextSubmit(s: String): Boolean {
                return true
            }
            override fun onQueryTextChange(s: String): Boolean {

                return true
            }
        })

    /* toujours retourner true */
    return true
}
```


Menu

Implémenter les actions associées aux items de menu dans la fonction `onOptionsItemSelected()` de l'activité :

```
override fun onOptionsItemSelected(item: MenuItem) : Boolean = when
( item.itemId ) {
    R.id.quitter -> {
        finish()
        true
    }
    R.id.byName ->{
        Log.d("AddAuthors", "menu byName")
        true
    }
    R.id.byFirstname ->{
        Log.d("AddAuthors", "menu byFirstname")
        true
    }
    else -> {
        super.onOptionsItemSelected(item)
    }
}
```

sélection d'action en fonction
de MenuItem actionné par
utilisateur

Notez que pas d'action pour le bouton "back", android implémente seul ce qu'il faut.