

TP5+6 : Composants Android(Suite)

II. Menus

Sur Android, les menus permettent d'ajouter des fonctionnalités à une application en fournissant des opérations supplémentaires, initialement cachées à l'utilisateur.

Il existe deux types de menus :

- Un menu *d'options* : déclenché par le bouton matériel *Menu* sur le téléphone.
- Un menu *contextuel* : déclenché par un évènement sur un élément de l'interface, par exemple un *long clic*.

Ces deux types de menus peuvent contenir :

- Du texte
- Des icônes
- Des boutons radios
- Des cases à cocher
- Des sous-menus
- Des raccourcis...

II. 1. Menu contextuel

Pour définir un menu contextuel, qui sera déclenché suite à un long clic sur un élément (que j'appelle ici *element* : il peut être un bouton, un textView, ou même le layout en entier) :

1. Créer un attribut pour l'élément auquel on veut ajouter un menu contextuel, et l'associer à l'élément graphique avec la méthode *findViewById(...)*.
2. Ajouter un écouteur pour le long clic, de la même manière que nous avons ajouté un écouteur pour le clic dans le TP précédent. Cet écouteur (*Listener*), va ordonner l'affichage du menu contextuel quand on fait un long clic sur *element* :

```
element.setOnLongClickListener(new OnLongClickListener() {  
  
    @Override  
    public boolean onLongClick(View v) {  
        v.showContextMenu();  
        return false;  
    }  
});  
element.setOnCreateContextMenuListener (this) ;
```

3. Indiquer le comportement de ce menu. Pour cela, générer la méthode *onCreateContextMenu* dans votre activité. Elle est appelée quand un menu contextuel va être affiché.
4. Dans le corps de cette méthode, on peut ajouter des menus. Pour cela, utiliser la méthode :

```
menu.add(groupID, itemID, ordre, "nom du menu") ;
```

Cette méthode permet de créer un nouveau menu. Les paramètres nécessaires sont les suivants :

- *groupID* : identifiant du groupe. Il est possible de regrouper les éléments, mais dans notre cas, on n'en a pas besoin, on lui donne donc la valeur 0 ;
- *itemID* : identifiant de ce menu. Il nous sera utilisé pour identifier ce menu parmi les autres. On doit donner un identifiant différent à chaque menu (1, 2, 3... par exemple).
- *ordre* : associer un ordre d'affichage au menu. On donnera toujours la valeur 0.
- *Nom du menu* : chaîne qui représente le titre du menu.

5. Nous devons dire quel est le comportement à faire quand on clique sur cet élément du menu. Pour cela, générer la méthode :

```
public boolean onOptionsItemSelected(MenuItem item) {...}
```

Dans le corps de cette méthode, indiquer le comportement à adopter, si on clique sur le menu 1 ou 2 :

```
switch(item.getItemId()){  
    case 1:  
        //traitement 1  
    case 2:  
        //traitement 2  
}
```

TAF-4 : Créer un menu contextuel sur les deux boutons radios. Il doit contenir deux menus :

1. « Taux dinar -> euro » : affiche dans un Toast le taux de conversion du dinar vers l'euro.
2. « Taux euro -> dinar » : affiche dans un Toast le taux de conversion de l'euro vers le dinar.

II. 2. Menu d'options

Le menu d'options se gère presque de la même manière qu'un menu contextuel, mais sans la gestion de l'élément graphique auquel est associé le menu contextuel.

1. Générer la méthode *onCreateOptionsMenu* dans votre activité. Cette méthode permet de déterminer le comportement de l'application quand le bouton *Menu* de votre téléphone est appuyé. Elle prend comme paramètre un objet de type *Menu* : c'est le menu que vous allez manipuler.
2. Dans le corps de la méthode *onCreateOptionsMenu* (avant l'instruction *return*), ajouter les différents menus dont vous avez besoin.
3. Générer la méthode *onOptionsItemSelected* pour définir le comportement au clic d'un menu.

TAF-5 : Créer un menu d'options qui contient deux menus :

1. *Conversion C <-> F* : qu'on laissera vide pour l'instant
2. *Quitter* : permet de quitter l'application.

Indication : Pour quitter l'application, il faut appeler la méthode *finish()*.