

## TP5\_ Android

# Traitement des Menus sous Android

### 1. Traitement des Menus:

Les widget et les layouts vus précédemment sont les briques de base de la construction d'interfaces. Cependant, Android, dans son kit de développement propose encore beaucoup plus : les menus. Les menus font partie des éléments d'interfaces ayant subi un remodelage complet depuis les différentes versions d'Android. Il propose deux catégories distinctes de menus :

Les « Option Menus »

Les « Context Menus »

Ces menus sont différents en nombreux points ; ils ne se déclenchent pas suite au même évènement, n'ont pas un visuel identique, ont leur propre règle de fonctionnement. Sur le plan ergonomique, ils ne servent pas non plus le même type d'interaction utilisateur.

### 2. Les « Option Menus »

Les « option Menus » sont les menus principaux de l'application. Ils sont rattachés aux activités qui ne peuvent qu'en définir qu'un seul. Ils apparaissent quand l'utilisateur clique sur le bouton « Menu » du téléphone ou sur l'ActionBar pour les tablettes et téléphones post la version 3 d'Android. Les éléments de ce type de menu ne peuvent pas contenir des cases à cocher ou des boutons radio. Ces menus sont destinés à présenter une liste d'actions ou à fournir des choix de navigation.

Créer un nouveau projet nommée Menu\_optionnel. Pour rappel, nous privilégions ici la méthode déclarative bien qu'il soit possible de créer ou modifier les menus directement depuis le code java. La première étape consiste à créer un fichier au format XML et de type Menu dans le répertoire res/menu. La syntaxe générée par défaut est la suivante :

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

Les options du menu sont ensuite ajoutées une par une en utilisant la balise item. Ses attributs permettent de paramétrer l'option. Voici les principaux :

Propriété	Description
android:title	Titre de l'option à afficher
android:icon	Image représentant l'icône de l'option
android:onClick	Nom de la méthode de l'activité à exécuter lorsque l'utilisateur clique sur la vue (apparue sous Android 3.0 API 11)
android:showAsAction	Permet de spécifier si cette option doit figurer dans la barre d'action et de quelle façon
android:titleCondensed	Utiliser lorsque le titre de l'item est très long et peut ne pas être affiché.

Sur les appareils en version Android 3.0 et plus, les entrées du menu apparaissent sur la barre d'action, ils seront par défaut regroupés dans le menu déroulant qui s'affiche suite à un clic sur l'icône place à droite. Pour intégrer l'item du menu directement dans la barre, il suffit d'ajouter dans le fichier xml la propriété :

Android:showAsAction= 'ifRoom' ou encore Android:showAsAction= 'ifRoom|withtext' pour afficher à la fois le texte et l'icône.

L'ordre d'apparition des options d'un menu est déterminé par celui de son apparition dans le fichier xml. Cependant, on peut modifier cet ordre moyennant l'attribut android:orderInCategory de l'option. Sa valeur commence par l'indice 0 qui précise dans son ordre d'apparition dans le menu. Il est possible de créer un niveau supplémentaire de sous-menu. Ce sont des menus qui s'affichent lorsque l'utilisateur a sélectionné un choix du premier menu. Cela permet de créer une arborescence de menus à deux niveaux maximum. Il ne peut pas y avoir des sous-menus de sous-menus. Pour créer un sous menu, il suffit d'insérer la balise menu dans un item du premier menu. Les items de ce sous menu doivent être indiqués dans cette nouvelle balise menu.

Le fichier menu final est le suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/item1"
        android:titleCondensed="chercher"
        android:title="chercher fichier ou dossier"
        android:showAsAction="ifRoom"
        android:orderInCategory="1"/>
    <item
        android:id="@+id/item2"
        android:title="Aide"
        android:showAsAction="ifRoom"
        android:enabled="false" ← cette option est affichée mais désactivée
        android:orderInCategory="3"/>

    <item
        android:id="@+id/item3"
        android:title="Parametres"
        android:showAsAction="ifRoom"
```

```

        android:orderInCategory="2">
<menu > ← création d'un sous menu
    <item
        android:id="@+id/item4"
        android:title="Reseaux"/>
    <item
        android:id="@+id/item5"
        android:title="telephone"/>
    </menu>
</item>
</menu>

```

Dans le code Java de l'activité du projet, il faut enlever tous les imports et le code java concernant les fragments et les widget et view pour ne pas encombrer le projet. Dans la partie importe de l'activité, il faut qu'il ait uniquement les imports suivants :

```

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu; ← Traitement du Menu
import android.view.MenuInflater; ← Traitement du Menu
import android.view.MenuItem; ← Traitement du Menu
import android.widget.Toast; ← traitement des notifications

```

La transformation de la description du menu depuis le format XML en instance de type Menu se fait depuis l'activité en utilisant un objet de type MenuInflater. Pour rattacher à chacun des options du menu des actions, il y a trois possibilités :

1. Enregistrer un listener de type MenuItem.OnMenuItemClickListener sur le MenuItem.
2. Redéfinir la méthode onOptionsItemSelected de l'activité (c'est la plus facile).
3. Référencer la méthode réagissant au clic de menu directement dans le XML avec l'attribut android:onClick. Cette méthode devra être définie dans l'activité et accepter un paramètre de type MenuItem.

Comme nous avons retenu la deuxième possible pour le traitement des options du menu le code java final de l'activité est le suivant:

```

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) { ← Fonction qui permet la création du menu d'options
    }
}

```

```

    MenuInflater inflater=getMenuInflater(); //création de l'objet de type MenuInflater
    inflater.inflate(R.menu.main, menu) //conversion du menu xml en objet MenuInflater
    return true;
}

```

```

@Override // fonction qui determine le traitement des options
public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {
        case R.id.item1:

            // Comportement a faire si item1 est selectionne
            Toast msg = Toast.makeText(this, "l'option chercher est selectionnee",
                Toast.LENGTH_SHORT);
            msg.show();

            return true;
        case R.id.item2:
            // Comportement a faire si item2 est selectionne
            Toast msg1 = Toast.makeText(this, "l'option Aide est
                selectionnee", Toast.LENGTH_SHORT);
            msg1.show();

            return true;
        case R.id.item3:
            // Comportement a faire si item3 est selectionne
            Toast msg2 = Toast.makeText(this, "l'option parametres est selectionnee",
                Toast.LENGTH_SHORT);
            msg2.show();

            return true;
        case R.id.item4:
            // Comportement a faire si item4 est selectionne
            Toast msg4 = Toast.makeText(this, "l'option Reaux est selectionnee",
                Toast.LENGTH_SHORT);
            msg4.show();

            return true;
        case R.id.item5:
            // Comportement a faire si item5 est selectionne
            Toast msg5 = Toast.makeText(this, "l'option Telephone est selectionnee",
                Toast.LENGTH_SHORT);
            msg5.show();

            return true;
        default:
            return false;
    }
}

```

ne sera  
car l'option  
2 est  
desactivee

### 3.Les « Contextes Menus »

Les menus contextuels ont un fonctionnement très proche des menus d'options. Ils ne sont néanmoins pas liés a une activité mais a un objet de type View. L'utilisateur active ce menu en faisant une pression long sur une vue. Ce sont les équivalents des menus qui apparaissent après un clic droit sur un élément graphique des applications traditionnelles de bureau. Sur un même écran, plusieurs menus contextuels pourront donc être enregistrés ce qui explique que, contrairement aux menus d'options, ces menus ne supportent pas les raccourcis clavier ni

d'ailleurs les icônes. Ce menu disparaît lorsque l'utilisateur sélectionne une option ou lorsqu'il presse la touche Retour.

Créer un nouveau projet nommé Menu\_Contextuel. Insérer un bouton dans le layout de ce projet intitulé « Test\_Menu\_Contextuel ». le fichier XML du layout du projet est le suivant :

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.test_menucontextuel.Menu_Contextuel"
    tools:ignore="MergeRootFrame" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Test Menu Contextuel" />

</FrameLayout>
```

Dans le répertoire res/menu créer un fichier XML nommé "menu\_contextuel" du type Menu et insérer les items suivant dans le fichier menu qui doit être comme suit :

```
<menu 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"
    tools:context="com.example.test_menucontextuel.Menu_Contextuel" >

    <item
        android:id="@+id/item1"
        android:orderInCategory="1"
        android:title="Rechercher"
        app:showAsAction="never"/>
    <item
        android:id="@+id/item2"
        android:orderInCategory="3"
        android:title="Aide"
        android:enabled="false"
        app:showAsAction="never"/>
    <item
        android:id="@+id/item3"
        android:orderInCategory="2"
        android:title="Parametres"
        app:showAsAction="never"/>

</menu>
```

Dans le code Java de l'activité du projet, il faut enlever tous les imports et le code java concernant les fragments et les widget pour ne pas encombrer le projet. Dans la partie import de l'activité, il faut qu'il ait uniquement les imports suivants :

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

```

import android.view.ContextMenu.ContextMenuInfo;
import android.view.MenuInflater;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

```

traitement du menu contextuel  
 traitement de la vue sur la quelle est rattachée le menu contextuel  
 traitement des notifications

La transformation de la description du menu depuis le format XML en instance de type Menu se fait depuis l'activité en utilisant, comme nous l'avons vu précédemment, un objet de type MenuInflater. Pour rattacher à chacun des options du menu contextuel des actions, nous devons redéfinir la méthode onOptionsItemSelected de l'activité. Le code java final de l'activité est le suivant:

```

public class Menu_Contextuel extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu_contextuel);
        Button b1= (Button) findViewById(R.id.button1);
        registerForContextMenu(b1);

    }

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo)
    {
        super.onCreateContextMenu(menu, v, menuInfo);

        switch(v.getId())
        {
            case R.id.button1:

                getMenuInflater().inflate(R.menu.menu_contextuel, menu);

                break;
        }
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {

        // traitement des actions des differentes options du menu contextuel
    }
}

```

Recuperation de l'id du bouton  
 Rattachement du menu contextuel au bouton  
 création de l'objet menu contextuel  
 conversion du menu xml en objet menu contextuel

```

int id = item.getItemId();
switch (id) {
    case R.id.item1:

        // Comportement a faire si item1 est selctionne
        Toast msg = Toast.makeText(this, "l'option chercher est selctionnee",
            Toast.LENGTH_SHORT);
        msg.show();

        return true;
    case R.id.item2:
        // Comportement a faire si item2 est selctionne
        Toast msg1 = Toast.makeText(this, "l'option Aide est selctionnee",
            Toast.LENGTH_SHORT);
        msg1.show();

        return true;
    case R.id.item3:
        // Comportement a faire si item3 est selctionne
        Toast msg2 = Toast.makeText(this, "l'option parametres est selctionnee",
            Toast.LENGTH_SHORT);
        msg2.show();

        return true;

    default:
        return super.onContextItemSelected(item);
        //return false;
}

}

}

```