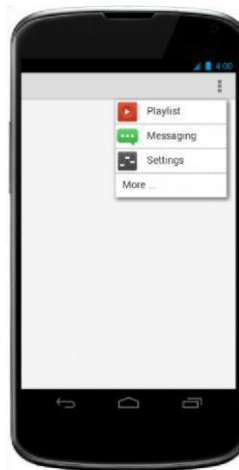




420-AM1-TT

les Menus et les WebViews



Enseignant : Rodrigue Tonga

Nom :

Prénom

Session : Automne 2023

Département d'informatique

Institut Teccart

1. But :

Le but de ce document guidé est de vous permettre de poursuivre votre découverte des outils permettant ou facilitant les interactions avec l'utilisateur.

2. Contexte :

En plus des différents types de boîtes de dialogues que nous avons étudiés dans le labo3, le système d'exploitation Android prévoit plusieurs autres mécanismes pour faciliter l'interaction entre votre application et ses utilisateurs. Nous allons donc découvrir comment les classes **Menus**, **Sub-menus**, **Context Menus**, **Popup Menus** et **WebView** participent à cet écosystème.

2.2 Les Menus

Les menus sont des composants standards permettant à un utilisateur de fournir des options ou d'exécuter des actions dans une activité. Depuis la version 3.0 du système d'exploitation Android, les menus font partie intégrante de la bar d'action des activités.



Chaque activité peut donc avoir son propre menu.

2.2.1 Création d'un Menu à partir du code Java

Pour être en mesure de créer un menu simple à partir d'un code en Java, il faut suivre les étapes suivantes :

- 1- Créez une nouvelle application Android et faites un Override de la fonction **onOptionsItemSelected()**.
-

```

public class MainActivity extends AppCompatActivity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {

    }

}

```

- 2- Pour chacun des items que vous voulez inclure dans votre menu, vous devez utiliser la fonction **add()**. **Menu.add()** prend 4 arguments, Group ID, menu item id, order et text.

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);
    MenuItem mi1 = menu.add(Menu.NONE, Menu.FIRST, 0, "VB.NET");
    MenuItem mi2 = menu.add(Menu.NONE, Menu.FIRST + 1, 1, "C#");
    MenuItem mi3 = menu.add(Menu.NONE, Menu.FIRST + 2, 3, "JAVA");
    MenuItem mi4 = menu.add(Menu.NONE, Menu.FIRST + 3, 4, "PHP");
    MenuItem mi5 = menu.add(Menu.NONE, Menu.FIRST + 4, 2, "ASP.NET");

    return true;
}

```

Exécutez votre application et représentez votre menu



- 3- Modifier la fonction **onCreateOptionsMenu()**, pour que les items soient présentés dans l'ordre de définition.
- 4- On peut ensuite personnaliser le menu en utilisant les fonction **setShortcut()**, **setIcon()**, etc.
- 5- Assigner des fonctionnalités aux items du menu en appelant **setIntent()** ou en faisant un override de la fonction **onOptionsItemSelected()** de votre activité.
-

```
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    super.onOptionsItemSelected(item);
    Intent i;
    switch (item.getItemId())
    {
        case Menu.FIRST:
            i = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.youtube.com/watch?v=Y016WzvFyz8"));
            startActivity(i);
            return true;
        case Menu.FIRST + 1:
            i = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.youtube.com/watch?v=QBQtUERSMI&t=36s"));
            startActivity(i);
            return true;
        case Menu.FIRST + 2:
            https://www.youtube.com/watch?v=fEDdxtTeL_g&t=936s
            i = new Intent(Intent.ACTION_VIEW, Uri.parse("https://www.youtube.com/watch?v=fEDdxtTeL_g&t=936s"));
            startActivity(i);
            return true;
        case Menu.FIRST + 3:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();
            return true;
        case Menu.FIRST + 4:
            Toast.makeText(this, item.getTitle(), Toast.LENGTH_LONG).show();
            return true;
    }
    return true;
}
```

6- *Que ce passe t'il quand vous sélectionnez les différents items du menu*

7- Ajoutez à votre menu un item permettant de faire jouer une chanson et un autre permettant de fermer votre application. Démontrez le bon fonctionnement de votre menu à votre enseignant.

2.2.2 Création d'un menu à partir d'un fichier de ressource

Il est possible de créer un menu à partir d'un fichier de ressource et d'un objet de type **MenuInflater**. Voici les étapes à suivre :

- 1- Créez un nouveau fichier XML dans le dossier **res/menu** si le fichier n'est pas déjà disponible dans votre projet.
-

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item
    android:id="@+id/settings"
    android:orderInCategory="100"
    android:title="Settings"
/>

<item
    android:id="@+id/exit"
    android:orderInCategory="101"
    android:title="exit"
/>

</menu>

```

2- Override la fonction **onCreateOptionsMenu** de votre activité

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);

    MenuInflater minf = getMenuInflater();
    minf.inflate(R.menu.mon_menu, menu);

    return true;
}

```

3- Exécutez l'application, que ce passe t'il si vous interchangez les valeurs de **android:orderInCategory** pour les deux items.

Si nous voulions réagir aux items de notre menu, les fonctions **setIntent** et **onOptionsItemSelected** doivent être utilisées comme précédemment. Dans l'éventualité où nous voudrions ajouter des sousmenus, il faudrait :

1- Créer un objet de type sous-menu en appelant la méthode **addSubMenu()**

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);

    MenuItem toto = menu.add(Menu.NONE, Menu.FIRST, Menu.NONE, "TOTO");
    MenuItem titi = menu.add(Menu.NONE, Menu.FIRST + 1, Menu.NONE, "TITI");
    SubMenu tata = menu.addSubMenu("TATA");
    SubMenu tutu = menu.addSubMenu("TUTU");

    return true;
}

```

2- Ajouter des branches aux sous-menus

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);

    MenuItem toto = menu.add(Menu.NONE, Menu.FIRST, Menu.NONE, "TOTO");
    MenuItem titi = menu.add(Menu.NONE, Menu.FIRST + 1, Menu.NONE, "TITI");
    SubMenu tata = menu.addSubMenu("TATA");
    tata.add("sous-menu 1");
    SubMenu tutu = menu.addSubMenu("TUTU");
    tutu.add("sous-menu 2");

    return true;
}
```

2.2.3 Les Menus contextuels

Un menu contextuel est associé à une vue et s'active à la sélection (un clic d'environ 3 sec) de cette dernière. Chaque activité peut donc avoir plusieurs menus contextuels, mais seulement un menu d'options.

Les étapes pour définir et peupler un menu contextuel sont relativement semblables à celles nécessaires pour la création d'un menu à option.

- 1- Déclarer un objet de type **View**, comme un **Button**, **TextView**, **EditView**, etc.

```
private TextView txt;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    txt = new TextView(this);
    txt.setText("vivre Teccart...les profs sont extraordinaires");
    setContentView(txt);
}
```

- 2- Il faut ensuite signifier que notre vue doit être associée à un menu contextuel.

```
registerForContextMenu(txt);
```

- 3- Override la méthode **onCreateContextMenu** qui reçoit trois paramètres.
-

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo)
{
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.setHeaderTitle("Mon premier menu contextuelle");
    menu.add(Menu.NONE, Menu.FIRST, Menu.NONE, "item 1");
    menu.add(Menu.NONE, Menu.FIRST + 1, Menu.NONE, "item 2");
    menu.add(Menu.NONE, Menu.FIRST + 2, Menu.NONE, "item 3");
}
```

- 4- Que se passe t'il si vous ajoutez l'option **.setCheckable(true)** à l'un des items de votre menu contextuel

```
menu.add(Menu.NONE, Menu.FIRST + 1, Menu.NONE, "item 2").setCheckable(true);
```

- 5- Que se passe t'il si vous effectuez la modification suivante. Que représentent les trois paramètres de la fonction **setGroupCheckable**.

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo)
{
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.setHeaderTitle("Mon premier menu contextuelle");
    menu.add(Menu.NONE, Menu.FIRST, Menu.NONE, "item 1");
    menu.add(1, Menu.FIRST + 1, Menu.NONE, "item 2");
    menu.add(1, Menu.FIRST + 2, Menu.NONE, "item 3");

    menu.setGroupCheckable(1, true, true);
}
```

Il est important de noter qu'il existe plusieurs mécanismes pour répondre aux sélections dans un menu contextuel.

- 1- Override la méthode **onContextItemSelected(MenuItem)**
-

```
@Override
public boolean onContextItemSelected(MenuItem item)
{
    return super.onContextItemSelected(item);
    //votre code ici
}
```

2- En utilisant un MenuItem ClickListener

```
menu.add(1, Menu.FIRST + 2, Menu.NONE, "item 3").setOnMenuItemClickListener(
    new MenuItem.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item) {
            //votre code ici
            return false;
        }
    });
```

- 3- En assignant un intent à votre **MenuItem**, à l'aide de **setIntent(new Intent(this, MonAutreActivite.class))**. Android exécute automatiquement **startActivity** quand l'item est sélectionné.

2.2.4 Création de Popup Menus

Apparentés aux menus contextuels, les popup menus peuvent être associés à une vue particulière et afficher un menu de type déroulant; pour les utiliser, il suffit de suivre les étapes suivantes :

- 1- Créer votre menu à l'aide d'une ressource XML

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/item1"
        android:orderInCategory="200"
        android:title="item1"/>
    <item android:id="@+id/item2"
        android:orderInCategory="201"
        android:title="item2"/>
</menu>
```

- 2- Ajoutez un événement **onClick**

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Pop me up"
    android:id="@+id/btn"
    android:onClick="showPopup"/>
```

- 3- Récupérer et afficher le menu

```
public void showPopup(View v)
{
    PopupMenu pop = new PopupMenu(this, v);
    MenuInflater m = pop.getMenuInflater();
    m.inflate(R.menu.mon_popup_menu, pop.getMenu());
    pop.show();
}
```

- 4- Que doit-on faire pour être en mesure de réagir aux sélections dans un Popup menu.

2.3 Création d'un WebView

Comme vous allez le découvrir sous peu, il est possible d'intégrer un navigateur web à votre application en utilisant un WebView. Imaginez une activité qui liste une série de titres d'article web dans un ListView et qui doit afficher l'article sélectionné par l'utilisateur; voilà le type de situation pour lesquelles un WebView est intéressant.

Voici un exemple simple d'utilisation d'un WebView :

- 1- Ajouté à votre manifeste la permission suivante

```
<uses-permission android:name="android.permission.INTERNET"/>
```

- 2- Ajoutez une balise **WebView** au fichier XML représentant l'interface de votre activité

```
<WebView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/web" />
```

- 3- Ajoutez le code suivant à votre activité

```
WebView w;
WebViewClient webc;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    w = (WebView) findViewById(R.id.web);
    w.getSettings().setJavaScriptEnabled(true);
    webc = new WebViewClient();
    w.setWebViewClient(webc);
    w.loadUrl("https://mail.aadmtl.com");
}
```