

TP N° 3

1. Composants Android

1.1 Gérer les boutons radios et cases à cocher

Un bouton radio est un bouton à deux états qui peut être soit coché (*checked*) ou décoché (*unchecked*). Les boutons radios sont en général utilisés dans un groupe *RadioGroup*. Au sein d'un même groupe, un seul bouton radio peut être coché.

Pour gérer l'état d'un bouton radio, il faut suivre les étapes suivantes :

- Créer un attribut de type *RadioButton* dans votre activité (par exemple *radio1*).
- L'associer au bouton radio approprié de votre interface en utilisant la méthode *findViewById*.
- Pour tester l'état de votre bouton radio, appeler la méthode *isChecked()*. Par exemple :

```
if (radio1.isChecked() ){  
    //traitement  
}
```

1.2 Gérer les cases à cocher

Tout comme les boutons radio, les cases à cocher ont deux états : coché ou décoché. Cependant, on peut avoir plusieurs cases qui sont cochées en même temps, et elle sont la plupart du temps indépendantes. Pour gérer l'état d'une case à cocher, il faut suivre les étapes suivantes :

- Créer un attribut de type *CheckBox* dans votre activité (par exemple *check1*).
- L'associer à la case à cocher appropriée de votre interface en utilisant la méthode *findViewById*.
- Pour tester l'état de votre case à cocher, appeler la méthode *isChecked()*. Par exemple :

```
if (check1.isChecked() ){  
    //traitement  
}
```
- Pour modifier l'état de la case à cocher, utiliser la méthode *setChecked(boolean etat)*. Par exemple :

```
check1.setChecked(false) ; //pour décocher la case  
check1.setChecked(true) ; //pour cocher la case
```

1.3 Menus

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

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*.

2.3.1 Menu contextuel

Pour définir un menu contextuel, qui sera déclenché suite à un long clic sur un élément (que j'appelle ici *element* :

- Créer un attribut pour l'élément auquel on veut ajouter un menu contextuel, et l'associer à l'élément graphique avec la méthode *findViewById(...)*.

- Ajouter un écouteur pour le long clic, écouteur(*Listener*), qui va ordonner l’affichage du menu contextuel quand on fait un long clic sur *element* comme suit :

```

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

```

```

element.setOnCreateContextMenuListener (this) ;

```

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

```

public boolean onContextItemSelected(MenuItem item){...}
public boolean onContextItemSelected(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
    }

```

1.3. 2. Menu d’options

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

- 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.

- Dans le corps de la méthode *onCreateOptionsMenu* (avant l'instruction *return*), ajouter les différents menus dont vous avez besoin.
- Générer la méthode *onOptionsItemSelected* pour définir le comportement au clic d'un menu.

1.3. Messages d'Alerte

Pour ajouter un message d'alerte, il suffit d'insérer le code suivant :

```
AlertDialog alertDialog;
alertDialog = new AlertDialog.Builder(this).create();
alertDialog.setTitle("titre de l'alerte");
alertDialog.setMessage("Message à afficher !!!");
alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
});
alertDialog.show();
```

Exercice 1

Créer une application android qui permet de convertir entre le dinar algérien et l'euro comme suit :

1. Créer une méthode appelée *convertir* et l'associer au bouton *Convertir* de votre interface
2. Créer deux méthodes : *dinarToEuro* et *euroToDinar*, qui prennent de convertir une valeur en entrée :

```
private float dinarsToEuro(float valeurDinar) {
    return (float) (valeurDinar * 140.45);
}
private float euroToDinar(float valeurEuro) {
    return (float) (valeurEuro * 0.0071);
}
```

3. Implémenter la méthode *convertir* pour qu'elle fasse la conversion nécessaire, selon le bouton radio qui est coché et mettre le résultat dans le champs de texte *Resultat*.

Indication : La valeur lue dans le champs de saisie (ici appelé **edt**) doit être convertie en float pour être manipulée. Pour cela, utiliser le code suivant :

```
EditText edt = (EditText) findViewById(R.id.edit_float);
float number = Float.valueOf(edt.getText().toString());
```

D'autre part, pour extraire la chaîne de caractères associée à une variable float (appelée ici **floatVar**), utiliser le code suivant :

```
String s = String.valueOf(floatVar) ;
```

4. Créer un menu contextuel sur les deux boutons radios. Il doit contenir deux menus :
 - *Taux dinar → euro* » : affiche dans un Toast le taux de conversion du dinar vers l'euro.
 - *Taux euro → dinar* » : affiche dans un Toast le taux de conversion de l'euro vers le dinar.
5. Créer un menu d'options qui contient deux menus :
 - *Conversion Dinar ↔ Dollar* : qu'on laissera vide.

- *Quitter* : permet de quitter l'application.

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

5. Créer un menu d'options qui contient deux menus : Créer une alerte appelée si l'utilisateur clique sur le bouton *conversion* alors que le champ en entrée est vide.

