Université Ferhat Abbas- Sétif 1
Année Universitaire : 2020/2021
Faculté des Sciences
Département d'informatique (3ème LMD)
Module : Applications Mobiles

TP1 *Création d'un Projet Android* (*Un premier projet : Hello World*):

Après l'installation du JDK et de l'Android Studio, vous procédez comme suit :

1. Lancez Android Studio.

2. Aller sous : File-> New project. Dans la boite de dialogue qui s'affiche, vous renseignez les détails à propos de votre projet comme suit:

Application Name: PremièreApplication, **Company Domain:** com.android.projet et vous cliquez sur **Next**. Sélectionnez Phone and Tablet et votre SDK version (6.0) et vous cliquez sur **Next.** Sélectionnez **Blank activity** et cliquez sur **Next**. Dans la boite de dialogue qui s'affiche vous donnez le **Activity Name:** PremièreActivité et le **Layout Name**: premier-layout et vous cliquez sur **Finish.**

3. Maintenant vous créez votre émulateur: Vous cliquez sur l'icone **AVD (A**ndroid **V**irtual **D**evice**)** de la boite d'outils et suivez toutes les étapes en personnalisant votre choix jusqu'à l'apparition de votre émulateur où s'est affiché le nom de votre application et le message "**Hello world**!".

4.Vous ouvrez maintenant le fichier.xml associé à votre application et vous voyez la partie de code (**TextView**) associée à ce message.

5. Vous allez maintenant supprimer ce message de l'émulateur et le remplacez par le message "**Mes salutations vont apparaitre ici**" et vous lui associez un identifiant en ajoutant cette ligne de code: **android:id="@+id/salutations_text_view"**

6.Vous allez maintenant ajouter un bouton à votre interface en ajoutant ce code:

```
<Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Montrez mes Salutations"
         />
```

7. Allez voir ce bouton sur votre émulateur avec le message associé. Pour le faire apparaitre en bas de votre message **"Mes salutations vont apparaitre ici"** identifié par: **"@+id/salutations_text_view" ,** ajouter à son code cette ligne :

android: layout_below="@+id/salutations_text_view"

En vérifiant votre émulateur, vous allez trouver le bouton aligné en dessous de votre message.

8. Modifiez votre code source de sorte que quand vous cliquez sur ce bouton, le message **"Bienvenue à ma première application"** apparaitra à la place du message **"Mes salutations vont apparaitre ici".** Pour cela associez une méthode **onClick** à ce bouton qui s'appelle **'montrerSalutations'** au niveau du fichier xml et la définir au niveau du fichier Java comme suit :

Fichier xml : android:onClick="montrerSalutations"

Vous allez maintenant définir cette méthode sur le fichier Java associé à ce premier_layout comme suit :

```
Public class PremièreActivité extends Activity {
        TextView   textView;


        @Override
         Protected void  onCreate (Bundle savedInstanceState)   {
                Super.onCreate(savedInstanceState);
                setContentView(R.layout.premier_layout);
                textView = (TextView) findViewByID(R.id.salutations_text_view);
}


Public void montrerSalutations (View view)
{
    String message= "Bienvenue à ma première application";
    textView.setText (message) ;
}
```

En exécutant ce programme (cliquez sur la flèche verte de la barre d'outils), vous allez voir le bouton sur votre émulateur et en cliquant la dessus ; le message **"Bienvenue à ma première application"** va apparaitre à la place du message **"Mes salutations vont apparaitre ici".**

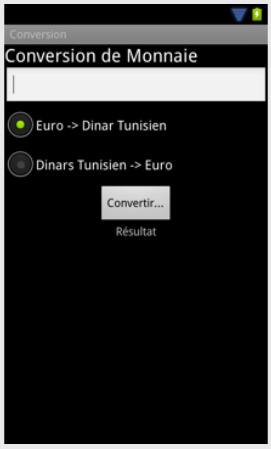
TP3 : Composants Android

Objectifs du TP

Ce TP a pour objectif de vous initier aux différents composants importants d'Android.

Nous allons dans ce TP réaliser progressivement une petite application de conversion.

TAF-1 **:** Créer un nouveau projet intitulé *Conversion*, qui contient le package *isi.conversion*. Créer les éléments nécessaires pour que l'interface soit comme la figure suivante :



I. Boutons, boutons radios et cases à cocher

I. 1. Associer un comportement à un bouton

1. Créer une méthode dans le code Java de l'activité qui définit le comportement du bouton. On l'appellera

```
public void auClicMethode(View v){…}
```

   Cette méthode doit obligatoirement être publique, retourner *void* et avoir un paramètre de type *android.view.View*.

2. Créer un bouton dans le fichier *layout* en utilisant la partie graphique.
3. Modifier son identifiant et son texte à votre guise

Remarque : Il vous est possible de modifier ces informations graphiquement. Pour cela, ouvrir la partie graphique du fichier *layout*, clic-droit sur l'élément à configurer, et choisir *Edit Text*, *Edit ID*…

4. Dans le code xml de votre bouton, ajouter l'attribut :

```
android:onClick = "@string/auClic"
```

5. Créer dans le fichier *strings.xml* un nouveau string dont le nom est *auClic* et la valeur est *auClicMethode* (qui est le nom de la méthode que vous avez créé dans 1.)

TAF-2 **:** Créer une méthode appelée *convertir* et l'associer au bouton *Convertir* de votre interface.

I. 2. Gérer les boutons radios

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 :

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

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

***TAF-3* :**

1. Créer deux méthodes : dinarToEuro et euroToDinar, qui prennent de convertir une valeur en entrée :

```
private float dinarsToEuro(float valeurDinar) {
        return (float) (valeurDinar * 1.9919);
}

private float euroToDinar(float valeurEuro) {
        return (float) (valeurEuro * 0.5020);
}
```

2. Implémenter la méthode *convertir* pour qu'elle fasse la conversion nécessaire, selon le bouton radio qui est coché. 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) ;
```

I. 3. 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 :

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

```
if (check1.isChecked() ){
          //traitement
}
```

4. 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
```

II. (TP5+6)

- Créer une nouvelle activité dans le même projet, qui s'appelle *ConversionTemperature*. Cette activité présente une interface similaire à l'activité précédente, et permet de convertir entre le Celcius et le Farenheit.

Indication :

Tc = (5/9)*(Tf-32)

Tf = (9/5)*Tc+32; avec Tc=température en Celcius et Tf=température en Farenheit

- Implémenter le menu d'option Conversion C <-> F de la première activité pour qu'il ouvre la deuxième
- Créer un menu d'options dans la deuxième activité qui permet de :
    - revenir à la conversion euro <-> dinar
    - quitter

TP 07 : Persistance des Données
--- Bases de données SQLite ---

Le but de ce TP est de créer une application permettant de créer une base de données embarquée dans le smartphone(Student.db) et de la manipuler à travers l'interface représentée dans la Figure1.

Student.db

ID	Name	Surname	Mark
1	Mark	Taylor	18
2	Tom	Smith	17
3	John	Mal	15
4	Max	Nickson	05

La base de données contient des étudiants. Un étudiant est défini par son id (un nombre qui s'incrémente automatiquement) et a un Prénom(Name), un Nom(Surname) et une Note(Mark). L'interface permet de rajouter un étudiant dans la base, le supprimer, visualiser ou modifier.

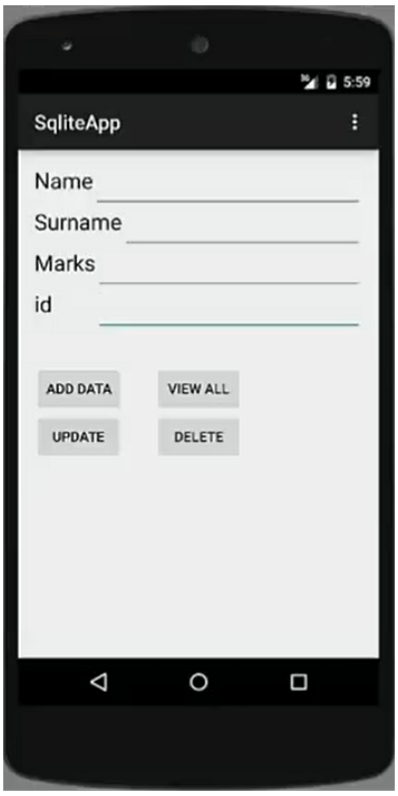


Figure 1

TD 07 : Persistance des Données
--- Bases de données SQLite ---

Il s'agit en séance de TD d'écrire les parties de codes nécessaires permettant de réaliser cette application.