Université d'Oran 1
Faculté des Sciences Exactes et Appliquées
Département d'Informatique

3ème année Licence – Semestre 2 **Année : 2019/2020**
Module : Applications Mobiles

Fiche TD/TP n°3 : Les éléments graphiques

Objectif : Le but de ce TD/TP est de se familiariser avec les éléments graphiques fournis par les librairies Android.

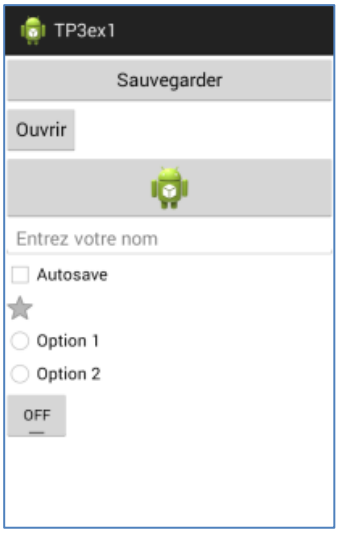
Prérequis : Cours 6 : Création d'interfaces simples

Remarque : Vous avez une semaine (**jusqu'à Samedi prochain 11/04**) pour résoudre la fiche et transmettre la solution à votre enseignant TD/TP (en envoyant vos projets TP en un seul fichier compressé et aussi des captures d'écran de l'exécution, une capture d'écran pour chaque exercice).

Exercice 1 :

• Créez un nouveau projet.

• Créez un nouveau fichier XML dans **res/layout/**, nommez-le **linearlayout.xml** et modifiez le fichier XML pour avoir le résultat suivant (Utilisez un linear layout):



Sous Eclipse

Sous Android Studio

• Les éléments graphiques contenus dans la figure sont respectivement : **Button**, **Button**, **ImageButton**, **EdiText**, **CheckBox**, **CheckBox**, **RadioGroup** (avec **deux RadioButton**) et enfin un **ToggleButton**.

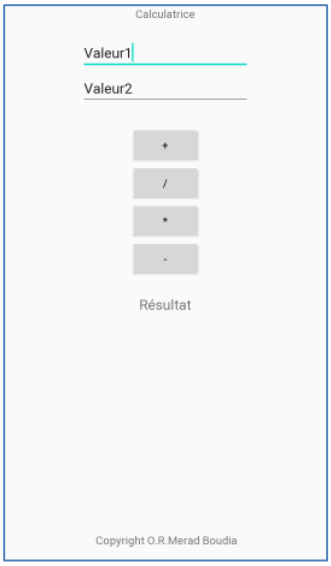
• Corrigez les erreurs. Pour avoir un élément **CheckBox** sous forme d'une étoile, on utilise le style suivant :

```
style="?android:attr/starStyle"
```

• Modifier l'interface à afficher dans **MainAtivity.java**.

• Exécutez l'application.

Exercice 2 :



- Créez un nouveau projet qui permet de réaliser une calculatrice simplifiée.
- Utilisez un **RelativeLayout** et ajoutez des **EditText** pour entrer les opérandes.
- Ajoutez des « **Button** » pour les opérations.
- Ajoutez des **TextView** pour le titre, le résultat et le copyright (votre nom).
- Pour la gestion du clic, voir le cours 6 – Création d'interfaces simples.

Exercice 3 :

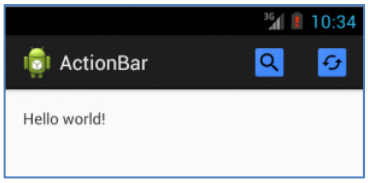
- Créez un nouveau projet qui permet d'afficher un **TimePicker**. Ce dernier permet d'afficher un composant sélecteur de temps.
- En utilisant un élément **TimePicker**, créez le fichier xml qui correspond à la figure.
- Gerez le clic sur le bouton afin d'afficher un toast affichant l'heure sélectionnée sur le **TimePicker**. Pour cela, utilisez les méthodes **getCurrentHour()** et **getCurrentMinute()** de la classe prédéfinie **TimePicker**.

Exercice 4 :

- Créez un nouveau projet et en se basant sur le cours 6, créez le menu (présenté sur la figure) affichant les noms des versions Android. Le clic sur un élément du menu affiche le numéro de la version dans un Toast.

Exercice 5:

Action Bar est un élément qui se place en haut de l'écran de l'app. Ce composant remplace les menus disponibles dans les anciennes versions d'Android (Avant 3.0) et permet à l'utilisateur d'accéder facilement aux fonctionnalités les plus importantes d'une application.



• Reprenez la classe Principale.

• Créez un fichier **main.xml** dans le dossier menu :

```
<item
    android:id="@+id/action_search"
    android:icon="@drawable/rechercher"
    android:orderInCategory="100"
    android:showAsAction="ifRoom"
    android:title="@string/action_settings"/>

<item
    android:id="@+id/action_refresh"
    android:icon="@drawable/actualiser"
    android:orderInCategory="100"
    android:showAsAction="ifRoom"
    android:title="@string/action_settings"/>
```

• Créez les ressources utilisées dans **strings.xml** et dans **drawable**, vérifiez le nom du package.

• Ajoutez les actions.

• Ajoutez dans la classe **MainActivity**, l'initialisation du **OptionsMenu** avec le fichier **main.xml**.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}
```

• Ajoutez les méthodes d'interaction sur le menu.

• Faites un Toast pour chaque action.

```
public boolean onOptionsItemSelected(MenuItem item){

    switch (item.getItemId()){
    case R.id.action_search:
        Toast.makeText(this, "Recherhce", Toast.LENGTH_SHORT).show();
        return true;
    case R.id.action_refresh:
        Toast.makeText(this, "Actualiser", Toast.LENGTH_SHORT).show();
        return true;
    default: return super.onOptionsItemSelected(item);
    }
    }
```