

## Série de TP n°1

### Exercice 1 :

Ouvrir l'IDE android studio. Puis Créez-y un nouveau projet. Vous devez avoir un code source semblable à celui-ci.

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloWorldActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
    }
}
```

1. Créer un AVD. Puis exécuter le code source.
2. Remplacer l'instruction `setContentView(R.layout.activity_main);` par le code suivant :

```
TextView tv = new TextView(this);
tv.setText("Hello, Android");
setContentView(tv);
```

### Exercice 2 :

1. Dans votre projet, créer une deuxième activité (empty activity) que vous nommez Main2Activity. Le fichier de mise en page résultant `activity_main2` doit ressembler au code suivant :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
</RelativeLayout>
```

2. Ajoutez un EditText et un bouton à votre mise en page.
3. Utilisez l'éditeur XML pour modifier l'ID du nouveau champ EditText à `main_input`. Dans le fichier XML, cela ressemble `@+id/main_input`.
4. Changez la couleur de fond de votre mise en page pour la couleur mauve (`@android:color/holo_purple`). Utilisez la propriété `android:background` pour cela.

5. Modifiez le texte du bouton pour « Start » avec la propriété android:text.
6. Assignez la fonction onClick à la propriété android:onClick de votre bouton. Ceci définit que la fonction « public void onClick (View view) » sera appelée dans l'activité quand le bouton est pressé.
7. Créez la méthode suivante dans votre classe MainActivity :

```
public void onClick (View view) {
    Toast.makeText(this, "Button 1 pressed",
        Toast.LENGTH_LONG).show();
}
```

8. Démarrez votre application et appuyez sur le bouton « Start ». Vérifiez qu'un message (Toast) est bien affiché.
9. Retournez dans votre code source et utilisez la fonction findViewById(id) avec l'identifiant correct en castant la valeur retournée en EditText. Par exemple, EditText text = (EditText) findViewById(id). Vous pouvez obtenir la valeur correcte de l'identifiant en utilisant la classe « R ». Il doit se trouver parmi les « ID » et s'appeler main\_input. Utilisez la méthode text.getText().toString() pour lire la valeur du champ EditText et ajouter ce texte au message affiché (Toast).

### Exercice 3 : Modifier la mise en page dynamiquement

Dans cet exercice, vous ajouterez des boutons radio de votre mise en page. Selon le choix de l'utilisateur, l'arrangement des boutons radio sera horizontal ou vertical.

1. Ouvrez votre fichier de mise en page (exercice 2) et ajoutez un groupe de boutons radio (RadioGroup) contenant deux boutons radio.

Le résultat de votre fichier de mise en page devrait être similaire à ceci :

```
<!-- this snippet is part of the larger layout file -->
<RadioGroup
    android:id="@+id/orientation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <RadioButton
        android:id="@+id/horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Horizontal" >
    </RadioButton>
    <RadioButton
        android:id="@+id/vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:checked="true"
        android:text="Vertical" >
    </RadioButton>
</RadioGroup>
```

2. Modifiez la fonction onCreate() de votre activité. Utilisez la méthode findViewById() pour retrouver votre RadioGroup.
3. Implémentez un listener sur ce groupe de boutons radio qui modifie son orientation en se basant sur la sélection courante du bouton. L'objet RadioGroup vous permet d'ajouter un RadioGroup.OnCheckedChangeListener (issu de android.widget.RadioGroup) avec la méthode setOnCheckedChangeListener(). Cet écouteur (Listener) est appelé lors de tout changement de sélection du groupe de boutons radio.

Vous pouvez utiliser le code suivant comme modèle pour créer le listener :

```
RadioGroup group1 = (RadioGroup) findViewById(R.id.orientation);
group1.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
```

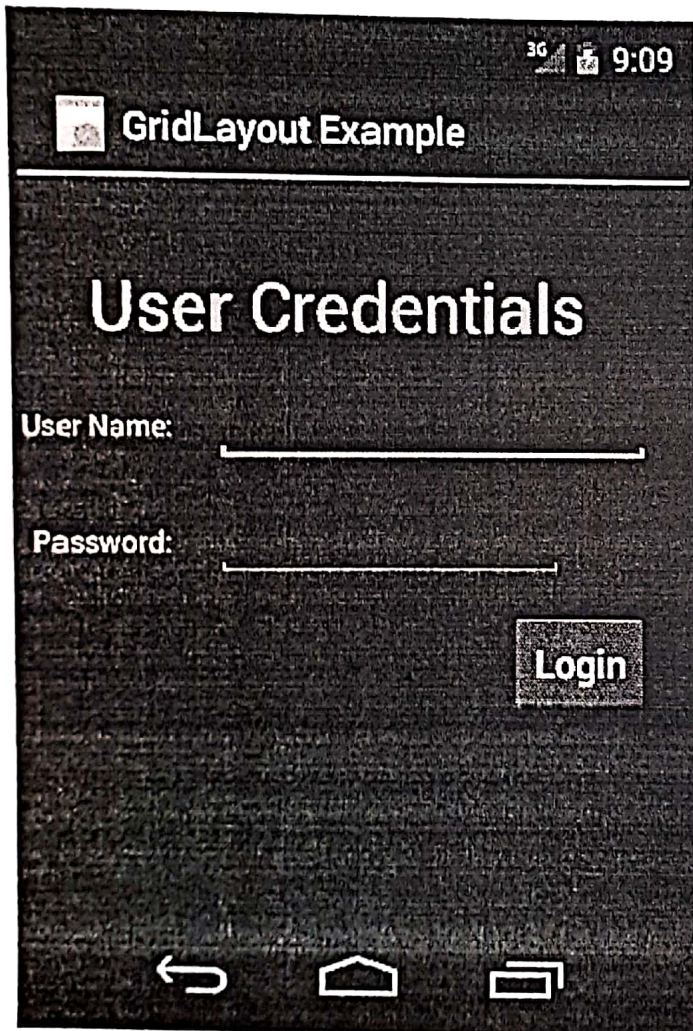
```

public void onCheckedChanged(RadioGroup group, int checkedId) {
    switch (checkedId) {
        case R.id.horizontal:
            group.setOrientation(LinearLayout.HORIZONTAL);
            break;
        case R.id.vertical:
            group.setOrientation(LinearLayout.VERTICAL);
            break;
    }
}
});

```

**Exercice 4:**

Créer une interface avec le gestionnaire de vue GridLayout similaire à la page d'écran ci-dessous.

**Exercice 2 : solution**

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"

```

```

android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >
<EditText
android:id="@+id/main_input"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginLeft="14dp"
android:ems="10" >
<requestFocus />
</EditText>
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText1"
android:layout_below="@+id/editText1"
android:layout_marginTop="31dp"
android:onClick="onClick"
android:text="Start" />
</RelativeLayout>

```

Et le code de votre activité devrait être semblable à ceci :

```

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;

import android.view.View;

import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (BuildConfig.DEBUG) {
            Log.d(Constants.LOG, "onCreated called");
        }
        setContentView(R.layout.activity_main);
    }
    // you may have here an onCreateOptionsMenu method
    // this method is not required for this exercise
    // therefore I deleted it
    public void onClick(View view) {
        EditText input = (EditText) findViewById(R.id.main_input);
        String string = input.getText().toString();
        Toast.makeText(this, string, Toast.LENGTH_LONG).show();
    }
}

```

#### Exercise 4: solution

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/GridLayout1"
android:layout_width="match_parent"

```

```

android:layout_height="match_parent"
- android:columnCount="4"
  android:useDefaultMargins="true" >
  <TextView
- android:layout_column="0"
  android:layout_columnSpan="3"
  android:layout_gravity="center_horizontal"
  android:layout_marginTop="40dp"
- android:layout_row="0"
  android:text="User Credentials"
  android:textSize="32dip" />
  <TextView
- android:layout_column="0"
  android:layout_gravity="right"
- android:layout_row="1"
  android:text="User Name: " >
  </TextView>
  <EditText
  android:id="@+id/input1"
- android:layout_column="1"
  android:layout_columnSpan="2"
- android:layout_row="1"
  android:ems="10" />
  <TextView
- android:layout_column="0"
  android:layout_gravity="right"
- android:layout_row="2"
  android:text="Password: " >
  </TextView>
  <EditText
  android:id="@+id/input2"
- android:layout_column="1"
  android:layout_columnSpan="2"
- android:layout_row="2"
  android:inputType="textPassword"
  android:ems="8" />
  <Button
  android:id="@+id/button1"
- android:layout_column="2"
- android:layout_row="3"
  android:text="Login" />
</GridLayout>

```