Contents

1 Exercice 1 - Les différentes parties d'une application Android

```
<!--../source/xml/exo1/manifest.xml-->

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">

        <activity
            android:name=".HelloAndroid"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

// ../source/java/exo1/HelloAndroid.java
```

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

public class HelloAndroid extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        LinearLayout ll = new LinearLayout(this);
        ll.setOrientation(LinearLayout.VERTICAL);
        ll.setLayoutParams(new LinearLayout.LayoutParams(
                ViewGroup.LayoutParams.MATCH_PARENT,
                ViewGroup.LayoutParams.MATCH_PARENT));
        TextView tv = new TextView(this);
        tv.setText("Hello World!");
        EditText ev = new EditText(this);
        ev.setHint("Enter text");
        ll.addView(tv);
        ll.addView(ev);
        setContentView(ll);
    }
}
```

Ces premiers bouts de code montrent le manifest d'une application Android en XML et le code Java de la logique métier de son activité principale `HelloAndroid.java`. L'interface graphique de l'activité est créée puis chargée lors de la création de l'activité.

```xml
<!--../source/xml/exo1/layout_file_name.xml-->

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:hint="Enter text" />
</LinearLayout>
```

```
// ../source/java/exo1/HelloAndroidWithLayout.java

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

public class HelloAndroid extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_file_name);
    }
}
```

Dans la deuxième version de l'activité, l'interface graphique est déclarée dans une ressource layout séparée en XML et est chargée ensuite dans l'activité lors de la création de cette dernière.

2 Exercice 2 - Les ressources

```
<!--../source/xml/exo2/string.xml-->

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Android Hello World</string>
    <string name="menu_settings">Settings</string>
    <string name="button_label">Show Message</string>
    <string name="image_content">image</string>
    <string name="text">Ma première application Android</string>
    <string name="hint">Écrire quelque chose</string>
</resources>
```

```
<!--../source/xml/exo2/layout_file_name.xml-->

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
    <Button
        android:id="@+id/mainbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_label"/>

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/text"/>

    <EditText
        android:id="@+id/edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="@string/edit"/>
</LinearLayout>
```

```
// ../source/java/exo2/HelloAndroid.java

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

public class HelloAndroid extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.layout_file_name);
    }
}
```

Dans cette version de l'activité, les strings utilisés pour définir le contenu textuelle des widgets sont externalisés dans le fichier de ressources simples `res/string.xml`. Ceci permettra une réutilisation des valeurs dans plusieurs vues si nécessaires. De plus, si le même string est écrit en dur en plusieurs endroits du code, il va falloir modifier chacune de ces occurrences individuellement lorsqu'on en aura besoin. En revanche, avec l'externalisation de cette valeur en une ressource string, cette modification est réalisée une seule fois au niveau de la resource elle-même et dont l'effet sera répércuté partout dans le code où elle est référencée.

3 Exercice 3 - Les gabarits (*layouts*)

3.1 Layout 1

```
<!--../source/xml/exo3/linear_layout.xml-->

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 1" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 2" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button 3"
        android:layout_weight="1"/>
</LinearLayout>
```

3.2 Layout 2

```
<!--../source/xml/exo3/relative_layout.xml-->

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
```

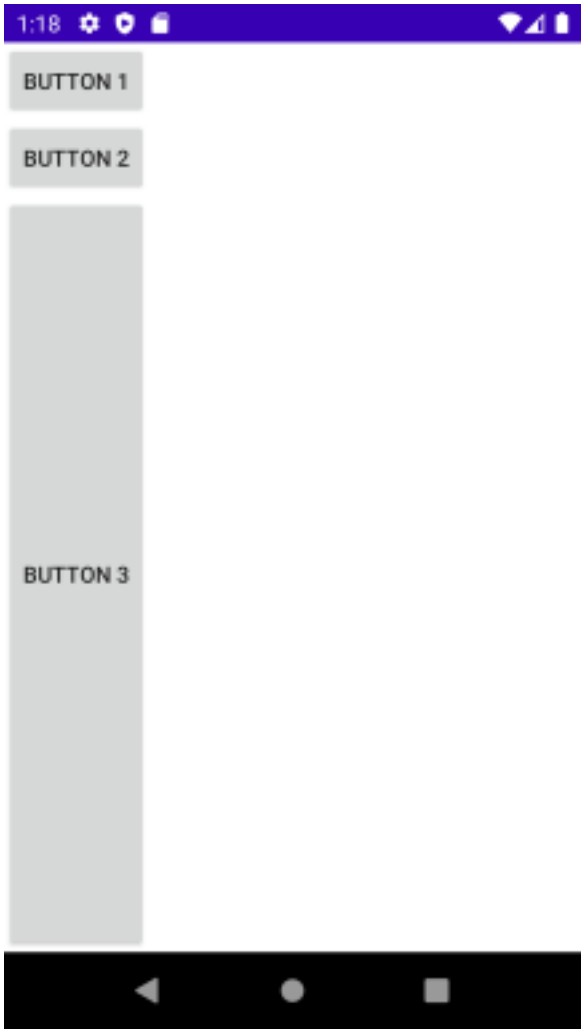

Figure 1: Linear Layout

```
        android:layout_height="wrap_content"
        android:text="Button 1" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button3"
        android:layout_marginTop="65dp"
        android:text="Username:"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@id/textView"
        android:layout_toRightOf="@+id/button2"
        android:inputType="text"
        android:hint="Enter a text"/>

    <Button
        android:id="@+id/button0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/editText"
        android:text="Submit:" />

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/button1"
        android:text="Button 3" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/button1"
        android:layout_toRightOf="@id/button1"
        android:text="Button 2" />
</RelativeLayout>
```

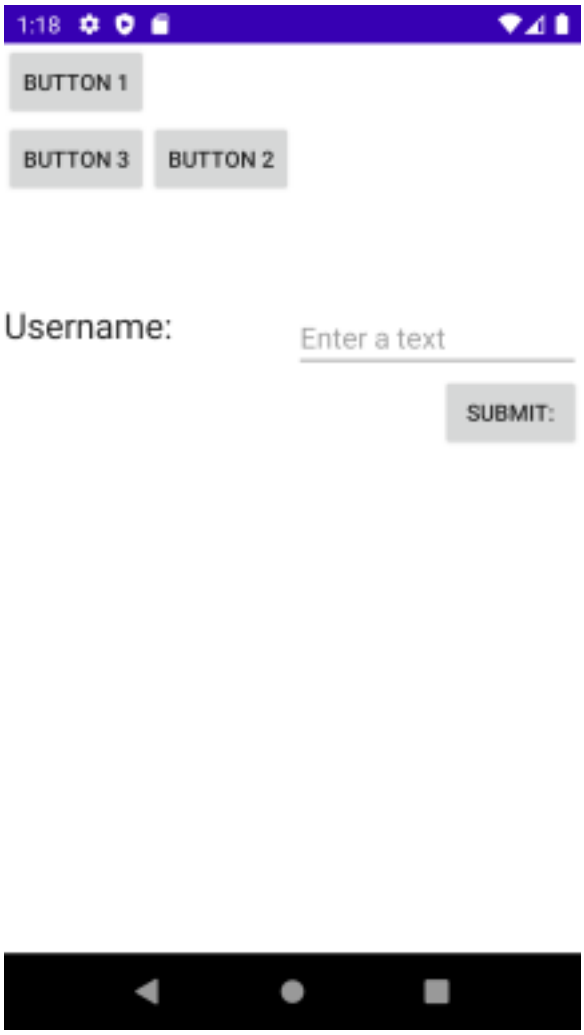

Figure 2: Relative Layout

3.3 Layout 3

```
<!--../source/xml/exo3/table_layout.xml-->

<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:shrinkColumns="*"
```

```
android:stretchColumns="*">

<!--2 columns-->
<TableRow
    android:id="@+id/tableRow1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dip">

    <TextView
        android:id="@+id/textView1"
        android:text="Col 1"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/button1"
        android:text="Col 2" />
</TableRow>

<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dip">

    <EditText
        android:id="@+id/editText1"
        android:layout_span="2"
        android:text="Col1 &amp;amp; 2" />
</TableRow>

<!--red line-->
<View
    android:layout_height="4dip"
    android:background="#FF00" />

<!--4 columns-->
<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="5dip">

    <TextView
        android:id="@+id/textView2"
        android:text="Col 1" />
```

```
        <Button
            android:id="@+id/button2"
            android:text="Col 2" />

        <Button
            android:id="@+id/button3"
            android:text="Col 3" />

        <Button
            android:id="@+id/button5"
            android:text="Col 4" />
    </TableRow>

    <TableRow
        android:id="@+id/tableRow4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip">

        <Button
            android:id="@+id/button4"
            android:layout_column="2"
            android:text="Col 3" />
    </TableRow>

    <TableRow
        android:id="@+id/tableRow5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dip">

        <Button
            android:id="@+id/button6"
            android:layout_column="1"
            android:text="Col 2" />
    </TableRow>
</TableLayout>
```

4 Exercice 4 - Évènements et utilisateurs

Dans cet exercice on dispose d'une application Android constituée d'une seule activité : son activité principale. L'interface graphique de ladite activité consiste d'un gabarit linéaire contenant un bouton. Lorsqu'on clique le bouton, un Toast

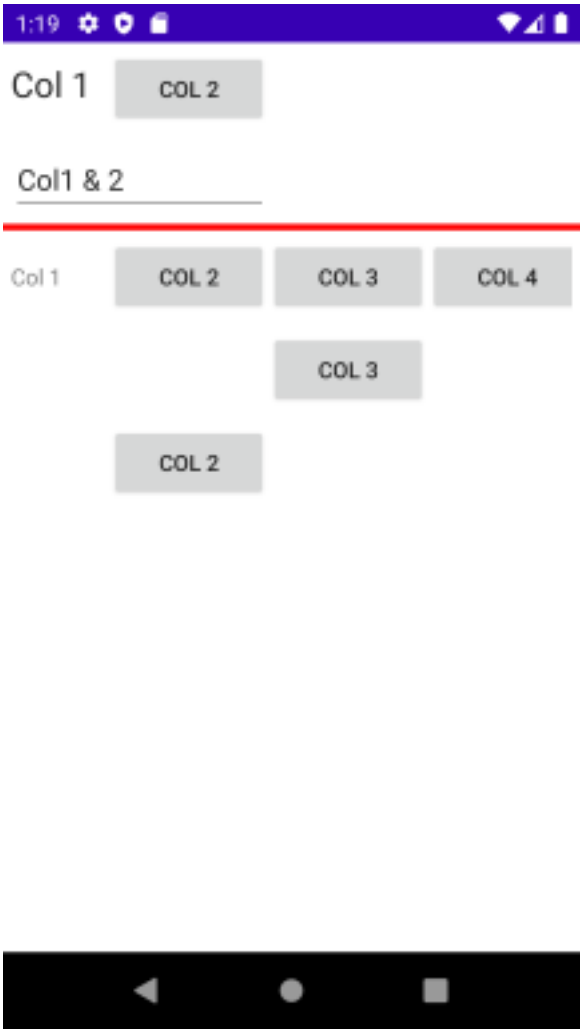

Figure 3: Table Layout

avec le message “Message Bouton 1” est affiché pendant une relativement longue durée.

On ajoute un deuxième bouton qui suite à un clic (`OnClickListener`) affichera un Textview s'il est invisible, ou le cachera s'il est visibile. Ensuite, on définit un écouteur de l'événement long clic (`OnLongClickListener`) pour le premier bouton qui affichera un Toast indiquant qu'il s'agit d'un long clic. Enfin pour assurer la portabilité de l'application à plusieurs langues, on externalise les messages et valeurs affichées dans deux fichiers de resources String : (1) l'anglais (*par défaut*), et (2) le français, si l'utilisateur utilise un système en français.

```
<!--../source/xml/exo4/main.xml-->

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/mainbutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/mainbutton" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button2" />

    <TextView
        android:id="@+id/hiddenTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="invisible"
        android:text="Exercice 4"/>
</LinearLayout>

<!--../source/xml/exo4/strings.xml-->

<resources>
    <string name="app_name">My Application</string>
    <string name="mainbutton">Toast me</string>
    <string name="button2">Toast me as well</string>
</resources>

<!--../source/xml/exo4/strings_fr.xml-->
```

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Mon Application</string>
    <string name="mainbutton">Toast moi</string>
    <string name="button2">Toast moi aussi</string>
</resources>
```

```
// ../source/java/exo4/MainActivity.java

package com.example.myapplication;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {

    private Button button;
    private Button button2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        button = (Button) findViewById(R.id.mainbutton);
        button2 = (Button) findViewById(R.id.button2);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(MainActivity.this,
                        "Message Bouton 1",
                        Toast.LENGTH_LONG).show();
            }
        });

        button.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View view) {
                Toast.makeText(MainActivity.this,
                        "Long Click",
                        Toast.LENGTH_LONG).show();
                return false;
```

```
            }
        });

        button2.setOnClickListener(new View.OnClickListener() {
            TextView text = (TextView) findViewById(R.id.hiddenTextView);
            @Override
            public void onClick(View view) {
                String message;
                if (!text.isShown()) {
                    text.setVisibility(View.VISIBLE);
                    message = "Showing Hidden TextView";
                }
                else {
                    text.setVisibility(View.INVISIBLE);
                    message = "Hiding Shown TextView";
                }
                Toast.makeText(MainActivity.this,
                        message,
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

5 Exercice 5 - Différentes vues et actions

Dans cet exercice on dispose d'une application Android constituée d'une seule activité : son activité principale. L'interface graphique de ladite activité consiste d'un gabarit linéaire contenant trois cases à cocher et un bouton. Lorsqu'on coche la case Windows, un Toast avec le message "Bro try Linux :)" est affiché pendant une relativement longue durée. De plus, quand on clique le bouton, un Toast avec les états des cases à cocher est affiché pendant une relativement longue durée.

```
<!--../source/xml/exo5/main.xml-->

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <CheckBox
        android:id="@+id/linux_option"
```

Figure 4: L'activité princiaple et son interface graphique

Figure 5: Le Toast “Message Bouton 1” est affiché quand le premier bouton est cliqué

Figure 6: Le Toast “Long Click” est affiché quand le premier bouton est cliqué pendant longtemps

Figure 7: Le TextView apparaît s'il est caché, avec le Toast "Showing Hidden TextView", quand le deuxième bouton est cliqué

Figure 8: Le TextView disparaît s'il est visibile, avec le Toast "Hiding Shown TextView", quand le deuxième bouton est cliqué

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/linux_option" />

    <CheckBox
        android:id="@+id/macos_option"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/macos_option" />

    <CheckBox
        android:id="@+id/windows_option"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/windows_option" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/display_label" />
</LinearLayout>
```

```
<!--../source/xml/exo5/strings.xml-->

<resources>
    <string name="app_name">My Application</string>
    <string name="linux_option">Linux</string>
    <string name="macos_option">Mac OS</string>
    <string name="windows_option">Windows</string>
    <string name="display_label">Display choices</string>
</resources>
```

```
// ../source/java/exo5/MainActivity.java

package com.example.myapplication;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends Activity {

    private CheckBox linux;
```

```
    private CheckBox macos;
    private CheckBox windows;
    private Button button;

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

    public void addListenerOnChkWindows(){
        windows = (CheckBox) findViewById(R.id.windows_option);
        windows.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (((CheckBox)view).isChecked()){
                    Toast.makeText(MainActivity.this,
                            "Bro, try Linux :)",
                            Toast.LENGTH_LONG).show();
                }
            }
        });
    }

    public void addListenerOnButton(){
        linux = (CheckBox) findViewById(R.id.linux_option);
        macos = (CheckBox) findViewById(R.id.macos_option);
        windows = (CheckBox) findViewById(R.id.windows_option);
        button = (Button) findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                StringBuffer result = new StringBuffer();
                result.append("Linux check: ").append(linux.isChecked());
                result.append("\nMac OS check: ").append(macos.isChecked());
                result.append("\nWindows check: ").append(windows.isChecked());
                Toast.makeText(MainActivity.this,
                        result.toString(),
                        Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Figure 9: L'activité princiaple et son interface graphique

6 Exercice 6 - Méta-modèle des applications Android

@TODO

Figure 10: Le Toast “Bro try Linux :)” est affiché quand la case Windows est cochée

Figure 11: Le Toast avec les états des cases à cocher est affiché quand le bouton est cliqué