
IHM (basiques) et Android

UE élective TAF

Thierry Duval, Mayte Segarra
Département Informatique

Hello World – récapitulatif

- Dérivation d'une classe Activity ou (AppCompatActivity)
- Description au travers de nombreux fichiers :
 - ✓ MainActivity.java
 - ✓ activity_main.xml
 - ✓ strings.xml
 - ✓ styles.xml
 - ✓ AndroidManifest.xml
 - ✓ ...
- Générés par Android Studio, à compléter...

MainActivity.java

```
package com.example.helloworld;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate (savedInstanceState) ;
        setContentView (R.layout.activity_main) ;
    }
}
```

Hello World

- setContentView (R.layout.activity_main) ;
 - ✓ Chargement du contenu du fichier comme vue de l'activité
- R.layout.activity_main
 - ✓ Accès à la ressource définie dans le paquetage
- R.layout
 - ✓ Fichiers de vue compilés

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
    ...  
</androidx.constraintlayout.widget.ConstraintLayout>
```

strings.xml

<resources>

<string name="app_name">HelloWorld</string>

</resources>

styles.xml

```
<resources>
```

```
    <!-- Base application theme. -->
```

```
    <style name="AppTheme"
```

```
        parent="Theme.AppCompat.Light.DarkActionBar">
```

```
        <!-- Customize your theme here. -->
```

```
        ...
```

```
    </style>
```

```
</resources>
```

AndroidManifest.xml

...

```
<application android:allowBackup="true"
  android:icon="@mipmap/ic_launcher"
  android:label="@string/app_name"
  android:roundIcon="@mipmap/ic_launcher_round"
  android:supportRtl="true" android:theme="@style/AppTheme" >
  <activity android:name=".MainActivity">
    <intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
  </activity>
</application>
```

...

Résumé...

app

manifests

AndroidManifest.xml

java

com.example.helloworld

MainActivity.java

res

mipmap

ic_launcher.png

ic_launcher_round.png

layout

activity_main.xml

values

strings.xml

styles.xml

Ajouter des composants graphiques simples

Ajout de composants dans l'interface de l'activité

- Directement en XML dans le fichier de type *layout*
 - ✓ Connaissance imposée de la syntaxe XML
- Via l'éditeur de vues (mode « *Design* » de l'édition du fichier XML)
 - ✓ Génération automatique de code dans le fichier XML
- Directement en Java
 - ✓ Connaissance imposée de l'API Java pour l'ajout/suppression des composants d'IHM

```
LinearLayout ll = (LinearLayout)findViewById (R.id.linearLayout) ;  
TextView tvPhone = new TextView (this) ;  
tvPhone.setText ("phone") ;  
ll.addView (tvPhone) ;
```

Hello World la suite : nouvelle application « Hello Universe »

- Déclaration d'un *widget* (composant) de visualisation de texte dans le fichier XML associé (activity_main.xml)
- Ajout du texte dans le fichier strings.xml
- Chargement du contenu du fichier comme vue de l'activité
 - ✓ setContentView (R.layout.activity_main) ;

strings.xml

<resources>

<string name="app_name">HelloWorld</string>

<string name="hello_text">Hello Universe!!!</string>

</resources>

activity_main.xml

...

```
<TextView android:text="@string/hello_text"  
    android:id="@+id/hello_elm"  
    ... />
```

...

Déclaration du contenu des vues

- Déclaration en XML dans le fichier de type *layout*

```
<TextView  
    android:text="@string/hello_text"  
    android:id="@+id/hello_elm" ... />
```

- android:text

- ✓ Texte associé au composant
- ✓ Ici défini dans un fichier de ressources (« strings.xml »)
 - ✗ Permet l'internationalisation...

- android:id

- ✓ Identifiant unique, numérotation automatique grâce à « @+id »
- ✓ Permet l'accès au composant à partir du code Java
 - ✗ `TextView msg = (TextView)findViewById (R.id.hello_elm) ;`

Associer un comportement à un composant IHM

- Directement en XML dans le fichier de type *layout*

```
<Button android:id="@+id/button"  
...  
    android:onClick="doIt"/>
```

- Via l'éditeur de vues (mode « *Design* » de l'édition du fichier XML)

- Directement en Java

```
Button button = (Button)findViewById (R.id.button) ;  
button.setOnClickListener (new View.OnClickListener () {  
    @Override  
    public void onClick (View v) {  
        doIt (v) ;  
    }  
}) ;
```

 Dans tous les cas, il faut coder en Java l'action à exécuter

```
public void doIt (View view) { ... }
```