



**utbm**  
université de technologie  
Belfort-Montbéliard

LO52

**INFORMATIQUE MOBILE ET  
COMMUNICATIONS COURTES  
PORTEES**

Rapport TPS

**COMPTE RENDU DES TPS : TP1, TP2, TP3 et TP4**

**Rédigé par :**

- NOUASSI Zechirine
- ABBALTU Manou

**Responsable de l'unité de valeur :**

- M. BRISSET Fabien

## TABLE DE MATIERES

TABLE DE MATIERES .....	2
TABLE D'ILLUSTRATIONS .....	3
INTRODUCTION .....	4
PARTIE I : MISE EN PLACE DE L'ENVIRONNEMENT DE DEVELOPPEMENT ..	5

## TABLE D'ILLUSTRATIONS

Figure 1: Configuration fichier String.xml .....	7
Figure 2 : Méthode afficher de l'activité MainActivity .....	8
Figure 3: Propriété OnClick affectée au bouton .....	8
Figure 4 : Activité 2 .....	9
Figure 5: Activité 1 .....	9

## INTRODUCTION

Android est un système d'exploitation mobile créé par google et qui équipe la majorité des smartphones actuels. D'où son utilisation dans notre unité de valeur LO52 (Informatique mobile et communication courtes portées) que nous suivons dans le cadre de notre cursus d'ingénierie informatique à l'UTBM (Université de Technologie de Belfort Montbéliard).

Pour ce semestre d'automne 2020, les objectifs sont les suivants : la mise en place de l'environnement de développement ensuite, la définition du modèle de données pour une application Android puis, la manipulation du kernel linux, la création d'un device Android enfin, le développement de l'application et l'utilisation du JNI. Ce document retrace les différentes étapes suivies pour la réalisation de ces objectifs.

## PARTIE I : MISE EN PLACE DE L'ENVIRONNEMENT DE DEVELOPPEMENT

Le but de cette partie est de nous familiariser avec le gestionnaire de code source Git et l'environnement de développement Android Studio. Pour cela nous avons évolué sous plusieurs étapes. Il sera donc question par la suite de développer toutes ces étapes qui nous ont permis de réaliser cette première partie.

### **Etape 1 : Téléchargement, Installation et configuration de Git**

Git étant déjà installé sur nos ordinateurs, ils nous restaient qu'à configurer git. Pour ce faire nous avons modifié le fichier de configuration de Git pour y faire figurer notre email UTBM et notre nom d'utilisateur au travers des commandes ci-après :

- `git config --global user.name "<user_name>"`
- `git config --global user.email "<email>"`

### **Etape 2 : Clonage du dépôt [https://github.com/gxfab/LO52\\_A2020](https://github.com/gxfab/LO52_A2020)**

Après la configuration de Git, nous avons procédé au clonage du dépôt du TP à travers la commande :

- `git clone https://github.com/gxfab/LO52\_A2020`

Cette commande crée un répertoire du même nom que le dépôt distant, initialise un répertoire `.git` à l'intérieur et récupère toutes les données de ce dépôt.

### **Etape 3 : Création d'une branche**

Le dépôt cloné, nous avons créé une branche pour notre groupe afin d'éviter les conflits lors des push des uns et des autres. Nous avons nommé notre branche **ManouAbbaltu\_NouassiZechirine** (combinaison de noms des membres de l'équipe). Une fois créé nous nous sommes positionnés dessus à travers la commande :

- `git checkout -b ManouAbbaltu_NouassiZechirine`

Nous avons exécutés les commandes suivantes afin de mettre à jour le dépôt distant en uploadant nos changements :

- `git add .` (pour indexer tout le contenu du répertoire courant)
- `git commit -m "Initial commit"`
- `git push origin ManouAbbaltu_NouassiZechirine`

### **Etape 3 : Création de notre première application HelloWorld**

Cette première application a pour but de constituer deux vues que nous allons appeler activité. La première activité doit afficher un bouton, au clic de ce bouton nous devons naviguer vers une autre activité affichant le label *Hello World*.

Pour le faire nous avons créer un projet sur Android Studio et nous l'avons sauvegardé dans le dossier TP1 du dépôt. A l'aide de l'assistant de création de projet, nous avons pu créer un projet avec déjà une activité vide appelé *Empty Activity*. Le langage de programmation choisi était *Kotlin* car nous avons déjà une bonne base en JAVA mais pas en KOTLIN et donc profiter pour apprendre davantage sur ce langage.

A l'ouverture du projet nous avons déjà le *MainActivity* avec ces composants *MainActivity.kt* (fichier kotlin) et *activity\_main.xml* (fichier pour le layout). Dans le fichier *activity\_main.xml* nous avons ajouter un bouton et un label.

Afin de mieux gérer nos valeurs, dans le fichier de ressource *String.xml* nous avons ajouté plusieurs valeurs :

```
<resources>
    <string name="app_name">HelloWorld</string>
    <string name="app_message">Hello World</string>
    <string name="welcome_message">Bienvenue dans notre Application</string>
    <string name="button_message">Cliquer ici</string>
</resources>
```

Figure 1: Configuration fichier *String.xml*

Nous avons également appliqué des contraintes au bouton pour qu'il soit au milieu et le label pour qu'il placé comme un titre.

Ensuite nous avons créé une nouvelle activité avec ces composants bien sûr *HelloWordActivity.kt* et *activity\_hello\_word.xml* afin qu'il affiche le texte « Hello World »

Une fois les deux activités misent en place, nous avons donc créer la méthode, *afficher* dans le fichier *MainActivity.kt* et lier cette méthode au layout *activity\_main.xml* en tant que propriété « *OnClick* ». Ci-après le code qui nous a permis de réaliser ceci.

```
1 package com.lo52.tp1.helloworld
2
3 import android.content.Intent
4 import androidx.appcompat.app.AppCompatActivity
5 import android.os.Bundle
6 import android.view.View
7
8 class MainActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12     }
13
14     fun afficher(view : View){
15         val intent = Intent(this, HelloWorldActivity::class.java)
16         startActivity(intent)
17     }
18 }
```

Figure 2 : Méthode afficher de l'activité MainActivity

```
<Button
    android:id="@+id/button"
    android:layout_width="174dp"
    android:layout_height="66dp"
    android:onClick="afficher"
    android:text="@string/button_message"
    android:textSize="14sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Figure 3: Propriété onClick affectée au bouton

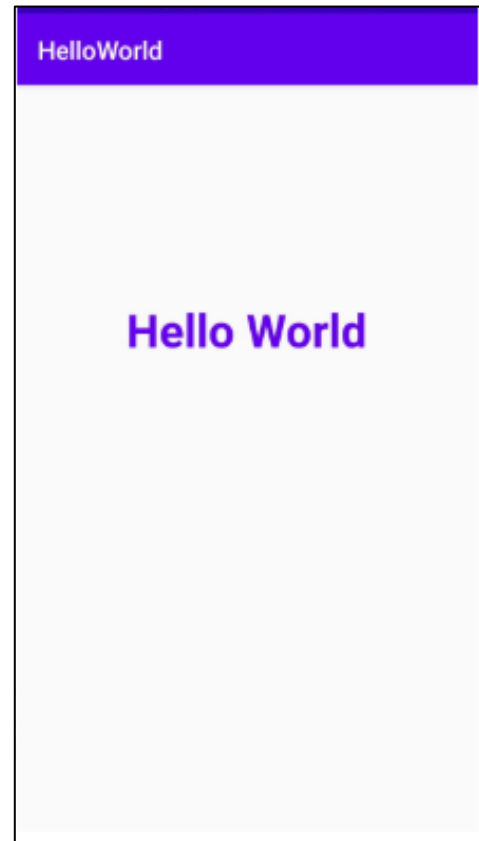
Cette méthode permet de passer de l'activité *MainActivity* à *HelloWordActivity* grâce à la méthode *startActivity* tout ceci une fois le bouton cliqué.

Nous avons pu tester l'application grâce à notre Android. Nous avons pu constater qu'au click du bouton, il affiche bien la deuxième activité.





*Figure 5: Activité 1*



*Figure 4 : Activité 2*