A dark blue vertical bar runs along the left edge of the page. A blue arrow points to the right from this bar, containing the date.

05/10/2020

# RAPPORT DU TP1

## LO52

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Ndeye Marie SALL & Mouhamed DIAGNE  
UTBM

# PLAN

Prérequis

TP réalisé

Commandes Réalisées

## I. Prérequis

Pour réaliser ce TP nous avons besoin d'installer :

- Android studio qui est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.
- Un SDK (Software Development Kit) qui permet de créer des applications sur la plateforme Android.
- La configuration du proxy :
  - ❖ HTTP
  - ❖ Hostname proxy.utbm.fr
  - ❖ Port 3128

## II. TP Réalisé

Le but de ce TP est de pouvoir charger une page qui affiche Hello World lorsque l'on clique sur le bouton de la page d'accueil.

**MainActivity.java**

```

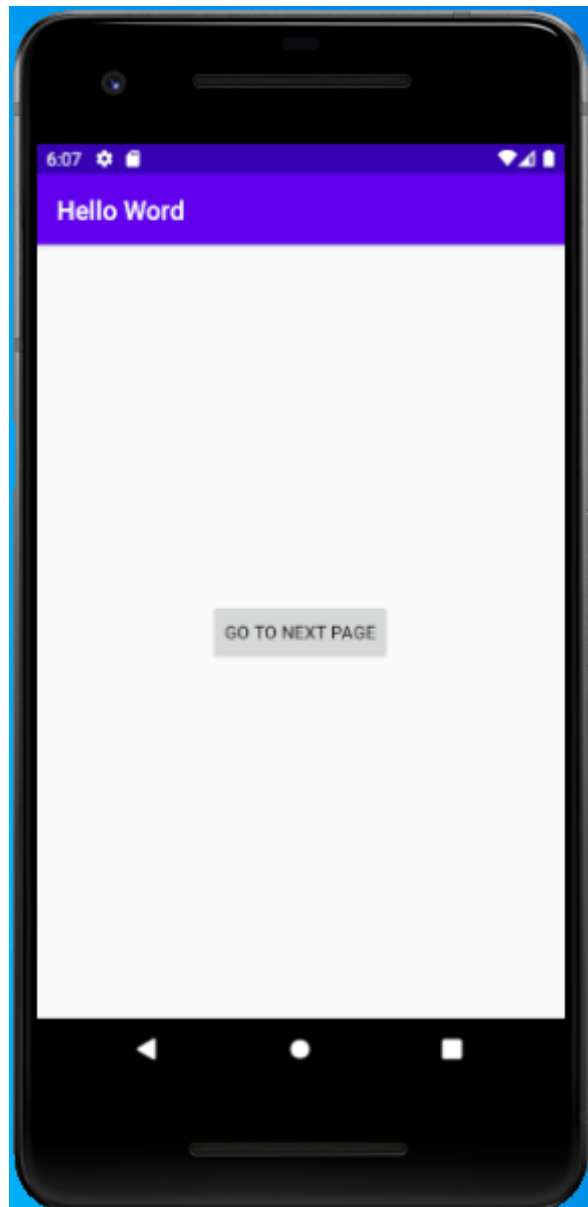
3  import androidx.appcompat.app.AppCompatActivity;
4
5  import android.content.Intent;
6  import android.os.Bundle;
7  import android.view.View;
8  import android.widget.Button;
9
10 public class MainActivity extends AppCompatActivity {
11
12     private Button button;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         button = (Button) findViewById(R.id.btn);
20
21         button.setOnClickListener(new View.OnClickListener() {
22             @Override
23             public void onClick(View v) {
24                 viewHelloWordActivity();
25             }
26         });
27     }
28
29     public void viewHelloWordActivity() {
30         Intent intent = new Intent(this, HelloWordActivity.class);
31         startActivity(intent);
32     }
33 }

```

Dans la classe MainActivity, la page activity\_main sera chargée. Ensuite nous avons créé une fonction setOnClickListener qui permet de détecter un clic sur un bouton.

Après un clic sur le bouton **Go to next page** la fonction viewHelloWordActivity() sera appelée. Cette méthode permet de charger HelloWordActivity.java qui affiche à son tour Hello Word.

### Design activity\_main.xml

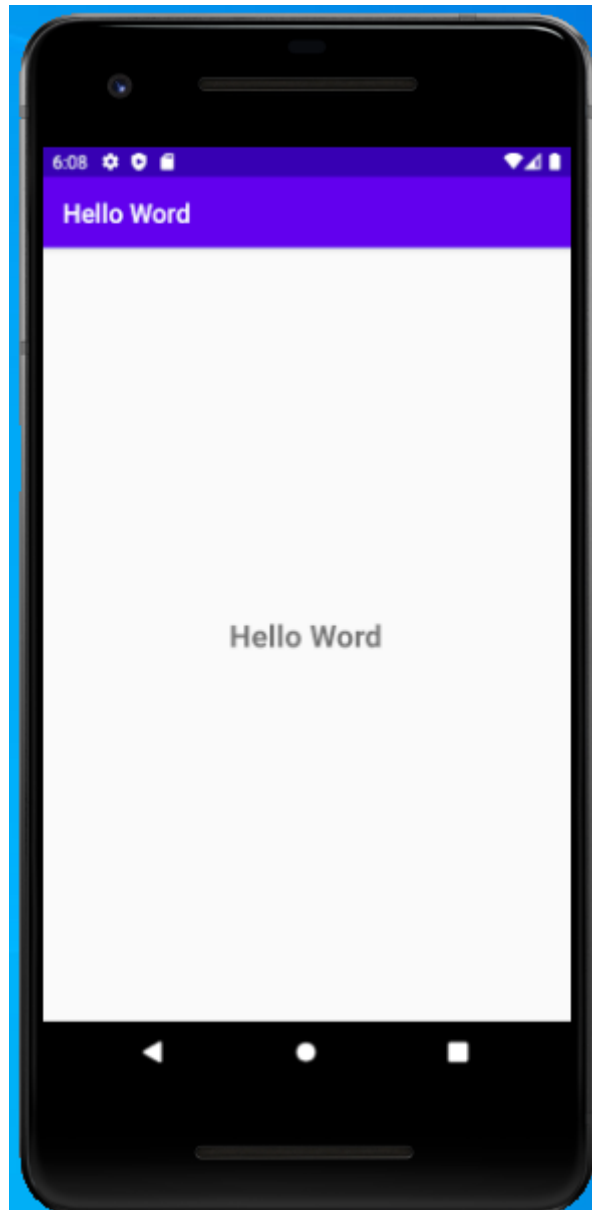


### HelloWordActivity.java

```
1  package com.nsallmdia01.helloworld;
2
3  import androidx.appcompat.app.AppCompatActivity;
4
5  import android.os.Bundle;
6
7  public class HelloWordActivity extends AppCompatActivity {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_hello_word);
13     }
14 }
```

---

## Design du fichier activity\_hello\_word.xml



### III. Commandes utilisées

**Git clone** est utilisé pour cloner un repository git distant dans un répertoire local. Par défaut la commande recrée le répertoire contenant le dossier. git et y télécharge le contenu du repository.

**Git branch** permet de lister les branches que nous avons, de créer une nouvelle branche en locale, supprimer et renommer des branches.

**Git checkout** permet de naviguer entre les branches créées par `git branch`. Lorsqu'on utilise cette commande, les fichiers présents dans le répertoire pour correspondre à la version stockée dans cette branche et `git` enregistrera tous les nouveaux commits de la branche.

**Git pull** : est en fait la combinaison de deux autres commandes, `git fetch` suivi de `git merge`. Lors de la première étape, `git pull` exécute une commande `git fetch` étendue à la branche locale vers laquelle HEAD pointe. Une fois le contenu téléchargé `git pull` entre en workflow de merge.

**Git push** est utilisée pour charger le contenu d'un dépôt local vers un dépôt distant. Le `push` vous permet de transférer les commits de votre dépôt local vers un dépôt distant.