Compte rendu :

TP ADB

Enseignant : Maddouri Faouzi

Réalisé par :

Guesmi Ikram

Farhaoui Eya

Année universitaire : 2024/2025

Etablissement: ISET Rades

Contents

Objectif

Ce compte rendu détaille les étapes suivies pour vérifier et utiliser le pont de débogage Android (ADB) sous Windows.

1 Configuration d'ADB

1.1 Vérification de l'installation

La première étape a consisté à vérifier la présence de l'outil ADB sur le système. La commande suivante a été exécutée :

```
C:\Users\tabar>adb version
'adb'  n est  pas reconnu en tant que commande interne
ou externe, un programme ex cutable
ou un fichier de commandes.
```

Pour résoudre ce problème, nous avons ajouté le chemin vers le dossier contenant `adb.exe` (par exemple, `C:\Users\tabar\AppData\Local\Android\Sdk\platform-tools`) dans les variables d'environnement du système.

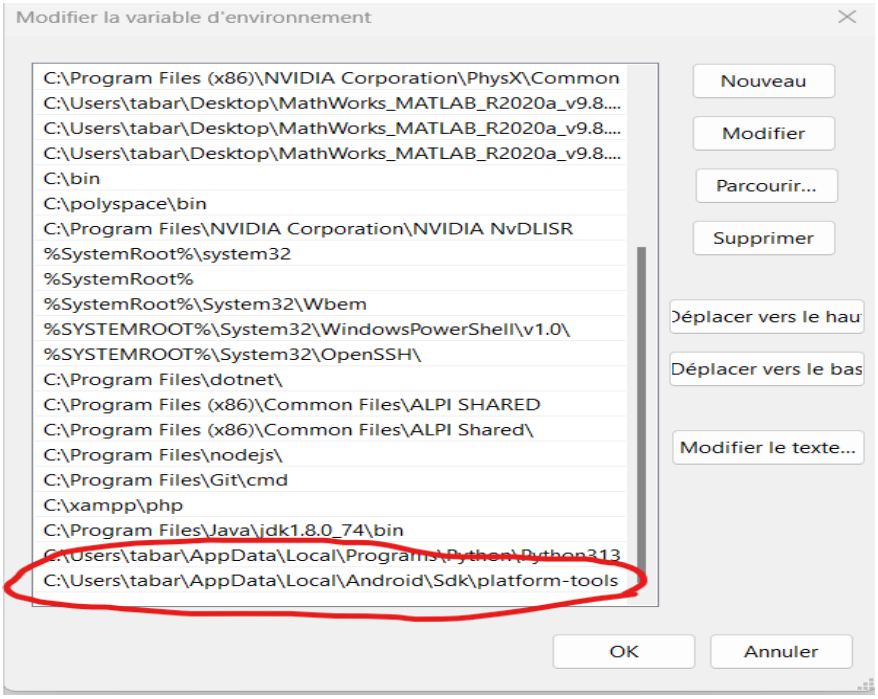

Figure 1: Configuration de l'environnement ADB sous Windows.

Après modification et redémarrage du terminal, la commande suivante a confirmé l'installation réussie :

```
C:\Users\tabar>adb version
Android Debug Bridge version 1.0.41
Version 35.0.2-12147458
Installed as
```

```
C:\Users\tabar\AppData\Local\Android\Sdk\
platform-tools\adb.exe
```

1.2 Connexion et Détection du Téléphone

Pour vérifier que le téléphone était détecté par ADB, nous avons utilisé la commande suivante :

```
C:\Users\tabar>adb devices
List of devices attached
DUJ6R20C10000746        device
```

2 Transfert de Fichiers avec ADB

2.1 Résolution et Transfert Réussi

la commande suivante a permis un transfert réussi d'une image du pc vers le Téléphone :

```
C:\Users\tabar>adb push C:\Users\tabar\Pictures
\images.jpeg /sdcard/Download/
C:\Users\tabar\Pictures\images.jpeg:...ed. 11.7 MB/s
(3870 bytes in 0.000s)
```

2.2 Transfert d'une image sur le Téléphone

Pour vérifier la présence du fichier transféré, nous avons utilisé la commande suivante :

```
C:\Users\tabar>adb shell ls -l /sdcard/Download/
```

Cela a confirmé que le fichier `images.jpeg` était bien présent :

```
-rw-rw---- 1 root sdcard_rw
3870 2024-01-04 17:58 images.jpeg
```

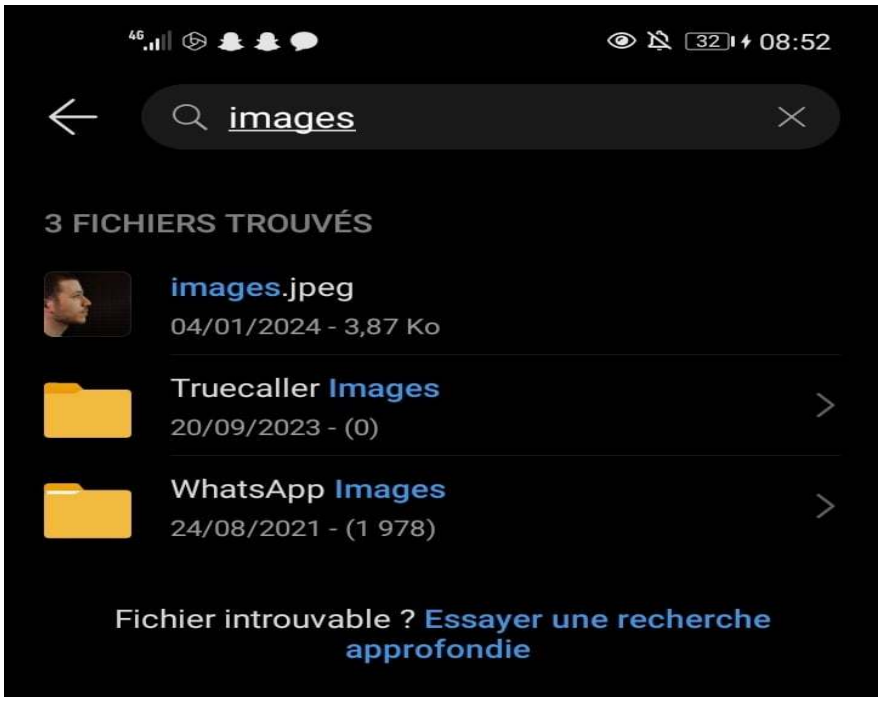

Figure 2: verification de lenvoi de l'image sur le telephne .

3 Transfert et Récupération de Résultats de Commandes Shell

3.1 Exécution des commandes Shell

Nous avons accédé au shell du terminal mobile avec la commande suivante :

```
C:\Users\tabar>adb shell
```

Dans le shell, nous avons exécuté deux commandes Shell ('ps -ef' et 'ls -l') et redirigé leurs sorties respectives vers des fichiers texte placés dans le répertoire `/sdcard/Download` :

```
HWART-H:/ $ ps -ef > /sdcard/Download/ps_output.txt
HWART-H:/ $ ls -l > /sdcard/Download/ls_output.txt
```

Remarque : La commande `ls -l` a affiché des erreurs d'autorisation pour certains fichiers, ce qui est normal en raison des restrictions d'accès aux fichiers système.

3.2 Vérification des fichiers créés

Nous avons vérifié que les fichiers avaient bien été créés dans le répertoire `/sdcard/Download` avec la commande suivante :

```
C:\Users\tabar>adb shell ls -l /sdcard/Download/
```

Voici un extrait des résultats obtenus :

```
-rw-rw---- 1 root sdcard_rw     2631 2024-11-23 09:23
ls_output.txt
-rw-rw---- 1 root sdcard_rw    40881 2024-11-23 09:22
ps_output.txt
```

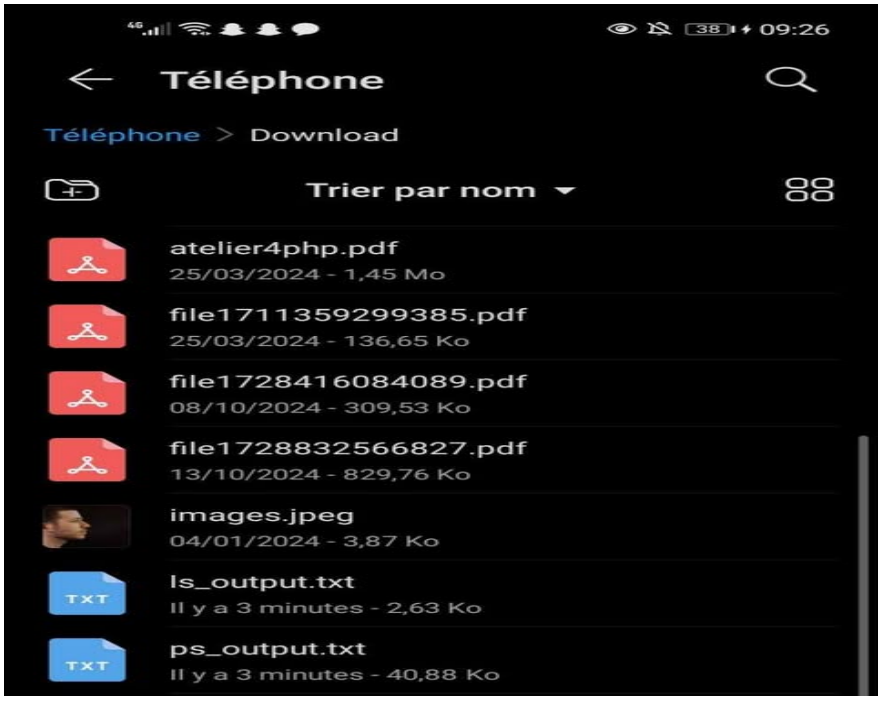

Figure 3: Vérification de la creation des deux fichiers sur telephone.

3.3 Transfert des fichiers vers le PC

Les fichiers ont été récupérés sur le PC avec les commandes suivantes :

```
C:\Users\tabar>adb pull /sdcard/Download/ps_output.txt
C:\Users\tabar\Documents\
C:\Users\tabar>adb pull /sdcard/Download/ls_output.txt
C:\Users\tabar\Documents\
```

Le transfert a été effectué avec succès, comme indiqué ci-dessous :

```
/sdcard/Download/ps_output.txt: 1 file pulled.
5.5 MB/s (40881 bytes in 0.007s)
```

```
/sdcard/Download/ls_output.txt: 1 file pulled.
0.6 MB/s (2631 bytes in 0.004s)
```

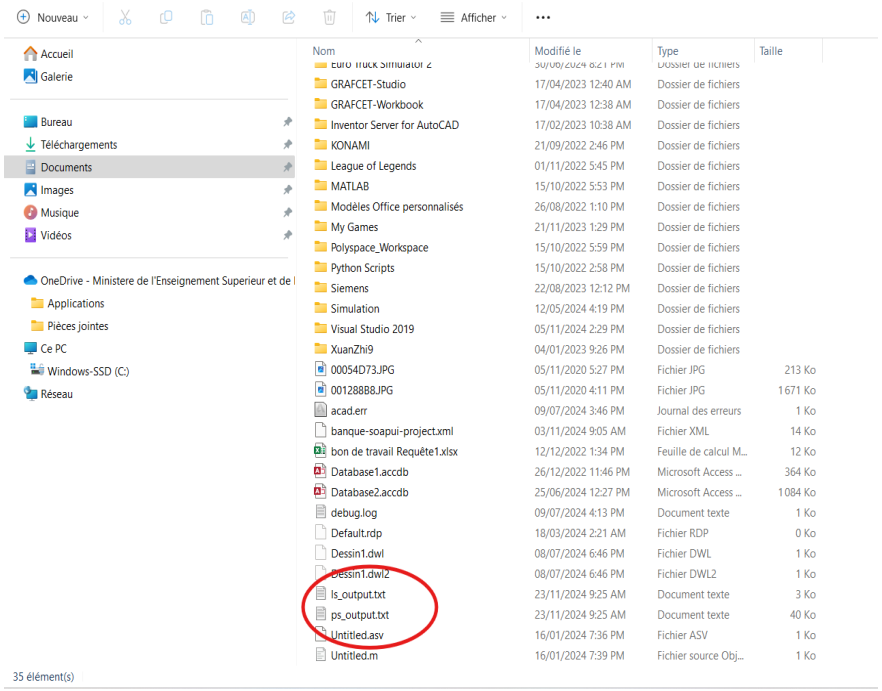

Figure 4: Vérification du transfert des deux fichiers sur PC.

4 Accès aux bases de données des contacts

4.1 Accès aux bases de données des contacts

Une tentative d'accès direct au répertoire des bases de données des contacts :

```
adb shell
cd /data/data/com.android.providers.contacts
/databases/
```

Résultat :

```
Permission denied
```

Cela est dû à des restrictions de permissions sur les appareils Android en production.

Tentative de basculer ADB en mode root :

```
adb root
```

Résultat :

```
adbd cannot run as root in production builds
```

4.2 Récupération des contacts via URI

Face à ces restrictions, nous avons utilisé la commande suivante pour accéder aux données des contacts via une requête URI :

```
adb shell content query --uri content://contacts
/phones/ --projection display_name:number
```

Résultat :

```
Row: 0 display_name=salah, number=+216 *******
Row: 1 display_name=Chiraz, number=+216 *******
...
```

Ces résultats incluent les noms des contacts et leurs numéros de téléphone, formatés ligne par ligne.

5 Installation manuelle d'une application Android (.apk)

5.1 Étape 1: Vérification préalable

Nous avons vérifié que le package n'était pas installé au préalable en exécutant la commande suivante:

```
adb shell pm list packages | findstr com.example.tpadb
```

Aucun résultat ne fut retourné, confirmant l'absence de l'application.

5.2 Étape 2: Compression du dossier

Une archive portant le même nom que le package a été créée pour simuler une installation manuelle.

5.3 Étape 3: Installation via ADB

L'application a été installée à l'aide de la commande suivante:

```
adb install C:\Users\tabar\AndroidStudioProjects
\TpADB\app\build\outputs\apk\debug\app-debug.apk
```

Lors de la première tentative, une erreur `INSTALL_FAILED_ABORTED: User rejected permissions` est survenue. Pour résoudre ce problème, nous avons activé l'option "Installer des applications depuis des ressources externes" sur le téléphone. Après cela, l'installation a été un succès.

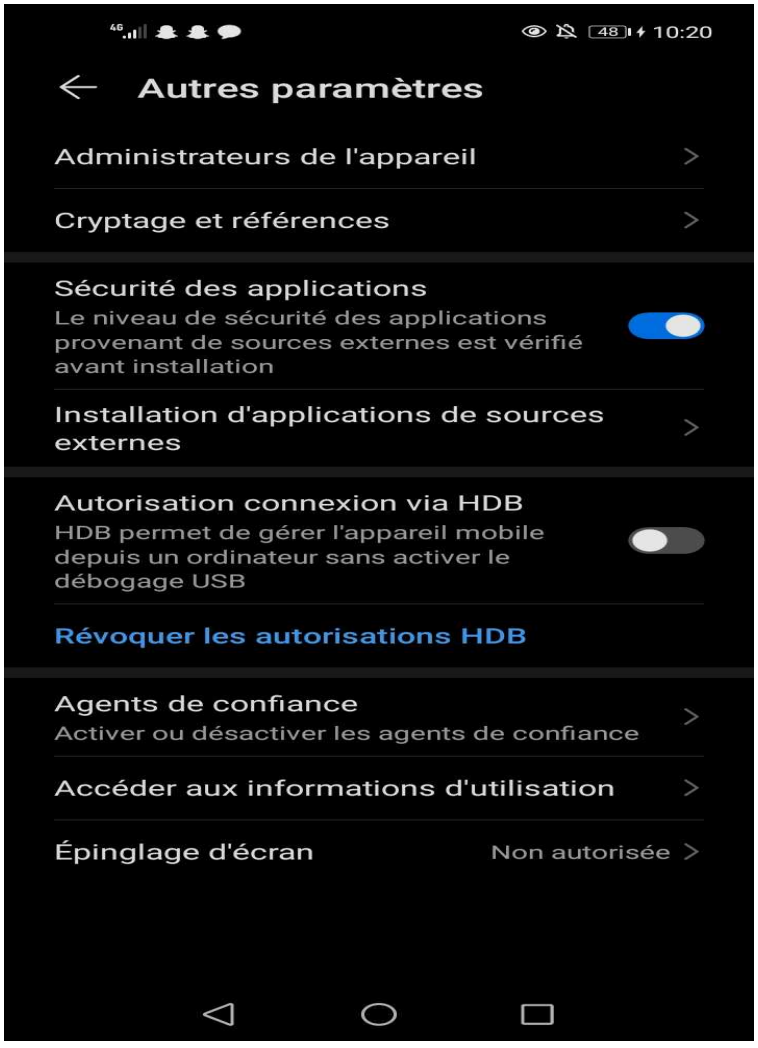

Figure 5: activation de l'option "Installer des applications depuis des ressources externes.

5.4 Étape 4: Vérification de l'installation

Nous avons confirmé que le package était bien installé:

```
adb shell pm list packages | findstr com.example.tpadb
```

Résultat:

```
package:com.example.tpadb
```

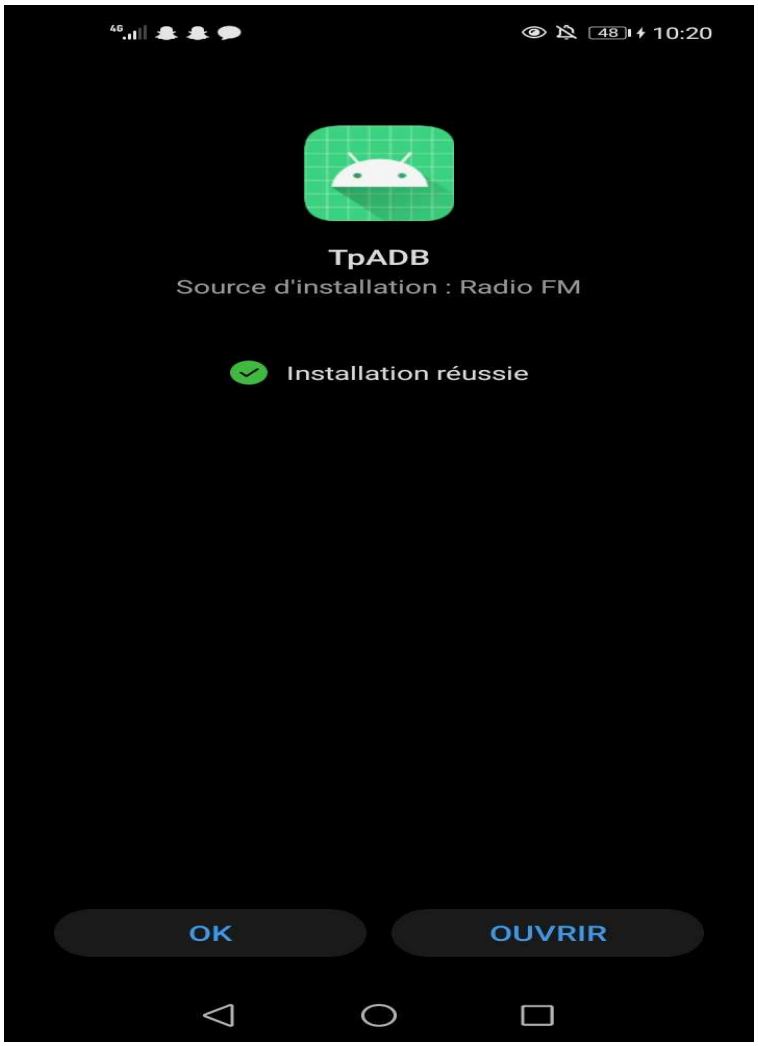

Figure 6: Installation de l'application sur le telephone

5.5 Étape 5: Lancement de l'activité principale

Pour tester le fonctionnement de l'application, nous avons lancé l'activité principale à l'aide de la commande suivante:

```
adb shell am start -n com.example.tpadb/com.example
.tpadb.MainActivity
```

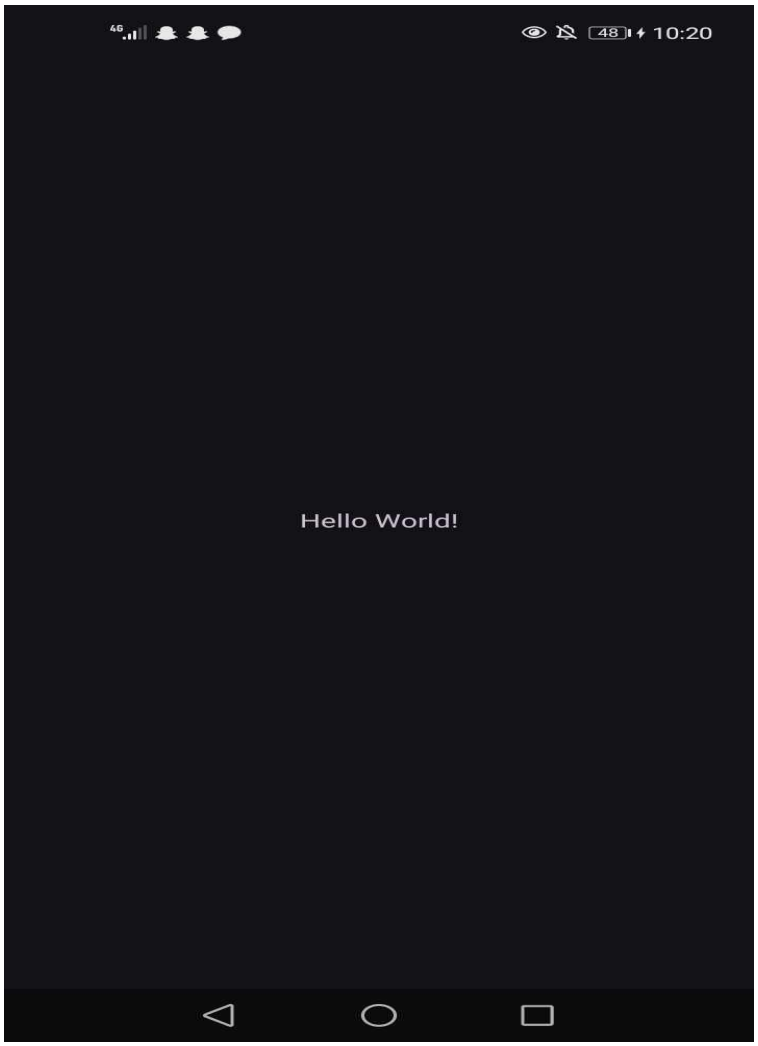

Figure 7: Lancement de l'activité principale

5.6 Étape 6: Exécutions répétées avec profilage

Nous avons lancé l'activité 20 fois de suite en activant le profilage avec l'option `-P`, et les résultats ont été redirigés vers un fichier texte:

```
for /l %x in (1, 1, 20) do adb shell am start -n
com.example.tpadb/com.example.tpadb.MainActivity -P
> profiling_output.txt
```

Le fichier `profiling_output.txt` a été récupéré sur le PC pour analyse.

6 Désinstallation d'un APK spécifique

6.1 Récupération de la liste des packages installés

Nous avons ensuite listé toutes les applications installées sur le terminal mobile en exécutant la commande suivante :

```
C:\Users\tabar>adb shell pm list packages
```

Le résultat était une liste des packages sous la forme :

```
package:com.huawei.hifolder
package:com.android.cts.priv.ctsshim
package:com.huawei.camera
package:com.huawei.permissioncontroller.overlay
package:com.huawei.android.tips
package:com.huawei.camerakit.impl
package:com.huawei.synergy
.
.
.
```

6.2 Récupération détaillée avec chemin d'installation

Pour obtenir des informations supplémentaires, telles que les chemins des fichiers APK installés, nous avons utilisé la commande :

```
C:\Users\tabar>adb shell pm list packages -f
```

Cela affiche chaque package avec son chemin d'installation sur le terminal.

6.3 Exportation des résultats vers un fichier texte

Nous avons redirigé la liste des packages dans un fichier texte pour une consultation plus facile :

- Sur le terminal mobile :

```
    adb shell 'pm␣list␣packages␣>
␣␣␣␣/sdcard/Download/installed_packages.txt'
```

- Sur l'ordinateur :

```
C:\Users\tabar>adb shell pm list packages >
C:\Users\tabar\Documents\installed_packages.txt
```

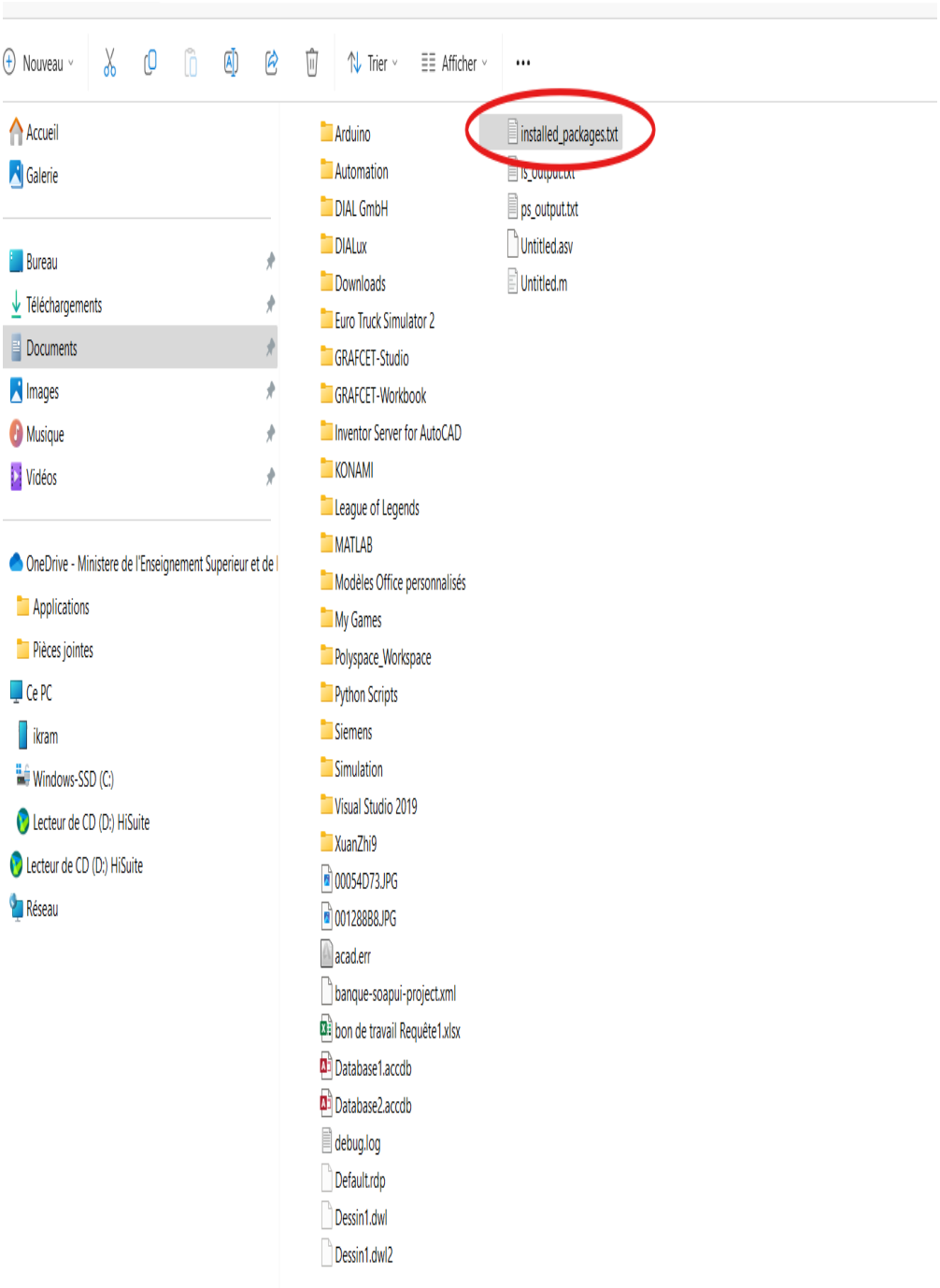

Figure 8: emplacement du fichier texte sur l'ordinateur

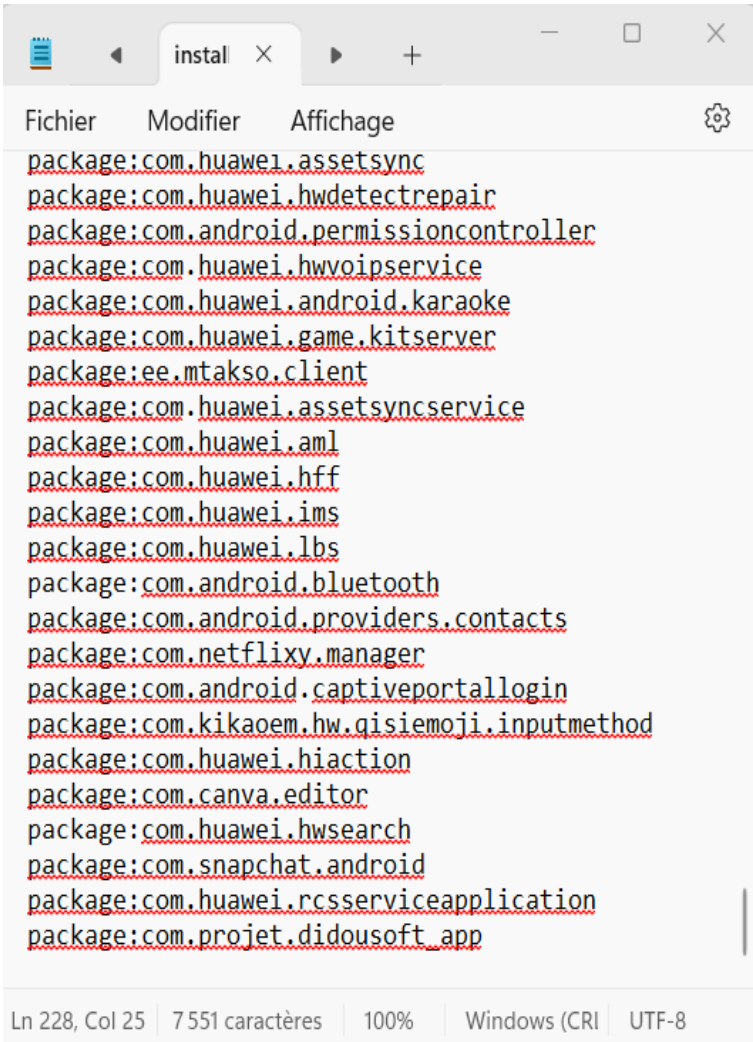

Figure 9: contenu du fichier texte

6.4 Désinstallation de l'application

Pour désinstaller l'application sélectionnée, nous avons exécuté la commande suivante :

```
adb shell pm uninstall -k --user 0 com.example.package1
```

Cette commande supprime l'application tout en conservant ses données (`-k`).

Conclusion

En conclusion, l'utilisation d'ADB sous Windows permet une gestion efficace des appareils Android, en offrant un contrôle complet sur divers aspects du système Android à distance.