1. Découverte de Android Studio

On va commencer par le principal outil de travail, l'atelier de création des applications : Android Studio. Il se trouve dans le menu démarrer.

Au tout premier lancement, il va vous poser quelques questions :

1. D'abord laissez Do not import settings coché : vous n'avez encore jamais travaillé avec Android Studio, donc rien à importer.
2. Ensuite, choisissez Don't send concernant les statistiques à envoyer à Google.

Il y a ensuite une phase de téléchargement des composants du SDK, un peu lente, il faut patienter.

3. Dans le dialogue qui apparaît « Android Studio First Run », cliquez sur le bouton grisé Setup Proxy
4. Dans la fenêtre «Welcome», cliquez sur Next.
5. Ensuite dans «Install Type», cochez installation de type *Custom*, cliquez sur Next.
6. Vérifiez que le chemin du *JDK* est bien C:\Program Files\Android\Android Studio\jbr (jbr = *JetBrains Runtime*, un SDK optimisé pour Android Studio), cliquez sur Next.
7. Choisissez l'esthétique des fenêtres (sombre ou clair), cliquez sur Next.
8. Puis, dans la fenêtre SDK Components Setup, ne cochez rien car tout est déjà installé. Simplement, vérifiez en bas l'emplacement du SDK Android : D:\android\sdk. Normalement, en bas, il vous affiche *An existing Android SDK was detected*. . . . Tapez sur Next. Il va y avoir une erreur Required Component Missing, mais tapez OK quand même.

9. Il va proposer d'installer le Android Emulator Hypervisor Driver, c'est le composant qui manque à l'étape précédente. Tapez Next, puis Finish et attendez encore un peu, s'il y a un téléchargement. À la fin, il va vous dire que *Android SDK* est à jour. Cliquez sur Finish.
10. La fenêtre principale d'Android Studio apparaît enfin, et elle apparaîtra directement les prochaines fois.

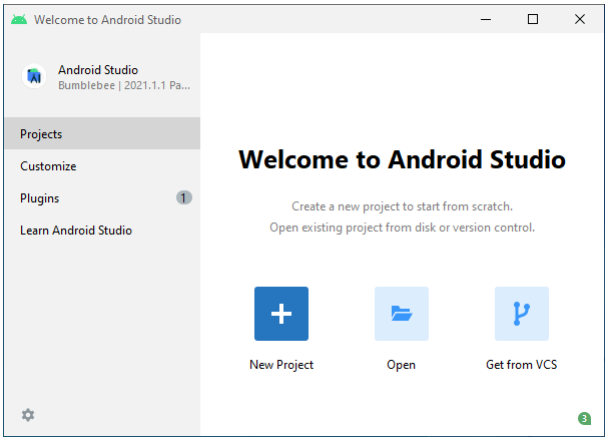

Figure 1: Fenêtre d'accueil

NB: Il y aura des différences entre les copies écran des énoncés de TP et votre installation. Android Studio change très régulièrement la position et l'apparence des vues.

1.1. Création d'un premier projet Android

1. Au tout début, vous n'aurez aucun projet Android. Dans la fenêtre d'accueil, choisissez New Project pour commencer à créer votre premier projet.
2. Dans la première boîte de dialogue, choisissez le type d'application souhaitée. Pour le TP1 et parce que c'est le modèle le plus simple, changez la sélection pour *Empty Views Activity*.

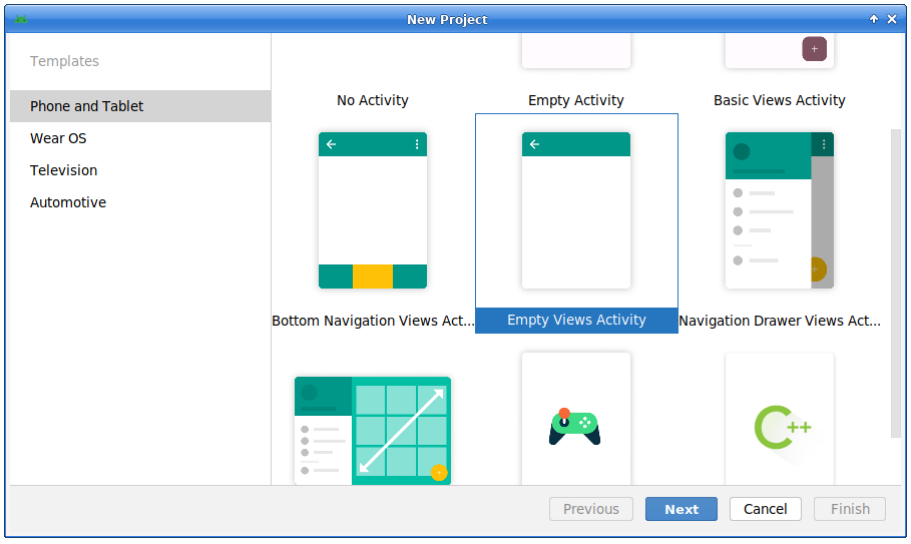

Figure 2: Type de projet

Attention, ce n'est pas *Empty Activity*, mais *Empty Views Activity*. Le type *Empty Activity* sélectionné par défaut, fait créer un projet avec Jetpack Compose qui est très similaire à Flutter (*Material Design*). On étudiera cette API dans un prochain TP, mais elle est trop complexe pour un premier TP.

3. Dans la seconde boîte de dialogue, on vous demande plusieurs choses :
   a. Le nom de l'application, mettez `TP1`,
   b. Son paquetage : mettez `fr.iutlan.tp1`,
   c. L'emplacement du projet, laissez l'emplacement proposé par défaut, `C:\Users\vous\AndroidStudioProjects\.....` Si vous êtes chez vous, laissez le dossier proposé par l'assistant.
   d. Le langage proposé est Kotlin ou java. Dans la version d'Android Studio.
   e. Le niveau d'API SDK voulu, par exemple la 16, 22, 24, et.. C'est la version minimale que devront avoir les smartphones pour installer votre application : ils devront être au moins aussi récents que cette API. Cela définit le niveau de fonctionnalités de votre application : s'il est trop élevé, peu de smartphones seront capables de faire tourner votre application ; trop bas, vous n'aurez pas beaucoup de fonctions agréables à votre disposition. Vous pouvez cliquer sur *Help me choose* pour avoir un graphique de la distribution des versions d'API dans le monde.

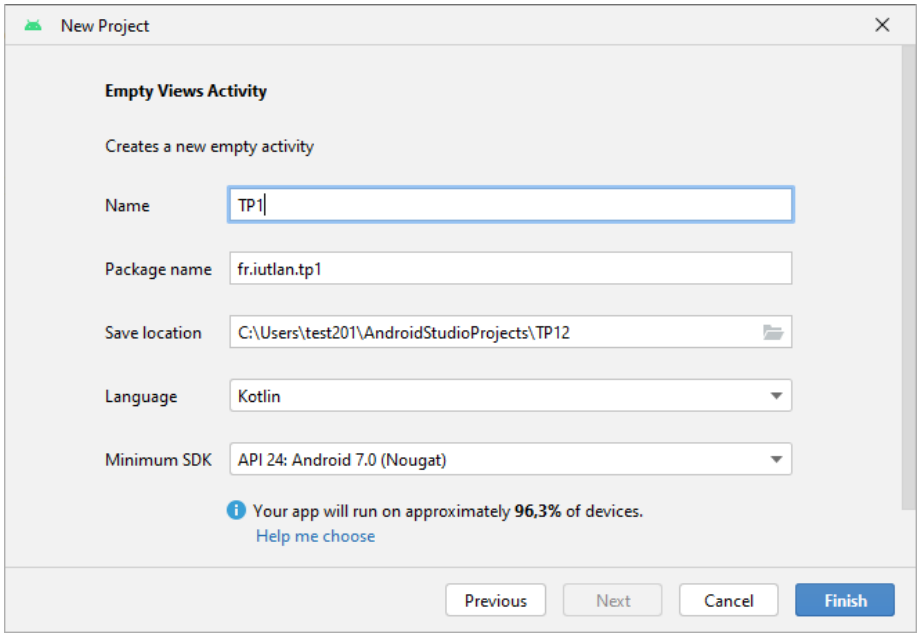

Figure 3: Détails du projet

4. Android Studio crée ensuite le projet.

1.2. Découverte de la fenêtre de travail

L'interface d'Android Studio est un peu déroutante au début. Elle repose sur IntelliJ qui est un concurrent d'Eclipse et de Visual Studio Code. Vous verrez rapidement qu'on ne se sert que d'une toute petite partie, assez vite apprise. Il y a des boutons tout autour de la fenêtre : Project, Structure, Captures, Favorites, Build Variants, Terminal, Android Monitor, Messages, TODO, etc. Ce sont des boutons à bascule, exclusifs entre eux sur un même côté, qui affichent des onglets ; celui qui est actif est grisé. Par exemple dans l'image ci-dessous, c'est l'onglet Project qui est ouvert.

La structure du projet est visible à gauche, comme dans Eclipse. Les fichiers sont arrangés de manière pratique : manifeste, sources java et ressources, grâce au choix déroulant Android juste au dessus. (NB: sur cette capture d'écran, les fichiers sont colorés car gérés par subversion).

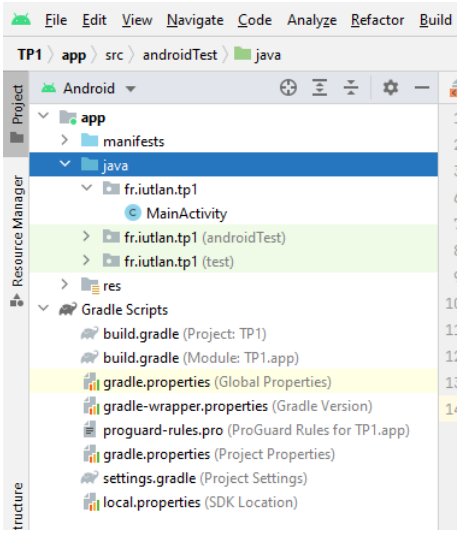

Figure 4: Inventaire du projet

Prenez quelques instants pour déplier les dossiers manifests, java et res et regarder ce qu'il y a dedans (ça a été décrit en cours).

1.3. Structure du projet

Allez voir le dossier correspondant au projet, dans le dossier AndroidStudioProjects\TP1 de vos documents, le nom complet est dans le titre de la fenêtre Android Studio :

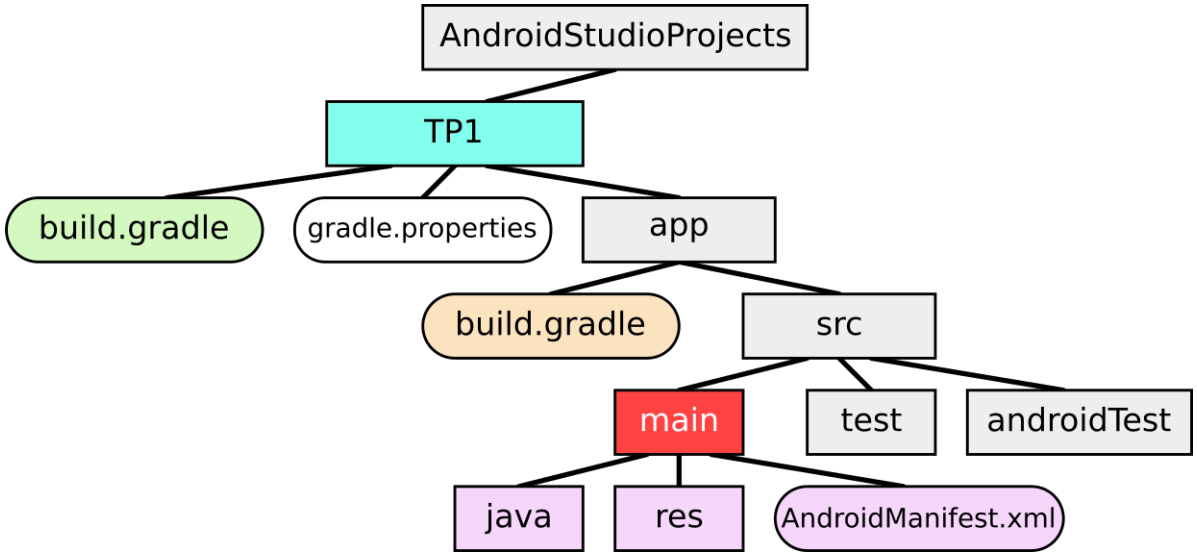

Figure 5: Structure des fichiers d'un projet Android

- Le dossier du projet contient le premier build.gradle, celui en vert qui indique les dépôts à utiliser pour la compilation (Google et mavenCentral). Le fichier gradle.properties contient la configuration du proxy. On trouve aussi un local.properties contenant le chemin du SDK.
- Le dossier app contient le second build.gradle, celui en orange qui décrit les caractéristiques du projet : API et dépendances.
- Le dossier app/src/main en rouge contient le cœur du projet : manifeste, sources et ressources.
- app/src/main/java contient les dossiers correspondant au package que vous avez déclaré, avec les sources Kotlin tout au bout du package, ici MainActivity.kt. Oui, les sources sont en Kotlin, mais placés dans le dossier Java.
- app/src/main/res/drawable contiendra des images vectorielles utilisées dans votre projet.
- app/src/main/res/layout contient les dispositions d'écran des activités de votre projet (voir le TP2).
- app/src/main/res/menu contient les menus contextuels et d'application du projet (voir le TP5).
- app/src/main/res/mipmap* contient les icônes bitmap dans diverses résolutions.
- app/src/main/res/values contient les constantes du projet, dont les chaînes de caractères avec toutes leurs traductions.

Ces fichiers n'apparaissent pas ainsi dans l'interface, ils sont regroupés autrement, à vous de voir comment.

2. Découverte du SDK Android

On va regarder les outils de développement : le *Software Development Toolkit* d'Android.

2.1. Gestionnaire de SDK

Ouvrez le gestionnaire de SDK en cliquant sur le bouton entouré en bleu (c'est un cube noir avec une flèche bleue descendante).

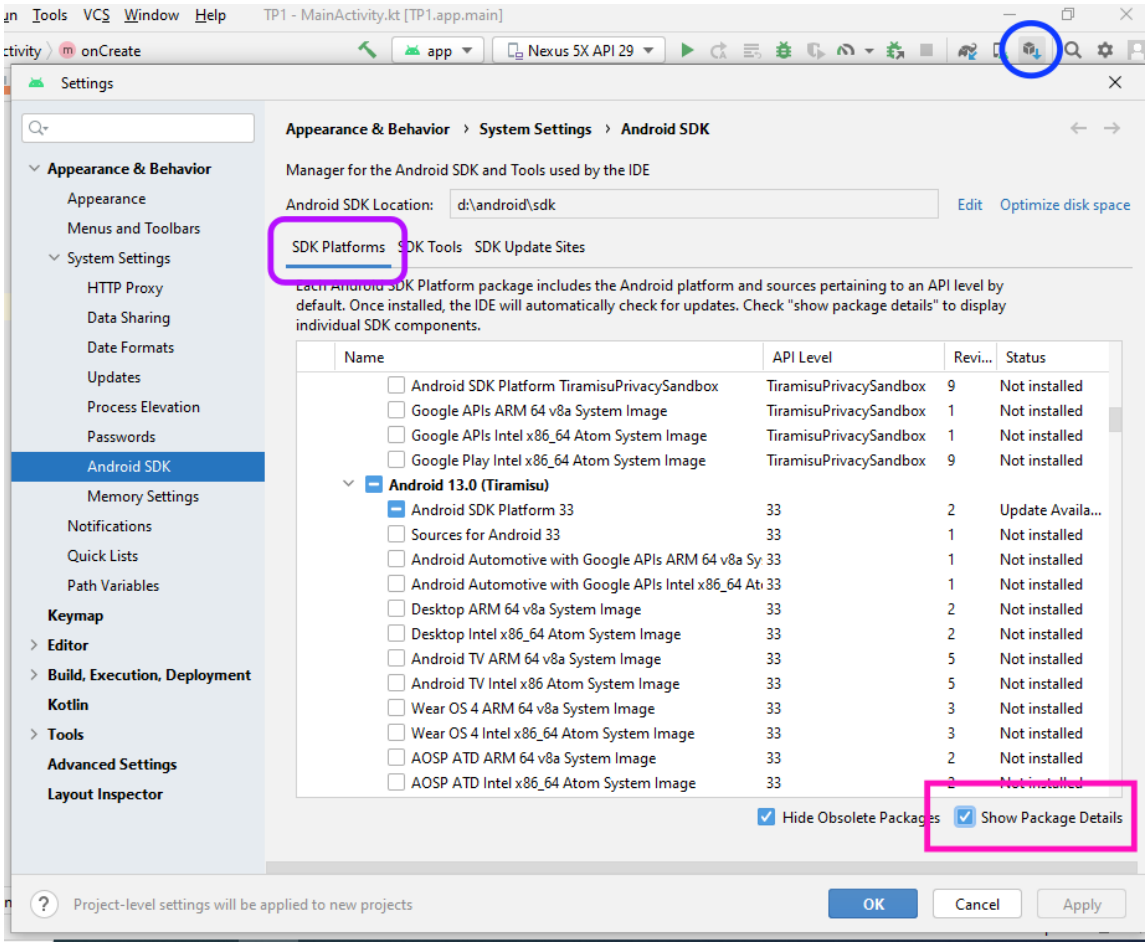

Figure 6: SDK Manager

Cochez Show Package Details pour voir tous les éléments installables : différentes machines virtuelles, avec ou sans Google Maps, etc.

L'onglet SDK Tools montre les outils installés, par exemple Android Emulator, Android SDK Tools. Ne vous inquiétez pas de ne rien y comprendre au début.

Notez l'emplacement du SDK pour l'exercice suivant, *Android SDK Location* : D:\android\sdk.

Fermez la fenêtre avec le bouton Cancel. **Vous devez ne rien changer du tout.**

2.2. Contenu du SDK

Avec le navigateur de fichiers du système, ouvrez le dossier du SDK et regardez ce qu'il y a dedans :

- tools qui contient les outils de gestion du SDK et des AVD (emulator.exe),
- platform-tools qui contient quelques programmes utiles comme adb.exe (adb sous Linux) Ces programmes sont des commandes qui sont normalement disponibles dans une fenêtre appelée *Terminal* en bas d'Android Studio. En principe, vous devriez pouvoir taper adb dans ce shell et ça vous affiche l'aide de cette commande. On va s'en servir un peu plus bas.
- system-images qui contient les images de la mémoire de tablettes virtuelles. Juste pour comprendre pourquoi il n'y a que ça, regardez la taille des fichiers .img. Ce sont des images de disques virtuels à lancer avec qemu.

3. Android Virtual Device : AVD

On revient encore dans Android Studio, pour lancer le gestionnaire d'AVD en cliquant sur le bouton entouré en bleu (c'est un rectangle représentant un smartphone avec le haut de l'Android vert).

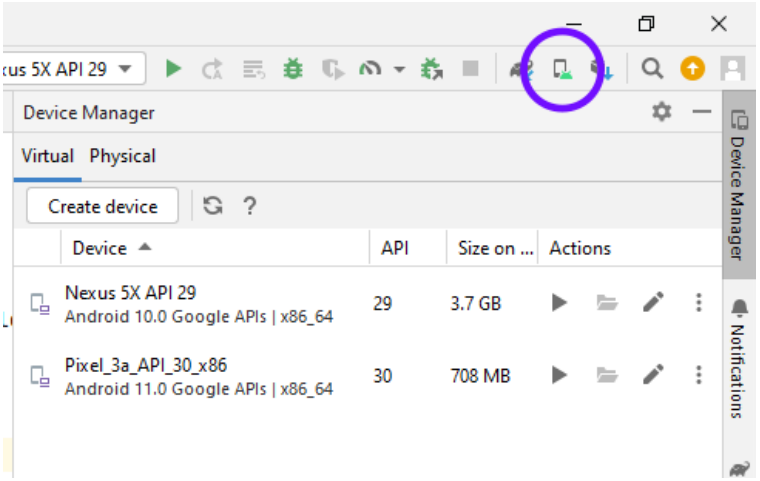

Figure 7: Gestionnaire d'AVD

Normalement, vous devez déjà trouver un AVD existant. Il a un écran un peu trop grand pour nos besoins.

3.1. Création d'un AVD si nécessaire

Commencez à créer un smartphone virtuel, avec le bouton + Create Virtual Device. Ensuite, vous avez beaucoup de choix possibles. Le principe est de ne pas créer un smartphone virtuel trop riche. Si le smartphone virtuel est trop puissant, trop gros (RAM et taille d'écran), il va occuper énormément de place en mémoire et sur le disque. Vous pourriez vous retrouver à court.

Un autre point important est le CPU du smartphone virtuel. Les vrais smartphones fonctionnent en général avec un processeur ARM. Ça désigne un modèle de programmation (registres et instructions) spécifiques qui doit être simulé sur votre ordinateur. Le mieux est de choisir un AVD ayant un processeur de type amd64 qui n'aura pas besoin d'être simulé, mais directement utilisé, comme avec Docker ou VirtualBox.

Une solution prudente consiste donc à choisir l'AVD le moins rutilant. Par exemple, choisissez le type Phone, modèle 5.1 480x800 en *mdpi* ; puis dans l'écran suivant, changez d'onglet pour x86 images et choisissez une des API installées, puis dans le dernier écran, activez Show Advanced Settings et changez la taille de la RAM : au lieu de 343 Mo, mettez 512 Mo, et décochez Enable Device Frame. Validez la création de ce smartphone virtuel.

La création d'un AVD prend du temps. Il faut générer l'image de la mémoire virtuelle, entre 1 et 4 Go. Une fois que c'est fini, l'AVD apparaît dans la liste et vous pouvez le lancer en cliquant sur le triangle vert à droite. Notez qu'elle met également beaucoup de temps à devenir disponible. Pour ne pas perdre de temps plus tard, vous devrez faire attention à ne pas la fermer.

Si jamais le lancement échoue, c'est qu'il y a eu un problème d'application des stratégies Windows lors du démarrage de la machine. Deux variables, %HOMEDRIVE% et %HOMEDIR% n'ont pas de valeur. Il faut redémarrer l'ordinateur et tout relancer.

Modifiez les préférences de cette tablette : ouvrez l'application Settings.

- Mettez-la en langue française : Language & Input, Language, ajoutez Français (France) et mettez-le en premier dans la liste (c'est selon la variante d'Android que vous avez choisie),
- Enlevez le verrouillage de l'écran en cas d'inaction : Sécurité, Verrouillage de l'écran, Aucun.

Alors par contre, vous ne pourrez pas conserver cette machine virtuelle si elle est créée dans votre *home dir* ou *profil*. Vérifiez cela. Vous devrez impérativement la supprimer de votre compte avant la fin du TP, parce qu'elle occupe plus d'1Go, faisant largement exploser vos quotas à l'IUT. Il y a un AVD prédéfini que vous pourrez utiliser pour vos TP.

3.2. Exécution

Le lancement de l'application est simple. Il suffit de cliquer sur le bouton triangulaire vert dans la barre d'outils (celui qui est entouré en vert ci-dessous).

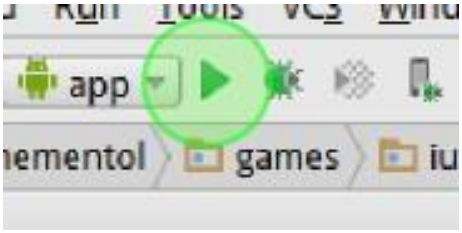

Figure 8: Bouton de lancement

La première compilation prend énormément de temps, car il y a de nombreuses dépendances à télécharger. Vous pouvez voir l'avancement dans l'onglet Build tout en bas.

Ensuite, s'il n'y a aucune tablette connectée ni AVD en cours, Studio vous demande quel AVD il faut lancer. Il en faut un qui soit au moins du niveau d'API de votre application. Cochez Use same device for future launches.

3.3. Android Debug Bridge : ADB

Pour finir, on va communiquer avec cette tablette virtuelle.

1. D'abord une phase de configuration :
   a. Fermez Android Studio
   b. Ouvrez les paramètres Windows (roue dentée dans le menu de lancement des applications).
   c. Tapez « environnement » dans la barre de recherche
   d. Cliquez sur Modifier les variables d'environnement pour votre compte
   e. Dans la liste, cliquez sur Path puis 'Modifier...
   f. Cliquez sur Nouveau
   g. Tapez D:\android\sdk\platform-tools dans la nouvelle ligne,

h. Cliquez OK en bas deux fois pour tout fermer en validant les changements.

2. Ensuite des expérimentations :
   a. Relancez Android Studio
   b. Démarrez un AVD, par exemple lancez l'application
   c. Ouvrez la fenêtre Terminal en bas d'Android Studio, puis essayez les commandes suivantes :

- adb devices Affiche la liste des smartphones connectés : les smartphones réels connnectés avec un câble USB et les AVD virtuels actifs.
- adb shell. Dans cette connexion à la tablette, tapez :
  - ls : c'est une arborescence Unix, mais on n'a pas beaucoup de droits, on n'est pas *root* alors il y a des erreurs dues aux protections.
  - cd /mnt/sdcard : c'est l'un des rares dossiers où on peut aller.
  - mkdir essais : crée un dossier sur la carte sd virtuelle (vérifiez la création avec ls).
  - exit pour sortir du shell et revenir dans Windows.
- il faudrait trouver l'une des images d'arrière-plan de Windows, peut-être dans C:\Windows\Web\Wallpaper\Theme1 à vous de farfouiller pour trouver une image JPG (attention le nom affiché ne correspond pas aux vrais fichiers, il faut cliquer sur la barre d'adresse en haut de la fenêtre). Ensuite faites par exemple adb push C:\Windows\Web\Wallpaper\Theme2\img8.jpg /sdcard/Pictures : ça envoie l'image sur la tablette, dans le dossier correspondant au stockage interne, pas celui nommé SD Card. L'image sera visible dans l'application Google Photos.

Vous voyez la ressemblance de adb avec des outils comme ssh et ftp.