

Atelier de professionnalisation 2 - Android - Informations techniques

Avant de commencer les missions, voici les étapes à suivre pour la mise en place et des informations utiles sur l'accès internet.

Désinstaller proprement Android Studio

L'application a été réalisée avec la version "Arctic Fox | 2020.3.1".

Dans un premier temps, il faut désinstaller complètement votre version actuelle ainsi que les SDK installés. Allez dans "Ajouter et supprimer des programmes" et demandez de désinstaller Android Studio.

S'il est toujours présent, supprimez le dossier "Android" qui se trouve dans le dossier des programmes (normalement : C:\Program Files\Android).

Dans votre dossier utilisateur, demandez d'afficher les fichiers et dossiers cachés, puis, s'ils existent, supprimez : dossier ".android", dossier ".androidStudio" (ou tout dossier commençant par ".android"), dossier ".gradle", dossier ".m2".

Toujours dans le dossier utilisateur, sous-dossier "AppData", cherchez dans les sous-dossiers s'il y a un dossier "JetBrains". Si oui, supprimez-le. Dans le sous-dossier "Local", supprimez les dossiers qui commencent par "android".

Installer Android Studio

En rappel, l'application a été réalisée avec la version "Arctic Fox | 2020.3.1".

Allez sur le site officiel d'Android Studio :

<https://developer.android.com/studio>

Si la version proposée est "2020.3.1", c'est parfait. Dans le cas contraire, téléchargez la dernière version et si par la suite vous rencontrez des difficultés pour faire tourner l'application, alors vous désinstallerez complètement cette version et chercherez la version "2020.3.1" pour l'installer. Il est possible que tout se passe bien et que l'IDE propose juste d'installer le SDK utilisé par l'application.

Une fois le fichier récupéré, installez Android Studio avec les options par défaut.

Intégrer SonarLint à Android Studio

Pour intégrer SonarLint à Android Studio :

"File > Settings > Pugins", onglet "Marketplace", dans la recherche, taper "sonarlint".

Sélectionner "SonarLint" puis cliquer sur "Install". À la fin de l'installation, "Restart IDE".

Lorsqu'un projet est ouvert, il faut cliquer sur l'onglet "sonarlint", tout en bas (au même niveau que "Logcat").

Les alertes sont données pour le fichier actuellement ouvert.

Attention : toutes les alertes ne sont pas forcément à prendre en compte.

Configurer l'AVD

Pour tester l'application, vous configurerez l'AVD "Pixel 3a API 30" :

Ouvrir l'AVD Manager > Create Virtual Device > "Pixel 3a" > "R (API Level 30 - Android 11.0)".

Au niveau de la dernière étape, cliquer sur "Show Advanced Settings" et dans la zone du dessus, faire défiler pour accéder à "Boot options" : sélectionner "Cold boot". Cela permet de éviter certaines erreurs.

L'application doit fonctionner en mettant le smartphone horizontal, dans l'activité de la vidéo. Pour faire la démonstration avec l'AVD, il faut le configurer comme un véritable smartphone : une fois l'AVD lancé, il faut faire glisser vers le bas la souris pour ouvrir le menu du haut et cliquer sur l'icône qui accepte que le smartphone soit mis aussi horizontalement.

Utiliser le mobile comme dispositif de test

Si vous avez un smartphone de type Android, vous pouvez aussi l'utiliser pour les tests, à la place de l'AVD.

Pour cela :

Dans Android Studio, ouvrez la fenêtre de gestion des SDK (clic sur l'icône "SDK Manager" de la barre du haut, à droite d l'icône "AVD Manager"). Dans la nouvelle fenêtre, au centre, sélectionnez l'onglet "SDK Tools". Si ce n'est pas déjà fait, sélectionnez "Android SDK Platform-Tools". Faites OK.

Il faut préciser dans les paramètres du smartphone, qu'il accepte le débogage USB (paramètres > à propos du téléphone > cliquer plusieurs fois sur le numéro du build, menu "options développeur" débloqué : choisir "débogage USB").

Il suffit alors de brancher le smartphone à l'ordinateur via USB et d'accepter le transfert de fichiers : le smartphone va apparaître dans la liste des AVD (combo à gauche du bouton flèche verte "Run 'app'").

Éviter l'erreur d'accès à internet

À titre informatif, avec cette version d'Android, pour éviter une erreur d'accès à internet ("java.io.IOException: Cleartext HTTP traffic to 192.168.0.xx not permitted"), le paramètre suivant a été ajouté à la balise "application" du manifest :

```
android:usesCleartextTraffic="true"
```

Gérer l'affichage d'images provenant d'internet

Sachant qu'il n'est pas possible de prévoir le temps de réponse lors d'une demande d'image provenant d'internet, cette fonctionnalité est gérée dans un thread indépendant. Cependant, la page présentant une formation s'affiche tout de même, et l'image est actualisée au moment de sa réception. C'est normalement très rapide.

La méthode qui s'occupe du chargement de l'image a été placée dans MesOutils. Elle reçoit l'objet graphique à valoriser avec l'image provenant de l'url (second paramètre). Un thread est créé et l'objet graphique est valorisé dès que l'image est reçue.

```
/**
 * Charge une image à partir d'une url
 * @param img
 * @param url
 */
public static void loadMapPreview (ImageButton img, String url) {
    //start a background thread for networking
    new Thread(new Runnable() {
        public void run(){
            try {
                //download the drawable
                final Drawable drawable = Drawable.createFromStream((InputStream) new URL(url).getContent(), srcName: "src");
                //edit the view in the UI thread
                img.post(new Runnable() {
                    public void run() { img.setImageDrawable(drawable); }
                });
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }).start();
}
```