

Informatique Mobile : Fiche 1

Objectifs

- Découverte d'Android Studio (AS)
 - Être capable de créer des projets, et de comprendre globalement leurs structures
 - Comprendre les Wizards
 - Savoir lancer une application
- Première approche d'une application Android
 - Comprendre la structure d'une application
 - Les activités
 - Les layouts
 - Tester le debugging

Android Studio(AS) n'est pas basé sur Eclipse, même s'ils se ressemblent très fort, ces deux IDE sont différents. Les quelques exercices suivants vous permettront d'apprivoiser ce nouvel environnement.

Configuration d'Android Studio

Si vous utilisez votre machine personnelle, référez-« InstallationAndroidStudio.docx » sur l'eCampus.

Si vous travaillez sur les machines de l'école, quelques configurations seront nécessaires :

Lors du démarrage d'Android Studio, il va vous demander d'importer des settings; choisissez "I do not have a previous version,"

Ensuite téléchargez sur l'eCampus le fichier settings.jar et mettez le n'importe où.

Lorsque vous aurez l'écran d'accueil, choisissez Settings => import Settings et sélectionnez le fichier settings.jar que vous venez de télécharger.

Ce fichier configure 3 choses:

1. La localisation du sdk dans le "Project Structure" : c:\Progs\Android\android-sdk. (Settings -> Project Structure -> SDK Localisation)
2. Il met Gradle Offline pour qu'il n'essaye pas de se mettre à jour avant de compiler le projet.
3. Il empêche le SDK manager de chercher des mises à jour quand on le lance.

Ces deux dernières configurations ne sont nécessaires que dans le cas où l'on travaille sans connexion internet ou que l'on est derrière un proxy tatillon.

Si vous avez déjà changé la configuration d'AS et que rien ne marche plus, fermez AS et supprimez le répertoire Z:\.AndroidStudio.3

Attention, le premier lancement du projet sera lent car il doit indexer plein de choses.

Exercice 1 : « Blank Activity »

Nous allons créer pas à pas une Application ne contenant qu'une activité affichant du texte.

Création du projet

1. Lancez Android Studio et créez un nouveau projet (File -> New Project). Le nom de l'application doit commencer par une majuscule ; c'est celui qui apparaîtra sur l'appareil (et google play ☺)
2. Notez que la notion de Workspace n'existe pas en Android Studio ! Choisissez donc la localisation de votre projet (ainsi que son nom et package). Pour plus de rapidité si vous êtes sur les machines de l'école, choisissez un répertoire dans votre Z:\ . Appuyez sur Next.
3. Sélectionnez la version minimale d'Android que votre application supportera. Dans un premier temps, nous ne supporterons pas les versions plus anciennes que 4.4 car cela impliquerait l'usage d'une librairie de compatibilité (app_compat dont nous verrons l'usage plus tard). Cliquez sur Next.
4. Choisissez « **Blank Activity** ». Next
5. Appuyez sur Finish et prenez patience.

Lancement de l'application

Lancez l'application (à l'aide de l'icône « play » sur la tool bar).

Après un certain temps AS vous demandera de lancer un émulateur s'il n'y en a pas déjà un qui tourne. Choisissez un de ceux proposés par défaut.

Prenez patience (entre 1 et 3 minutes).

Une fois l'application lancée, cliquez sur les trois points verticaux en haut à droite de l'écran.

Dans la fenêtre d'AS, en bas, observez le logcat, la mémoire et le cpu qui évoluent.

Observation de la structure

Votre application comporte trois folders :

1. Manifest qui contient la description de haut niveau de votre application :
 - ce qu'elle est capable de faire
 - les permissions demandées
 - les Activity (écrans) qu'elle contient
2. Java
3. res qui comprend les ressources du projet :
 - les images, les icônes
 - la description des écrans
 - des valeurs, et chaînes de caractères utilisées dans l'application

Prenez le temps d'observer le contenu des différents fichiers et répondez à ces quelques questions :

1. Quel est le fichier contenant la description de l'unique activité (écran) de l'application ?
2. Ouvrez le fichier MainActivity.java. A quel endroit dans le code fait-on référence au fichier du point précédent ?

Modification de l'application

1. Changez le texte affiché sur l'écran et relancez l'application.
2. Ajoutez une deuxième ligne de texte à l'aide de l'outil de conception graphique d'écrans.
3. Observez le changement dans le xml (vue non graphique).
4. Dans la méthode *onOptionsItemSelected*, rajoutez un log comme par exemple :

```
Log.d("MonTag", "id="+id);
```
5. Relancez l'application.
6. Cliquez sur les trois points en haut à droite puis sur settings.
7. Observez le logcat.
8. Placez un point d'arrêt sur l'instruction que vous venez d'ajouter.
9. Relancez l'application en mode debug et recommencez la manipulation.

Plus sur Android Studio

1. Observez la fenêtre à gauche dans AS. Elle présente votre projet. Il existe différentes vues possibles pour cette fenêtre : Android, Project, Problems, Production, etc. Jouez avec ces vues.
2. Vous vous souvenez des makefile ? Tout projet qui se respecte a besoin d'un outil de construction. Eclipse utilise ant. Android Studio utilise Gradle.
Cliquez sur les fichiers build.gradle et regardez leur contenu.
3. Cliquez sur la Gradle Console en bas à droite et regardez son contenu.