

1 Mise en place

Le développement d'applications Android utilise principalement quatre composants :

- L'EDI Android Studio (l'environnement officiel actuel).
- Le SDK Android qui offre un accès aux bibliothèques existantes sous Android.
- Le compoant SDK manager qui permet de gérer les versions de sdk.
- La machine virtuelle ADT permet l'exécution des logiciels développés à fin de test.

Tous ces éléments peuvent être téléchargés et déployés en quelques étapes. Concernant l'IUT, cet environnement a été déployés dans le répertoire `/usr/local/android/studio` de votre PC.

Il a fallu cependant prendre certains détours, en particulier pour survivre à la limitation d'espace disque utilisé (quota) car cet environnement est très gourmand en espace disque. Heureusement, nous vous proposons un script qui automatise bon nombre de manipulations. Le détail de cet environnement est expliqué dans la sous-section Configuration.

1.1 Lancement

Lancez simplement la commande :

```
android.run
```

. Celle-ci configurera automatiquement un nouvel environnement.

Vous pouvez garder le workspace dans votre répertoire personnel afin qu'il soit accessible de tous les postes de l'IUT. Les fichiers correspondant sont beaucoup moins encombrants.

1.2 Configuration

L'environnement Android Studio suppose que les réglages de l'utilisateur sont dans un sous dossier du HOME directory de nom `.AndroidStudio`. Par ailleurs, le (ou les) dispositif(s) android émulé(s) sont également dans un sous-répertoire `$HOME/.android`. Ceci peut poser problème car l'ensemble est relativement encombrant (de l'ordre du Go) Nous avons donc imaginé de placer ce dossier dans des répertoires locaux (au PC) et d'établir un lien symbolique entre les nom attendus et ces répertoires. Si un utilisateur change de pc (lundi sur ayou01 et mardi sur ayou02 par exemple) il faudra donc recopier ces repertoires de l'ancien pc vers le nouveau. L'ensemble de ces opérations est intégré dans le script qui démarre l'environnement.

Quand l'environnement est lancé, nous vous demandons de vérifier un certains nombre de points de configuration :

- Options Configure / Setting de l'EDI : `http_proxy` : (Manual proxy configuration) `cache.univ-lille1.fr` :3128, port 3128.
- Option Configure / Project defaults / Project Structure
 - `sdk` : `/usr/local/android/studio/android-sdk-linux`
 - `jdk` : `/usr/lib/jvm/java-1.7.0-openjdk-amd64`
- Configuration / SDK Manager / Menu Tools/Options : HTTP Proxy Server : `cache.univ-lille1.fr`
- Configuration / SDK Manager / Menu Tools/Options : HTTP Proxy Port : 3128

2 Architecture d'un projet Android

Dans cette section, nous créerons un projet android, puis nous expliquerons ensuite la structure et les fichiers clés. Pour finir nous montrerons la création d'une machine virtuelle android à fin d'exécution du programme.

2.1 Génération du projet

Nous supposons que vous avez devant vous la fenêtre 'Welcome to Android Studio'

Effectuez les opérations suivantes :

- Start a new Android Studio project.
- Entrez les réglages suivants :

Application name *PremS*

Compagny Domain `α .iut-info.univ-lille1.fr` (où α est votre login utilisateur)

Project location un de vos répertoires personnels

- puis *Next*
- Choix de la cible :

Phone and Tablet (cochez).

Minimum SDK La valeur par défaut (4.0.3) convient. *Si vous souhaitez programmer sur votre téléphone personnel, vérifiez la compatibilité de la version.*

- puis *Next*
 - Configuration initiale : Blank Activity. (*Next*) puis gardez toutes les valeurs par défaut. (*Finish*)
- Le projet met quelques instants à se créer.

2.2 Structure

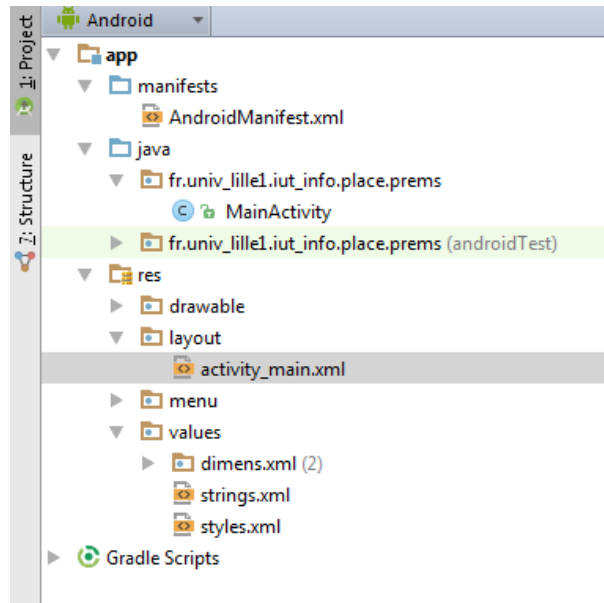


FIGURE 1 – Structure d'un projet Android

La structure du projet est représentée figure 1. Elle est normalement visible dans le volet de gauche. On y distingue les éléments suivants :

Le manifeste Il s'agit d'un fichier au format XML qui donne la description de l'application et des éléments nécessaires à son fonctionnement. On y trouve entre autres :

- Le nom de l'application (qui devrait être unique, d'où la construction impliquant le nom de domaine (i.e. *fr.univlille1.iutinfo.login*).
- Les versions Android supportées (mini, maxi,...)
- Les droits requis pour un fonctionnement correct de l'application (Accès internet, droit d'écriture sur carte SD, ...).
- La liste des composants de l'application (Activités, Services, Content Providers,).

Le code applicatif Java C'est ce code qui va donner l'essentiel des fonctions applicatives du logiciel. L'édition du code utilise l'environnement de développement traditionnel Java sous eclipse.

Les ressources Les ressources permettent à l'application de disposer d'un certain nombre d'éléments sans avoir à en intégrer la gestion physique. En outre les ressources peuvent exister en plusieurs versions (langage, résolution, ...). Ceci permet à un même programme de s'adapter automatiquement à son environnement d'installation. Une ressource peut représenter un chaîne de caractères, une image, un son, un menu, un design d'écran. L'édition d'un écran peut se faire soit d'une boîte de dialogue, soit directement sur le texte XML. Nous utiliserons aujourd'hui deux types de ressources : layout et valeurs.

Les écrans IHM (layout) Un écran IHM est une ressource. C'est un fichier au format XML qui décrit l'apparence, la disposition et les comportements qui serviront de cadre à l'interaction entre l'utilisateur en l'application. Ce cadre est appelé Activité (de fait une activité est composé d'un écran et de code java). L'édition d'un écran peut se faire soit au moyen d'un éditeur graphique, soit directement sur le texte XML.

Les valeurs Représentent les constantes chaînes de caractères manipulées par le programme. Le fait de déporter leur déclaration à l'extérieur permet une traduction (internationalisation soit i18n) plus rapide et plus sûre du logiciel.

Nous allons explorer point par point chacun des ces éléments pour construire une première application.

2.2.1 Le manifeste

Ouvrez le manifeste. L'écran présente alors le source XML qui décrit les caractéristiques générales de l'application. La version actuelle est assez pauvre (on y voit le nom unique de l'application et la description de l'activité initiale).

2.2.2 Les ressources

Ouvrez le fichier `strings.xml`. La liste des ressources présentes initialement apparaît :

- Le nom de l'application `app_name`.
- Le message de bienvenue `hello_world`.
- Le menu par défaut `action_settings`.

Dans ce contexte, vous pouvez changer la chaîne de caractère (changez "Hello world!" en "Bonjour!" par exemple) ou encore ajouter vos propres chaînes.

2.2.3 Le layout (activity_main.xml)

La ressource layout donne l'aspect et le comportement d'une activité un peu comme les formulaires sous Microsoft Access. Il se construit essentiellement de manière graphique (bien que l'édition XML soit permise). La configuration par défaut affiche juste un message ("Bonjour!") si vous avez modifié la ressource dans la sous-section précédente au centre de l'écran. Cette sous-fenêtre se structure comme suit :

- A Gauche, la palette des composants disponibles (un trainer-lâcher) permet de placer ces composants sur le layout. Les composants sont regroupés par catégories (Form Widgets, Text Fields, ...) qui se déplient/replient à la demande.
- En haut on peut choisir différentes définitions d'écran pour évaluer l'impact de la résolution sur le dessin. Le thème par défaut est aussi affiché. Choisissez le thème 'Holo.Light'
- A Droite la structure du layout. Les composants sont intégrés dans des layout (il en existent plusieurs sortes comme les `LayoutManager` de Swing) Dont l'imbrication détermine la géométrie du dessin. Par défaut le composant affichant le message d'accueil (`TextView` est placé seul à l'intérieur d'un `RelativeLayout`).
- En bas et à droite, les propriétés du composant sélectionné sont affichées et modifiables sur la partie inférieure droite de la fenêtre. On peut également ajouter/modifier une propriété par clic droit sur le composant proprement dit. Exemple : modifiez l'Id du `textView` pour mettre `Accueil`. Observez, dans le fichier XML (Onglet `Text` en bas de la sous-fenêtre), comment le description fait référence aux ressources id ou chaîne au moyen des expressions `@+id/ACCUEIL` et `@string/hello_world` (lisez l'infobulle du message Bonjour!)

2.2.4 R.java

Vous aurez compris que les équivalences entre les noms symboliques (comme `ACCUEIL`) et les valeurs entières réellement utilisées par le système et l'application sont créées puis compilées dans un fichier caché.

2.2.5 Le code source Java

Le code source de l'activité créée par défaut est simplement composé d'une classe dérivée de `Activity` et surchargeant la méthode `onCreate`. Cette méthode est automatiquement appelée par l'environnement lors de la création d'une activité. L'implémentation par défaut de cette méthode met seulement en place le layout désiré (spécifié comme nom de ressource (`R.layout.activity_main`)). Vous pourrez bien sûr enrichir cette classe par la suite de façon à obtenir un comportement plus intéressant.

3 L'environnement

3.1 Création de la machine virtuelle.

Les logiciels créés peuvent être testés/exécutés :

- Sur un dispositif android connecté via une liaison usb Il faut pour cela avoir installé le pilote correspondant sur l'ordinateur de développement.
- Une solution alternative est l'utilisation d'une machine virtuelle, certes un peu plus lente, mais qui permet de tout faire sur le même poste. L'environnement offre (dans le menu `Window` l'accès à `Android Virtual Device Manager` qui permet de créer et d'exécuter une machine virtuelle. Ce dispositif virtuel étant un processus autonome, la connexion se fait au moyen de la pile TCP/IP (port par défaut 5554).

Pratiquement :

- Lancer l'Android Virtual Device Manager `Tools/Android/AVD manager` ou dans la barre d'outil.
- Créez une machine virtuelle (via le bouton `Create a virtual device`. Tablet : Nexus 7 (2012).
System image : Lollipop (API 21) ABI : x86, Google API
- La machine virtuelle est créée (soyez patient) puis intégrée dans la liste qui apparaît.
- Demandez l'exécution de cette machine (par l'icône `Launch this AVD in the emulator - triangle vert vers la droite`. Cette étape peut être particulièrement longue (surtout la première fois à cause de la création des structures de données et de la carte SD).
- Quand l'écran d'accueil apparaît (enfin!), déverrouillez le dispositif. Vous pourrez conserver ce processus en arrière plan tout au long de votre session de travail.

3.2 Exécution du programme.

à ce niveau là du TP (vérifiez que vous avez bien l'écran d'accueil du dispositif android), vous pouvez demander l'exécution du projet.

- Retournez sous l'environnement Android Studio.
- Excitez la commande *Run app*. puis choisissez *Choose a running device*.
- Replacer la machine virtuelle au premier plan. Au bout de quelques/plusieurs secondes, vous verrez l'écran d'accueil de votre application sur le dispositif android.

4 exercice

4.1 Ajout des composants.

Ajoutez à votre layout initial les composants suivants :

- Un textfield (Person Name) d'ID "nom"
- Un small Button à droite du textfield précédent.
- Un large text d'ID "message" sous les deux précédents.

Exécutez ensuite votre projet.

4.2 Paramétrage des composants

- Changez la propriété "Hint" du composant 'nom' pour mettre la chaîne "Votre nom ?"
- Remarquez l'ampoule à gauche de la propriété. C'est pour vous signaler que cette chaîne n'est pas stockée comme ressource valeur et sera donc plus difficilement traduisible.
- en activant les points de suspension à droite de la propriété (puis bouton "new ressource", vous pouvez créer une ressource chaîne (nom : prompt, valeur : "Votre nom ?"). Vous procéderez ainsi pour toutes les chaînes de caractères.
- Effacez la propriété Text du composant "nom".
- Changez le texte du bouton pour "Ok"
- Changez le texte du LargeText par "Bienvenue"
- Changer la couleur du prompt à "#FF0000" (faites en aussi une ressource).

Vous devriez obtenir le résultat de la figure 2

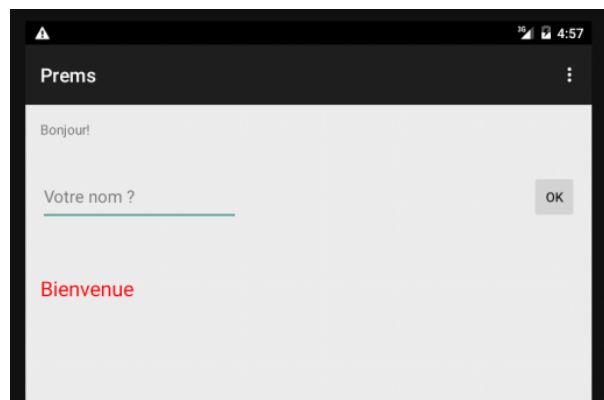


FIGURE 2 – Exécution de l'étape 1

4.3 Liaison entre le code Java et l'IHM.

Pour l'instant le bouton ne sert à rien. Les actions à effectuer lors de son activation pourront être spécifiées par du code Java.

1. L'appui sur un bouton peut déclencher l'exécution d'une méthode de l'activité.
2. Le java peut obtenir des références vers les composants visuel et les manipuler comme tout autre objet.

Pratiquement :

- Changez la propriété onClick du bouton par "doOk"

Insérez la méthode suivante dans le source *MainActivity.java* :

```
public void doOk(View view) {  
}
```

1
2

Cette méthode sera appelée lors de chaque appui sur le bouton.

- Ajouter les imports suivants :

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;
```

1
2

```
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
```

- Ajouter à la méthode, le code qui calcule le message de bienvenue :

```
public void doOk(View view) {
    TextView tvNom ;
    TextView tvMessage ;
    String msg ;
    tvNom = (TextView) findViewById(R.id.nom) ;
    tvMessage = (TextView) findViewById(R.id.message) ;
    msg = tvMessage.getText()+"_"+tvNom.getText()+" !" ;
    tvMessage.setText(msg);
}
```

Remarques :

- Les constantes repérant les composants sont importée depuis un package R (`R.id.nom` par exemple).
- La méthode `findViewById` permet d'obtenir une référence vers le composant désigné mais il faut imposer le type précis du composant.
- La documentation en ligne Android Developer vous donne toutes les informations sur ces méthodes.

4.4 Questions et exercices.

4.4.1 Exercice 1.

Appuyez plusieurs fois sur le bouton Ok. Que se passe t il? Corrigez.

4.4.2 Exercice 2.

Laissez en rouge le message initial, mais forcez par programme un texte bleu dans le composant `message`. Bonus :

- Créer une nouvelle ressource de type couleur de nom `color_changed` (bleu).
- Ecrivez les lignes de code permettant de récupérer par programme la valeur de cette couleur.
- Utilisez cette valeur dans la modification.

Quel avantage voyez-vous à cette construction?

4.4.3 Question 3.

Une fois le programme lancé, repassez à l'écran d'accueil (icône ronde en bas et au centre). Puis relancez le programme (icône en bas au centre puis choix de l'application Prems). Que voyez vous? Qu'en déduisez-vous?

4.4.4 Question 4.

Lors de l'exécution, appuyez sur la touche 5 puis 7 du pavé numérique^{1 2}. Que se passe t il? Quel impact obtient t on sur le résultat du programme?

4.4.5 Exercice 5.

- Ajouter un bouton qui réinitialise cette application :
- Texte du `message` en rouge et à Bienvenue.
- Zone `nom` mise à blanc.

4.4.6 Exercice 6.

Écrivez l'application qui accumule les valeurs saisies dans un champ de type `EditText` (entier signé) et affiche le résultat dans un champ `TextView`. L'ajout sera validé par un bouton.

Contrainte : la valeur cumulé sera mémorisée au moyen d'un attribut de la classe *Activity*

Note : Vous aurez besoin des méthode de conversion de chaîne de caractères en entier et vice-versa.

1. fn + ctrl gauche + F12 sur MacOS

2. Ceci revient à faire pivoter le téléphone.