

Introduction à Android

Version du 13 septembre 2020, 10:22

Objectifs

Le but de cette séance est de vous permettre de prendre en main l'environnement de développement Android et de réaliser une première application.

▷ Exercice 1 *Mise en place de l'environnement*

Dans le cadre des travaux pratiques vous pouvez soit utiliser votre ordinateur portable soit utiliser une machine de la salle de TP. Si vous utilisez votre ordinateur personnel, privilégiez l'IDE IntelliJ IDEA Pro et installez les API (à partir de la version 22).

Dans les salles des TPs :

- **IntelliJ** : Dans IntelliJ, créez un nouveau projet Android. Si ce n'est pas déjà fait, il vous installera la librairie Android. Pour cela dans "Platform SDK", donnez le chemin de la SDK qui se trouve dans : `/local-storage/`
L'IDE IntelliJ est installé sur les machines de TPs mais doit encore être configuré. Pour cela suivez les instructions suivantes :
 - Lancez IntelliJ à d'un terminal avec la commande `# intelliJ &`
 - (Si cela ne fonctionne pas, le script de lancement se trouve dans le répertoire `/local-storage/android/idea-IU-171.4694.70/bin/idea.sh`)
 - Lors du premier lancement, vous devez spécifier la licence que vous souhaitez utiliser. Choisissez **serveur de licence**
 - Donnez l'adresse `http://agate.unilim.fr:1111`
 - Normalement, l'IDE doit être enregistré à partir de cette étape
 - Lors du premier lancement il peut être également nécessaire de spécifier le chemin de la JDK.
 - Pour cela, avant de créer votre premier projet, cliquez sur *Configure* puis sur *Project Default -> Project Structure*.
 - Ajoutez ensuite une nouvelle JDK en spécifiant le chemin : `/opt/jdk1.8.0_131`
 - N'oubliez pas ensuite d'ajouter la SDK Android par défaut située dans `/local-storage/android/platforms/`
- Pour les autres séances vous devriez pouvoir retrouver IntelliJ dans le menu de la barre de tâche dans la section développement.
- **Android Studio** : Pour les personnes qui souhaitent installer Android Studio sur leur ordinateur personnel, suivez les indications de l'assistant d'installation et pensez à télécharger l'API 22 (mais également 26 et 28 si vous n'avez pas de problème de place sur votre machine.

Une autre solution pour avoir les mêmes conditions qu'en TP est de télécharger la version Pro d'IntelliJ. L'IDE IntelliJ est proposé en deux versions. Une version community (gratuite) qui permet de faire du Java ou de l'Android mais malheureusement pas de JEE (pour ceux qui ont pris l'UE). Pour cela il est nécessaire d'installer la version Ultimate. Celle-ci est normalement payante mais vous pouvez en tant qu'étudiant obtenir une licence gratuite en vous connectant à l'adresse <https://www.jetbrains.com/shop/eform/students>. Vous devez utiliser votre adresse unilim pour que cela fonctionne.

Une fois l'IDE configuré, il faut encore créer un émulateur pour pouvoir tester vos programmes. Pour cela, lancez l'Android Virtual Device Manager (AVD Manager) depuis le menu présent sur votre IDE (à vous de le retrouver en fonction de votre IDE :)).

- Cliquez sur le bouton **Create Virtual Device** ou **New** suivant la version d'AVD dont vous disposez
- Donnez un nom à votre Android Virtual Device
- Choisissez une petite résolution d'écran (240x432 est largement suffisant pour ce premier TP)
- Pour la RAM, 512M sont suffisants
- Ne cochez pas snapshot si la question vous est posée

Une fois votre émulateur configuré, **lancez le et laissez le fonctionner pendant toute la séance.**

Vous devez normalement disposer d'un environnement Android fonctionnel. Si ce n'est pas le cas appelez votre encadrant de TP.

▷ Exercice 2 *Hello World*

Reprenez l'exemple vu en cours et réalisez une application HelloWorld. Cette application dispose :

- d'une activité principale
- d'un bouton au milieu de l'interface qui permet de générer un **Toast** avec un message de votre choix

Modifiez ensuite l'exemple pour qu'un compteur soit incrémenté à chaque fois que l'utilisateur appuie sur le bouton.

▷ Exercice 3 *Nombre Aléatoire*

En reprenant l'exercice précédent, réalisez une application android permettant à un utilisateur de deviner un nombre entre 0 et 100 tiré aléatoirement au lancement de l'application.

Le cahier des charges est le suivant :

- L'application donnera des indications à chaque tentative pour le guide vers la solution (c'est plus ; c'est moins).
- Les indications et le résultat peuvent apparaître sous la forme de **Toast**

Pour réaliser les interactions utilisateurs vous aurez besoin de manipuler un champ texte. Pour l'affichage on privilégiera un **TextView** et pour récupérer les valeurs saisies par l'utilisateur un **EditText**.

Pour récupérer la valeur d'un widget, il est nécessaire d'y avoir accès dans votre code Java. Pour cela, il faut tout d'abord faire le lien avec le layout que vous avez défini dans le fichier XML. Par exemple, un `EditText` nommé `monEditText` dans le fichier XML (`android:id="@+id/monEditText"`) se retrouvera à l'aide du code suivant :

```
EditText champValeurSaisie = (EditText) findViewById(R.id.monEditText);  
int valUser = Integer.parseInt(champValeurSaisie.getText().toString());
```

▷ **Exercice 4** *Convertisseur de devise*

Proposez une nouvelle application toujours composée d'une seule activité qui permette d'effectuer une conversion de dollar en euros ($1\$ = 0.88\text{€}$).

Le cahier des charges est le suivant :

- Il devra y avoir au moins deux zones de texte, l'une éditable et l'autre servant à l'affichage du résultat
- Une liste avec deux boutons radios permettant de choisir le sens de la conversion. Regardez la documentation de `RadioButton` et `RadioGroup` pour réaliser ce widget.