

Fiche TD/TP n°2 : Prise en main Android

Objectif : Ce TP est une initiation à Android ADT que vous utiliserez au cours de ce module. Il a pour but de vous familiariser avec cet environnement et d'illustrer les concepts du SDK Android.

Exercice 1 :

- Lancez Eclipse.
- Dans le menu File > New, sélectionnez Android Application Project.
- Donnez le nom « MonApplication1 » à votre application.
- Utilisez le même nom pour le projet d'Eclipse.
- Spécifiez un package à utiliser, qui doit être unique. Ce package doit contenir au moins deux niveaux. Dans notre cas, on peut taper oran.monapplication1.
- Choisissez la version 4.2 (Jelly Bean) comme SDK cible, et 2.2 (Froyo) comme la version minimale sur laquelle l'application peut tourner. A noter qu'il faut éviter de remonter trop en arrière ça réduirait les fonctionnalités que vous pourriez donner à votre application.
- Cliquez sur Next et laissez les choix par défaut.
- Cliquez sur Next. La fenêtre suivante vous propose de définir une icône pour votre application. Nous laisserons l'icône proposée par défaut.
- Cliquez sur Next. Sélectionnez Blank Activity et cliquez Next.
- Donnez le nom « Principale » à votre activité et cliquez sur Finish, Un nouveau projet apparaît. □ On va maintenant configurer un périphérique Android Virtuel.
- Dans Window > Android SDK and AVD Manager cliquez sur New.
- Dans la nouvelle fenêtre, rentrez les informations suivantes : – Name : TP2 – Device: Nexus One – Target : Android 4.2.2 - API Level 17 – RAM : 728
- Cliquez sur OK puis sélectionnez le périphérique créé et cliquez sur Start.
- Allez dans le navigateur de fichier et ouvrez la classe **Principale.java** située dans le sous dossier oran.monapplication1 de src.
- Exécutez l'application en cliquant sur Run.

Remarque : Gardez l'émulateur ouvert (ou réduit) après l'avoir lancé. Lorsque vous testerez une nouvelle modification de votre application, Eclipse ne fera que d'installer la nouvelle version de votre application dans l'émulateur. Cela améliorera sensiblement la rapidité de vos tests.

Exercice 2 :

- On va essayer de changer le message qui apparaît lors de l'exécution de l'application. Dans le fichier *activity_principale.xml*, le message Hello world! apparaît dans la balise TextView. Essayez de modifier puis testez l'application. Qu'est-ce que vous remarquez ?
- Dans Android, "@string/" fait référence au fichier Strings.xml dans l'emplacement *res > values*. String.xml contient un fichier xml qui se réfère à une chaîne avec un id. Par exemple:

```
<string name="hello_world">Hello world!</string>
```

Ici « *name="hello_world"* » est l'id, et « Hello world! » est sa valeur. Lorsque "@string/hello" est utilisée, la valeur sera affichée.

- Changez la valeur puis testez votre application.
- On peut aussi changer le texte directement dans notre classe principale en utilisant la méthode : *setContentView* (View vue). Modifiez la classe principale en utilisant le code suivant :

```
TextView text = new TextView(this);  
text.setText("Bonjour, Android");  
setContentView(text);
```

- On peut aussi récupérer la chaîne de caractères « Hello world! » pour la manipuler, grâce à la méthode *public Resources getResources()*.
- Utilisez l'instruction suivante qui renvoie une chaîne de caractères, puis testez à nouveau l'application :

```
getResources().getString(R.string.hello_world);
```

Exercice 3 :

- On va à présent modifier l'icône de lancement de l'application. Aller au fichier de l'application dans le navigateur de fichiers, un clic droit, sélectionnez *new > Other > Android Icon Set*.
- Récupérez ou créez une icône. Placez la dans le répertoire *res* et modifiez le fichier *AndroidManifest.xml*.
- Testez à nouveau votre application.

Exercice 4 :

- Par défaut, seule la méthode **onCreate** est générée dans la classe principale. Ajoutez un affichage (en utilisant **System.out.println**) dans la console lorsque la méthode est appelée.
- Ajoutez les autres méthodes du cycle de vie d'une activité : **onStart**, **onPause**, **onStop**, **onResume**, **onRestart**, ou encore **onSaveInstanceState** (**Bundle outState**) et **onRestoreInstanceState** (**Bundle savedInstanceState**).
- Testez votre application.

➤ Par exemple pour la méthode *onStart*:

```
protected void onStart() {  
    super.onStart();  
}
```