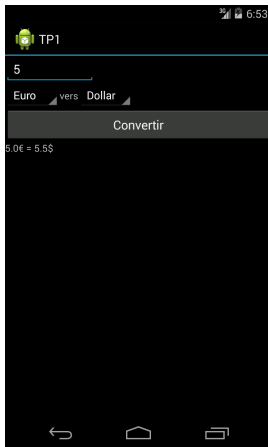


ANDROID - TP1

Source, pdf et corrigé de ce TP :
<http://tiny.cc/techmob>

Ce TP a deux objectifs :

- Se familiariser avec le SDK Android et l'IDE Android Studio
- Construire une petite application de conversion de devises



Il est fortement conseillé d'avoir le cours à portée de main et à ne pas hésiter à se référer à la documentation officielle <http://d.android.com> ainsi qu'à la multitude de tutoriaux disponibles.

1 Mise en place

1.1 Verifier l'installation

- Lancer Android Studio
 - Vérifier dans **Tools/Android/SDK Manager** que le chemin vers le SDK est bien renseigné
1. Quels sont les niveaux d'API disponibles ?

1.2 Hello world

- Créer un nouveau projet android via **File/New/New Project**
- On souhaite cibler plus de 95% des utilisateurs actifs d'Android. Choisir le bon niveau minimum d'API
- Notre application ne supportera pour l'instant que les téléphones et les tablettes
- Pour simplifier, choisir une Empty Activity

Android Studio génère automatiquement toute la structure d'un projet android gradleisé.

- Vérifier que la structure générée correspond à la structure vue dans le cours et que votre projet contient déjà une première activity
- Lancer l'application

2. Pourquoi l'application ne se lance t'elle pas ?

- L'Android Virtual Device manager permet de créer un émulateur afin d'y déployer votre application

L'AVD manager est aussi disponible via **Tools / Android / AVD Manager**

1.3 Créer un appareil virtuel

Un émulateur a déjà été créé sur vos postes

- Démarrer l'appareil
- Manipuler un peu l'émulateur
- Lancer l'application sur l'émulateur

3. Comment voir la log de l'émulateur ? (2 façons)

2 Construire notre convertisseur de devises

2.1 Logcat

- Logger chaque appel à onCreate() en niveau info
- Vérifier dans logcat que les messages apparaissent bien

2.2 Un peu d'IHM

- Simplifier le layout main pour utiliser un LinearLayout

Modifier le layout pour y ajouter quelques composants :

- Un champ de texte modifiable : montant (en euros)
- Un bouton : convertir

4. Que faut-il prévoir pour pouvoir manipuler ces vues en Java ?

2.3 Rendre l'interface vivante

- Écouter les clicks sur le bouton et déclencher un toast lors d'un click sur le bouton
- Dans le toast, afficher le contenu du champ de texte montant

Aide : la méthode getText() de EditText permet de récupérer le contenu du champ

2.4 Implémenter la logique métier

- Lors d'un click sur le bouton, récupérer la valeur numérique du montant

Aide : Double.parseDouble(String) permet de convertir une chaîne en nombre à virgule flottante

- Implémenter la règle métier compliquée suivante : 1€ = 1.1\$
- Afficher le résultat dans un toast
- Afficher le résultat dans une vue

5. Comment cacher la vue contenant le résultat tant qu'il n'a pas été calculé ?

6. Que proposez-vous si l'utilisateur n'a pas entré de montant ou un montant invalide ?

7. Que proposez-vous pour éviter que l'utilisateur ne rentre un montant invalide ?

2.5 Retravailler l'interface

- Choisir un ViewGroup simple et adapté pour disposer tous les éléments
- Mettre le bouton en dessous du champ de texte
- Modifier les attributs du Button pour lui faire prendre tout l'écran en largeur
- Ajouter un attribut "Hint" au champ de texte

2.6 En multidevise c'est mieux

On souhaite maintenant donner le choix à l'utilisateur entre plusieurs devises.

La vue correspondant à un menu déroulant est Spinner.

- Ajouter deux vues Spinner au layout de l'activity

Pour les Spinner basiques, on peut préciser les valeurs possibles en mettant un string-array dans l'attribut XML "entries" de la vue Spinner.

- Créer, dans strings.xml, un string-array contenant les noms de plusieurs devises
- Mettre ce string-array en attribut "entries" des Spinner
- En utilisant la méthode `getSelectedItemPosition`, récupérer les langues choisies par l'utilisateur
- Faire varier le taux de change en fonction des langues choisies
- En utilisant le fait que des ViewGroup peuvent contenir d'autres ViewGroup, positionner les spinner sur la même ligne

3 Aller plus loin

3.1 Un peu de style

- Changer le nom de l'application
- Changer l'icône de l'application

3.2 Traduire l'application

- S'assurer que toutes les strings ont été définies en XML
- Créer un dossier values-fr et y copier le fichier strings.xml
- Traduire le contenu de strings.xml
- Tester sur émulateur en changeant la langue dans le menu

3.3 Partager le résultat

- Ajouter un bouton : partager
 - Mettre une icône de partage à ce bouton (`setBackgroundResource`)
 - Lors d'un click sur le bouton, lancer un intent implicite pour partager le résultat
8. Que se passe-t-il si aucune activity n'est capable de répondre à l'intent ?