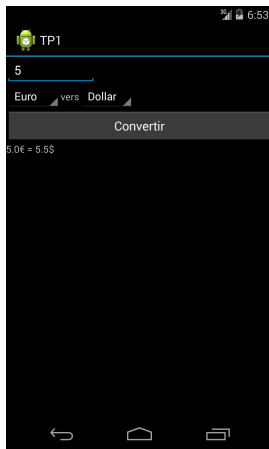


ANDROID - TP1

Source, pdf et corrigé de ce TP :
<http://goo.gl/rA09Sp>

Ce TP a deux objectifs :

- Se familiariser avec le SDK android et le plugin ADT pour eclipse
- 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 lire la documentation officielle <http://d.android.com> ainsi que la multitude de tutoriaux disponibles.

1 Mise en place

1.1 Verifier l'installation

- Lancer Eclipse
- Vérifier dans **Window/Preferences** la présence du menu android
- Vérifier que le chemin du SDK est bien rempli

1. Quels sont les niveaux d'API disponibles ?

1.2 Hello world

- Créer un nouveau projet android via File/New/Other/Android/Android Project. **Décocher “Create custom launcher icon” et choisir “Empty activity” plutôt que “Blank activity”**

ADT génère automatiquement toute la structure d’un projet android.

- Vérifier que la structure générée correspond à la structure vue dans le cours et que ADT a bien créé une activity basique
 - Lancer l’appliation
2. Pourquoi l’application ne se lance t’elle pas ?
- L’Android Virtual Device manager s’ouvre et permet de créer un émulateur

L’AVD manager est aussi disponible via **Window / Android virtual device manager**

1.3 Créer un appareil virtuel

- Créer un nouvel appareil avec le plus haut niveau d’API disponible
 - Démarrer l’appareil
 - Manipuler un peu l’émulateur
 - Lancer l’application sur l’émulateur
3. Comment voir la log de l’émulateur ?

2 Construire notre convertisseur de devises

2.1 Logcat

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

2.2 Un peu d’IHM

Modifier le layout main pour y ajouter quelques composants :

- Un champ de texte modifiable : montant en euros
 - Un bouton : convertir
4. Que faut t’il prévoir pour pouvoir manipuler ces vues en java ?

2.3 Rendre l'interface vivante

- Ecouter 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 entier

- 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 View-Group, positionner les spinner sur la même ligne

3 Aller plus loin

Attention : une partie de ce qui suit fait appel à des concepts qui seront abordés plus tard dans le cours.

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

7. Que se passe t'il si aucune activity n'est capable de répondre à l'intent ?

3.4 Créer un historique des calculs

8. Quelle solution de stockage de l'historique des calculs vous paraît la mieux adaptée ? (Rappel : on a le choix entre préférences, BDD SQLite et fichier plat)
 - Mettre en place la solution choisie
 - A chaque calcul, rajouter un élément à l'historique

3.5 Afficher l'historique

- L'historique sera affiché dans un écran séparé
 - Créer une nouvelle Activity qui contiendra l'historique (penser à la déclarer dans le Manifest)
9. Quel ViewGroup vous paraît le plus adapté pour afficher l'historique (nombre indéfini d'éléments) ?
 10. Comment lancer cette activité ?