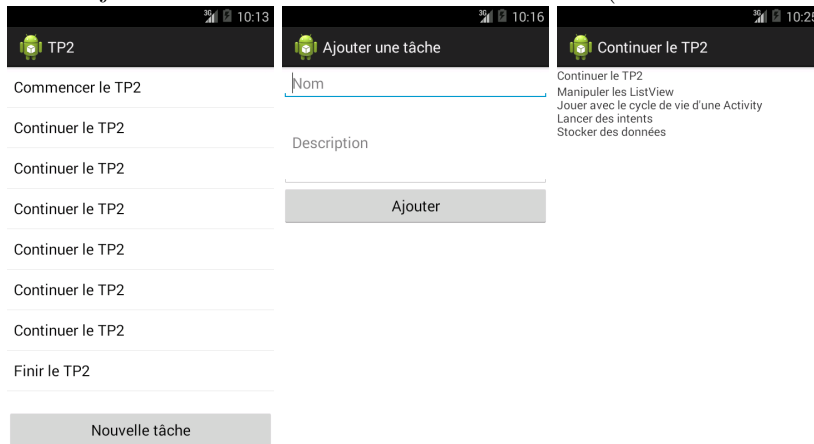


Android - TP2

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

L'objectif de ce TP est de réaliser une todo list (liste de choses à faire).



Ce TP fera appel aux notions suivantes :

- ListView
- Listeners
- Cycle de vie de l'activity
- Intents
- Stockage de données

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 L'écran principal : affichage des tâches

- Créer un nouveau projet android avec les mêmes paramètres que dans le TP1 (EmptyActivity et Android 4.4 comme API min, max, cible)
- Modifier le layout de cette activity pour y placer une ListView qui prendra toute la largeur et toute la hauteur

1. Quel composant est chargé de faire le lien entre les données et la ListView ? (on l'implémentera plus tard)

On souhaite ajouter un bouton "ajouter une tâche" en dessous de la ListView.

2. Pourquoi un LinearLayout "naïf" ne convient t'il pas ?

- En utilisant un RelativeLayout, placer le bouton en bas de l'écran
- Obtenir le même résultat en utilisant la propriété layout_weight de LinearLayout au lieu du RelativeLayout
- Ecouter les clicks sur le bouton et afficher un Toast à chaque click

2 Un écran secondaire : ajout d'une tâche

- Créer une deuxième activity
- Ajouter deux champs de texte modifiables (nom et commentaires) ainsi qu'un bouton "ajouter" à cette activity

3 Enchainement des écrans

3.1 Dans un sens ...

3. On veut lancer l'activity d'ajout d'une tâche lors d'un click sur le bouton "ajouter une tache" de l'activity principale. Quel concept android va-t-on utiliser ?
 4. Dans ce cas, est-il implicite ou explicite ?
 5. Pensez vous qu'il faille passer des données lors de l'appel à l'activity "ajouter une tâche" ? Si oui, lesquelles ?
- Lancer l'activity d'ajout de tâche lors d'un click sur le bouton.

3.2 ...et dans l'autre

Lors d'un click sur le bouton "ajouter" de l'activity d'ajout, on souhaite retourner à l'écran principal.

6. Est-il judicieux de lancer un intent vers l'activity principale ?

En réalité, android construit une pile des activités lancées (stack). Documentation officielle sur le stack

- En faire l'expérience en allant sur l'activity d'ajout puis en appuyant sur le bouton retour de l'appareil.
- Utiliser la méthode finish() de Activity pour fermer l'activity et ainsi revenir à l'écran principal lors d'un click sur le bouton ajouter.

4 Place aux données !

- Créer une classe métier destinée à contenir les données : Element. Chaque élément de la todo list contiendra au moins un nom et une description.

4.1 Afficher les données dans la ListView

- Dans onCreate, créer un ArrayAdapter utilisant pour chaque view le layout android.R.layout.simple_list_item_1
- Ouvrir le fichier XML de ce layout à l'aide de CTRL+click et constater sa simplicité
- Initialiser cet adapter avec une liste de Element générés à la volée
- Affecter cet adapter à la ListView
- Optionnel : compléter l'affichage de chaque élément en surchargeant la méthode getView

4.2 Stocker les données

7. Quelle solution de stockage des données vous paraît la plus adaptée ?

On va opter ici pour une base de données SQLite

- Étendre la classe SQLiteOpenHelper pour mettre en place la base
- La base contiendra une unique table éléments
- Lors de la création de la base, insérer quelques éléments dans la table
- Dans la méthode onCreate() de l'activité principale, remplir la ListView à partir des données contenues dans la base.

4.3 Remplir les données

- Lors d'un click sur le bouton "ajouter" de l'activité d'ajout, créer une instance d'Element et l'insérer en base.

8. Pourquoi la ListView n'est-elle pas mise à jour lors du retour sur l'activité principale ?

- Utiliser le cycle de vie des activités pour recharger les données de la ListView lors du retour sur l'activité principale.

5 Voir le détail d'un élément

- Créer une activité de visualisation d'élément avec les champs utiles (nom, description ...)
- Écouter les clicks sur les éléments de la ListView
- Lancer l'activité de visualisation lors d'un click sur un item de la liste
- Transmettre à l'activité le nom de l'item choisi
- Récupérer les données dans l'activité de visualisation et afficher les données de l'élément

6 Pour aller plus loin

Wow, déjà tout fait ? Voici quelques pistes d'améliorations :

- Ajouter, sur l'activité principale, un bouton "Exporter les données" qui génère un fichier contenant les éléments. Proposer à l'utilisateur de partager ce fichier.
- Les données associées aux items sont pour l'instant minimales, pourquoi ne pas ajouter un identifiant, une date d'ajout, un statut (en cours, fait ...), une priorité, une catégorie ... ?