ANDROID - TP2

Source et pdf du cours et de ce TP :
https://bitbucket.org/olevitt/technologies-mobiles/src

L'objectif de ce TP est de réaliser une todo list (liste de "choses à faire").
Ce TP fera appel aux notions suivantes :

- ListView
- RelativeLayout
- Intents
- Stockage de données

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 L'écran principal : affichage des tâches

- Créer un nouveau projet android avec une activity
- Modifier le layout de cette activity pour y ajouter 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 ListActivity ? (on l'implémentera plus tard)
2. On souhaite ajouter un bouton "ajouter une tâche" en dessous de la ListView. Quel layout proposez vous ?
3. Pourquoi un LinearLayout avec orientation "vertical" ne convient-il pas ?

- Ajouter le bouton en lui faisant prendre la place nécessaire en hauteur et toute la largeur.

2 Un écran secondaire : ajout d'une tâche

- Créer une deuxième activity (rappel : ne pas oublier de l'ajouter au manifest !)
- Ajouter deux champs de texte (nom et commentaires) ainsi qu'un bouton "ajouter" à cette activity (libre à vous de définir le layout)

4. Cette activity n'étant pour l'instant appelable que depuis l'écran principal, quels intent-filters proposez vous ?

3 Enchainement des écrans

3.1 Dans un sens ...

- Sur l'activity principale, récupérer les clicks sur le bouton "ajouter une tâche"

5. On veut lancer l'activity d'ajout d'une tâche lors d'un click sur ce bouton. Quel concept android va t'on utiliser ? (indice : il peut être soit implicite soit explicite)
6. Dans ce cas, est-il implicite ou explicite ?
7. 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.

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

- En réalité, android construit une pile des activités lancées (stack). Documentation officielle sur les stacks
- En faire l'éxpérience en lançant un intent vers l'activity principale depuis cette activity et en appuyant ensuite sur la touche 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 déstinée à contenir les données : Item. Chaque élément de la todo list contiendra au moins un nom et une description.

4.1 Afficher les données dans la ListView

9. Au vu de la simplicité des données à afficher pour chque élément (2 champs de texte), quel type d'adapter vous parait le plus intéressant ? (Rappel : on a le choix entre ArrayAdapter, CursorAdapter et faire sa propre implémentation à partir de BaseAdapter)

- Implémenter la solution choisie en utilisant un jeu de de données de test (initialiser les données à la volée)

4.2 Stocker les données

10. Quelle solution de stockage des données vous parait le plus adaptée ? (Rappel : on a le choix entre Preferences, fichier plat, base de données SQLite et stockage distant)

- Pour des raisons de simplicité, on va opter dans la suite pour un stockage en fichier plat (libre à vous de le remplacer par une base SQLite)
- Créer une méthode qui écrit une liste d'éléments Item dans un fichier (utiliser un format de stockage primitif par exemple nom#description et un item par ligne. (Ce format est un très mauvais choix en pratique car peu robuste et peu extensible mais ça suffira largement ici).
- Créer une méthode qui instancie une liste d'éléments Item à partir d'un fichier.
- Dans la méthode onCreate(), remplir la ListView à partir des données contenues dans le fichier.

4.3 Remplir les données

- Lors d'un click sur le bouton "ajouter" de l'activity d'ajout, créer une instance d'Item et la rajouter au fichier. (pour simplifier, réecrire la totalité du fichier à chaque fois)

11. Pourquoi la ListView n'est elle pas mise à jour lors du retour sur l'activity principale ?

- Utiliser le cycle de vie des activities pour recharger les données de la ListView lors du retour sur l'activity principale. (implémenter la bonne méthode onXXXXX)

5 Voir le détail d'un item

- Créer une activity de visualisation d'item avec les champs utiles (nom, description ...)
- Récupérer les clicks sur les éléments de la ListView (Attention à bien utiliser onItemClickListener et non onClickListener).
- Lancer l'activity de visualisation lors d'un click sur un item de la liste
- Transmettre à l'activity soit l'identifiant de l'item soit les données de l'item
- Récupérer les données dans l'activity de visualisation et afficher les données de l'item

6 Pour aller plus loin

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

- Le stockage en fichier plat n'est pas très judicieux. Mettre en place une base de données SQLite à la place.
- 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 ... ?
- L'affichage des éléments dans la liste est basique mais on peut difficilement faire mieux avec ArrayAdapter. Créer un adapter maison en héritant de BaseAdapter
- Partager les items par mail, sms, twitter ... en ajoutant un intent implicite dans l'activity de visualisation ou de modification.