Interface Homme/Machine 2

Rapport de projet Android – Hugo MOUGARD & Noémi SALAÜN

1. Cas d'utilisation

Gestions des listes de tâches

Consultation des listes

- Une ViewList permet un affichage rapide de l'ensemble des listes
- Une barre de progression indique l'avancement d'une liste en fonction des tâches qui la composent

Création d'une liste

- Le dernier élément de la ViewList est un bouton permettant d'ajouter une nouvelle liste
- Le menu du téléphone permet aussi d'ajouter une nouvelle liste

Suppression d'une liste

- Un appui long sur une liste permet d'accéder au menu contextuel, il est ensuite possible de supprimer la liste sélectionnée

Renommage d'une liste

- Le menu contextuel d'une liste permet aussi de renommer cette liste

Suppression de toutes les listes

- Le menu du téléphone permet de supprimer d'un seul coup toutes les listes

Gestion des tâches

Consultation des tâches

- Un clic sur un élément de la ListView contenant les listes de tâches permet d'ouvrir une nouvelle fenêtre similaire contenant uniquement les tâches de la liste sélectionnée

Création d'une tâche

- De la même façon que pour les listes, le dernier élément de la liste de tâches est un bouton pour créer une nouvelle tâche
- Une fois sur la page de consultation des tâches, le menu du téléphone permet aussi de créer une nouvelle tâche

Suppression d'une tâche

- Comme pour les listes, un appui long sur une tâche permet d'accéder à son menu contextuel, et ainsi la supprimer

Renommage d'une tâche

- Le menu contextuel d'une tâche permet aussi de la renommer

Suppression de toutes les tâches d'une liste

- Il est possible de supprimer l'ensemble des tâches d'une liste directement via le menu du téléphone

Validation d'une tâche

- Une boîte à cocher permet de valider chaque tâche individuellement

2. Exemple de scénarios complet

Utilisation d'une liste de course

Le principe d'une liste de course est de fournir un aide-mémoire juste le temps d'aller au magasin, sans avoir forcément besoin de garder la liste indéfiniment.

- Sur la fenêtre des listes, cliquer sur « Créer une liste… » puis valoriser son nom
- Cliquer sur la liste ainsi crée, pour consulter la liste de ses tâches
- Dans la fenêtre des tâches, cliquer sur « Créer une tâche… » puis valoriser son nom
- Il est possible de répéter cette étape autant de fois que l'on veut pour constituer sa liste
- Une fois un élément acheté, un simple clic sur la boîte à cocher correspondante permet de marquer la tâche comme étant effectuée
- Le bouton retour du téléphone permet de retourner à la fenêtre des listes, où l'on peut voir l'avancement de notre liste de course directement avec sa jauge de progression
- Une fois les courses terminées, il est possible de supprimer la liste correspondante par un appui long sur celle-ci

Utilisation d'une TODO-List

Le principe d'une TODO-List est un peu différente, il s'agit ici de fournir une liste de chose à faire, qui peut être conservée et complétée durant longtemps.

- La création de la liste se fait comme dans l'exemple précédent
- Lorsque l'on consulte la liste de tâche de notre TODO-List, on peut ajouter de nouvelle tâche à faire, ou invalider des tâches redondantes que l'on doit de nouveau faire
- Dans le cas où l'on voudrait préciser une tâche, il est possible de la renommer grâce à un appui long sur celle-ci
- A la fermeture de l'application, l'ensemble des listes est enregistré sur le téléphone, il n'est donc pas obligatoire de compléter sa TODO-List d'un seul coup
- A la réouverture de l'application, on retrouve ainsi la liste de nos tâches à effectuer

Réinitialisation de l'application

Ce scénario n'est pas très complexe, mais il peut être intéressant

- Sur la fenêtre contenant les listes de tâches, un clic sur le bouton « Menu » du téléphone permet d'effectuer l'action « Supprimer tout », ce qui aura pour effet de supprimer l'ensemble des listes enregistrées
- A noté qu'il est aussi possible d'effectuer cette opération lors de la consultation des tâches d'une liste particulière. La suppression ne portera alors que sur le contenu de cette liste.

3. Implémentation

Les classes Java

Package activity

DomainsActivity

Il s'agit de l'***Activity*** principale qui se lance avec l'application. Elle permet de consulter la liste des listes de tâches.

Son cycle de vie est standard, mis à part le ***onResume()*** qui s'occupe de charger les données enregistrées avec la persistance. Ce qui permet de mettre à jour l'affichage de la progression d'une liste après avoir modifié ses tâches dans une autre ***Activity***.

TasksActivity

Il s'agit de l'***Activity*** qui est appelée pour afficher la liste des tâches composant une liste.

Son cycle de vie est complétement standard, et l'application s'ouvre correctement sur cette ***Activity*** après avoir été mise en pause.

Package adapter

DomainsAdapter

Il s'agit d'un ***BaseAdapter*** modifié pour gérer l'affichage de la liste de listes en se basant sur les classes d'entités. C'est lui qui instancie le design d'un élément de la liste pour le valoriser correctement avec le nom et la progression d'une liste. Il gère aussi le cas particulier du dernier élément de la liste qui correspond au bouton d'ajout d'une nouvelle liste.

TasksAdapter

De la même façon, il s'agit d'un ***BaseAdapter*** modifié pour gérer l'affichage d'une liste de tâches. Il s'occupe d'afficher le nom d'une tâche ainsi que sa boîte à cocher correctement valorisée. Le fait d'utiliser un ***Adapter*** permet de gérer automatiquement la mise à jour de l'affichage en cas de changement de valeur, il suffit pour cela d'appeler sa méthode ***notifyDataSetChanged()***. Le TasksAdapter s'occupe aussi d'instancié le ***Listener*** des boîtes à cocher.

Package entity

Les classes d'entités correspondent à la modélisation des éléments que l'on manipule. Il s'agit dans notre cas des tâches (***Task***), des listes (***Domain***) et de l'ensemble de nos listes (***Data***). Grâce aux ***Adapters***, l'affichage est ainsi directement calqué sur les valeurs des entités.

Package persistence

Il s'agit des classes s'occupant de sauvegarder sur le téléphone l'état de nos entités. Une interface Java ***PersistenceManager*** comprend uniquement les méthodes ***Load()*** et ***save(data)***. Nous avons ensuite choisi d'en faire une implémentation simple en enregistrant les données dans un fichier texte. En travaillant de cette manière, il serait extrêmement aisé d'implémenter un nouveau gestionnaire travaillant sur une base de données ***SQLite*** par exemple.

Pour éviter d'avoir des soucis de perte de données, par exemple si le téléphone s'éteint, la persistance est faite en temps réel à chaque modification des données.

Les fichiers de design XML

Pour les designs, la logique est la même pour les listes et pour les tâches.

Les layouts

- **Layout principal :** Il gère l'affichage de la fenêtre complète
- **Layout d'un élément :** Il gère l'affichage d'un élément de la ***ListView***
- **Layout du bouton :** Il gère le bouton d'ajout d'un nouvel élément dans la ***ListView***

Les menus

- **Menu contextuel :** Il s'agit du menu ouvert par un appui long sur un élément
- **Menu d'options :** Il s'agit du menu ouvert avec le bouton ***Menu*** du téléphone

4. Les fichiers sources

Un dépôt GitHub contient l'ensemble des fichiers nécessaire à notre projet :

`git://github.com/AtalM1/tachoid.git`