Rapport de Projet

Projet : Gestionnaire simple de tâches

Taskinator Android

COUTABLE Guillaume, RULLIER Noémie
2 avril 2013

UNIVERSITÉ DE NANTES

Table des matières

1 Introduction

L'objectif de ce projet fut de développer un gestionnaire simple de tâches. Celui-ci devait permettre de créer des listes de tâches et de suivre l'avancement de celles-ci.

Afin de créer cette application que nous avons appelé *Taskinator*, nous avons établit plusieurs étapes dans l'avancement du projet. Ce rapport présentera ces étapes les unes après les autres.

2 Les fonctionnalités

La première étape fut d'analyser l'ensemble des fonctionnalités que notre application devait proposer.

2.1 Fonctionnalités principales

Voici dans un premier temps les fonctionnalités principales :

Créer une liste : cette fonctionnalité permet à l'utilisateur de créer une liste vide.

Créer une tâche : cette fonctionnalité permet à l'utilisateur de créer une tâche.

Supprimer un élément : cette fonctionnalité permet de supprimer une tâche ou une liste. Cette fonctionnalité est à manipuler avec précaution, en effet dans le cas d'une liste, la suppression de celle-ci implique aussi la suppression de toutes ses tâches.

2.2 Fonctionnalités secondaires

Voici les fonctionnalités secondaires :

Paramètre : cette fonctionnalité permet à l'utilisateur de modifier le type de l'élément sélectionné. Il pourra par exemple choisir de modifier une liste en liste ordonnée ou en une tâche. Cette fonctionnalité est à manipuler avec précaution, en effet si l'utilisateur décide de transformer une liste en tâche l'ensemble des éléments de la liste seront supprimés.

Monter / Descendre : cette fonctionnalité permet de monter ou descendre un élément dans l'arborescence de la liste. Dans le cas d'une liste, toutes ces tâches sont aussi montées/descendues d'un rang.

Aperçu : cette fonctionnalité permet à l'utilisateur d'avoir un aperçu de sa liste, se qui permettra l'affichage d'un nombre plus important de tâches.

3 Storyboard, *paper prototyping*et Scénarios

3.1 Storyboard

Le storyboard permet de montrer à quoi sert l'application. Il utilise les star people de Bill VerPlank. A la fin su storyboard, le personnage atteint son but et est satisfait. Nous avons donc ici créé notre storyboard pour notre application :

FIGURE 1 – Le storyboard

3.2 Scénarios

Nous avons imaginé différents scénarios d'utilisation de notre application.

3.2.1 Scénario 1 - Création/Utilisation/Suppression d'une liste

Ce premier scénario permet de créer une liste et d'y ajouté des tâches.

1. L'utilisateur crée une liste et lui donne un nom *Ski*.
2. Il crée ensuite une tâche dans cette liste et lui donne un nom *Bonnet*.
3. Il crée ensuite une autre tâche dans cette liste et lui donne un nom *Echarpe*.
4. Il souhaite maintenant échanger les tâches *Bonnet* et *Echarpe*, il reste longtemps appuyé sur la tâche *Bonnet* et la glisse un rang plus bas.
5. L'utilisateur supprime la tâche *Bonnet*.
6. L'utilisateur supprime la liste *Ski*. Une popup apparaît pour l'avertir que cette suppression supprimera aussi toutes les tâche de la liste.

3.2.2 Scénario 2 - Gestion des listes et sauvegarde

Ce scénario permet de créer des listes et de les modifiers. Il permet aussi de constater que l'état de l'application est sauvegardé.

1. L'utilisateur choisit de créer une liste et lui donne un nom *Course*.
2. Il décide ensuite de créer une nouvelle liste et lui donne un nom *Sac de voyage*.
3. Il décide ensuite d'inverser ces deux listes, pour cela il reste longtemps appuyé sur la liste *Sac de voyage* et la glisse un rangplus haut.
4. L'utilisateur va ensuite supprimer la liste *Course*.
5. L'utilisateur va ensuite ouvrir une autre application et revenir sur *Taskinator*. Il peut constater que l'application est ouverte avec l'état dans lequel on l'a quittée.
6. Cette fois-ci, l'utilisateur quitte l'application. De même s'il l'a lance à nouveau, celle-ci est ouverte avec le dernier état dans lequel on l'a quittée.

4 L'IHM

4.0.3 Choix réalisés (Gestures, ...)

Dans cette section, nous expliquons les choix effectués et pourquoi nous avons choisi de mettre en place ces solutions :

- **Fonctionnalité de création (liste et tâche) :** nous avons décidé d'adopter un système de sélection de l'élément "fils". Expliquons ce système. Lorsque l'utilisateur créé une liste, un élément vide de cette liste apparaît. L'utilisateur doit alors sélectionner cet élément et définir de quel type l'élément sera. Lorsque cet élément à un type, un nouvel élément vide apparaît ; l'utilisateur peut alors définir un autre élément et ainsi de suite. Si l'utilisateur ne souhaite pas continuer à remplir cette liste, il ne tient plus compte des éléments vides.
- **Fonctionnalité *Monter* et *Descendre* :** nous avons décidé de placer ces fonctionnalités dans la boîte à outils plutôt que sur l'élément à déplacer pour un gain de temps au niveau de l'utilisation. En effet, si nous avions fait le choix de mettre les boutons *monter* et *descendre* sur l'élément, lorsque l'utilisateur aurait déplacé l'élément, il aurait perdu le focus de la souris sur ces boutons. Attention, lors de l'utilisation de cette fonctionnalité dans une liste ordonnée ; si on monte ou descend un élément de tel façon qu'un élément coché se trouve être en dessous d'un élément non coché dans la liste, alors le déplacement ne sera pas activé. (Cette dernière précaution n'a pas été implémentée)
- **Fonctionnalité *Supprimer* :** nous avons choisi d'ajouter cette possibilité de suppression sur chaque élément pour que l'utilisateur puisse supprimer plus rapidement. De plus nous avons décidé de faire apparaître une fenêtre d'avertissement car la suppression d'une liste sélectionnée peut entraîner la suppression d'éléments. Pour plus de sécurité, le focus de la validation de la suppression est par défaut sur *Non*.
- **Fonctionnalité *Paramètre* :** nous avons décidé pour cette fonctionnalité (qui peut entraîner la suppression de différents éléments) de faire, également, apparaître une fenêtre d'avertissement. Dans cette fenêtre le focus de la validation du changement est mise à *Non*. On perd cependant un peu de vitesse d'exécution (si l'utilisateur souhaite effectivement bien effectuer ce changement) au profit de plus de sécurité.

L'application est tout d'abord ? ? ?composée d'un menu, d'une boite à outils et d'une zone réservée à l'affichage de la liste en cours de création.

FIGURE 2 – La fenêtre principale

4.1 Affichage des listes et tâches

Afin d'afficher l'ensemble des listes créées et de leur(s) tâche(s), on a décider d'utiliser une/-deux ? ? ? ? ? ?ListeView. La première permettant d'afficher l'ensemble des listes et une pour chaque liste permettant d'afficher sa ou ses tâche(s). La tâche est représentée par un ? ? ? qui est composé d'une checkbox, d'un EditLine permettant de taper le nom de la tâche et d'un button permettant d'ajouter la fonctionnalité de suppression sur la tâche.

5 Le modèle

Afin de réaliser cette application nous nous sommes basés sur le modèle suivant dont voici le diagramme de classes :

Le modèle de cette application est composé d'un pattern composite permettant d'implémenter une structure d'arbre et de composer les différents objets ensemble. Nous avons donc ici un *Component* qui peut être, soit une *Task* (une tâche), soit une *List* (une liste) qui elle-même peut ensuite être une *SortedList* (une liste ordonnée), celle-ci hérite de la classe *List*. Ce modèle permet donc comme expliqué ci-dessus de composer ces éléments et d'obtenir par exemple des listes de listes de tâches ...Nous pouvons donc gérer tous ces composants au sein de ce modèle et ainsi, ajouter un composant dans une liste, le supprimer, le déplacer d'un rang (dans les deux sens). Nous disposons aussi de toutes les fonctions permettant de déterminer si un composant est cochable ou non.

6 Limite de l'application

Notre application à ses propres limites.

7 Conclusion générale

Ce projet nous a permis de réfléchir à la structure de l'IHM afin que celle-ci soit la plus intuitive possible pour l'utilisateur. Réfléchir à la position des boutons, le nom ou l'icône les représentant, comment l'ajout des composants se fera au sein de la liste, la gesture à utiliser, l'utilisation d'un menu ... Et donc, d'orienter notre conception sur l'IHM, plus que sur le modèle, et ainsi expérimenter une autre approche de conception.