

IHM - Rapport Labo Pebble

Ce document constitue le rapport du laboratoire II du cours d'IHM portant sur la Pebble.

Introduction

Le but de ce laboratoire est de développer une application permettant à l'utilisateur d'afficher sur une Pebble différents types de données provenant du smartphone associé.

Utilisation

Cette partie décrit comment utiliser l'application.

Mode de lecture

Ceci est la vue par défaut au lancement de l'app. Elle présente un écran composé de deux parties. Les deux parties sont similaires et présentent les valeurs que l'utilisateur souhaite afficher. Par défaut, un texte explique à l'utilisateur comment éditer la section de l'écran.

En utilisant les boutons *haut* et *bas* de la Pebble, l'utilisateur peut naviguer d'écran en écran. Cela lui permet d'avoir 3 autres écrans de données vers lesquels il peut aisément changer.

En cliquant sur le bouton de *sélection*, l'utilisateur peut passer en **mode de sélection**.

Mode de sélection

Dans ce mode, l'utilisateur peut choisir quel section il va éditer par la suite.

À l'aide des boutons *haut* et *bas* de la Pebble, il peut changer la section qu'il souhaite éditer, puis, à l'aide du bouton de *sélection*, il valide son choix.

Le bouton *retour* permet de revenir au **mode de lecture**.

Mode d'édition

Ce mode affiche un menu permettant à l'utilisateur de changer le type de donnée à afficher sur la section en cours d'édition.

Il s'agit d'un simple menu composé de 5 sections:

- Location
- Weather
- Transport
- Time
- Battery

Choix

Durant le développement de cette application, nous avons relativement peu de choix à faire. De plus, nous avons préféré garder un aspect minimaliste à l'application afin de ne pas perdre l'utilisateur dans des séries de menus et sous-menus.

Navigation

Le fait d'avoir 3 modes d'affichage permet à l'utilisateur de facilement savoir ce qu'il est en train de faire, quel mode il est en train d'utiliser.

La navigation d'écran en écran se fait à l'aide des boutons *haut* et *bas* de la Pebble. Nous trouvions cela relativement intuitif de cette manière.

En effet, même s'il n'y a pas d'indicateur direct de l'écran sur lequel on se situe, l'utilisateur est largement capable de se souvenir sur lequel il se trouve. De plus, cela n'a pas une grande influence sur son utilisation.

Sélection de la section à changer

La sélection de la section à changer se fait à l'aide des boutons *haut* et *bas* de la Pebble. Quand la section du bas est pré-sélectionnée elle son texte devient blanc sur fond noir. Cela respecte ainsi les conventions des menus de la Pebble. On évite alors de perdre l'utilisateur, car il est déjà habitué à ce comportement.

De plus, si la section du haut est pré-sélectionnée et que l'on appuie sur le bouton *haut*, la pré-sélection va sur la section du bas. On évite alors que certains boutons ne servent à rien et beaucoup d'utilisateur sont habitués à pouvoir "boucler" à travers les menus.

Menu

La séparation du menu en diverses sections permet une meilleure expérience à l'utilisateur qui saura s'y retrouver plus facilement. Les différentes catégories qui le compose ont été créées intuitivement.

Rafraichissement des données

L'application comporte 8 sections d'écran qui peuvent comporter des données pouvant potentiellement changer.

Le dilemme auquel nous avons été confronté est le suivant: doit-on aussi rafraichir les données qui ne sont pas actuellement affichées (dans les 6 autres sections) ou doit-on attendre que l'utilisateur change d'écran?

En effet, la première solution est plus gourmande en batterie, mais les données sont lisibles instantanément au changement d'écran. Alors que la deuxième solution économise un peu de batterie mais peut faire attendre l'utilisateur pendant plus de 5 secondes (ce qui peut être désagréable pour son expérience).

Bugs connus

Nous n'avons pas décelé de bug à proprement parlé, mais notre application comporte deux interactions involontaires ou indésirées.

Menu

Nous avons voulu faire en sorte que lors de l'édition d'une section, la sélection par défaut du menu soit identique au type de donnée courant.

Par exemple, si la section actuelle est **Uptime**, lorsque l'on tente d'éditer la section, le menu pré-sélectionne déjà **Uptime**.

Or, la fonction `simple_menu_layer_set_selected_index` de l'API de Pebble ne permet de pré-sélectionner que dans la première section du menu et pas au-delà. La pré-sélection sera alors sur le dernier élément de la première section (dans notre cas: **Elevation**).

Start/Stop Location Thread

L'API et les fonctionnalités **Start/Stop Location Thread** n'étant pas précisément documentées, et n'ayant pas de téléphone pour tester, nous ne savions pas ce que le téléphone allait répondre à ces fonctions. Il s'avère que nous ne devrions pas attendre de réponse.

Conclusion

Ce laboratoire nous a permis de faire une application sur smartwatch, ce qui était une toute première pour nous. Nous avons pu découvrir le SDK et les outils de développement et de déploiement de Pebble.

Le fait de travailler sur un écran aussi petit que celui de la Pebble peut s'avérer être un véritable challenge. En effet, il n'est pas possible d'écrire beaucoup de texte, et même si c'était le cas, ce n'est pas un très bon support de lecture. Il faut donc faire attention aux informations que l'on va afficher.

Étant donné que l'on ne peut pas afficher beaucoup d'information, il est d'autant plus important de respecter les différentes conventions et d'utiliser les différents modules établis par Pebble.

Annexes

Le code source se trouve à l'adresse suivante: <https://github.com/BinaryBrain/Pebble>