

L'Obedix

<https://github.com/oumar-fall/lobedix>

Présentation de l'application

Cette application a été réalisée par :

- Oumar Fall
- Domitille Dommergues
- Julie Keisler
- Marion Candelier

Elle a été codée sur Android Studio.

Elle a pour but de remplacer la carte papier d'un restaurant par une carte numérique plus interactive. Bien que nous souhaitions au départ réaliser cette application pour tablette, elle est aujourd'hui adaptée aux smartphones (plus facile à tester pour nous dans le temps imparti).

Différentes fonctionnalités ont été implémentées :

- La carte classique
- Les menus
- Le mode coup de pouce
- Le mode enfant (plus ludique et avec des plats adaptés)
- Des informations précises sur chacun des plats proposés : boisson pour accompagner, valeurs nutritionnelles...
- Barre de recherche fonctionnelle

En lançant l'application, le premier choix que vous avez à faire est de choisir l'environnement dans lequel vous souhaitez évoluer (enfant/adulte), puis le choix de l'apéro arrive. Pour ces 2 activités il est possible de swiper afin d'accéder aux pages suggérées sur l'écran (sinon vous pouvez aussi cliquer sur le petit rond correspondant en bas de l'écran). Enfin vous arrivez sur le menu général qui vous permet de choisir entre les menus, la carte classique, le coup de pouce (choisir un plat à la façon de tinder), la barre de recherche (afin d'accéder plus rapidement au plat souhaité).

Pour valider la commande il faut aller dans le menu situé en haut à gauche.

Nous n'avons pas implémenté le paiement.

L'apéro n'est pas implémenté jusqu'au bout (nous avons fait juste le visuel par manque de temps, un manque de temps en partie dû au manque d'automatisation de l'application)

Il y a certainement des petits bugs mais nous avons fait notre maximum pour en éliminer un maximum (de toute façon pendant le confinement les restaurants sont fermés, il nous reste encore un peu de temps pour corriger ça ;)).

Organisation du code :

Nous avons mis dans le dossier ressource (fichier strings.xml) la plupart des textes utilisés afin de faciliter l'extension ainsi que la maintenance de l'application.

Le fichier margins.xml contient toutes les marges utilisées dans nos layouts afin de pouvoir les modifier simplement mais également afin de pouvoir garder une certaine cohérence entre les différents layouts.

Le fichier sizes.xml contient les informations relatives aux dimensions des différentes objets dans l'application (les textes et les images notamment). Il permet également de garder une certaine cohérence visuelle des différents layouts.

Une classe DataHolder permet de garder toutes les informations concernant la commande du client.

Cependant, l'ajout de plats supplémentaire reste compliqué car nous n'avons pas assez généralisé les différentes activités de l'application (ce qui explique le nombre important d'activités créées). Nous aurions pu par exemple utiliser des recycler view pour gérer la liste des plats ...

Pour chaque plat 4 activités ont été créées :

- Dish_Information_Nomduplat (contient les informations essentielles sur le plat)
- Dish_Informations_Nutri_Nomduplat (contient les informations nutritionnelles du plat en question, on y accède en cliquant sur le nutriscore de la page Dish_Information_Nomduplat)
- Ingredient_Nomduplat (contient les ingrédients du plat en question, on y accède depuis la page Dish_Information_Nomduplat)
- Boisson_Nomduplat (conseille la boisson à boire avec ce plat, on y accède depuis la page Dish_Information_Nomduplat)

Le coup de pouce est divisé en 11 classes différentes. Le but est de pouvoir accéder à tous les plats en répondant à des questions. Chaque classe correspond à une question. L'ajout de plats supplémentaires nécessiterait également de revoir l'ensemble de la structure du coup de pouce, qui repose entièrement sur les plats proposés de la carte. Néanmoins on peut facilement voir comment généraliser cette méthode avec une carte de restaurant classique, quitte à proposer plusieurs choix de plats après une série de réponse, au lieu de juste un seul comme nous avons voulu faire ici. Cette méthode du choix du plat façon tinder permet de donner des idées aux indécis.

L'interface enfant nécessite un prétraitement de toutes les images afin que celles-ci représentent des pièces de puzzle. Nous avons essayé de rendre ce traitement dynamique mais en vain.

Objectifs non atteints :

Nous avons pour objectif d'implémenter également le changement de langue. L'idée initiale était d'utiliser le fichier strings.xml et d'en changer son contenu. Nous nous sommes finalement rendus compte que les valeurs renseignées dans les fichiers XML annexes étaient en fait encodées "en dur" et qu'il était impossible de les modifier directement depuis le code.

Description de l'interface :

Comme voulu dans le cahier des charges, nous avons essayé de faire l'interface la plus sobre et intuitive qui soit. Nous nous sommes volontairement appuyées sur des modèles d'interfaces bien connues du grand public : tripadvisor pour la présentation des

plats, Yuka pour les valeurs nutritionnelles et tinder pour les swipes au début de l'application et pour le coup de pouce. Le but recherché est que l'utilisateur se sente en confiance avec notre application et trouve rapidement ses repères. De plus, ce système nous permet de ne pas surcharger l'application d'informations.

Le swipe que nous avons fait a été limité par nos compétences techniques et ne reflète pas vraiment ce que nous avions en tête.

En ce qui concerne l'interface enfant, celle-ci est relativement fidèle à l'idée de départ. Il s'agit d'un espace où les plats proposés sont adaptés au contenu du menu enfant. L'interface est également plus simple, avec moins de possibilités. La manière de composer le menu est ludique, sous forme d'un puzzle à recomposer (chaque pièce correspond à une partie du menu). On y trouve une zone délimitée en pointillés indiquant où placer les pièces. Ces dernières sont "aimantées" à cette zone : si on la relâche à proximité de la bonne position, elle se replace directement alignée sur celle-ci.