



Réalisation d'une application de gestion de ToDoList

- ToDouDou -

Introduction

L'objectif de la séquence est de créer une application permettant une gestion basique de ToDoLists. L'application doit permettre à l'utilisateur de créer un pseudo, de créer des listes et d'y ajouter des tâches. L'utilisateur doit pouvoir indiquer les tâches réalisées ou non. Enfin l'application doit garder en mémoire les changements effectués par l'utilisateur.

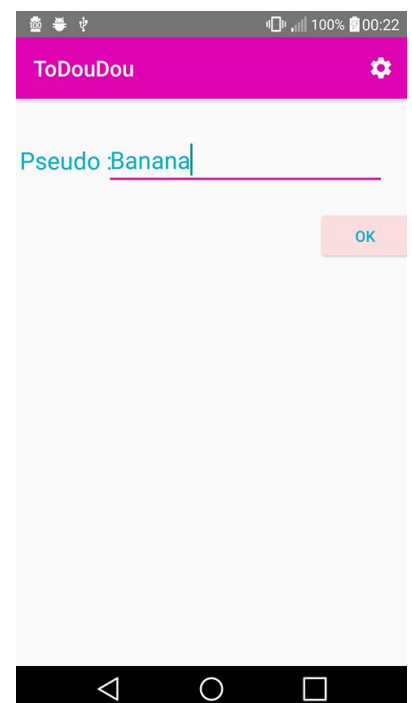
Analyse

Fonctionnalités

✓ Pseudos :

La création de pseudo se fait sur l'activité principale `MainActivity`.

Celle-ci présente un champ `EditText` où l'utilisateur doit renseigner un pseudo, un bouton `OK` pour valider le pseudo, ainsi qu'une icône d'engrenage qui permet d'accéder à la vue `SettingsActivity`. Si le pseudo existe déjà dans les données de l'application, lorsqu'il clique sur le bouton `OK`, l'application passe à la vue `ChoixListActivity` qui affiche les `ToDoList` déjà créées avec ce pseudo. Sinon, il enregistre ce nouveau pseudo et la vue `ChoixListActivity` n'affiche aucune `ToDoList`.

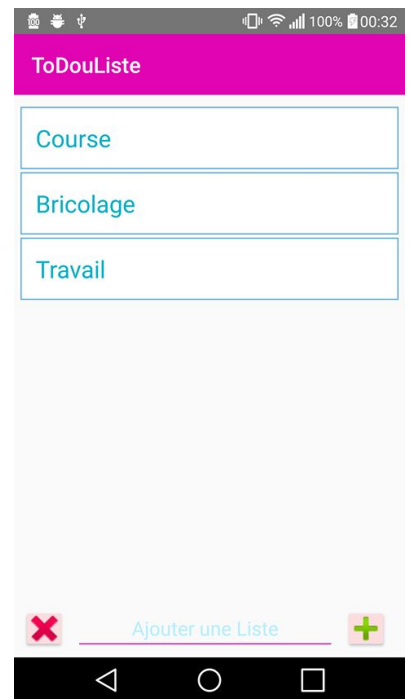




✓ **ToDoList :**

La gestion des ToDoList se fait sur l'activité `ChoixListActivity`.

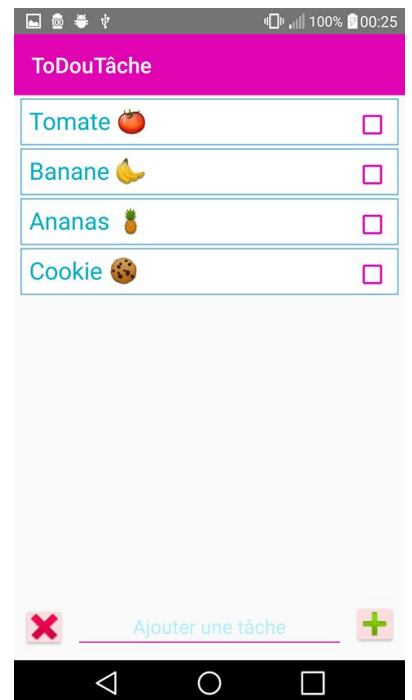
Celle-ci présente une liste (éventuellement vide) de noms de listes (objets `ListeToDo`) affichés dans un `RecyclerView`, un bouton en croix rouge permettant d'effacer toutes les listes et de retourner à la `MainActivity`, un champ `EditText` permettant de renseigner un nom de liste et un bouton plus vert permettant de créer une liste ayant pour nom la chaîne de caractères dans le champ `EditText` et une liste de tâches (objets `ItemToDo`) vide. Cliquer sur l'un de ces noms de liste permet de passer à l'activité `ShowListActivity` et d'accéder aux tâches créées dans cette liste.



✓ **Tâches :**

La gestion des tâches se fait sur l'activité `ShowListActivity`.

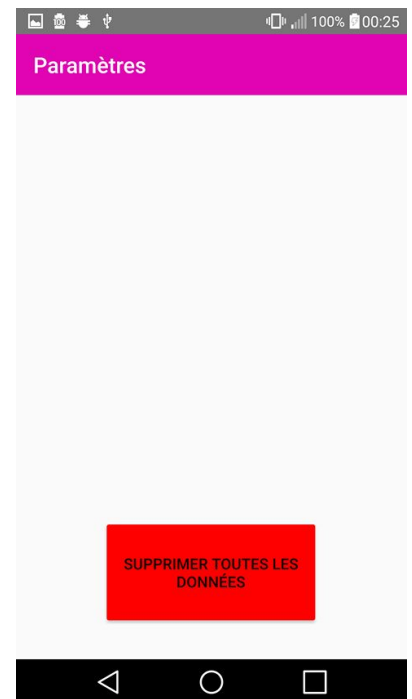
Celle-ci présente une liste (éventuellement vide) de noms de tâches (objets `ItemToDo`) et de `CheckBox` affichés dans un `RecyclerView`, un bouton en croix rouge permettant d'effacer toutes les listes et de retourner à la `ChoixListActivity`, un champ `EditText` permettant de renseigner un nom de tâche et un bouton plus vert permettant de créer une tâche ayant pour nom la chaîne de caractères dans le champ `EditText` et dont l'attribut `fait` est initialisé sur `false`. Cliquer sur une `CheckBox` permet de la cocher et de passer l'attribut `fait` de la tâche correspondante sur `true`.





✓ Paramètres :

L'activité `SettingsActivity` présente un unique bouton permettant de supprimer toutes les données enregistrées dans l'application (pseudos, listes et tâches). Le dernier pseudo enregistré reste cependant proposé dans le champ principal de `MainActivity`.



✓ Gestion et persistance des données : Utilisation de JSON et des `SharedPreferences` :

Pour réaliser la persistance, nous avons créé une classe `GestionDonnees` qui permet de manipuler un fichier xml `SharedPreferences` contenant les différents objets (profil, listesToDo, tâches) au format JSON.

Il suffit alors d'appeler :

```
GestionDonnees ges = new GestionDonnees(getBaseContext());  
ges.nomDeMaMethode();
```

pour réaliser le traitement voulu. Cette classe permet de récupérer le pseudo en cours, créer un nouveau profil, récupérer la liste des profils, récupérer la liste des ToDos du profil courant, récupérer les tâches de la liste ToDos courante, ajouter une liste/tâche, cocher ou décocher une tâche, supprimer les tâches ou les listes ToDos, ou les profils.

Perspectives

Actuellement, l'utilisateur ne peut pas supprimer individuellement une liste ou une tâche, il est obligé de supprimer toutes les tâches d'une liste ou toutes les listes, ce qui n'est pas pratique. Une alerte demandant la confirmation de suppression d'une liste ou d'une tâche pourrait être affichée après un appui long sur l'un de ces objets.

Pour ce qui est de l'organisation, l'activité principale pourrait être remplacée par un système de choix de profil (Par exemple : travail, maison, perso etc.).



Les activités ShowListActivity et ChoixListActivity pourraient être munies de drawer affichant des options de création/modification/filtrage.

D'après la documentation Android, les PreferenceActivity sont dépréciés, il pourrait être judicieux d'utiliser des fragments.

L'application n'est disponible qu'en français, et aucune disposition n'a été prise pour pouvoir traduire simplement l'application dans une autre langue.

Conclusion

Cette application est notre premier projet Android, et elle nous a donné l'occasion de mettre en oeuvre ce que nous avons appris dans les différents cours. La particularité par rapport aux cours de Live Coding auxquels nous avons assisté, est que si le raisonnement permettant de mettre en place les différents objets pour obtenir un résultat (par exemple un RecyclerView nécessitant la création d'un Adapter contenant Holder et Binder) paraît simple lorsqu'il est expliqué en cours, il est beaucoup plus compliqué de le mettre en place dans un cas plus complexe lorsqu'on est en autonomie. En effet, lorsqu'on se retrouve bloqué (le projet ne compile pas, l'application ne peut pas s'ouvrir, ou ne réagit pas comme on le pensait), les sources d'erreurs sont alors multiples et parfois peu compréhensibles malgré l'aide de l'IDE.

En plus du travail demandé qui nous a obligé à maîtriser des concepts clés, on s'est essayé à 'améliorer' le design de notre application en reprenant les couleurs du Pokémon TouDoudou (pour ToDoList). Cela nous a permis de découvrir par nous mêmes comment donner des couleurs/formes à notre application. Cependant, dans un soucis de temps et par manque de connaissances, nous avons modifié individuellement les paramètres d'affichage de chaque objet plutôt que de créer un style personnalisé héritant du style par défaut.

Ce projet nous a permis de nous familiariser avec l'IDE Android-Studio, d'appréhender la programmation en équipe, de comprendre certaines des problématiques liées au développement mobile et d'en comprendre les concepts fondamentaux (vues, activités, classes etc.)

Bibliographie

Sitographie :

- JSON : <https://developer.android.com/reference/kotlin/org/json/JSONObject>
- SharedPreferences : <https://developer.android.com/reference/android/content/SharedPreferences>
- RecyclerView : <https://developer.android.com/guide/topics/ui/layout/recyclerview>

Nathan BOISARD
Maxime LERAITRE



- PreferenceActivity
<https://developer.android.com/reference/android/preference/PreferenceActivity>

Ressource :

- Electif PMR 2019 TD 13 mai 2019.pdf