

CR Séquence 1

Introduction

La séquence 1 consiste à créer en Kotlin une application permettant aux utilisateurs de créer des listes de choses à faire, implémentées avec RecyclerView, avec persistance des données en préférences utilisant le format JSON.

Analyse

Grâce au langage Kotlin, les classes dédiées à la gestion des listes (définies par le diagramme de classes donné dans l'énoncé) sont plus succinctes : elles se réduisent à un constructeur primaire en en-tête de la classe.

Après avoir compris cela, j'ai rencontré principalement trois difficultés.

La première, résolue à l'aide de la vidéo [1], a été de gérer la persistance des données en préférences pour ChoixListActivity. Pour cela, il a fallu considérer une chaîne de caractère extraite des préférences prenant une valeur par défaut correspondant à une liste vide. Ensuite, à chaque fois qu'une liste est ajoutée par l'utilisateur, cette chaîne est actualisée et enregistrée dans les préférences. On fait de même pour ShowListActivity, en faisant attention à enlever les espaces dans le titre des listes dans la chaîne définie par défaut, pour être conforme au format JSON.

La deuxième, résolue à l'aide de l'article [2], a été de gérer les clics sur les listes. Pour cela, on définit dans l'adaptateur ListAdapter une interface OnItemClickListener, munie d'une méthode onItemClick prenant en paramètre une liste de type List<Item>, qui est implémentée par l'activité ChoixListActivity. Comme on a accès à l'item cliqué dans onBindViewHolder, on utilise onItemClick dans cette méthode. Ainsi, on peut ouvrir ShowListActivity et lui passer le pseudo et le titre de la liste cliquée. Il faut remarquer que le titre ne suffit pas à identifier une liste, puisque deux utilisateurs peuvent avoir créé une liste avec le même titre, d'où la nécessité de passer le pseudo.

La troisième, résolue à l'aide du *repository* GitHub [3], a été de gérer la sauvegarde de l'état des checkbox dans ShowListActivity après un clic sur ceux-ci. Pour cela, on procède de façon similaire à la gestion des clics sur les listes, en utilisant une interface OnCheckedChangeListener implémentée par ShowListActivity, mais dont la méthode onItemClick prend en paramètre une vue correspondant au checkbox cliqué et un entier correspondant à la position.

Conclusion

Ce TP m'a permis d'appliquer mes connaissances en Kotlin, de manier des listes RecyclerView, et notamment de gérer le clic sur un élément d'une liste, ainsi que de me familiariser avec la persistance des données en préférences avec le format JSON. Ayant réalisé ce TP sans binôme, j'ai rencontré des difficultés (ce qui explique que je n'ai pas eu le temps de réaliser les améliorations facultatives) mais j'ai réussi à les surmonter par tâtonnement et à l'aide de ressources trouvées sur Internet (cf Bibliographie).

Perspectives

En plus des améliorations facultatives proposées dans l'énoncé, il serait utile de pouvoir supprimer les listes et items créés par l'utilisateur.

Bibliographie

[1] Flutter - Todo App with Data Persistence using Shared Preferences - HOW TO :

https://www.youtube.com/watch?v=oPrpk_JiGvc&ab_channel=Jo%C3%A3oSoares

[2] RecyclerView Item Click Listener the Right Way : <https://medium.com/android-gate/recyclerview-item-click-listener-the-right-way-daecc838fbb9>

[3] Kotlin RecyclerView Checkbox : https://github.com/Oclemmy/Kotlin-RecyclerView-CheckBox/blob/master/app/src/main/java/com/devosha/kotlin_recyclerview_checkbox/MainActivity.kt