eFrei
PARIS
Ecole d'ingénieurs du numérique

BRANDAO Matthieu
CAO Anthony
BAUDIN Alexandre

PROJET PYTHON : SYSTEME DE RECOMMANDATION DE LIVRES

L1 – Groupe A

2021/2022

Sommaire

Sommaire 1
Introduction 2
Présentation fonctionnelle du projet 2
Fonctionnalités réalisées 2
Fonctionnalités ajoutées 3
Présentation technique du projet 4
Choix des structures de données 4
Profil 4
Dépôt 7
Recommandation 9
Difficultés rencontrées et solutions 9
Présentation des résultats 10
Présentation (non exhaustive) de l'interface utilisateur 10
Données de reproduction des tests 10
Conclusion 11

Introduction

Durant un semestre, les élèves de L1 dans l'école du numérique EFREI ont été introduits au langage de programmation Python lors de cours magistraux et de travaux dirigés. Afin d'évaluer nos compétences, l'école nous a demandé de rendre un projet nommé « Système de recommandation de livres ».
Le projet consiste dans la programmation en Python d'une bibliothèque virtuelle. Elle doit être capable de gérer des utilisateurs (afficher, ajouter, modifier et supprimer), de gérer un dépôt de livres (afficher, ajouter, modifier et supprimer) et de recommander des livres à un lecteur (noter, recommander).
Les objectifs de ce projet sont multiples. D'abord, se familiariser avec le langage de programmation Python ; mettre en application les compétences acquises à travers nos cours et apprendre de nouvelles fonctionnalités. Ensuite apprendre à gérer un projet en groupe ; répartir les tâches à faire, mettre au point les fonctionnalités faites et à faire, faire cohabiter les programmes des uns et des autres ou alors respecter des deadlines. Et enfin savoir réaliser un rapport de notre projet ; synthétiser les fonctionnalités créées et savoir les expliquer clairement.

Présentation fonctionnelle du projet

Fonctionnalités réalisées

Ajout d'un lecteur

Cette fonctionnalité permet d'ajouter un nouveau lecteur. On entre les informations d'un lecteur (nom, sexe, tranche d'âge, style de lecture, livres lus) et le programme vérifie si celui-ci existe déjà, sinon il ajoute le lecteur et ses informations dans la base de données.

Affichage d'un lecteur

Cette fonctionnalité permet d'afficher les informations d'un lecteur. On entre le nom du lecteur à afficher. S'il existe dans la base de données, le programme affiche les informations du lecteur.

Modification d'un lecteur

Cette fonctionnalité permet de modifier les informations d'un lecteur. On entre le nom du lecteur à modifier. S'il existe dans la base de données, on entre l'information à modifier puis on renseigne la nouvelle valeur. L'information du lecteur est alors mise à jour.

Suppression d'un lecteur

Cette fonctionnalité permet de supprimer un lecteur déjà existant. On entre le nom de l'utilisateur à supprimer. S'il existe dans la base de données, l'intégralité de ses informations enregistrées sont supprimées (nom, sexe, tranche d'âge, style de lecture, livres lus, notation des livres).

Affichage des livres

Cette fonctionnalité permet d'afficher les titres des livres présents dans le dépôt.

Ajout d'un livre au dépôt

Cette fonctionnalité permet d'ajouter un nouveau livre au dépôt. On entre le nouveau titre, le programme vérifie si le livre existe déjà, sinon il ajoute le livre au dépôt.

Modification du titre d'un livre

Cette fonctionnalité permet de modifier le titre d'un livre. On entre le titre du livre à modifier. S'il existe dans le dépôt, on renseigne le nouveau titre. Le titre du livre est alors mis à jour.

Suppression d'un livre

Cette fonctionnalité permet de supprimer un livre déjà existant. On entre le titre du livre à supprimer. S'il existe dans le dépôt, le livre est supprimées ainsi que ce qui lui était rattaché (titre, place dans la liste des livres lus par les lecteurs, notes données par les lecteurs).

Notation d'un livre

Cette fonctionnalité permet de noter un livre déjà lu. On entre le nom du lecteur ainsi que le livre à noter. Le programme vérifie si le lecteur et le livre existent dans la base de données. Si oui nous pouvons entrer la note comprise entre 1 et 5. La note est enregistrée pour ce livre.

Recommandation de livres

Cette fonctionnalité permet de recommander des livres non lus par le lecteur en fonction de sa similarité avec les autres lecteurs. On entre le nom du lecteur. Si celui-ci est dans la base de données, le programme calcule le lecteur le plus proche du lecteur entré en fonction des livres lus et notés. S'affiche alors les livres recommandés. S'en suivent 2 options : noter un des livres recommandé (Si on l'a lu) ou revenir au menu principal.

Fonctionnalités ajoutées

Fonctionnalité « Retour sur le menu principal »

Dans le menu principal, lors de la sélection d'une fonctionnalité il est possible de revenir en arrière, permettant de s'échapper d'une mauvaise manipulation.

Fonctionnalité « Proposition de notation si aucune note n'existe »

Si l'utilisateur n'a pas noté de livre, il sera proposé d'en noter un au moment de la recommandation de livres.

Fonctionnalité « Aucun livre lu »

Le programme continue de fonctionner correctement même si un utilisateur n'a pas lu de livres.

Fonctionnalité « Affichage du temps de calcul de la matrice similarité »

Dans la partie recommandation, une ligne exposant le temps de calcul de la matrice de similarité est affichée.

Fonctionnalité « Affichage des notes »

Au moment de l'affichage du lecteur, est affiché chaque titre avec une note donnée par le lecteur.

Fonctionnalité « Enregistrement des notes »

Après le calcul de la matrice de notation, celle-ci est enregistrée dans le fichier notation_matrix.txt. Elle fait donc partie intégrante de la base de données. Les notes sont conservées entre chaque lancement du programme.

Fonctionnalité « Affichage des livres entrés »

Lors de la saisie des livres lus par un lecteur, une liste contenant les livres déjà entrés est affiché.

Présentation technique du projet

Choix des structures de données

Nous avons décidé :

- D'utiliser des fichiers « .txt » car ils étaient recommandés et par souci de simplicité.
- D'enregistrer la matrice de notation dans un fichier texte afin de ne pas perdre les notes données par les utilisateurs lorsqu'on ferme le programme.
- D'utiliser des dictionnaires au lieu de listes pour stocker les informations relatives aux utilisateurs. Il est en effet plus simple d'utiliser le pseudo de l'utilisateur comme clé pour accéder à ses informations plutôt que de stocker le pseudo de l'utilisateur et ses informations dans des listes 2D, exemple :

```
[['Gilbert', 5, 9], ['William', 1, 2, 3, 4, 8], ['AlienRoXoR17', 5]]
```

Ici on doit parcourir la 1re grande liste puis parcourir toutes les sous-listes afin de chercher laquelle contient les informations relatives à un certain utilisateur, ex : Gilbert

```
{'Gilbert': [5, 9], 'William': [1, 2, 3, 4, 8], 'AlienRoXoR17': [5]}
```

Ici, en utilisant dict[«Gilbert»] on accède directement aux informations de l'utilisateur désiré.

- D'utiliser des listes pour stocker les titres des livres (lorsque le programme est en cours d'utilisation) plutôt que d'utiliser une chaine de caractères contenant tout le fichier books.txt

```
'Débuter la programmation Java
Apprendre Python
Les Citations du Président Mao Tse-Toung'
```

VS

```
['Débuter la programmation Java', 'Apprendre Python', 'Les Citations du Président Mao Tse-Toung']
```

En utilisant une liste on peut utiliser list[0] pour afficher le 1er titre, c'est utile lorsque on veut afficher tous les titres en utilisant une boucle.

Profil

addNewReader

- On entre un pseudo à ajouter.
    - Le programme vérifie si le pseudo existe dans le fichier readers.txt (grâce à ConditionalInput).
- Le programme affiche successivement les listes d'informations possibles (Genre, Tranche d'âge, style de lecture), d'après la liste suivante :

Liste contenant les informations possibles

```
["Homme", "Femme", "Autre"],
["A moins de 18 ans", "A entre 18 et 25 ans", "A plus de 25 ans"],
["Science-fiction ", "Biographie", "Horreur", "Romance", "Fable", "Histoire", "Comédie"]
```

Affichages

```
1- Homme
2- Femme
3- Autre
```

```
1- A moins de 18 ans
2- A entre 18 et 25 ans
3- A plus de 25 ans
```

```
1- Science-fiction
2- Biographie
3- Horreur
```

...

- Les entrées sont enregistrées dans un tableau de chaines de caractères, sous la forme **[nom, indice genre, indice âge, indice style]**.
- Le fichier readers.txt est mis à jour avec le nouvel utilisateur/lecteur et les indices de ses informations, sous la forme ci-dessous.

Format d'enregistrement des données du lecteur dans readers.txt

```
Xavier,1,3,5
```

- Grâce à _readBooksLine (fonction interne) :
    - Affichage de la liste des livres du dépôt
    - Récupération des entrées (correspondant aux indices des livres lus) dans un tableau de chaines de caractères, sous la forme **[nom, indice, indice, indice, …]**.
    - Vérification par le programme que les indices ne se répètent pas et qu'ils sont compris dans ceux possibles.
- Le fichier booksread.txt est mis à jour avec le nouvel utilisateur et les indices des livres lus, sous la forme ci-dessous :

Forme d'enregistrement des données du lecteur dans booksread.txt

```
anonyme,4,12,15,8
```

printReaders

- Récupération des informations présentes dans readers.txt, booksread.txt et books.txt, sous forme de trois dictionnaires (grâce à FileDict).
- Le programme demande d'entrer un nom s'il n'a pas été renseigné lors de l'appel à la fonction (en vérifiant l'existence du pseudo entré dans la base de données).
- On récupère les informations personnelles du lecteur dans les dictionnaires précédemment créés, ainsi que les livres lus par le lecteur.
- Affiche ces informations, et teste si le nombre de livre lu est nul ou non.

- S'il est nul, affichage d'un message ; sinon, affiche les livres.

Message si aucun livre n'a été lu

```
A lu :
rien pour l'instant...
```

Message d'une liste des livres lus

```
A lu :
- Débuter la programmation Java
- Un conte de deux villes
- Le Hobbit
- Le rêve dans le Pavillon rouge
```

- S'il existe des notes pour les livres, on affiche les notes.
- Retourne les informations personnelles du lecteur.

ModifyReader

- Grace à rewriteFile (fonction interne) :

Protocole :

- Ouverture du fichier en mode lecture.
- Utilisation de FileDict pour enregistrer les informations du fichier sous forme d'un dictionnaire.
- Dans le dictionnaire, suppression des données correspondantes à la clef renseignées.
- Dans le dictonnaire, ajout des informations renseignées.
- Fermeture du fichier en mode lecture et ouverture du fichier en mode écriture.
- Réécriture du fichier avec les données à jour du dictionnaire.
- Fermeture du fichier écrit.

- Affichage du lecteur et récupération de ses données grâce à printReaders.
- Affichage de la liste des informations modifiables, comme ci-dessous :

Informations modifiables

```
1- Nom
2- Genre
3- Âge
4- Style de lecture
5- Livres lus
```

- Récupération de l'indice correspondant à l'information à modifier et de la nouvelle valeur.
- Réécriture des fichiers effectés par la modification.

removeReader

- On demande le pseudonyme de l'utilisateur à supprimer.
- On lit les fichiers booksread et readers ; on les convertit en 2 dictionnaires dont les clés sont les pseudos et les valeurs sont le reste de la ligne correspondant au pseudo (en utilisant la

fonction FileDict), et dont les valeurs sont des listes d'entiers ; cf. illustrations ci-dessous :

Contenu d'un fichier

```
Gilbert,5,9
William,1,2,3,4,8
```

Contenu du dictionnaire correspondant

```
{'Gilbert': ['5', '9'], 'William': ['1', '2', '3', '4', '8']}
```

- On supprime le couple clé-valeur du pseudo entré des 2 dictionnaires,
- Et on reconvertit les dictionnaires en 2 chaines de caractères et on replace le contenu des fichiers textes par ces deux chaines de caractères.

Dépôt

PrintBookList

- On lit le fichier books.txt ;
- On le convertit en une liste dont les valeurs sont les noms des livres et les indices sont la ligne du livre dans le fichier (- 1) ;
- On parcourt cette liste par énumération. La variable « index » parcourt les indices de la liste et « title » parcourt les titres puis on affiche la liste des livres sous un format spécifique.

addNewBook

- On demande à l'utilisateur d'entrer le nom d'un livre qui n'est pas déjà dans la base de données (on redemande un titre tant qu'il le faut) ;
- On écrit le titre à la fin du fichier books.txt.

modifyBook

- On affiche la liste des livres et on affecte cette liste à la variable bookslist (grâce à la fonction printBookList) ;
- On demande à l'utilisateur d'entrer l'indice (titleindex) du livre dans la liste affichée qu'il veut modifier ;
- On demande à l'utilisateur d'entrer le nouveau nom du livre ;
- On affecte ce nouveau nom à la variable d'indice (titleindex – 1) dans la liste booklist ;
- On convertit la liste booklist en chaine de caractères et on remplace le contenu de books.txt par cette chaîne de caractères.

removeBook

Partie books.txt :

- On affiche la liste des livres et on affecte cette liste à la variable bookslist (grâce à la fonction printBookList) ;
- On demande à l'utilisateur d'entrer l'indice (bookindex_to_remove) du livre dans la liste affichée qu'il veut supprimer ;
- On supprime la valeur d'indice (bookindex_to_remove – 1) de la liste booklist ;
- On convertit la liste booklist en chaine de charactères et on remplace le contenu de book.txt par cette chaîne de charactères.

Partie booksread.txt :

- On lit le fichier booksread.txt qu'on convertit en un dictionnaire (content) dont les clés sont les pseudos et les valeurs sont le reste de la ligne correspondant au pseudo (en utilisant la fonction FileDict).

Les valeurs sont des listes de chaine de caractères (qui sont en fait des entiers), cf. illustrations ci-dessous :

Contenu du fichier

```
Gilbert,5,9
William,1,2,3,4,8
```

Contenu du dictionnaire (content)

```
{'Gilbert': ['5', '9'], 'William': ['1', '2', '3', '4', '8']}
```

pseudo infos

- On parcourt le dictionnaire content par énumération. La variable pseudo parcourt les clés et la variable infos parcourt les valeurs :
- On regarde si l'indice du livre à supprimer (bookindex_to_remove) est dans la liste infos, si c'est le cas alors on supprime cet indice de la liste infos.
- On parcourt ensuite la liste infos par énumération (par indice et élément) :

```
i : indice      0     1     2     3     4
William': ['1', '2', '3', '4', '8']
           number
```

La variable i parcourt les indices de la liste et number parcourt les valeurs (chaines de caractères) de la liste info.

- On regarde pour chaque valeur de number (on regarde int(number) car c'est une chaine de caractères) si elle est supérieure à bookindex_to_remove, si c'est le cas alors on la décrémente de 1.

On a donc modifié le dictionnaire content de sorte qu'il ne contienne plus l'indice du livre qu'on supprime (bookindex_to_remove) et de sorte que les indices des livres supérieurs à bookindex_to_remove soient décrémentés de 1.

Partie notation_matrix.txt :

- On lit le fichier notation_matrix.txt et on en fait un dictionnaire ayant pour clés le pseudo de l'utilisateur et pour valeurs la liste des notes données par l'utilisateur :

Contenu du fichier

```
Gilbert, 0, 0, 0, 0, 1, 0, 2, 3, 4, 0, 0, 0, 0, 0, 0, 0, 0
William, 2, 0, 0, 0, 0, 0, 0, 4, 1, 2, 0, 0, 0, 0, 0, 0, 0
```

Contenu du dictionnaire correspondant

```
{'Gilbert': [0, 0, 0, 0, 1, 0, 2, 3, 4, 0, 0, 0, 0, 0, 0, 0, 0],
 'William': [2, 0, 0, 0, 0, 0, 0, 4, 1, 2, 0, 0, 0, 0, 0, 0, 0]}
```

listrates

- On parcourt les valeurs du dictionnaires (les listes de notes : listrates) et pour chaque liste on supprime la note d'indice bookindex_to_remove -1.

Recommandation

rateBook

- Si le pseudo n'est pas donné ou incorrect alors on le demande (typiquement en passant par le menu).
- Si le nombre de livres lus est zéro alors on annule l'opération.
- Si l'indice du livre à noter n'est pas donné (ou incorrect, quoique cela n'est théoriquement jamais le cas), alors on affiche une liste des livres lus de l'utilisateur et on le lui demande un livre parmi ladite liste. Finalement on lui demande la note entre 1 à 5 :

```
4- Don Quichotte de la Manche
8- Harry Potter à l'école des sorciers
12- Le Lion, la Sorcière blanche et l'Armoire magique
15- Réfléchissez et devenez riche
Quel livre lu souhaitez-vous noter ? Indice : 3
Vous n'avez pas lu ce livre. Indice d'un autre livre lu : 0
Vous n'avez pas lu ce livre. Indice d'un autre livre lu : 4
Entre 1 et 5, quelle est votre appréciation de ce livre ? 5
Le livre 'Don Quichotte de la Manche' a bien été noté !
```

- On inclut la note dans la matrice de notation.

recommendBooks

- On récupère le pseudo de l'utilisateur en le lui demandant. Ensuite, on calcule la matrice de similarité et on actualise la variable globale liée ; le temps de calcul est aussi donné.
    - Si l'utilisateur n'a pas noté de livres alors on lui recommande de le faire (on bascule dans la fonction rateBook).
- On affiche une liste des livres recommandés (ceux lus par le lecteur le plus similaire mais non lus par l'utilisateur) et on demande un livre de cette liste. L'indice du livre est ajouté dans le fichier des livres lus, puis on demande si l'utilisateur souhaite noter maintenant.
    - Si oui, alors on bascule dans la fonction rateBook ; sinon on s'arrête là.

Difficultés rencontrées et solutions

Encodage des fichiers

L'encodage des fichiers en lecture/écriture diffère sur Windows (utilisant ainsi un encodage spécifique), donc le contenu lu était illisible avec des caractères spéciaux → solution : spécifier un même encodage à chaque ouverture de fichier texte.

modifyReader

Optimiser les parties communes des différentes bifurcations de la fonction → solutions : définition de la fonction rewriteFile et utilisation d'un dictionnaire pour associer nom du fichier et ligne à écrire.

removeReader

Adapter les indices des livres lus → solution : pour chaque ligne, on supprime le livre à supprimer et on réduit les indices plus grands que ledit livre.

Présentation des résultats

Présentation (non exhaustive) de l'interface utilisateur

Affichage de la lectrice Lili

```
------------------------------

Le lecteur "Lili" :
- Est de genre Femme ;
- A entre 18 et 25 ans ;
- Aime lire des livres de Biographie ;

A lu :
- Débuter la programmation Java
- Un conte de deux villes
- Le Hobbit
- Le rêve dans le Pavillon rouge
Notes :
Débuter la programmation Java - 5/5
Apprendre Python - 2/5
Don Quichotte de la Manche - 4/5
Le Seigneur des Anneaux - 2/5
Harry Potter à l'école des sorciers - 2/5
Dix Petits Nègres - 2/5

------------------------------
```

Interface de recommendation pour Lili

```
Pseudonyme entré : "Lili"
------------------------------

Matrice de similarité calculée en 0 min 0 s 1 ms 843 µs

4- Don Quichotte de la Manche
8- Harry Potter à l'école des sorciers
12- Le Lion, la Sorcière blanche et l'Armoire magique
15- Réfléchissez et devenez riche
Saisir l'indice d'un livre recommandé que vous n'avez pas encore lu : 4
Livre ajouté : Don Quichotte de la Manche
Voulez-vous noter le livre maintenant ? 0 ou 1 :
1 --> Oui, sans tarder
0 --> Pas maintenant
 1


------------------------------

Entre 1 et 5, quelle est votre appréciation de ce livre ? 4
Le livre 'Don Quichotte de la Manche' a bien été noté !

------------------------------
```

Test d'entrées avec des caractères spéciaux non autorisées

```
Pseudonyme : ,,,,,,,,,,,,\
Le pseudo ne peut pas être vide. Pseudonyme : ,,,,,,,,,,,,,,,,,,,,\n\n,,,,,,,ArchiBald\fx\,,,,\n,
Pseudonyme entré : "ArchiBald\fx\"
------------------------------
```

Données de reproduction des tests

Profil

Ajouter un lecteur : 1 / 1 / « un pseudo » / 2 / 1 / 4 / 0 / 1 / 3 / 1 / 100 / 5 / 1 / 3

Afficher un lecteur : 1 / 2 / « un pseudo »

Modifier un lecteur :

- 1 / 3 / « un pseudo » / 1 / « un, deux » (normalement interprété comme « un deux »)
- 1 / 3 / « un deux » / 2 / 1
- 1 / 3 / « un deux » / 3 / 2

- 1 / 3 / « un deux » / 4 / 3
- 1 / 3 / « un deux » / 5 / 1 / 0 / 7

Supprimer un lecteur : 1 / 4 / « un deux »

Dépôt

Afficher la liste du dépôt : 2 / 1
Ajouter un livre au dépôt : 2 / 2 / « Le Temps des Tempêtes »
Modifier le titre d'un livre dans le dépôt : 2 / 3 / 20 / « Pas d'idées »
Supprimer un livre du dépôt : 2 / 4 / 20

Recommandations

Noter un livre : 3 / 1 / « Xavier » / 1 / 6 / 4
Suggérer des livres : 3 / 2 / « Xavier » / 10 / 9 / 0

Conclusion

Ce projet nous a d'abord permis d'utiliser nos connaissances et d'affiner nos compétences techniques en programmation, nous avons pu approfondir notre compréhension des collections, des matrices, des fonctions, des fichiers, des modules, etc.
Nous avons aussi pu développer de nouvelles compétences : nous avons découvert le débogage sur PyCharm, nous avons utilisé Discord et Google Docs pour communiquer notre avancement, nous partager les tâches, ainsi que GitHub pour metter le travail en commun.
Ainsi, tous ces outils et compétences nous ont permis de terminer ce projet à temps et de vous rapporter notre expérience.