

# Compte rendu : SAE 1.02

## Comparaison d'approches algorithmiques

JACUELIN Bastien

DE LA FUENTE Axel

### Résumer des fonctionnalités de l'application :

Ici le besoin du client était la production d'une application permettant une gestion complète des logements, des locataires et des demandes de logement de HML63. Pour répondre au mieux à ce besoin nous avons mis en place de nombreuses fonctionnalités décrites ci-dessous.

#### Pour la gestion des logements de HML63 :

- Il est possible de créer un logement qui sera stocké dans le fichier. Ce logement est décrit selon plusieurs critères : son identifiant, son adresse, son type, son nombre de place et son occupation (occupé : 1 ou non : 0).
- Il est possible de supprimer un logement du fichier. Cela fonctionne grâce à l'identifiant du logement.
- Il est possible d'afficher tous les logements contenus dans le fichier. Il y a trois manières de les afficher. (Seulement afficher : ils ne sont pas triés dans le fichier).
  - o Afficher de manière non-trié (comme dans le fichier)
  - o Afficher de manière triée selon le nombre de place du logement (du plus petit au plus grand)
  - o Afficher de manière triée selon le type (T6, T5, T4, T3, T2, T1)

- Il est possible d'afficher seulement un type de logement, que les T1, que les T2 ect...

#### Pour la gestion des locataires :

- Il est possible d'afficher tous les locataires actuels (pas de tris)
- Il est possible de rechercher un locataire précis et avoir ses informations. On recherche par son identifiant de locataire.
- En ce qui concerne la création des locataires, elle passe par une fonction qui permet de les créer automatiquement. Il suffit de la lancer et une actualisation des points se fait, il faut ensuite trier la file des demandes. Si un logement est libre ET correspond aux critères du demandeur(nombre de personnes qui vivent dans le logement), alors la demande est supprimée, un locataire est créé et le logement devient occupé. (Un locataire est défini selon son nom, son prénom, sa nationalité, ses ressources annuelles, le nombre de personnes dans le foyer, l'identifiant de son logement, et un identifiant de locataire.)

#### Pour la gestion des demandes de logement :

- Il est possible d'enregistrer une demande et de l'enregistrer dans un fichier. La demande est mise en fin de file et sera triée lors de la prochaine actualisation des points qui a lieu tous les mois. Une demande est définie selon un identifiant, le nom, le prénom, la nationalité, s'il possède ou non une carte de séjour valide, ses ressources annuelles, le nombre de personnes qui composent le foyer, et le nombre de points attribués (handicap ect...)
- Il est possible d'effectuer une recherche d'une demande selon son identifiant et d'afficher les informations la concernant.
- Il est possible d'afficher toute la file des demandes. (Elle est dans l'ordre dans laquelle elle est enregistrée dans le fichier, donc triée depuis la dernière actualisation.)
- Il est possible de trier la file en fonction des points. (la file n'est pas triée automatiquement, mais seulement quand on le demande)

### **Structure choisie pour les fichiers :**

Tous les fichiers sont faits avec la même structure interne. Il s'agit de fichier texte simple avec une entité par ligne composé de plusieurs éléments. Dans chaque cas l'entité prends une ligne et possède un identifiant qui lui est propre, il en suit après toutes les informations concernant l'entité. Le fait d'utiliser un fichier avec du texte et non pas un fichier binaire permet que ces données soit exportable plus simplement, comme sur un tableur par exemple.

### **Structure choisie pour les données :**

#### Pour les logements :

Les logements n'ont pas de contrainte particulière si ce n'est que d'être stockés et triés efficacement. L'option la plus évidente et que nous gérons le mieux est le tableau de structure. Les tableaux permettent un chargement du fichier de manière simple ainsi que son tri et son affichage. C'était selon nous la meilleure option pour la gestion des données des logements.

#### Pour les locataires :

Les données sur les locataires sont le fruit de la concaténation des données des logements et des demandes. Ces données n'ont pas de contraintes particulières non plus. Le tableau de structure a donc été l'option retenue, sachant que c'est le type de structure de données que nous gérons le mieux.

#### Pour les demandes :

Pour les demandes une contrainte apparaissait, car il s'agissait d'une file d'attente pour les demandes de logement. Pour gérer une file d'attente nous avons opté pour une liste chaînée de type file.

### **Les différents algorithmes de tris ainsi que les différents algorithmes de recherche :**

En ce qui concerne les algorithmes de tris, nous n'utilisons qu'un seul type de tri, il s'agit du tri à bulle, il est plutôt simple à mettre en œuvre. Nous avons plusieurs fonctions de tris car certains tris en fonction des entiers et d'autres en fonction de chaînes de caractères (tris alphabétique), mais se sont tous des tris à bulle. Nous avons choisis de sélectionner un tri à bulle car les données à traiter restent relativement faibles et sa complexité bien que supérieure aux autres tris n'en fait pas une contrainte.

Pour les algorithmes de recherche nous n'utilisons qu'un seul type de recherche, c'est la recherche séquentielle ce n'est pas la plus efficace mais elle est très simple à mettre en place et fonctionne à tous les coups. De plus dans certains cas nous recherchons la plus petite valeur qui se trouve au tout début du tableau. La recherche est donc plus rapide dans ce cas là.