

Compte rendu TPC++ n°3

I. Description de notre format de fichier

Liste des commandes en rapport avec le fichier Makefile :

- Make TP3
- Make build_MAP
- Make clear
- Make run

Pour le format du fichier de sauvegarde/restitution des trajets, nous avons fait le choix d'écrire un trajet par ligne, peu importe son type. Un trajet possède obligatoirement en début de ligne une lettre permettant l'identification du type de trajet : S pour simple et C pour composé. Chaque trajet simple se présente sous la forme suivante :

Format d'un Trajet Simple
S VilleDepart1 VilleArrivee1 MoyenTransport1

Les caractères | jouent le rôle de séparateur entre les villes et aussi les moyens de transport. Un trajet composé contient N trajets (simples), ce nombre devra apparaître clairement à la suite de l'identificateur de trajet composé (lettre C en début de ligne) sans espace. Il ne faudra pas répéter la ville de départ entre chaque trajet simple composant un trajet composé car nous considérons qu'elle se propage automatiquement d'un trajet à l'autre, ce qui gagne de l'espace au niveau de la saisie et assure la cohérence des trajets.

Format d'un Trajet Composé
CN VilleDepart1 VilleArrivee1 MoyenTransport1 VilleArrivee2 VilleArriveeN MoyenTransportN

Dans le menu, deux options supplémentaires sont apparues : la sauvegarde sélective ainsi que la lecture sélective. Pour chacune de ces deux options, il est proposé à l'utilisateur de rentrer le nom du fichier en question, puis tous les critères de sélection lui sont proposés les uns à la suite des autres. Nous avons rendu la combinaison des critères possible : l'utilisateur peut juxtaposer le(s) critère(s) qu'il désire pour la sélection ou bien ne pas en appliquer du tout. Lors de la sauvegarde, les trajets sont rajoutés à la fin du fichier ce qui permet de conserver les anciens trajets. Si le fichier de sauvegarde indiqué n'existe pas, il sera automatiquement créé avec le nom donné. En revanche, pour la récupération des trajets, si le fichier n'existe pas, un message d'erreur s'affichera sur le terminal pour indiquer que la lecture du fichier souhaité n'a pas pu se dérouler correctement. Il n'y a aucune contrainte sur l'ordre des trajets en fonction de leur type. En ce qui concerne la sélection par indices, le premier trajet porte le numéro 1 ainsi si l'utilisateur veut les n premiers trajets, l'intervalle correspondant sera [1, n].

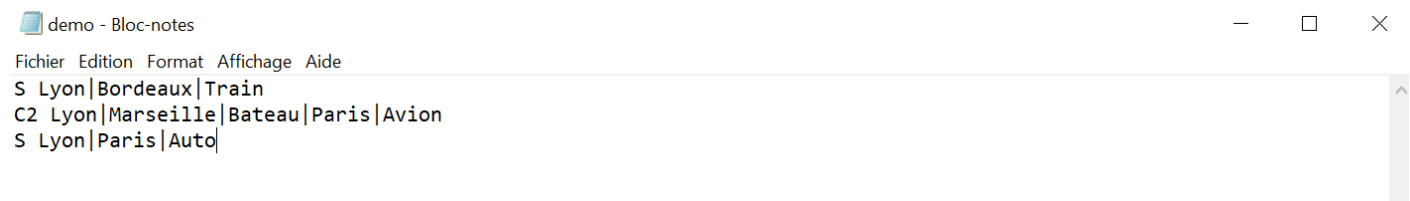
Pour le chargement de trajets, notre programme regarde d'abord s'il n'a pas dépassé l'indice de fin (s'il est renseigné) afin d'éviter de parcourir inutilement la fin du fichier. Ensuite, il y a une deuxième vérification sur le numéro de la ligne : il n'est pas nécessaire de lire et stocker les trajets avec un indice inférieur à celui de la borne minimale demandée donc on peut se contenter de passer à la ligne suivante si le cas se présente. Une fois que l'indice est vérifié, on se préoccupe de la sélection du type de trajet si elle est spécifiée. En effet, il

est facile d'éliminer les types de trajets non voulus en comparant avec l'identificateur de type de trajet qui apparaît au début de chaque ligne dans le fichier texte. Enfin, si tous les critères précédents sont validés, on compare la ville de départ voulue avec celle en cours. Si elle passe ce test, on distingue 2 cas : c'est un trajet simple et on a donc accès directement à la ville d'arrivée où l'on peut facilement savoir si c'est celle sélectionnée. Si c'est un trajet composé, un tableau temporaire de pointeurs de trajets est créé dans lequel sont stockés les trajets intermédiaires. Si la ville de d'arrivée n'est pas celle voulue, il y a destruction de la mémoire associée au tableau de pointeurs de trajets sinon on ajoute effectivement au catalogue le trajet composé préalablement instancié grâce au tableau temporaire. Ainsi, il n'y a pas d'occupation d'espace mémoire inutile lors du parcours du fichier de chargement, seuls les trajets correspondant à la sélection (à moins que tous les critères soient respectés sauf la ville d'arrivée) sont effectivement créés, ce qui limite l'accès au TAS et nous fait gagner en rapidité.

Pour la sauvegarde des trajets, nous comparons aussi les critères de sélection dans le même ordre que pour le chargement. Cette fois-ci, la vérification des critères est bien plus simple car nous n'avons pas à lire les trajets dans un fichier mais dans le catalogue. Ainsi, une fonction membre nommée `ecriture()` est surdéfinie pour un trajet simple et pour un trajet composé. Elle permet d'écrire le trajet dans le fichier voulu. Nous avons utilisé le « dynamic cast » afin d'appliquer la méthode d'écriture correspondant au type de trajet en cours.

II. Contenu de notre fichier « demo »

Figure 1 : Capture d'écran du fichier demo



III. Conclusion

III.1. Problèmes rencontrés

Dans un premier temps, il nous a été difficile de faire un choix sur le type du format, plusieurs nous semblaient pertinents avec leurs avantages et inconvénients. En conséquence, nous avons dû trouver un compromis entre lisibilité pour l'utilisateur et facilité de lecture/écriture d'un point de vue logiciel. L'association des critères de sélection (facultative) a été également contraignante notamment pour le chargement des trajets : nous ne pouvions plus découper chaque type de sélection en une seule méthode. En effet, si nous nous étions contentés de juxtaposer les méthodes de sélection pour les combiner, cela aurait nécessité plusieurs parcours du fichier. Nous les avons donc tous réunis afin d'obtenir un unique parcours du fichier, ce qui est plus rapide surtout avec des fichiers volumineux. Il reste tout de même un petit bémol en terme de complexité à cause de notre format : nous ne pouvons pas exclure facilement les trajets composés avec une ville de départ ne correspondant pas à la sélection car nous devons d'abord lire toutes les villes intermédiaires qui se trouvent sur la même ligne. En revanche, la sélection sur la ville de départ est très optimisée car elle apparaît en premier sur chaque trajet composé.

III.2. Axes d'amélioration

Au niveau des améliorations possibles, nous avons pensé qu'il pourrait être utile que plusieurs utilisateurs puissent bénéficier du même catalogue. Ainsi, l'utilisation d'une base de données au lieu d'un fichier texte permettrait une évolution commune des trajets sauvegardés s'il y a par exemple plusieurs sites. D'autres fonctionnalités pourraient être rajoutées comme la suppression sélective ou la modification de trajets, que ce soit au niveau du chargement ou de la sauvegarde. Les critères de sélection pourraient aussi être plus nombreux : selon le moyen de transport, le nombre de trajets simples maximum composant un trajet composé, toutes les villes commençant par une lettre voulue... Toutes les améliorations proposées pour le TP2 peuvent également s'appliquer sur ce TP en rajoutant par exemple plus de caractéristiques sur les trajets dans les fichiers (durée, distance, coût etc...).