

TP C++ n°3 : Gestion des entrées / sorties

Damien Carreau

Jérôme Hue

8 janvier 2020

Table des matières

1	Nouvelles fonctionnalités	1
1.1	Description du fichier de sauvegarde/chargement	1
1.2	Cas particuliers	1
2	Conclusion	2
2.1	Problèmes rencontrés	2
2.2	Axes d'amélioration et d'évolution	2

1 Nouvelles fonctionnalités

1.1 Description du fichier de sauvegarde/chargement

On a choisi de sauvegarder nos trajets dans un fichier texte. On saisira donc son nom de la manière suivante : *monfichier.txt*. Le fichier servant à charger des trajets est un fichier texte dans lequel chaque trajet est spécifié sur une ligne. Un trajet est spécifié par ses attributs séparés par des points-virgules (Il y a donc $n+1$ éléments pour n points- virgules), comme suit pour un trajet simple :

ville de départ ; ville d'arrivée ; moyen de transport

et pour un trajet composé :

ville de départ ; ville d'arrivée ; moyen de transport ; ville de départ ; ville d'arrivée ; moyen de transport...

Il n'est donc pas nécessaire de spécifier si un trajet est simple ou composé, le programme le détecte de lui même, ce qui implique qu'un trajet qui comporte 3 points-virgules ou plus est considéré comme trajet composé. Un trajet composé est composé de trajets simples COMPLETS, voir la sous-section suivante pour plus de détails.

1.2 Cas particuliers

Voici une liste de différents cas particuliers :

- *ville de départ ; ville d'arrivée ; moyen de transport ;*
Ici, on a terminé la liste par un point virgule. Le programme va considérer que ce trajet est composé et logiquement insérer un trajet composé d'un seul trajet simple (De *ville de départ* à *ville d'arrivée* par *moyen de transport*). Ce cas nécessite d'être vigilant, car si l'on charge uniquement les trajets simples, ce trajet n'est évidemment pas chargé.
- *ville de départ ; ville d'arrivée ; moyen de transport ; ville de départ ; ville d'arrivée ; moyen de transport ; ville de départ*
Ici la ligne est bien terminée mais il manque deux éléments. Sera chargé dans le catalogue un trajet composé des deux premiers trajets simples.
- *ville de départ ; ville d'arrivée ; moyen de transport ; ville de départ ; ville d'arrivée ; moyen de transport ; ville de départ ; ville d'arrivée ;*
Ce trajet comporte 9 éléments, donc on charge un trajet composé de 3 trajets simples, avec le moyen de transport du dernier trajet simple non renseigné.

Enfin, concernant l'intervalle, il est possible de choisir un intervalle plus grand que le nombre de trajets, ou bien un intervalle qui ne comporte aucun trajet.

2 Conclusion

2.1 Problèmes rencontrés

Ce TP ne vit pas d'obstacles majeurs, contrairement au premier. Concernant le chargement de trajets, la difficulté fut la gestion des différents formats de chaînes de caractères : *string*, *char **, *const char ** ; on finira par modifier la classe *Trajet*. Une fois un seul mode de sauvegarde/chargement programmé, il est facile d'implémenter les 3 autres. Si parvenir à un programme s'exécutant sans fuites de mémoires ni erreurs fut complexe lors du précédent TP, ce ne fut ici que l'objet de rapides corrections de code.

2.2 Axes d'amélioration et d'évolution

Sur le plan technique, le code peut grandement être amélioré en factorisant le contenu des différentes méthodes de sauvegarde et de chargement. On peut aussi utiliser la STL et la classe *string*, là où cela nous était défendu auparavant. Sur le plan fonctionnel, des ajouts appréciables pourraient être le croisement de plusieurs critères de recherche (par exemple rechercher uniquement les trajets simples au départ de Paris), ou encore la sauvegarde d'une recherche.