



Documentation calculateur C

Projet d'intégration 2015

Nicolas Medar

v1.0

Sommaire

I-Utilisation	3
II-Fonctionnement	3
1-Algorithme aléatoire	3
2-Algorithme Alphabétique	4
3-Algorithme Hamiltonien	5

I-Utilisation

Ce programme sert à convertir un nombre de points géographiques en une route déterminée de façon à ce qu'elle soit exploitable par le serveur pour diriger les drones de la société.

II-Fonctionnement

Le programme reçoit un fichier de type Json (format de document web) fournit par le serveur. Ce fichier contient des POI (point of interest) constitués d'un nom (généralement une ville) et de ses coordonnées GPS. Le programme reçoit également en paramètre le type d'algorithme à employer ainsi qu'un nom à allouer au fichier de sortie. Chaque partie de la commande d'exécution du programme est indépendante, autrement dit on peut les placer dans l'ordre de son choix. En effet chaque commande est identifiée par le programme via des options de commande. Une commande type ressemble donc à `./programme -i fichier.json -o fichier.kml -a algoChoisi`.

La liste des options :

-h : -en cas de problème l'ensemble des commandes du programme sont accessibles via l'option -h ou `./programme -h`. cette commande renvoie la liste des options à transmettre aux programmes ainsi que les mots clefs à utiliser pour sélectionner l'algorithme à employer.

-i ou -in : précède le nom du fichier Json.

-o ou -out : précède le nom que l'on souhaite donner au fichier dans lequel on sauvegarde la route générée.

-v : active le mode verbeux.

-a ou -algo : indique l'algorithme à utiliser parmi les trois disponibles, soit :

algo_aleat pour une génération aléatoire.

algo_alpha pour un tri alphabétique du nom des villes

algo_hamil pour la génération de la route la plus courte entre un point de départ (premier POI reçu) et un point d'arrivée(dernier POI reçu).

1-Algorithme aléatoire

Dans ce premier cas nous partons du principe que le programme doit traiter un fichier appelé test.json est fournir une route aléatoire dans un fichier nommé test.kml

Dans un premier temps il faut exécuter le programme en lui transmettant les informations dont il a besoin pour fonctionner. Pour ce faire, tapez la commande suivante :

« chemin d'accès au programme »/NomDuProgramme « identifiant d'option x » « option correspondante »

Dans notre cas la commande ressemblera donc à :

```
./programme -i test.json -o test.kml -a algo_aleat.
```

Une fois ses consignes acquises, le programme s'exécutera de la façon suivante :

- 1-récupération des informations essentielles stocké dans le fichier Json
- 2-vérification de ses informations (les noms de fichier correctes et aucun élément manquant)
- 3-utilisation de l'algorithme sélectionné, ici un trie aléatoire.
- 4-création d'un fichier KML portant le nom transmet à l'étape 1.
- 5-sauvegarde de la route triée dans le fichier créé précédemment.
- 6-message de vérification (kml done). Puis fermeture du programme.

2-Algorithme Alphabétique

Dans ce deuxième cas nous partirons du principe que le programme doit traiter un fichier appelé test.json est fournir une route aléatoire dans un fichier appelé test.klm

Dans un premier temps il faut exécuter le programme en lui transmettant les informations dont il a besoin pour fonctionner. Pour ce faire tapez la commande suivante :

```
./programme -i test.json -o test.kml -a algo_alpha.
```

Vous pouvez constater que seule la dernière partie de la commande a changé par rapport à la précédente. Cette modification porte sur l'option -a qui détermine l'algorithme à utiliser.

Une fois ses consignes acquises, le programme s'exécutera de la façon suivante :

- 1-récupération des informations essentielles stocké dans le fichier Json
- 2-vérification de ses informations
- 3-utilisation de l'algorithme sélectionné, ici un trie alphabétique.
- 4-création d'un fichier KML portant le nom transmet à l'étape 1.
- 5-sauvegarde de la route triée dans le fichier créé précédemment.
- 6-message de vérification (kml done). Puis fermeture du programme.

3-Algorithmme Hamiltonien

Dans ce troisième cas nous partirons du principe que les paramètres en dehors de l'algorithme employé ne change pas.

Dans un premier temps il faut exécuter le programme en lui transmettant les informations dont il a besoin pour fonctionner. Pour ce faire tapez la commande suivante :

```
./programme 5 -i test.json -o test.kml -a algo_hamil.
```

Une fois ses consignes acquises, le programme s'exécutera de la même façon pour la saisie et la sauvegarde des données. Seule la façon de les traiter change.

Pour chaque algorithme sélectionnés il est possible de demander au programme d'afficher en temps réel les actions qu'il entreprend, pour ce faire il suffit d'ajouter à la commande d'exécution l'option -v comme pour toutes les commandes en dehors du nombre de villes il est possible de la placer dans n'importe quel ordre.