

Transformation de fichier .json en fichier .kml avec le langage Python et C

Victor VITCHEFF

5 avril 2022

Table des matières

Introduction	2
I Présentation des solutions	3
I.1 Solution Python	3
I.2 Solution C	3
II Choix des solutions	3
III Validation des choix	4
Conclusion	5

Introduction

Le but de ce test est de transformer des fichiers .json en fichier .kml. J'ai décidé de réaliser cette conversion en python et en C. Les fichiers json se présentaient sous la forme de dictionnaires avec 4 paramètres : l'ID, la longitude, la latitude et la confiance. Dans ce petit document, je vous propose de présenter ma solution (en python et en C), de justifier pourquoi j'ai utilisé ces langages et de vous montrer comment j'ai vérifié que mes fichiers obtenus étaient fonctionnels.

I Présentation des solutions

Pour les 2 solutions, j'ai décidé de les réaliser de la même manière. J'ai mis du temps à trouver comment utiliser la confiance. Finalement, j'ai décidé de calculer la moyenne de toutes les confiances et de ne sélectionner dans le fichier kml que les points dont la confiance était supérieure à la moyenne. Dans le fichier kml, j'ai décidé de remplacer le nom du point par son ID et de placer toutes les altitudes à 0.0m.

I.1 Solution Python

Ma solution python se présente en 2 fichiers. L'un des fichiers est le fichier "module.py". Il contient toutes les fonctions nécessaires à la conversion du fichier json en fichier kml. Le deuxième fichier quant à lui est le fichier "main.py". C'est celui que l'on doit exécuter dans la ligne de commande, il utilise les fonctions présentes dans le fichier "module.py".

La solution python fut la plus simple à programmer. En effet, la récupération du fichier json est plutôt simple en python. De plus, python peut facilement transformer la chaîne de caractères en un dictionnaire et facilement accéder à l'ID ou la longitude de chaque point par exemple. J'ai utilisé le module simpleKML, module qui m'a permis de facilement créer des fichiers kml. Il existait une fonction qui se nommait "newPoint" qui prenait en paramètre le nom, la longitude, la latitude et l'altitude et qui crée un point et l'ajouter dans le fichier KML. Donc après avoir récupéré le dictionnaire présent dans le fichier json, il ne reste qu'à sélectionner les points dont la confiance est supérieure à la moyenne de toutes les confiances puis de les ajouter dans le fichier kml à l'aide de la fonction newPoint du module simpleKML.

I.2 Solution C

La réalisation de la solution en langage C fut plus difficile. En effet, C ne dispose pas de module pour créer des fichiers kml. Il m'a donc fallu créer le fichier à partir de 0. De plus le langage C n'a pas de type dictionnaire. Donc pour chaque ligne du fichier json, il m'a fallu récupérer sous la forme d'entier l'ID, la longitude, la latitude et la confiance. Je pense que c'est cela qui m'a pris le plus de temps. Une fois toutes ces données récupérées, il m'a été facile de transformer le fichier json en fichier kml puisque j'ai pu faire comme avec le langage python. La création du fichier kml a la même configuration pour chaque fichier. Il m'a fallu faire 2 fonctions de plus permettant d'ajouter tous les points nécessaires dans le fichier kml. Bien sûr, cette conversion m'a pris plus de temps que le langage python. Le code est fait de la même manière que python. C'est à dire qu'il y a 2 fichiers, l'un se nommant "module.c" où se trouvent toutes les fonctions nécessaires à la conversion et l'autre se nommant "main.c" qu'il faut compiler et lancer pour que cela fonctionne.

II Choix des solutions

J'ai choisi d'utiliser le langage python car celui est très facile d'utilisation. Il possède beaucoup de modules comme par exemple le module simpleKML. Ce module m'a permis de réaliser sans problème un fichier kml. Je sais aussi utiliser le langage de programmation Python, c'est pour moi l'un des plus faciles car il est très complet. En revanche, du fait qu'il soit très complet, il ne faut pas se perdre lorsqu'on l'utilise. L'un des inconvénients que j'ai trouvé à utiliser ce langage est que je n'ai pas réussi à placer le nom du fichier directement dans le terminal.

Pour le langage C, je l'ai utilisé car on peut facilement mettre des fichiers dans la ligne de commande du terminal et ceci sont faciles à traiter. En effet, les fichiers placés après l'exécutable sont directement des arguments de la fonction main. Le seul défaut de C est qu'on ne peut pas créer de tableau avant d'avoir la taille du tableau. Cela rajoute donc une fonction qui calcule la taille du tableau. On peut ensuite allouer de la mémoire au tableau puis initialiser notre tableau. J'ai aussi utilisé le langage C car c'est l'un des langages dans lequel je me sens le plus à l'aise. Je suis aussi très à l'aise avec la programmation modulaire, je dirais même que je suis plus à l'aise avec cette programmation qu'avec la programmation orientée objet.

III Validation des choix

Pour vérifier si mes fichiers .kml obtenus fonctionnaient, j'ai décidé les charger sur google earth en ligne. Je suis arrivé à des trajectoires cohérentes et c'est pour cette raison que j'ai approuvé la génération des fichiers kml. Ci joint l'impression écran obtenu à gauche pour le langage python et à droite pour le langage C. Il s'agit du fichier traj1.json.



On remarque que ceci sont différents, je pense que cela est du à un traitement de la moyenne de la confiance différents entre les 2 langages.

Pour lancer le programme python, il faut au préalablement se placer dans le repertoire ou se situe le fichier "main.py". On ouvre ensuite un terminal et on écrit "python3 main.py". Il faut aussi modifier dans le code le repertoire dans lequelle se trouve le fichier json que vous souhaitez convertir.

De la même manière que pour python, il faut changer dans le code le répertoire dans lequel se trouve votre fichier .json. Pour lancer le programme C, placer vous dans le répertoire où se trouve le fichier main.c. Lancer la commande "gcc -o main main.c" pour compiler le programme. Ensuite, dans le même terminal, lancer la commande "./main nomfichier.json". Si le fichier existe, un fichier.kml sera créé dans le même répertoire où vous vous situez.

Conclusion

Pour conclure, je pense avoir passé environ 5h sur ce test. Plus précisément, j'ai du faire ce rapport pendant une heure, coder en python pendant une heure, faire des recherches sur ce qu'est un fichier json et fichier kml pendant une heure et coder en C pendant 2 heures. Je l'ai trouvé intéressant car j'ai pu moi même interpréter ce qu'est la confiance dans le fichier json étant donné que je n'ai pas pu le faire apparaître dans le fichier kml.