Python
Affichage de données GPS dans google.maps

tp_kml_2ap

Affichage de parcours au format kml

Le site https://www.google.com/maps permet d'afficher des images satellites de la terre à haute résolution et d'y superposer ses propres données en téléchargeant en fichier au format kml (Menu : Vos adresses → cartes → créer une carte).
KML (*Keyhole Markup Language*) est un format de fichier pour la modélisation et le stockage de caractéristiques géographiques comme les points, les lignes, les images, les polygones pour l'affichage sur ce site.
Une introduction au langage *kml* en français se trouve à :
https://developers.google.com/kml/documentation/kml_tut

Un récepteur **GPS** (http://fr.wikipedia.org/wiki/Global_Positioning_System) délivre diverses informations (position, direction, vitesse, date, heure, nombre de satellites en vue,....) sous forme de trames (suite de caractères) au format **nmea** (http://aprs.gids.nl/nmea/) . Il est facile de connecter un gps à un ordinateur[1] par un port série et de stocker toutes les informations délivrées par le récepteur GPS durant un parcours dans un fichier. On obtient un fichier constitué de trames qui commencent toutes par $ et une suite de 5 caractères qui identifie la nature des informations présentes dans la trame.
Voici un extrait du fichier obtenu :

```
$GPZDA,062238,03,02,2022,,*46
$GPGSV,1,1,00,,,,,,,,,,,,,,,,*79
$GPVTG,000.0,T,,M,000.0,N,000.0,K*60
$GPGGA,062239,3537.8333,N,13944.6667,E,0,00,99.9,0100,M,,M,000,0000*74
$GPGLL,3537.8333,N,13944.6667,E,062239,V*32
$GPRMC,062239,V,3537.8333,N,13944.6667,E,000.0,000.0,030222,,*04
$GPZDA,062239,03,02,2022,,*47
$GPGSA,A,1,,,,,,,,,,,,,99.9,99.9,
```

On va s'intéresser pour commencer uniquement aux trames **$GPGGA** car elles contiennent les principales informations dont nous avons besoin pour afficher le parcours.

La structure de cette trame est la suivante :

```
$GPGGA,062239,3537.8333,N,13944.6667,E,0,00,99.9,0100,M,,M,000,0000*74
```

heure latitude Nord/Sud longitude Est/Ouest qualité Altitude (invalide si 0)

1Avec un téléphone Android, vous pouvez enregistrer vos déplacements à partir du GPS intégré avec les applications "GPS logger". L'enregistrement peut se faire directement en kml. Les formats des données GPS peuvent être légèrement différents de celui de l'exercice.

Python
Affichage de données GPS dans google.maps

Voici un exemple de fichier au format *kml* qui permet l'affichage d'un parcours sous forme de lignes jaunes reliant les différents points relevés.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
  <Document>
    <name>Paths</name>
    <description> Mon parcours </description>
    <Style id="yellowLineGreenPoly">
      <LineStyle>
        <color>7f00ffff</color>
        <width>4</width>
      </LineStyle>
      <PolyStyle>
        <color>7f00ff00</color>
      </PolyStyle>
    </Style>
    <Placemark>
      <name>Absolute Extruded</name>
      <description> </description>
      <styleUrl>#yellowLineGreenPoly</styleUrl>
      <LineString>
        <extrude>1</extrude>
        <tessellate>1</tessellate>
        <altitudeMode>absolute</altitudeMode>
        <coordinates>
                  6.175453,43.115388,3
                  6.175447,43.115353,3
                  6.175480,43.115350,3
                  6.175508,43.115388,3
                  6.175493,43.115428,3
                  6.175415,43.115462,3
                  6.175327,43.115482,3
                  6.175203,43.115522,4
        </coordinates>
      </LineString>
    </Placemark>
  </Document>
</kml>
```

longitude, latitude, altitude

Le fichier se compose d'un en-tête, de balises décrivant le type et la couleur des traits et enfin entre les balises <coordinates> et </coordinates> des valeurs des coordonnées (dans l'ordre longitude,latitude,altitude) des points à relier.

La latitude doit être en degrés décimaux comprise entre -90 et +90 (positive nord).
La longitude doit être en degrés décimaux comprise entre -180 et +180 (positive est).
L'altitude doit être en mètres.

Un fois chargé sur le site ce fichier produit l'affichage du parcours superposé à l'image satellite de la zone.

Python
Affichage de données GPS dans google.maps

1/ Afficher un premier parcours

Votre travail consiste à écrire un programme capable d'extraire les informations latitude, longitude et altitude présentes dans les trames $GPGGA des fichiers fournis par un gps et de créer un fichier kml afin d'afficher le parcours.
Attention la première et la dernière ligne peuvent contenir des trames incomplètes.

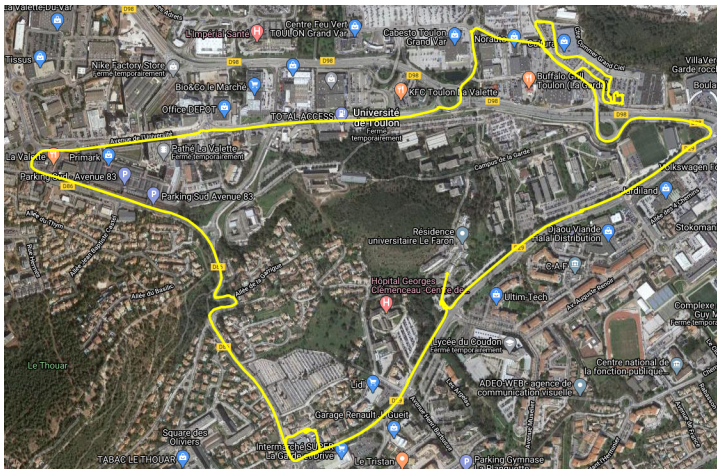
Voici quelques indications sur la manière de procéder :

On commence par créer un fichier *kml* contenant la partie invariante (depuis le début jusqu'à <coordinates> (par exemple en recopiant un fichier préalablement créé).
Ensuite il faut ensuite lire chaque ligne du fichier pour repérer les trames $GPGGA.
Pour les lignes $GPGGA, il faut d'abord voir si la qualité est correcte (champ qualité non nul).
Si c'est le cas il faut extraire les informations latitude,longitude et altitude, les convertir dans le format kml (degrés décimaux) et les recopier dans le fichier kml.
Lorsqu'on atteint la fin du fichier gps, il suffit d'écrire la fermeture des balises dans le fichier kml (</coordinates> </LineString> </Placemark> </Document></kml>).

Attention *kml* utilise les degrés décimaux alors que le gps donne ici des degrés sur les deux premiers chiffres puis des minutes fractionnaires (base 60).
Par exemple, on a dans le fichier GPS 4306.9122 pour 43° et 06.9122 minutes (minutes fractionnaires) qu'il faut convertir en 43.11520 degrés décimaux soit 43+(6.9122/60) à stocker dans le fichier KML.
-Écrire une fonction pour la conversion de format.

-Écrire une fonction vérifie la validité d'une trame GGA.

-Écrire une fonction qui retourne la latitude, la longitude et l'altitude d'une trame GGA.



2/ Exploitation de l'information de vitesse

On souhaite maintenant utiliser l'information de vitesse présente par les trames $GPVTG pour colorer le parcours en fonction de la vitesse. La couleur d'un segment doit être fonction de la vitesse à cet endroit du parcours. Le "style" va maintenant être intégré dans l'objet "Placemark" de manière à pouvoir faire varier la couleur :

```
<Placemark>
	<Style>
		<LineStyle>
			<color>7f0081ff</color>
			<width>8</width>
		</LineStyle>
	</Style>
	<LineString>
		<extrude>1</extrude>
		<tessellate>1</tessellate>
		<altitudeMode>absolute</altitudeMode>
		<coordinates>
			6.016946666666667,43.13399833333333,72.8
			6.016933333333332,43.13401,70.2
		</coordinates>
	</LineString>
</Placemark>
<Placemark>
	<Style>
		<LineStyle>
			<color>7f0037ff</color>
			<width>8</width>
		</LineStyle>
	</Style>
	<LineString>
		<extrude>1</extrude>
		<tessellate>1</tessellate>
		<altitudeMode>absolute</altitudeMode>
		<coordinates>
			6.016933333333332,43.13401,70.2
			6.016933333333332,43.13401,68.8
		</coordinates>
	</LineString>
</Placemark>
```

Tout les points d'un même "placemark" ont le même style et donc la même couleur.
A chaque changement significatif de la vitesse il faudra fermer le placemark et en créer un nouveau avec la nouvelle couleur.

2.1/ Conversion vitesse ↔ couleur

La couleur est définie en ABGR[2] :

Color and opacity (alpha) values are expressed in hexadecimal notation. The range of values for any one color is 0 to 255 (00 to ff). For alpha, 00 is fully transparent and ff is fully opaque. The order of expression is *aabbggrr*, where *aa=alpha* (00 to ff); *bb=blue* (00 to ff); *gg=green* (00 to ff); *rr=red* (00 to ff). For example, if you want to apply a blue color with 50 percent opacity to an overlay, you would specify the following: <color>7fff0000</color>, where *alpha*=0x7f, *blue*=0xff, *green*=0x00, and *red*=0x00.

2 Chercher "couleur html" pour avoir des sites qui permettent la visualisation en ligne. Attention html utilise l'ordre RGB et kml BGR

On cherche une loi de conversion vitesse vers couleur.
Comme la vitesse est un scalaire et la couleur un vecteur à trois composantes trouver une loi simple n'est pas facile. On va utiliser le système de couleur HSV[3].
Dans ce système la couleur (H=hue) est un scalaire[4]. Il va être facile de trouver une loi H=f(V). Ensuite il faudra convertir la couleur HSV en BGR.

Écrire la fonction qui convertie linéairement la vitesse en teinte :
Par exemple [0km/h ;100km] → [140;0] ([turquoise;rouge])

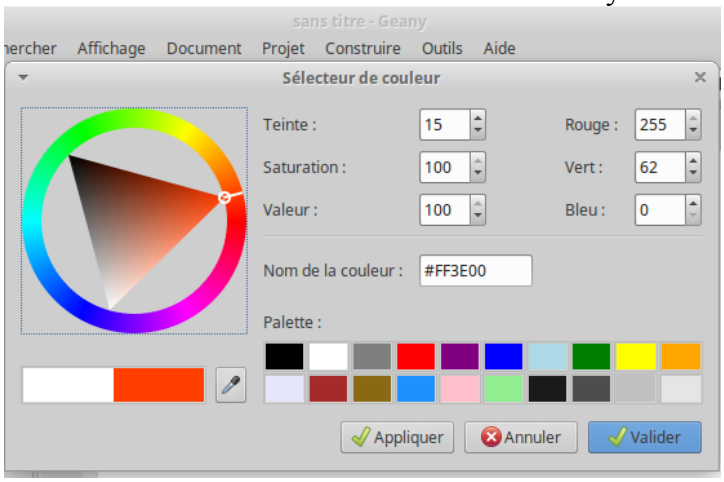
```
def speedToHue( speed, speedMin=0 , speedMax=100, hueMin=140,hueMax=0):
                ...
                return hue
```

Python permet la conversion entre systèmes de couleur
https://docs.python.org/3.7/library/colorsys.html

Écrire la fonction qui convertie une teinte en couleur kml. Attention les fonctions python travaillent entre 0 et 1 alors que en kml on travaille en 0 et 255 pour R,G,B.
De même pour HSV en python les valeurs sont entre 0 et 1 alors que la teinte H est un angle (0-360) et la valeur et la saturation sont en pourcentage (0-100)[5].

```
def colorKml( hue , sat=100,value=100):
                        …
                        return kmlColor
```

La fonction retourne une chaîne hexadécimale BGR au format kml (ou html en ajoutant # et en changeant l'ordre en RGB).
Vérifier vos résultats sur un site internet ou avec un logiciel permettant la sélection de couleur.
Par exemple, ci dessous avec le sélecteur de couleur html de Geany.



3 Voir aussi : http://arlotto.univ-tln.fr/arduino/article/thermometre-colore
4 On prendra S=saturation=1 (ou 100) et V=niveau=1 (ou 100)
5 Remarquez qui si la saturation et la valeur sont maximales alors une couleur RGB est maximale et une autre est nulle.

2.2 Création du fichier kml avec les couleurs fonction de la vitesse

- Écrire une fonction qui extrait la vitesse en km/h d'une trame VTG

On va changer la couleur du segment lorsque la vitesse change de plus ou moins N km/h.
Par exemple N = 5 km/h.

Algorithme général :

Recopier l'entête kml jusqu'à `<Placemark>`
Extraire les coordonnées de la première trame GGA valide
Extraire la vitesse de la trame VTG qui suit la trame GGA
vitesse_du_segment_courant ← vitesse
Déterminer la couleur
Écrire le "style" du premier segment
Jusqu'à la fin du fichier GPS faire :
    Extraire les coordonnées d'une trame GGA valide
    Extraire la vitesse de la trame VTG qui suit la trame GGA
    Si la vitesse diffère de la vitesse_du_segment_courant de plus de N km/h
        Fermer le segment courant
        Calculer la nouvelle couleur
        Ouvrir un nouveau segment[6,7] avec la nouvelle couleur
    Placer le point courant dans le segment
Fermer le segment courant
Écrire la fin du fichier kml

Voici un exemple de ce vous devriez obtenir, les couleurs peuvent varier en fonction des paramètres choisis.

6 Pour obtenir un parcours contigu il faut ré-écrire le dernier point du segment précédent lors de l'ouverture du nouveau segment.
7 Il peut être judicieux de faire une fonction pour fermer un segment et une autre pour ouvrir un segment d'une couleur passée en paramètre.

3. Exploiter les données de son téléphone

Avec un téléphone Android, vous pouvez enregistrer vos déplacements à partir du GPS intégré avec des applications de type "GPS logger".

L'enregistrement peut se faire directement en kml mais à ma connaissance la variation de la couleur avec la vitesse n'est pas proposée directement.

Les formats proposés (kml,gpx, txt, xml,...) peuvent différer de celui de l'exercice. Il faudra adapter votre programme. Vous pouvez avoir des trames préfixés différemment : GNGGA, BDGGA, GLGGA,… Il s'agit de différent systèmes similaires au GPS américain :

BD ou GB – Beidou (chinois),
GA – Galileo (européen),
GP – GPS (américain),
GL – GLONASS (russe).

Le préfixe GN est utilisé dans le cas de signaux mixés GPS + GLONASS.

3.1/ Colorisez votre propre parcours

Installer une application de ce type sur votre téléphone, sauvegarder un parcours puis générer un fichier kml colorisé avec la vitesse.

3.2/ Vérifier le respect du confinement

Durant le confinement, on peut sortir faire de l'exercice physique pendant une heure maximum sans s'éloigner de plus de 1km de son domicile.

- Écrire une fonction qui calcule la distance entre deux points en fonction de leurs coordonnées géographiques.

- Écrire un programme vérifiant le respect de la règle de confinement à partir du fichier issu de votre téléphone.
On considère que le premier point est le lieu du domicile.