Génération automatique de livrets de cartes OpenStreetMap
Mise en route et documentation

Table des matières

1 Introduction

Le script cree_carnet.py permet essentiellement de générer un fichier pdf rassemblant un succession de cartes le long d'une trace gps fournie en entrée. Il a été développé par JB (compte OpenStreetMap : JBacc1), et est mis à disposition sous licence FTWPL (licence publique + non-responsabilité). Évidemment, vous arriverez à y trouver des bugs, des suggestions d'améliorations, n'hésitez pas à me contacter, par messagerie OSM ou autre, ou sur le futur git du projet.

1.1 Principe de fonctionnement

Le principe de fonctionnement de ce script est le suivant :

1. L'utilisateur fournit une trace gps, un fichier de réglages et un fichier de données OSM ou un lien de téléchargement.
2. Des cadres qui représentent les cartes successives sont positionnés le long de la trace gps.
3. Une image des cadres sur la trace gps est générée.
4. Si les données OSM ne sont pas fournies, elles sont téléchargées.
5. Pour chaque cadre :
   (a) Le fichier de données OSM est découpé selon les limites du cadres.
   (b) La carte correspondant au cadre est générée.
6. Un fichier PDF est généré avec :
   (a) Une première de couverture (optionnelle)
   (b) Un texte d'introduction (optionnel)
   (c) Les cartes successives
   (d) Un récapitulatif des noms ou numéros de cartes
   (e) L'image de l'assemblage des cadres sur la trace gps

1.2 Outils utilisés

Le tout est géré par du code python qui fait appel aux ressources suivantes :
– 2 : Script Python et paquet gpxpy
– 3, 5b : Maperitive avec une feuille de style
– 5a : Osmosis (Java)
– 6 : Latex et divers paquets complémentaires

1.3 Autres informations et avertissements

Maperitive a été développé en C#. Le portage des bibliothèques graphiques de .NET vers Mono est assez pauvre. Les rendus de cartes sous Linux et Mac sont dégradés par rapport à un fonctionnement sous Windows.

Le code a été développé et testé sous Windows, une utilisation réussie sous Linux d'une première version du code a été rapportée.

Le code contient probablement des bugs, n'hésitez pas à les rapporter en utilisant les contacts fournis ici ou sur la future page git du projet. À ceux qui oseront y mettre le nez, il est crado.

1.4 Licences

Le code source du projet est mis à disposition sous licence FTWPL (licence publique + non-responsabilité).

Cette documentation est mise à disposition sous licence CC-by-sa 4.0.

1.5 Contact

Contact OpenStreetMap : JBacc1, http ://openstreetmap.org/user/JBacc1

2 Téléchargements et installation

2.1 Installation de Maperitive

À télécharger à partir de http://maperitive.net/download/Maperitive-latest.zip. À décompresser, l'exécutable Maperitive.exe et la console Maperitive.Console.exe sont dans le répertoire principal. Sous Linux/Mac, nécessite Mono.

Pour rappel, le rendu Maperitive sous Linux/Mac est dégradé par rapport à une utilisation sous Windows.

2.2 Installation du rendu R25

Si vous comptez utiliser le rendu R25 pour vos cartes (ce qui n'est pas indispensable !), à télécharger sous http://jb.tradfrance.com/R25.zip.

À décompresser et à placer dans le dossier Maperitive pour respecter l'enchaînement : (Maperitive-latest/)Maperitive/R25(/icons, /textures, /rules). Si vous observez des grosses taches noires dans vos cartes, c'est que le dossier R25 n'est pas situé au bon endroit.

Repérez l'emplacement du fichier R25.mrules dans le dossier R25/rules/.

2.3 Installation de Latex

Par exemple à partir de : http://latex-project.org/ftp.html. Repérez l'empacement de l'exécutable pdflatex.exe, peut-être dans Program Files/MiKTeX 2.9/ miktex/bin/x64/.

Lors de la première utilisation, l'installation de divers paquets complémentaires sera demandée.

2.4 Installation de Java

Osmosis nécessite un runtime Java. Installation à partir de : http://www.java.com/fr/download/

2.5 Installation d'Osmosis

Les données OpenStreetMap sont prétraitées à partir des scripts Osmosis. Renseignements sur la page du wiki d'OSM : http://wiki.openstreetmap.org/wiki/Osmosis. Téléchargement à partir de : http://bretth.dev.openstreetmap.org/osmosis-build/osmosis-latest.zip, à décompresser. Repérer l'emplacement du fichier osmosis.bat, sous osmosis-latest/bin/.

2.6 Installation de Python 3

Voir les détails à l'adresse : https://www.python.org/downloads/

2.7 Installation du paquet gpxpy

Le paquet gpxpy est utilisé pour la gestion des traces gps. Téléchargement à partir de : https://pypi.python.org/pypi/gpxpy/0.9.8 puis à installer.

2.8 Téléchargement des scripts

À partir de http://jb.tradfrance.com/CarnetRando.zip, à décompresser. Le script principal cree_carnet.py et les fichiers de ressources et de réglages ressources.ini et CarnetRando.ini sont dans le répertoire principal.

2.9 Renseignement du fichier ressources.ini

Le fichier ressources.ini indique aux scripts où trouver les ressources nécessaires à leur fonctionnement. Voici la liste des paramètres à renseigner.

[**RESSOURCES**]

osmosis Préciser le chemin d'accès au fichier osmosis.bat.
osmosis = C :/OSM/osmosis-latest/bin/osmosis.bat

maperitiveconsole Chemin d'accès à Maperitive.Console.exe.
maperitiveconsole = C :/OSM/Maperitive/Maperitive.Console.exe

pdflatex Chemin d'accès au fichier pdflatex.exe.
pdflatex = C :/Program Files/MiKTeX 2.9/miktex/bin/x64/pdflatex.exe

pdfreader Chemin d'accès à un lecteur pdf.
pdfreader = C :/Program Files/Sumatra/SumatraPDF.exe

pngreader [non-indispensable] Chemin d'accès à un lecteur de fichier png.

defaultmrules Chemin d'accès à la feuille de style Maperitive utilisée pour le rendu des cartes.
defaultmrules = C :/OSM/Maperitive/R25/rules/R25.mrules

assemblagemrules Chemin d'accès à la feuille de style Maperitive utilisée pour le rendu de la carte d'assemblage.
assemblagemrules = C :/OSM/CarnetRando/Cadres.mrules

2.10 Test de l'installation

Dans une fenêtre de commandes et à partir de l'emplacement du script cree_carnet.py, lancer :
python cree_carnet.py -gpx=demo.gpx -ini=demo.ini

Si le script fonctionne sans erreur, un carnet contenant la description d'une traversée de la Chaine des Puys devrait s'ouvrir en fin d'exécution. Sinon, vérifiez l'installation des différents éléments et le contenu du fichier ressources.ini.

3 Préparation de la première utilisation

3.1 Renseignement des données principales du fichier .ini

Pour fonctionner correctement, le fichier CarnetRando.ini nécessite d'être renseigné convenablement. Ici sont listés uniquement les paramètres essentiels. Une description complète est proposée dans la suite de la documentation.

3.1.1 [DATA]

Données OpenStreetMap qui seront utilisées pour le rendu des cartes. Voir également la partie 3.2 « Où trouver les données OpenStreetMap ? »

osmdata Fichier de données OSM qui sera utilisé pour la création des cartes. Format .pbf (ou .osm.pbf) préféré, .osm également supporté. Si le fichier n'est pas présent, il sera téléchargé et enregistré sous ce nom.
osmdata = bourgogne-latest.osm.pbf

osmdata_downloadurl Lien de téléchargement de données openstreetmap si le fichier renseigné sous [DATA][osmdata] n'est pas présent.
osmdata_downloadurl = http ://download.geofabrik.de/europe/france/bourgogne-latest.osm.pbf

elevation_source Source pour les données de relief. La source doit être reconnue par Maperitive. Par défaut, utiliser SRTMV3R3.
elevation_source = SRTMV3R3

3.1.2 [PAGE]

Formatage des cartes et du document final.

format Format de la page, du A6 au A3.
format = A5

Il est également possible d'utiliser un format de page personnalisé. La syntaxe est : custom_dimension1_dimension2, les dimensions étant précisées en centimètres. L'ordre des dimensions est sans importance, le format portrait/paysage est précisé dans [PDF][pdf_portrait]
format = custom_20.3_16.4

marginouter, margintop, marginouter, marginbottom Marges respectivement de l'extérieur de la page, du haut, de l'intérieur et du bas en centimètres.
marginouter = 0
margintop = 0
margininner = 0.75
marginbottom = 0

3.1.3 [PDF]

premiere_page Permet de renseigner une image .png à utiliser pour la première de couverture du fichier PDF. Si non renseigné, le pdf commence directement à la section suivante.
premiere_page = C :/OSM/CarnetRando/Cartes/volcan_1er.png

introduction Permet de renseigner un fichier au formatage .tex qui sera placé après la première de couverture (si fournie). Si non renseigné, le PDF passe directement à la section suivante. Le fichier doit être encodé en utf-8 pour être lu correctement !
introduction = ./Latex/intro_vosges.txt

pdf_portrait Indique si le carnet est proposé en format portrait ou non (paysage).
pdf_portrait = False

3.1.4 [MAP]

printgpx Imprime la trace gpx sur les cartes.
printgpx = true

mapscale Échelle des cartes produites. Renseigner 25000 pour des cartes au 1:25000.
mapscale = 25000

dpi_map Résolution (en DPI) des cartes. En cas d'incertitude, une valeur de 400 peut faire une bonne moyenne.
dpi_map = 400

recouvrement_senscourt, recouvrement_senslong Longueur de recouvrement en centimètres des cartes successives lors de la découpe de traces gps. La valeur représente la somme des recouvrements des deux cotés de la page.
recouvrement_senscourt = 3.5
recouvrement_senslong = 4

optimise_recouvrement Augmente les paramètres de recouvrement au maximum en gardant le même nombre de cadres. Permet une meilleure répartition des cartes le long de la trace gps.
optimise_recouvrement = yes

forceportrait Si à True, lors de la découpe d'une trace gps, les cartes seront toutes en format portrait. Sans effet si forcelandscape est à True.
forceportrait = false

forcelandscape Idem, force les cartes en format paysage.
forcelandscape = false

3.1.5 [ASSEMBLAGE]

printgpxonassemblage Imprime la trace gpx sur l'image de l'assemblage des cadres.
printgpxonassemblage = true

printshadingonassemblage Imprime l'ombrage du relief en fond de la carte d'assemblage.
printshadingonassemblage = true

assemblagemapscale Échelle de la carte d'assemblage. Renseigner 300000 pour une échelle de 1:300000.
assemblagemapscale = 300000

dpi_assemblage Résolution (en DPI) de l'assemblage. À adapter selon l'étendue de l'assemblage et la présence éventuelle d'ombrage.
dpi_assemblage = 400

assemblage_leftmargin, assemblage_topmargin, assemblage_rightmargin, assemblage_bottommargin Marge ajoutée autour de la trace gps sur l'image d'assemblage des cadres. En unité de latitude et longitude.
assemblage_leftmargin = 0.005
assemblage_topmargin = 0.005
assemblage_rightmargin = 0.005
assemblage_bottommargin = 0.005

Les sections [PATH], [BEHAVIOUR], [OSM] peuvent rester inchangées dans un premier temps. Elles sont décrites dans la suite de la documentation.

3.2 Où trouver les données OpenStreetMap ?

La solution la plus pratique, rapide, à jour, est probablement d'utiliser les extraits .pbf fournis gratuitement par Geofabrik. Si vos cartes restent à l'intérieur

d'une région, utilisez l'extrait correspondant sur : http://download.geofabrik.de/europe/france.html. Dans les autres cas, osmosis permet de rassembler les extraits de deux régions en un seul avec une petite ligne de commande ; en dernier recours, utilisez l'extrait de la France complète, en conservant activée l'option de prédécoupe dans le fichier .ini.

Le format .pbf est préféré pour un traitement accéléré, mais le format .osm est également géré.

3.3 Où trouver une trace GPS ?

Dans votre gps ?

Sinon, internet peut vous aider, par exemple le site de routage GraphHopper : https://graphhopper.com/maps vous permet de trouver un initéraire piéton (également vélo et voiture) entre un point de départ et d'arrivée, en forçant le passage par des points intermédiaires. Le fichier .gpx exportable peut être directement utilisé. Le site d'OpenRouteService peut également être utilisé : http://openrouteservice.org/.

La partie exploitée du fichier .gpx doit être structurée en : gpx/trk/trkseg/trkpt

Note : un fichier gpx exporté par OSRM n'est pas conditionné de cette manière et n'est pas géré.

4 Utilisation

4.1 Utilisation basique

python cree_carnet.py -gpx=Volvic_Aydat.gpx

4.2 Utilisation normale

4.2.1 Création d'un fichier .ini spécifique

Dupliquer le fichier CarnetRando.ini et le renommer au nom du projet (volcans.ini). Renseigner les champs [DATA].

Renseigner également les champs [PAGE].

4.2.2 Vérifier les paramètres de page

Note : l'essentiel du travail est réalisé automatiquement lorsque le paramètre optimise_recouvrement du fichier .ini est à True.

Lancer le script avec l'option -ao (--assemblageonly) en précisant le fichier .ini : python cree_carnet.py -gpx=Volvic_Aydat.gpx -ini=volcans.ini -ao

Le fichier assemblage.png est ouvert en fin de génération si vous avez indiqué un exécutable [FILES][pngreader], sinon, ouvrez-le manuellement. Vous pouvez alors ajuster les paramètres de mise en page, notamment : [ASSEMBLAGE][assemblagemapscale], [MAP][recouvrement_senslong] et [MAP][recouvrement_senscourt].

Recommencez jusqu'à ce que le résultat vous convienne.

4.2.3 Lancer une première génération

python cree_carnet.py -gpx=Volvic_Aydat.gpx -ini=volcans.ini

Si vous avez renseigné un exécutable [FILES][pdfreader], le fichier Carnet.pdf est ouvert en fin de script. Vous pouvez alors :

- Ajouter un nom à chaque carte : ce nom sera indiqué dans le pdf sur les cartes en haut à gauche et dans le récapitulatif. Ouvrez le fichier cadres.csv (ou [PATH][cadrescsv] si renseigné) et modifiez les valeurs de la dernière colonne, par exemple « 1. Gare de Volvic ». La plupart des caractères spéciaux devraient être reconnus (mais pas le séparateur du fichier csv !).
  Si vous souhaitez modifier le nom sous lesquels les images des cartes (et les fichiers de découpe .osm) seront enregistrés lors d'une nouvelle exécution du script, modifiez la première colonne du fichier. Attention à ne pas y utiliser de caractères spéciaux !
- Créer une première de couverture (fichier .png), en respectant les dimensions format de la page (et donc les marges), renseignez-le dans le fichier .ini sous [PDF][premiere_page]
- Éventuellement créer un texte d'introduction (syntaxe latex), renseignez [PDF][introduction]. Vous pouvez prendre exemple sur les fichiers fournis dans le répertoire /Latex.

4.2.4 Lancer la génération finale

Si vous n'avez pas modifié la première colonne du fichier [PATH][cadrescsv] qui contient les noms sous lesquelles les cartes sont enregistrées, relancez la génération de l'assemblage (pour y afficher les noms de cadres) et la génération du pdf avec :
python cree_carnet.py -pdf=cadres.csv -gpx=Volvic_Aydat.gpx -ini=Volcans.ini

Cette ligne de commande contenant l'option -gpx force la régénération de l'assemblage, elle remplace les deux commandes :
python cree_carnet.py -csv=cadres.csv -gpx=Volvic_Aydat.gpx -ini=Volcans.ini -ao
python cree_carnet.py -pdf=cadres.csv -ini=Volcans.ini

Si vous avez modifié la première colonne du fichier de cadres, vous devez générer à nouveau les cartes avant de produire le pdf. Utilisez la commande :
python cree_carnet.py -csv=cadres.csv -ini=Volcans.ini

4.2.5 Corriger une petite erreur

Zut, j'ai une faute d'orthographe dans mon introduction/une boulette sur ma première de couverture.

Corrigez l'erreur puis générez uniquement le pdf à partir des cartes (et du fichier de cadres) existantes :
python cree_carnet.py -pdf=cadres.csv -ini=volcans.ini

4.2.6 Autres options

Quelques options sont désactivables (activables ?) à partir de la ligne de commande.

--assemblageonly, -ao Stoppe le script après la génération de l'image de l'assemblage et ouvre l'image.

--nopdf, -np Ne génère pas le pdf en fin de script.

--rotatepage, -rp Les pages contenant des cartes en format paysage (ou en format inversé) sont pivotées automatiquement dans le pdf final (pdf contenant des pages dans les deux formats).

--nooddrotation, -nor Sans cette option, les cartes en format paysage (ou en format inversé) sur les pages impaires sont pivotées pour présenter le nord du coté de la reliure. Avec -nor, ces pages ne sont plus pivotées. Sans effet si --rotatepage.

--forceportrait, --forcelandscape, -fp, -fl Ces options pour forcer le format des cadres lors de la découpe de la trace gps peuvent également être activées à partir de la ligne de commande. --forceportrait sans effet si --forcelandscape est activé.

Exemple d'utilisation
python cree_carnet.py -gpx=Volvic_Aydat.gpx -ini=Volcans.ini -rp -fp

5 Fichier .ini : dernières sections

Les sections [FILES], [DATA], [PAGE], [PDF], [MAP] et [ASSEMBLAGE] ont été décrites dans la section 3.1. Ici sont renseignées les sections [PATH], [BEHAVIOUR] et [OSM], ainsi que le détail pour un format de page personnalisé.

5.1 [PATH]

Chemin d'accès utilisé pour la création des fichiers nécessaires lors de l'exécution des scripts.

osmfiles osmfiles = ./osmfiles

scripts scripts = ./scripts

latex latex = ./Latex

cartes cartes = ./Cartes

5.2 [FILES]

cadrescsv Nom (et chemin d'accès éventuel) du fichier de cadres qui sera utilisé s'il est créé lors du script.
cadrescsv = cadres.csv

assemblageosm Nom (et chemin d'accès éventuel) du fichier d'assemblage qui sera utilisé s'il est créé lors du script.
assemblageosm = ./osmfiles/assemblage.osm

5.3 [BEHAVIOUR]

Comportement du script.

forcedownload_osmdata Force le téléchargement du fichier [DATA][osmdata] même si ce dernier est déjà présent.
forcedownload_osmdata = false

remove_tempfiles Supprime les fichiers intermédiaires non utilisés pour le traitement final.
remove_tempfiles = true

remove_osmfiles Supprime les fichiers .osm créés pour la génération des cartes après leur utilisation.
remove_osmfiles = false

5.4 [OSM]

Paramètres de découpe du fichier [DATA][osmdata] selon les différents cadres pour générer les cartes. En cas d'incertitude, conservez les paramètres initiaux.

cut1 Indique si le fichier [DATA][osmdata] doit être découpé pour rendre les différentes cartes. Nécessaire dès que le fichier [DATA][osmdata] est lourd (c'est-à-dire toujours).
cut1 = true

cut1_completerelations Indique à osmosis l'option completeRelations=yes lors de la découpe du cadre.
cut1_completerelations = true

cut1_horizontalmargin Largeur des marges à droite et à gauche ajoutées au cadre lors de la découpe des données osm. Unité de longitude.
cut1_horizontalmargin = 0.0015

cut1_verticalmargin Largeur des marges du haut et du bas ajoutées au cadre lors de la découpe des données osm. Unité de latitude.
cut1_verticalmargin = 0.002

cut2 Indique si une seconde découpe doit être effectuée sur le fichier issu de [cut1]. Sans effet si [cut1] est à False.
cut2 = true

cut2_completerelations, cut2_horizontalmargin, cut2_verticalmargin Identiques à [cut1]. Utile pour alléger un fichier lorsque des relations de type sub-area, collection sont trop lourdes même si des multipolygones de landuses sont étendus.
cut2_completerelations = false
cut2_horizontalmargin = 0.05
cut2_verticalmargin = 0.05

precut1, precut2 Permet une prédécoupe du fichier [DATA][osmdata]. Les découpes [cut1] des cadres successifs se font sur cet extrait plus léger et sont plus rapides. À privilégier en cas de fichier [DATA][osmdata] de grande taille ou d'un grand nombre de cartes à générer (c'est-à-dire toujours). precut2 sans effet si precut1 = False

Les paramètres suivants sont comparables à ceux de [DATA][cut1].
precut1 = true
precut1_completerelations = true
precut1_horizontalmargin = 0.01
precut1_verticalmargin = 0.015
precut2 = true
precut2_completerelations = false
precut2_horizontalmargin = 0.15
precut2_verticalmargin = 0.15

6 Usages spécifiques

6.1 Mode Atlas

Le mode Atlas permet de couvrir une zone avec un quadrillage de cartes. Il est appelé avec l'option --atlas (-atl). La méthode à utiliser est comparable à celle de l'utilisation basique :

- Déterminer les limites géographiques à couvrir en longitude, latitude. Vous pouvez utiliser par exemple Maperitive, en ajustant la zone d'impression (clic droit sur la carte, Place Printing Bounds Here), puis en copiant les paramètres Bounds de la fenêtres de propriétés (F4 si non affichée). L'ordre à passer au script est identique à celui de Maperitive : gauche, bas, droite, haut (sans espace entre les virgules et les nombres dans la ligne de commande).
- Lancer la génération de l'assemblage avec l'option --assemblageonly :
  python cree_carnet.py -atl=2.92,45.68,3.02,45.87 -ao
- Ajuster les paramètres de recouvrement dans le fichier .ini si nécessaire et relancer la pré-génération.
- Lancer la génération globale avec :
  python cree_carnet.py -atl=2.92,45.68,3.02,45.87
- Si nécessaire, modifier les noms des cartes dans le fichier csv, la première de couverture, l'introduction dans le fichier .ini, et relancer la génération de l'assemblage et du pdf :
  python cree_carnet.py -csv=cadres.csv -ao
  python cree_carnet.py -pdf=cadres.csv

À noter également :

- Le format des cartes est en portrait par défaut. Passage en paysage avec l'option --forcelandscape (-fl).
- Par défaut, les cartes sont classées par lignes. Pour les classer par colonne dans le document, utiliser l'option --atlascolumn (-col).

Les autres options --inifile (-ini), --nopdf (-np), --rotatepage (-rp) et --noodd-rotation (-nor) restent accessible. Il est également possible de sur-imprimer une trace gps sur les cartes ou l'assemblage en la passant en argument :
python cree_carnet.py -atl=2.92,45.68,3.02,45.87 -gpx=Volvic_Aydat.gpx

Note : On rappellera que la terre n'est pas plate... et que pour la couverture d'une zone géographique étendue, le recouvrement entre les cartes du nord sera plus important que celui des cartes du sud. Les paramètres indiqués dans le fichier .ini sont portés sur la ligne de cartes du sud afin de toujours garantir une zone de recouvrement.

6.2 Créer manuellement le fichier de cadres

Je ne veux pas qu'un pauvre script sans intelligence crée des cadres le long d'un trace gps. Je veux positionner moi-même les cadres, là où je veux. Et même avoir des échelles différentes pour certaines cartes, pour bien perdre le randonneur. Renseigner intégralement le fichier cadres.csv à la main serait laborieux, voilà une méthode simplifiée.

6.2.1 Renseigner le tableau pour l'importation

Ouvrir le fichier imp_cadres.csv fourni, l'en-tête vous indique les champs à renseigner, utilisez une ligne par carte.

Pour vous guider, ouvrez Maperitive (Maperitive.exe), puis :

- Clic droit sur la carte, « Place Printing Bounds Here »
- F4 pour afficher la fenêtre Properties (si non visible)
- Régler « Fixed Paper » à True
- Renseigner Paper Type et Margins (attention, ici en millimètres), ils doivent correspondre aux informations renseignées dans le fichier .ini pour le pdf final.
- Renseigner Orientation.
- Renseigner la valeur de Map Scale. Notez que les cartes peuvent être à des échelles différentes les unes des autres.
- Si l'emplacement du cadre ne vous convient pas, le déplacer (il s'agit du cadre bleu, glissable à partir des bordures lorsque sélectionné, quand le pointeur passe en quadruple flèche).
- Renseigner à nouveau Map Scale, qui est recalculé quand le cadre est déplacé.

Lorsque l'emplacement et l'échelle du cadre vous conviennent, copiez les valeurs X et Y de « Map Center » dans la ligne et les colonnes correspondantes du tableau imp_cadres.csv. Renseignez également les autres cases de la ligne :

- Référence : Ne pas laisser vide, ne pas renseigner deux fois la même valeur. Dans le doute, utilisez une numérotation 1, 2, 3... Pas de caractères spéciaux,

mais le « _ » est supporté.
- Scale : 25000 pour une échelle de 1:25000
- Portrait/Landscape
- Format : De A6 à A3
- DPI : En cas d'incertitude, une valeur de 400 peut faire une bonne moyenne haute
- Description : Texte qui sera utilisé comme titre de la carte dans le pdf. La plupart des caractères spéciaux devrait être prise en compte

6.2.2 Importer le tableau

Lancer le script cree_carnet.py avec l'option d'importation --import2csv (-imp) pour générer le fichier de cadres [FILES][cadrescsv] au format standard :
python cree_carnet.py -imp=imp_cadres.csv

L'option --assemblageonly permet de générer également directement le fichier .osm d'assemblage [FILES][assemblageosm] et l'image de l'assemblage :
python cree_carnet.py -imp=imp_cadres.csv -ao

L'ouverture sous Maperitive du fichier .osm d'assemblage avec la feuille de style [FILES][assemblagemrules] peut aider à positionner les cadres successivement les uns par rapport aux autres. La représentation du fichier de cadres sous Maperitive sera automatiquement mise à jour lors sa ré-importation par le script.

6.2.3 Lancer la génération à partir des cadres importés

Une fois le fichier de cadres complet et importé, vous pouvez le passer au script et lancer la génération des cartes et du pdf à partir de lui en utilisant l'option --csvfile (-csv) :
python cree_carnet.py -csv=cadres.csv

Notez que la surimpression d'une trace gps sur les cartes et l'assemblage reste possible si renseigné dans [PDF][printgpx] et [PDF][printgpxonassemblage] avec l'option -gpx :
python cree_carnet.py -csv=cadres.csv -gpx=Volvic_Aydat.gpx

Les autres options --inifile (-ini), --assemblageonly (-ao), --nopdf (-np), --rotatepage (-rp) et --nooddrotation (-nor) restent disponibles.

6.3 Modifier un fichier de cadres généré automatiquement

Vous avez lancé la découpe d'une trace gpx en cadres successifs, mais malheur, le cadre numéro 32 serait mieux situé deux kilomètres plus à l'ouest ? Et le fichier cadres.csv est tellement mal fichu qu'il en est impossible à manipuler ? Pas de

problème, voilà comment l'exporter, libre à vous de le modifier et de le ré-importer ensuite selon la méthode décrite au point 6.2.

Pour exporter un fichier .csv de cadres, utilisez l'option --exportcsv (-exp) : python cree_carnet.py -exp=cadres.csv

Le fichier exporté est créé sous le nom (cadres)_export.csv et est au format nécessaire pour permettre sa ré-importation avec --import2csv.

7 Problèmes classiques

7.1 Les cartes sont pleines de tâches noires

Le rendu R25 est mal installé, vérifiez l'emplacement du dossier R25 comme indiqué au point 2.2.

7.2 Caractères spéciaux mal rendus dans l'introduction

Vérifiez l'encodage du fichier [PDF][introduction], il doit être en utf-8. Notepad ++ par exemple permet de modifier l'encodage aisément (Menu Encoding/ Encode in UTF-8) : https://notepad-plus-plus.org/

7.3 Autres problèmes

N'hésitez pas à me faire remonter d'autres problèmes, par les contacts proposés ou sur la future page git du projet.