université virtuelle Burkina Faso

Sujet	Projet de fin de cours
Professeur	Dr. W. OUEDRAOGO
Titre	Mini Google-Maps
Auteurs	Coulibaly Cheick Ahmed et Traoré Soungalo

1. Recherche de Lieux sur Google Maps (Gmaps) ou OpenStreetMaps (OSM)

Pour la recherche des lieux nous avons utilisé Google Maps, en enregistrant les 5 lieux dans les chefs-lieux des 13 régions que voici :
1. Boucle du Mouhoun : Dédougou
2. Cascades : Banfora
3. Centre : Ouagadougou
4. Centre-Est : Tenkodogo
5. Centre-Nord : Kaya
6. Centre-Ouest : Koudougou
7. Centre-Sud : Manga
8. Est : Fada N'Gourma
9. Hauts-Bassins : Bobo-Dioulasso
10. Nord : Ouahigouya
11. Plateau-Central : Ziniaré
12. Sahel : Dori
13. Sud-Ouest : Gaoua

Nous avons d'abord enregistré les lieux parmi nos favoris, puis nous les avons exportés avec Google Takeout dans un fichier nommé adresses_enregistrees.json.

2. Informations concernant chaque lieux

Vu le nombre important d'informations à noter, nous avons préféré les enregistrer dans le fichier lieux.json.
Les données exportées avec takeout (adresses_enregistrees.json), on été formatées à l'aide du script python preprocess.py afin d'obtenir un format conforme aux exigences de l'énoncé et compatible avec une insertion dans une base de données MongoDB : lieux.geojson.

De plus, des modifications manuelles ont été apportées au fichier lieux.json, notamment pour l'ajout des différents horaires d'ouverture des lieux, ces informations n'étant pas exportées avec Takeout.

3. Importation des données dans MongoDB

Les données ont été importés dans notre BD : minigmaps, dans la collection lieux à l'aide d'un script python bd_import.py.

4. Fonction de filtre

La fonction `filtrer(s, t, lon, lat)` du fichier filtrer.py :
1/ récupère les lieux depuis MongoDB filtrés par `s` (nom ou type).
2/ calcule la distance entre chaque lieu et (`lon`, `lat`).
3/ vérifie si le lieu est ouvert à la date et l'heure données (`t`).
4/ trie les lieux par distance et retourner les résultats : `nom`, `ville`, coordonnées (`longitude`, `latitude`), disponibilité (`ouvert` ou `fermé`).

5. Chargements des Données dans QGIS

Pour l'analyse des données, nous avons utilisé le découpage administratif du Burkina Faso, bfa_adm_igb_20200323_shp, utilisé lors du TP5.

Les données lieux.json ont été exportées en lieux.csv en rajoutant la région correspondant à chaque lieu, à l'aide du script formater.py.

La division administrative bfa_admbnda_adm1_igb_20200323.shp a été chargée sur QGIS, et le fichier lieux.csv a été ajouté comme une couche de texte délimité.

Nom du projet QGIS : minigmaps.qgz

6. Génération de la carte Choroplèthe

Pour la création de la carte choroplèthe des régions du Burkina en fonction du nombre de restaurants au kilomètre carré, on a d'abord filtré notre lieux.csv pour ne garder que les restaurants dans le fichier statistics, ensuite compté le nombre de restaurants par région et enfin effectué une jointure entre statistics et bfa_admbnda_adm1_igb_20200323.shp.

La carte choroplèthe a été exportée sous forme image sous le nom de restaurants.png

7. Script python dans QGIS

Pour ordonner les régions du Burkina par le nombre de restaurants au km², il faut :

1. Avoir les deux couches dans QGIS :
- la couche des régions (bfa_admbnda_adm1_igb_20200323.shp) contenant les limites et superficies des régions
- la couche des statistiques (statistics) avec le nombre de restaurants par région.

2. Associer les restaurants aux régions :
- pour chaque région, récupérer le nombre de restaurants depuis statistics
- récupérer sa superficie en km² depuis bfa_admbnda_adm1_igb_20200323.shp

3. Calculer la densité (restaurants/km²) pour chaque région

4. Trier les résultats par densité décroissante (du plus grand au plus petit).
Le script python : ordonner.py
Le résultat est exporté dans le fichier : densite_restaurants.csv (dans le dossier projet_&_plans_QGIS)