

ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

www.polytech.univ-tours.fr

Projet Recherche & Développement 2015-2016

GéoPiéton : Génération d'un réseau piéton à partir d'un réseau routier

Entreprise

GéoVélo

Tuteurs entreprise

Antoine GIRET

Étudiants

Maxime BRASSART (DI5)

Tuteurs académiques

Antoine GIRET

Liste des intervenants

Entreprise

GéoVélo

antoine.giret@geovelo.fr

| Nom | Mail | Qualité |
|-----------------|--|-------------------|
| Maxime BRASSART | maxime.brassart@univ-tours.fr | Étudiant DI5 |
| Antoine GIRET | antoine.giret@geovelo.fr | Tuteur académique |
| Antoine GIRET | antoine.giret@geovelo.fr | Tuteur entreprise |

Avertissement

Ce document a été rédigé par Maxime Brassart susnommé l'auteur.

L'entreprise GéoVélo est représentée par Antoine Giret susnommé le tuteur entreprise.

L'école polytechnique de l'université François Rabelais de Tours est représentée par Antoine Giret susnommé le tuteur académique.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable du tuteur académique et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.

Pour citer ce document :

Maxime Brassart, *GéoPiéton : Génération d'un réseau piéton à partir d'un réseau routier*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2015-2016.

```
@mastersthesis{
  author={Brassart, Maxime},
  title={GéoPiéton : Génération d'un réseau piéton à partir d'un réseau routier},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2015-2016}
}
```

Table des matières

| | |
|---|----------|
| Introduction | 1 |
| I Recherche | 2 |
| 1 Présentation | 3 |
| 1 Contexte..... | 3 |
| 2 Problématique | 3 |
| 3 Objectif final..... | 3 |
| 2 Génération du réseau piéton | 5 |
| 1 Généralités sur les réseaux | 5 |
| 2 Le réseau piéton..... | 6 |
| 3 Application existantes..... | 8 |
| 3.1 Google Map..... | 8 |
| 3.2 Mappy | 9 |
| 3.3 Université américaine..... | 10 |
| 4 Valeurs ajoutées par GéoPiéton..... | 10 |
| 5 Recherche et algorithme concernant la génération du réseau piéton | 10 |
| 5.1 Approche..... | 10 |
| 5.2 Méthode : "Network Buffering" | 12 |
| 5.2.1 Principe..... | 12 |
| 5.2.2 Données d'entrées..... | 13 |
| 5.2.3 Algorithme..... | 14 |
| 5.3 Méthode : "Collaborative Mapping"..... | 15 |
| 5.3.1 Principe..... | 16 |
| 5.3.2 Données d'entrées..... | 16 |
| 5.3.3 Algorithme..... | 16 |
| 5.4 Méthode : "Image Processing" | 17 |
| 5.4.1 Principe..... | 17 |

| | | |
|-----------|--|-----------|
| 3 | Solution retenue et adaptée | 18 |
| 1 | Données en entrées..... | 18 |
| 2 | Algorithme adapté..... | 19 |
| 3 | Critiques et améliorations | 23 |
| 4 | Outils et technologies utilisées | 24 |
| 1 | Open Street Map (OSM) | 24 |
| 2 | GeoFabrik | 24 |
| 3 | PostgreSQL et PostGIS | 25 |
| 4 | OSMOSIS | 26 |
| 5 | QGIS | 26 |
| II | Développement | 27 |
| 5 | Méthodologie | 28 |
| 1 | Méthode de travail..... | 28 |
| 2 | Planning..... | 28 |
| 3 | Outils | 29 |
| 4 | Qualité du code | 29 |
| 5 | Versionning..... | 30 |
| 6 | Mise en oeuvre | 31 |
| 1 | Structure du système..... | 31 |
| 2 | Base de données..... | 32 |
| 3 | Interface web | 32 |
| 3.1 | Technologies..... | 32 |
| 3.2 | La tuile : représentation du réseau piéton | 33 |
| 3.3 | L'interface | 34 |
| 3.4 | Structuration du site..... | 35 |
| 4 | Web serveur | 35 |
| 4.1 | Technologies..... | 35 |
| 4.2 | Rôle | 36 |
| 4.3 | Structuration du web-serveur..... | 36 |
| 7 | Tests | 37 |
| 1 | Tests fonctionnels | 37 |
| 2 | Tests de performance..... | 38 |
| 8 | Reproductibilité | 40 |
| 1 | Configuration de l'environnement..... | 40 |
| 2 | Lancement de l'application | 40 |
| | Conclusion | 42 |

| | |
|---|-----------|
| Annexes | 43 |
| A Cahier de spécification | 44 |
| B Interfaces | 45 |
| C Diagramme de cas d'utilisation | 47 |

Table des figures

2 Génération du réseau piéton

| | | |
|----|--|----|
| 1 | Exemple de reseau : le réseau routier..... | 5 |
| 2 | Autre représentation du réseau routier | 6 |
| 3 | Différence entre réseau routier/piéton | 7 |
| 4 | Répartition des voies piétonnes utilisées..... | 8 |
| 5 | Exemple de Google Map..... | 9 |
| 6 | Exemple de Mappy | 9 |
| 7 | Étape de construction du réseau piéton..... | 11 |
| 8 | Principe du 'buffering' | 12 |
| 9 | Exemple d'un réseau 'bufferisé' | 12 |
| 10 | Exemple de structuration de voies..... | 13 |
| 11 | Algorithme de construction du réseau piéton via la méthode "Buffering"..... | 14 |
| 12 | Construction des points pour la "bufferisation" | 15 |
| 13 | Exemple de trace GPS (source : Open Street Map)..... | 16 |
| 14 | Étapes de l'algorithme de construction du réseau piéton à partir de traces GPS | 17 |
| 15 | Étapes de l'algorithme de construction du réseau piéton à partir d'image | 17 |

3 Solution retenue et adaptée

| | | |
|---|--|----|
| 1 | Base de données géographique de la solution | 18 |
| 2 | Etape 1 : Récupération des routes et voies piétonnes | 19 |
| 3 | Etape 2 : Création des polygones autour des routes | 20 |
| 4 | Etape 3 : Récupération des contours des polygones | 20 |
| 5 | Etape 4 : Détermination des intersections | 21 |
| 6 | Etape 5 : Split des contours des polygones | 21 |
| 7 | Etape 6 :Suppression des segments résidus..... | 22 |

4 Outils et technologies utilisées

| | | |
|---|---------------------------|----|
| 1 | Open Street Map..... | 24 |
| 2 | PostgreSQL / PostGIS..... | 25 |
| 3 | Types de géométries | 25 |
| 4 | QGIS | 26 |

5 Méthodologie

| | | |
|---|---|----|
| 1 | Planning du semestre 9 | 28 |
| 2 | Planning prévisionnel du semestre 10..... | 29 |
| 3 | Planning réel du semestre 10..... | 29 |
| 4 | WebStorm | 29 |
| 5 | JSHint..... | 29 |

6 Mise en oeuvre

| | | |
|---|--|----|
| 1 | StructureDuSystème..... | 31 |
| 2 | Création de la tuile depuis TileMill | 34 |
| 3 | Page d'accueil..... | 34 |
| 4 | Structuration du site web | 35 |
| 5 | Rôle du web-serveur..... | 36 |
| 6 | Structuration du serveur | 36 |

7 Tests

| | | |
|---|--|----|
| 1 | Test 1 de performance du web-serveur | 38 |
| 2 | Test 2 de performance du web-serveur | 39 |
| 3 | Test 3 de performance du web-serveur | 39 |

B Interfaces

| | | |
|---|---|----|
| 1 | Page d'accueil..... | 45 |
| 2 | Affichage du réseau piéton..... | 45 |
| 3 | Affichage des voies à proximité du clique | 46 |
| 4 | Affichage des informations de la voie sélectionnée..... | 46 |

C Diagramme de cas d'utilisation

| | | |
|---|--------------------------------------|----|
| 1 | Diagramme de cas d'utilisation | 47 |
|---|--------------------------------------|----|



Liste des tableaux

2 Génération du réseau piéton

| | | |
|---|--|---|
| 1 | Voies accessibles par les piétons..... | 7 |
|---|--|---|

Introduction

Ce projet a été réalisé dans le cadre des projets recherches et développement de 5e année. Le projet est décomposé en deux parties, une partie recherche, c'est-à-dire l'étude de l'état de l'art et une partie développement.

Le projet a été encadré par Antoine Giret, qui travaille pour la société GéoVélo, dont les locaux sont situés à Tours. GéoVélo est une société qui propose un service (web) d'itinéraires pour les trajets à vélo. La société a déjà des partenariats avec différentes villes, dont Tours. La ville de Lyon a contacté la société GéoVélo car elle a un besoin fort sur le calcul d'itinéraire piéton.

Il existe déjà des services qui permettent d'effectuer des calculs d'itinéraire piéton (ex : Google Map). Le problème c'est que ces services considèrent le piéton comme un véhicule. C'est-à-dire que l'outil va utiliser le même réseau que celui des voitures pour effectuer l'itinéraire. Les passages piétons ne sont pas pris en compte par exemple.

Le but de ce projet est, à partir d'un réseau routier, sous la forme d'une base de données géographiques, construire un réseau piéton. Dans le but d'effectuer des calculs d'itinéraires appropriés aux piétons via une interface web. C'est-à-dire en prenant en compte les passages piétons à emprunter, la présence d'escaliers ou non etc.. Le but de ce projet est donc d'étendre le principe de GéoVélo aux piétons.

Dans ce projet j'interviens dans la partie de génération du réseau piéton à partir du réseau routier. Le but est de construire et d'implémenter un algorithme qui permet de générer un tel réseau. J'interviens également sur le développement de l'interface web qui permet de visualiser le réseau piéton de la ville de Lyon sur une carte, avec la possibilité de modifier les attribues des voies.

Première partie

Recherche

1

Présentation

1 Contexte

Ce projet est destiné à la société GéoVélo, située à Tours. La société propose un service web qui aide les personnes à trouver simplement et rapidement un itinéraire à vélo. Les itinéraires sont calculés selon différents critères (confort des pistes cyclables, distance, dénivelés...). GéoVélo possède déjà plusieurs partenariats avec de grandes villes telles que Tours, Paris et Lyon. La ville de Lyon a demandé à GéoVélo de développer un outil permettant, ce coup-ci, d'effectuer des itinéraires piétons. Cet outil aurait pour but de proposer des itinéraires touristiques, dans la ville de Lyon. Les itinéraires seraient calculés selon l'attrait des bâtiments, la présence ou non d'escalier... tout en utilisant uniquement les voies praticables par les piétons. C'est-à-dire les trottoirs, les passages piétons, les rues piétonnes...

Un tel service n'existe pas actuellement sur le marché. Car la majorité des outils permettant le calcul d'itinéraires piétons se basent essentiellement sur le réseau routier, et non en se basant réellement sur les vraies voies piétonnes.

GéoVélo a donc proposé un sujet de projet de fin d'études à Polytech' Tours afin de développer un algorithme qui génère le réseau piéton d'une ville. Et de rendre accessible sous forme visuelle à travers une interface web.

Mon rôle est de rechercher et développer cet algorithme, ainsi que cette interface pour GéoVélo. Qui s'en servira pour leur projet avec la ville de Lyon.

Le projet a été lancé cette année. Il n'y a donc aucun existant.

2 Problématique

La problématique de ce projet est : comment générer le réseau piéton à partir du réseau routier et comment le représenter. De nos jours, l'accessibilité aux données d'un réseau piéton est très difficile, parfois payante (organisme qui vend les données d'un réseau particulier). De plus, il n'existe aucun outil ou technologie permettant de le générer.

3 Objectif final

L'objectif de ce projet est de concevoir et d'implémenter un algorithme permettant de générer le réseau piéton de la ville de Lyon, en se basant sur son réseau routier.

Une fois la génération du réseau piéton effectuée, il faut représenter le résultat sur la carte de Lyon, à travers une interface web. Cette interface servira dans un premier temps à visualiser le réseau piéton

généralisé sur la carte de Lyon. On pourra choisir d'afficher ou non le réseau piéton (qui sera une couche sur la carte). Dans un second temps, cette interface permettra aux utilisateurs d'effectuer des calculs d'itinéraires piétons. Le calcul d'itinéraire sera multicritères. La partie concernant le calcul d'itinéraire sera réalisée par Antoine Giret.

À la fin de ce projet, un script sera développé. Qui permettra de générer le réseau piéton d'une ville à partir de son réseau routier (sous forme de base de données géographique). Et une interface web sera développée également. Elle servira à visualiser ce réseau sur une carte de France.

2

Génération du réseau piéton

1 Généralités sur les réseaux

Tout d'abord, qu'est qu'un réseau ? Un réseau se définit comme une structure composée d'éléments ou de points, souvent qualifiés de nœuds ou de sommets, reliés entre eux par des liens ou liaisons. Il est très souvent représenté avec un graphe. Les domaines d'utilisation du concept de réseau sont extrêmement variés. On le retrouve en sciences humaines (relations entre personnes), économie et gestion (relations entre organisations et entre acteurs économiques), informatique et télécommunications (interconnexion d'équipements), mathématiques (avec en particulier la théorie des graphes), toutes les autres sciences exactes (cristallographie, informatique abstraite, électricité, optique, biologie...), en géographie, en transports...

Voici un exemple de réseau :

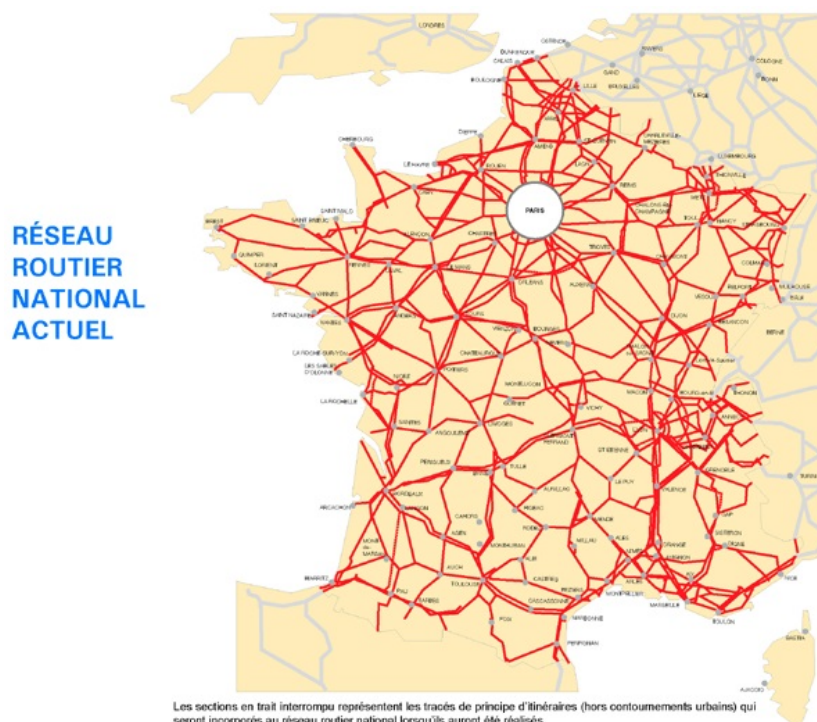


Figure 1 – Exemple de réseau : le réseau routier

Cette image représente une partie du réseau routier de la France. Les noeuds sont les villes, et les liaisons sont les routes. Seules les grandes voies sont représentées. On peut donc voir qu'un réseau peut devenir rapidement complexe et contenir une énorme quantité de données.

Voici une autre représentation du réseau routier d'un quartier d'une ville :



Figure 2 – Autre représentation du réseau routier

Dans cette représentation, les bâtiments sont représentés par des polygones (bleus) en plus des routes (rouge).

Avec le succès des systèmes de navigation, les données liées aux réseaux routiers sont maintenant très développées et disponibles dans la quasi-totalité des pays (En Amérique, Asie, Europe...). Ces données peuvent être open source (ex : Open Street Map) ou payante (ex : Navteq). Grâce à ces données, il est possible d'obtenir toutes informations concernant chaque route du réseau (vitesse autorisée, type de voie, longueur...). Des outils de calcul d'itinéraire sont également disponibles et peuvent être utilisés par quiconque souhaitant effectuer des calculs d'itinéraires sur ses propres graphes.

2 Le réseau piéton

Le réseau piéton diffère du réseau routier. En effet, un piéton n'a pas les mêmes accès qu'une voiture sur les différentes voies existantes. Par exemple, pour une route donnée, un piéton peut, par exemple, marcher du côté droit ou du côté gauche (s'il existe un trottoir de chaque côté) pour aller dans la même direction. Contrairement aux voitures qui doivent obligatoirement (dans la majeure partie des cas) prendre la voie de droite. De plus, les piétons ont accès à des voies particulières dont les voitures n'ont pas accès. Par exemple les voies piétonnes ou chemin forestier. L'article [3] traite le sujet de l'accessibilité des piétons.

La représentation du réseau piéton est donc différente de celle du réseau routier. Voici un schéma montrant la différence entre la représentation d'un carrefour sur un réseau routier, et un carrefour sur un réseau piéton :

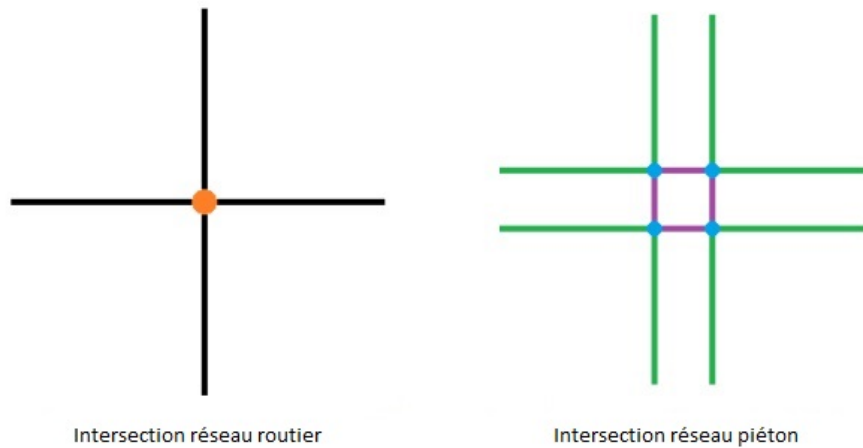


Figure 3 – Différence entre réseau routier/piéton

À gauche, il y a quatre voies, chacune représentée par un trait noir. À droite, les traits verts représentent les trottoirs et les traits violets les passages piétons. On peut voir que pour une voie donnée (trait noir) sur le réseau routier, il existe en réalité deux voies sur le réseau piéton. En effet, un piéton peut marcher à gauche ou à droite le long d'une voie, peu importe la direction où il se dirige. On peut voir également que pour une intersection de voies, cela se modélise par quatre passages piétons.

Comme nous l'avons évoqué plus haut, les voies accessibles par les piétons ne sont pas les mêmes que celles accessibles par les véhicules. Il est donc important de savoir qu'elles sont les différentes voies que peut emprunter un piéton. Voici une liste exhaustive des différents types de chemins piétons répertoriés sur OpenStreetMap (présenté plus tard) :

Table 1 – Voies accessibles par les piétons

| Type de chemin | Description |
|-------------------------------|---|
| Trottoir | Un chemin conçu pour la circulation des piétons le long d'une route |
| Voie piétonne | Un chemin se situant pas nécessairement sur le côté d'une route. Cela peut être une passerelle entre les bâtiments , un chemin traversant une place.. |
| Voie et rampe d'accessibilité | Une partie d'un trottoir , mais spécifiquement l'entrée des bâtiments ou une rampe pour les personnes handicapés |
| Passage piéton | Un marquage situé sur une route pour indiquer où les piétons doivent traverser , généralement à une intersection |
| Pont piéton | Pont construit au dessus d'une route ou fleuve permettant aux piétons de traverser |
| Tunnel pour piéton | Tunnel permettant aux piétons de traverser une route |
| Sentier | Un chemin qui est le plus souvent désigné comme de loisirs , comme les sentiers de course ou des sentiers naturels |

Le réseau piéton contient tous ces types de voie. Mais chacune ne représente pas la même recouvrance sur le réseau. En effet, il existe beaucoup plus de trottoir que de tunnel par exemple. Voici la répartition moyenne des différents types de voie piétonne au sein d'un réseau :

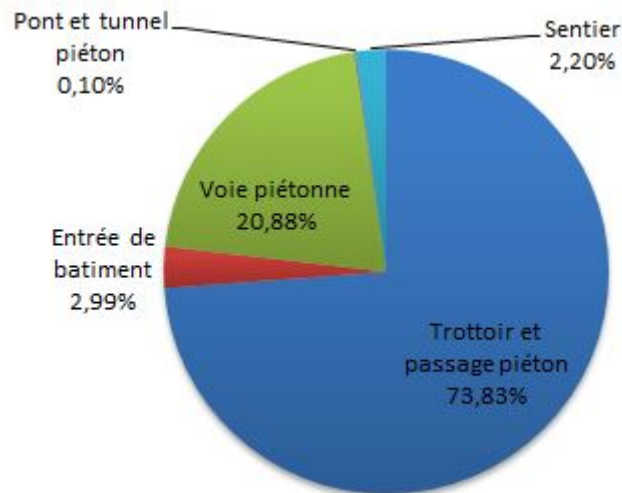


Figure 4 – Répartition des voies piétonnes utilisées

La majeure partie d'un réseau piéton est constituée de trottoir et de passage piéton. Au second rang nous avons les voies réservées exclusivement aux piétons. Les autres types de voies telles que les ponts... sont très peu utilisées. Ces données peuvent être intéressantes car dans les bases de données géographiques (détaillées dans la suite), les caractéristiques de toutes les routes ne sont pas complètement renseignées. Ce qui peut être très gênant lors de la conception du réseau piéton.

3 Application existantes

La demande pour les services de navigation piétonnière (PNS) est assez récente et ne cesse d'augmenter au fil des années. Les PNS, bien qu'ils chevauchent en fonctionnalité les systèmes de navigation automobile, doivent être conçus spécifiquement pour les besoins particuliers en orientation et en navigation des piétons ainsi que leurs préférences. L'un des principaux défauts de la plupart des PNS existants est qu'ils utilisent et offrent des services en fonction des réseaux routiers, ce qui fait en sorte que les PNS ne fassent pas un suivi efficace et approprié à des piétons étant donné qu'ils marchent sur les trottoirs, lesquels ont plus de segments que les routes et sont plus étroits que celles-ci.

3.1 Google Map

Le service de Google, Google Map, est un service gratuit de cartographie en ligne très connu et utilisé de tous. Il a été lancé pour la première fois en 2004 aux États-Unis. Deux types de vue sont disponibles dans Google Maps : une vue en plan classique, avec nom des rues, quartiers, villes et une vue en image satellite, qui couvre aujourd'hui le monde entier. On peut effectuer des calculs d'itinéraire routier, contenant des détails sur la longueur, le temps nécessaire... Il est également possible d'effectuer des itinéraires pour piéton. Analysons comment fonctionne le calcul d'itinéraire sous Google Map et comment est représenté l'itinéraire.

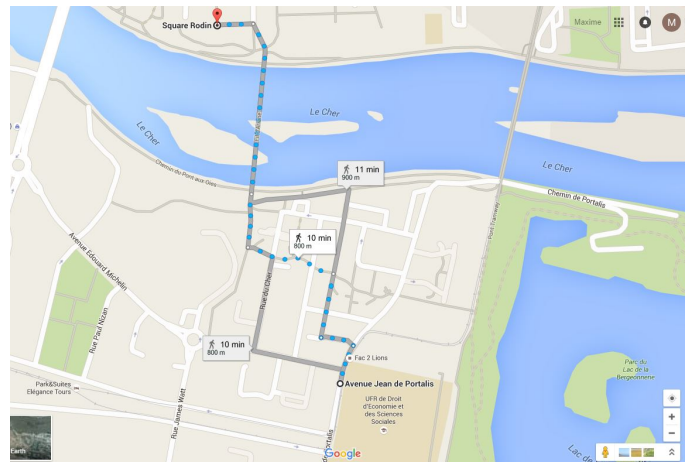


Figure 5 – Exemple de Google Map

On constate plusieurs choses :

- Le trajet est représenté sur les routes, et non sur les trottoirs.
- Les passages piétons ne sont ni représentés, ni utilisés.
- Les voies piétonnes, ponts cyclables sont utilisés.

Le système ne prend donc pas en compte toutes les voies qu'un piéton peut emprunter. Il se base essentiellement sur le réseau routier.

3.2 Mappy

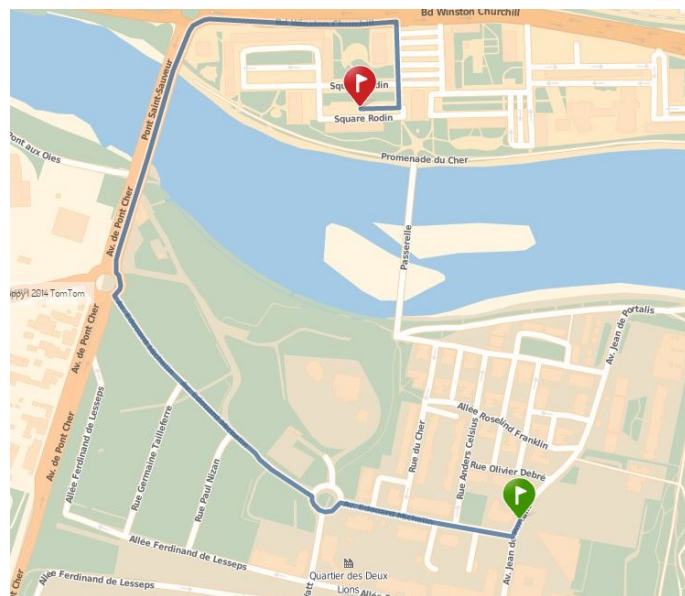


Figure 6 – Exemple de Mappy

Le résultat est globalement le même au niveau de la représentation. Les trottoirs et passage piétons sont inexistant. Une chose diffère dans cet exemple. Avec Google Map, le pont cyclable est utilisé dans l'itinéraire contrairement à Mappy qui ne l'utilise pas.

3.3 Université américaine

Exemple d'un site [WWW5] d'une université américaine qui propose un calcul d'itinéraire piéton au sein de leur université. On y constate que les passages piétons, escaliers sont représentés comme une route et sont pris en compte dans leur calcul d'itinéraire. Leur outil est donc plus précis que ceux des mastodontes dans ce domaine. L'objectif de GéoPiéton est proche de celui qu'a réalisé cette université.

4 Valeurs ajoutées par GéoPiéton

On a pu remarquer que la quasi-totalité des services proposant des calculs d'itinéraire piéton ne sont pas optimisés à 100%. C'est-à-dire qu'ils collent aux besoins réels des piétons en se rapprochant aux plus proches du réseau auquel les piétons sont confrontés dans la vie de tous les jours.

C'est pourquoi GéoVélo souhaite proposer un service qui prend réellement en compte toutes les contraintes qu'un piéton peut rencontrer à la ville de Lyon, car ils ont un besoin réel dans ce domaine. Et il n'existe pas de tel service à ce jour.

Les plus de GéoPiéton sont donc :

- Prise en compte des passages piétons.
- Visualisation exacte du chemin à emprunter (notamment les trottoirs).
- Effectuer des itinéraires personnalisés selon différents critères.

5 Recherche et algorithme concernant la génération du réseau piéton

Dans cette partie, nous allons voir les différentes méthodes pour, à partir d'un réseau routier comme ceux vus précédemment, construire le réseau piéton correspondant.

Ces méthodes proviennent de recherches déjà existantes. Pour avoir des informations supplémentaires et détaillées, veuillez vous référer à la dissertation [2] et à l'article [1].

5.1 Approche

Les données sur les réseaux piétons sont minimales (comparé à celles des réseaux routiers), et ne sont pas disponibles dans le domaine public. Dans certaines zones, ces données sont disponibles mais de façon commerciale. Malgré que les deux réseaux paraissent ressemblants, il existe des différences qui ne peuvent être négligées si l'on souhaite se rapprocher au plus au réseau piéton réel.

L'idée est donc d'utiliser le réseau routier, en considérant tous les attributs qu'il comporte, afin de construire le réseau piéton.

Quelques recherches/études ont été effectuées afin de régler le problème de la construction du réseau piéton, mais celles-ci restent très peu nombreuses et aucun outil n'a répondu totalement à notre problématique n'existe. Cependant, quelques méthodes ont été réfléchies et étudiées. Sur tous les documents traitant ce sujet, il en ressort trois méthodes.

Afin de générer le réseau piéton, ces différentes méthodes se décomposent toujours en trois grandes étapes :

- Collecte et préparation des données
- Construction du réseau piéton
- Présentation et visualisation du réseau

La collecte et préparation des données consistent à formater les données pour qu'elles soient utilisables et exploitables par un algorithme. Ces données peuvent provenir de différentes sources :

- Données structurées et stockées dans une BDD

- Images représentant le réseau routier, qui couplé avec du traitement d'images, permet d'en extraire le réseau piéton
- Des données GPS : coordonnées de piétons qui permettent de retracer approximativement les voies qu'ils ont empruntées afin de construire un réseau.

L'étape de construction utilise ces données en entrée, préalablement traitées, afin de construire le réseau piéton, qui est composé de segments et de points, en utilisant l'algorithme choisi. Les données en sorties sont stockées au minimum sous trois formes : les noeuds, les segments, les relations entre points et segments. Cette structuration doit être respectée car, pour pouvoir appliquer un calcul d'itinéraire sur le réseau routier, il doit être représenté sous forme de graphe (noeuds, arcs)...

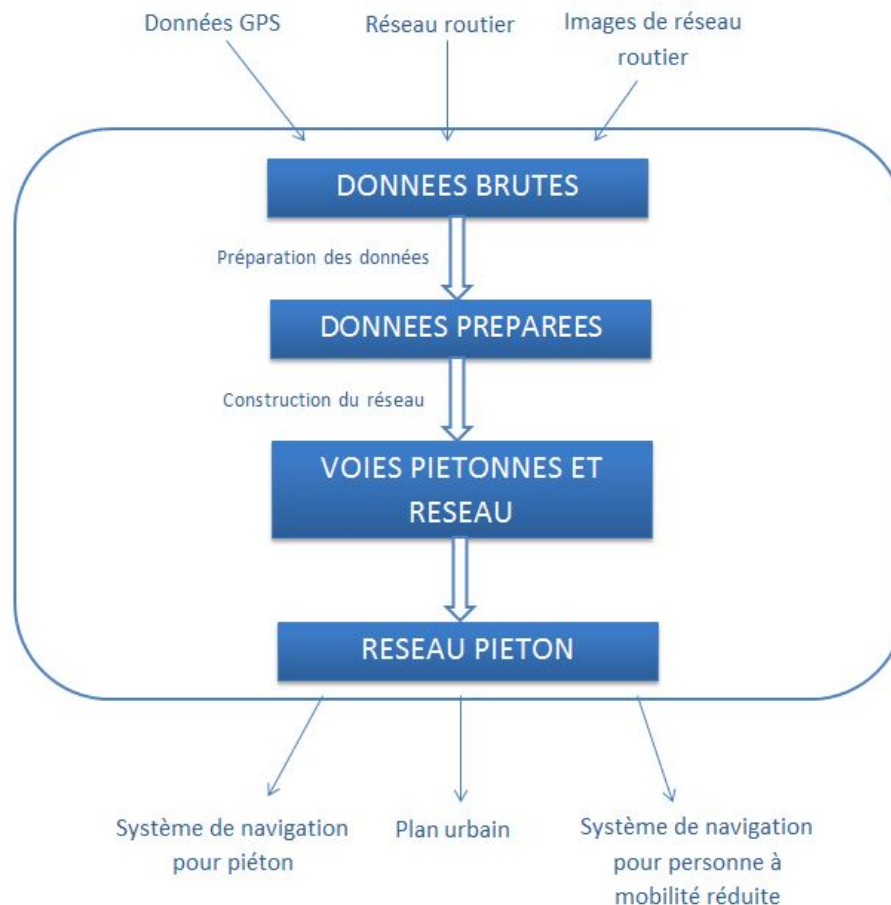


Figure 7 – Étape de construction du réseau piéton

5.2 Méthode : "Network Buffering"

5.2.1 Principe

La méthode "Network Buffering" consiste à utiliser le réseau routier et de "bufferiser" chaque voie. Cela va créer des contours autour de cette voie qui vont représenter approximativement la localisation des trottoirs.

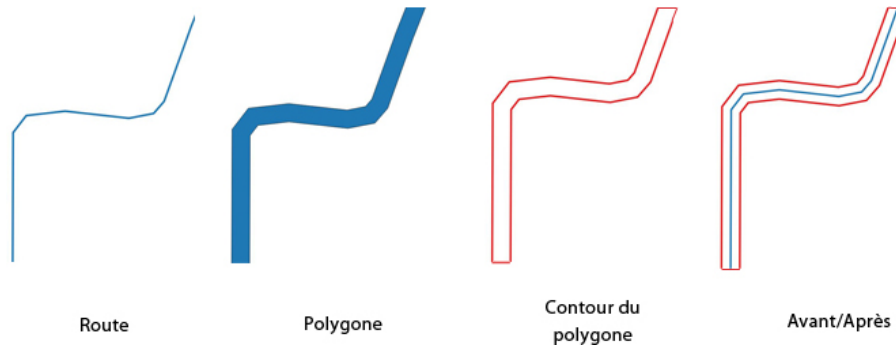


Figure 8 – Principe du 'buffering'

Le but, est donc de "bufferiser" chaque route où il y a présence de trottoirs. Pour les routes qui sont déjà des voies piétonnes, il n'est pas nécessaire d'effectuer cette démarche. En effet, construire des trottoirs sur une voie piétonne n'a aucun sens. Une fois le traitement effectué pour chaque voie, on obtient un réseau de cette forme-là (exemple sur une partie) :

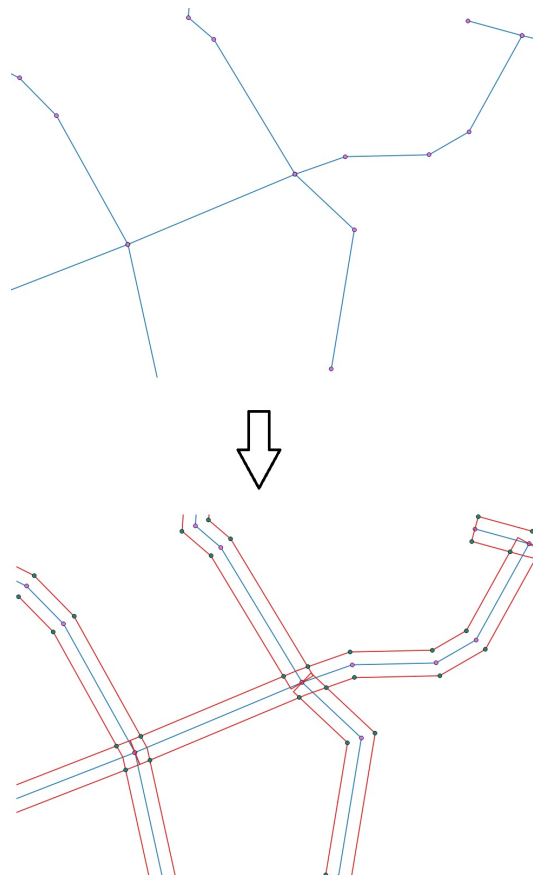


Figure 9 – Exemple d'un réseau 'bufferisé'

À la fin du traitement, nous obtenons bien les trottoirs situés autour des voies routières. Cependant, on constate qu'il y a des incohérences au niveau des intersections. En effet, les trottoirs viennent se chevaucher sur les routes, ce qui n'est pas correct. Cependant, ce problème dépend seulement de la manière dont on "bufferise" les routes. Sur cet exemple, nous avons juste créé un polygone autour de chaque route et avons récupéré le contour. Nous allons voir dans la suite qu'il existe une autre approche qui permet d'éviter ce problème.

5.2.2 Données d'entrées

Afin de pouvoir construire un polygone autour d'une voie donnée ("bufferiser"), il est nécessaire que cette voie soit représentée d'une manière exploitable. Les voies doivent être constituées de segment et de point. Chaque point de la voie est relié par des segments :

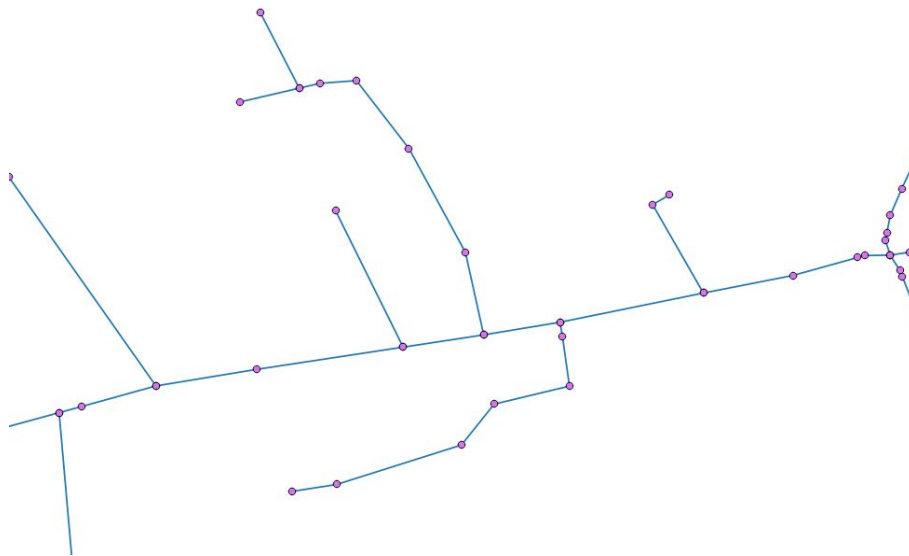


Figure 10 – Exemple de structuration de voies

Chaque point représente une coordonnée (x,y) . Ces coordonnées sont très importantes car ce sont à partir de celles-ci que nous allons pouvoir construire les trottoirs.

Dans la majorité des cas, ces données sont représentées dans une base de données. Il existe différentes technologies qui permettent de stocker et traiter des données de type géographiques (exemple : [PostGIS](#)).

5.2.3 Algorithme

Dans cette partie nous allons voir comment il est possible de construire le réseau piéton à partir du réseau routier.

Voici l'algorithme :

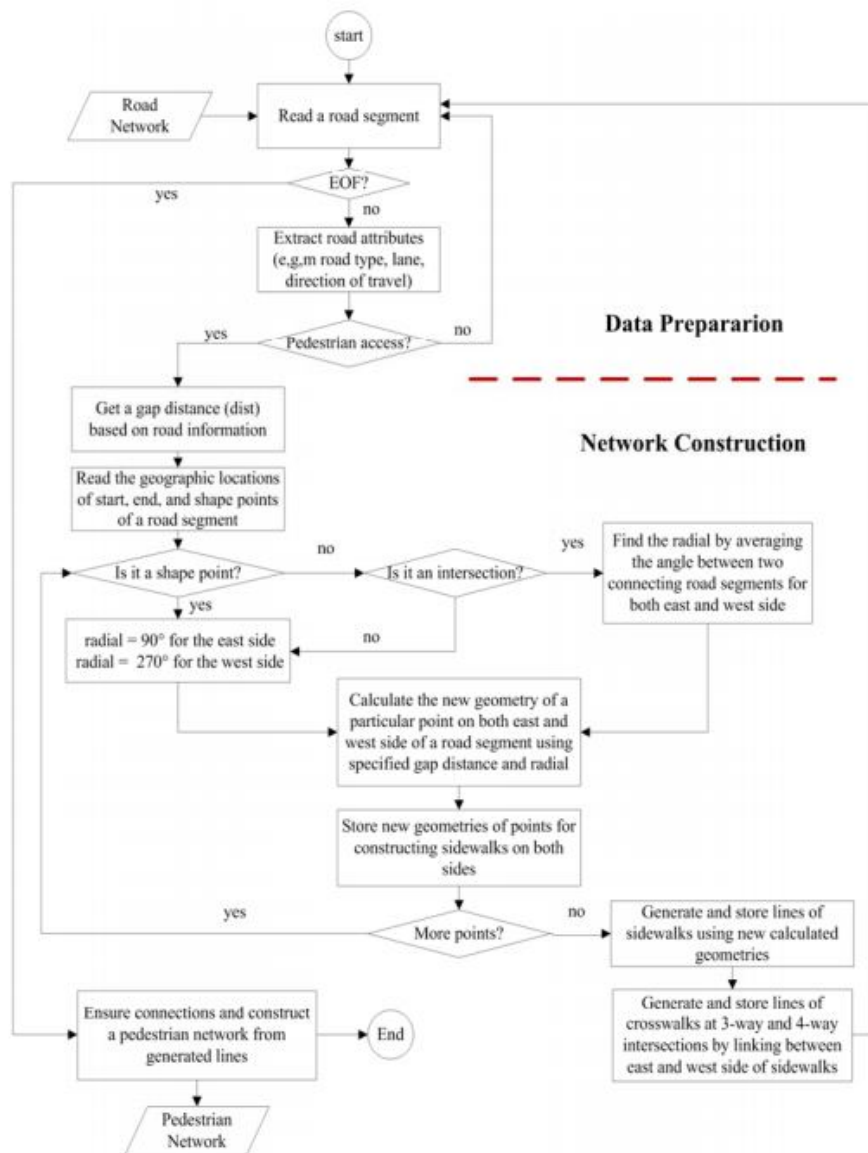


Figure 11 – Algorithme de construction du réseau piéton via la méthode "Buffering"

Phase de préparation des données :

Pour chaque segment de route, on extrait tous les attributs que l'on possède sur celle-ci. Ces attributs vont permettre de déterminer la largeur de la route, de déterminer s'il y a présence de trottoir ou non, s'il il s'agit d'une voie accessible aux piétons ou non...

Phase de construction du réseau piéton :

1 : On récupère la largeur de la route, si cela est possible, grâce aux différentes informations que l'on a sur celle-ci. Selon le type de route, la distance entre le trottoir et la route n'est pas la même. C'est pourquoi cette information est importante si l'on souhaite se rapprocher au plus près de la réalité.

2 : On récupère les différentes localisations de chaque point que comporte la route.

3 : Pour chaque point, on regarde si celui-ci représente une intersection ou non.

3-a : S'il s'agit d'une intersection, on récupère l'angle entre les deux routes concernées par l'intersection. Puis on récupère la moitié de cet angle qui sera utilisé pour la création de deux points, un pour le trottoir de gauche, et un pour celui de droite (voir schéma ci-contre).

3-b : S'il ne s'agit pas d'une intersection, les deux angles sont : 90° et 270° . En effet, les trottoirs se situent de façon parallèle à la route, il faut donc construire les points de façon perpendiculaire à la route (voir schéma ci-contre).

4 : Puis on construit le point de gauche et droite en utilisant l'angle déterminé précédemment. Voici la représentation des cas 3-a et 3-b, où l'angle n'est pas le même :

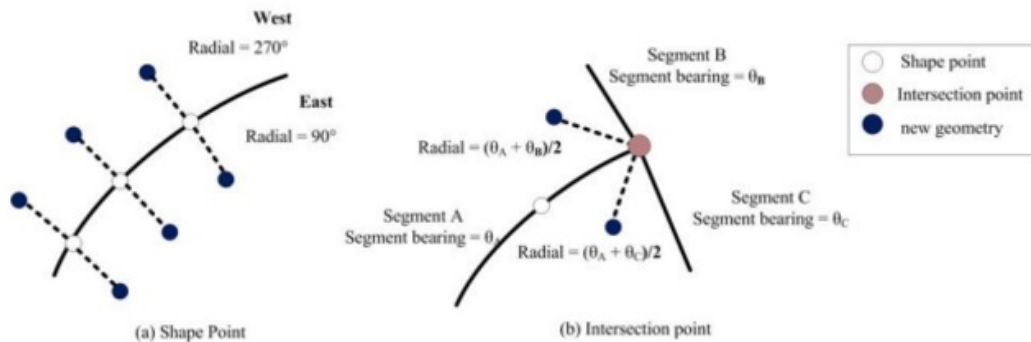


Figure 12 – Construction des points pour la "bufferisation"

Les points sont construits grâce aux formules suivantes :

$$\text{newlat} = \text{asin}(\sin(\text{lat}) * \cos(\text{dist}) + \cos(\text{lat}) * \sin(\text{dist}) * \cos(\text{radial})) \quad (1)$$

$$\text{newlong} = \text{long} + \text{atan2}(\sin(\text{radial}) * \sin(\text{dist}) * \cos(\text{lat}), \cos(\text{dist}) - \sin(\text{lat}) * \sin(\text{newlat})) \quad (2)$$

lat : la latitude du point traité

long : la longitude du point traité

newlat : la latitude du point crée

newlong : la longitude du point crée

dist : la distance entre le point traité et le point crée

radial : l'angle calculé précédemment, qui correspond à l'angle entre le point traité et celui crée

5 : Une fois chaque point de la route traité, on construit les deux trottoirs en reliant les différents points créer précédemment, puis on stocke les deux voies. Cette démarche représente la "bufferisation" de la voie.

Cette démarche est appropriée lorsqu'il existe un trottoir de chaque côté de la route. Cependant, le réseau routier comporte évidemment des voies de type différents (voir le [tableau récapitulatif](#)). Selon ce type de voie, les caractéristiques du réseau piéton varient :

- Il peut exister 0, 1 ou 2 trottoirs présents sur une voie
- La largeur des routes est variable. Donc la distance des trottoirs à la route est également variable.

Il faut donc, ajouter à la méthode précédente, un test vérifiant l'existence ou non de trottoir, ainsi que le nombre, et d'adapter le traitement de construction.

5.3 Méthode : "Collaborative Mapping"

Les deux méthodes qui vont suivre ("Collaborative Mapping" et "Image Processing") vont être expliquées et étudié plus brièvement. Car les données sur lesquelles se basent ces méthodes ne sont pas du tout les mêmes que celles que nous disposons. Nous possédons et devons utiliser les données sous forme de base de données et non des images ou des traces GPS. Cependant, il est intéressant d'étudier d'autres méthodes, car cela pourrait nous apporter des éléments supplémentaires pour la conception de notre algorithme.

5.3.1 Principe

Le "Collaborative Mapping" consiste à utiliser des localisations réelles. C'est-à-dire, des coordonnées GPS représentant des trajectoires de piéton. Grâce à ces trajectoires, il est possible de retrouver les voies piétonnes, et ainsi, construire un réseau.

Cependant, il faut supposer que ces coordonnées représentent bien les voies piétonnes. En effet, un piéton, s'il le souhaite, peut emprunter des voies qui ne sont pas des voies destinées aux piétons.

5.3.2 Données d'entrées

Les données en entrée de cet algorithme sont des traces GPS. Une trace GPS est un ensemble de points (représenté par des coordonnées GPS), qui représente un parcours, une route, qu'une personne (ou véhicule) a empruntée.

Voici à quoi ressemble une trace GPS :

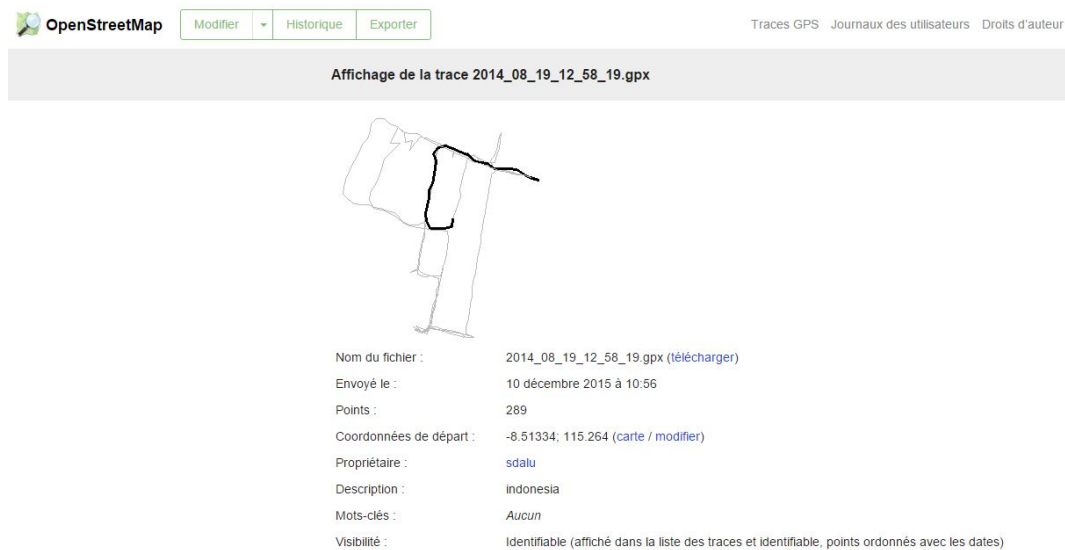


Figure 13 – Exemple de trace GPS (source : Open Street Map)

Chaque personne qui contribue à fournir des traces GPS, peut en fournir plusieurs, à différents moments, et plusieurs traces peuvent recouvrir une seule et une voie (si trois personnes empruntent la même voie, cela se traduira par trois traces GPS qui seront superposées les unes aux autres).

Les traces GPS peuvent provenir de téléphone mobile, de carte digitale, de sites internet collectant des données géographiques telles qu'Open Street Map et WikiLoc. Il s'agit de site où toute personne peut contribuer à ajouter/modifier des données géographiques. Ces sites forment de grosses communautés qui rendent les informations très fiables et abondantes. Cependant, les données concernant les traces GPS sont très peu utilisées par les utilisateurs, et donc il n'y en existe qu'un nombre réduit.

5.3.3 Algorithme

Ceci sont les étapes à suivre afin de construire le réseau piéton à partir de données GPS, cela pour une zone définie. Il faut savoir qu'il est possible d'éditer en ligne les données concernant les traces GPS sur les sites précédemment cités. Mais cela implique qu'une personne effectue la démarche d'elle-même. Renseigner la totalité des traces GPS, à la main, d'un réseau piéton d'une ville est une tâche plus que fastidieuse en temps. C'est pourquoi nous allons nous intéresser essentiellement au moyen de générer automatiquement ce réseau à partir des traces GPS, grâce à un algorithme.

Le but est donc, à partir de traces GPS en entrée, de construire le réseau piéton automatiquement.

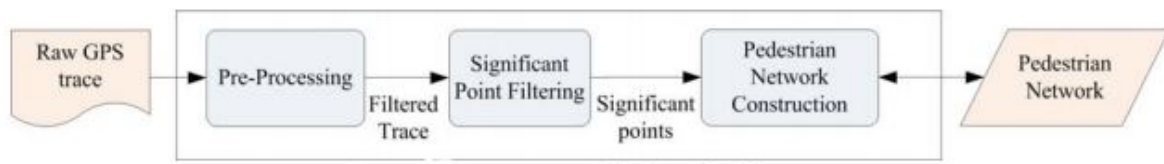


Figure 14 – Étapes de l'algorithme de construction du réseau piéton à partir de traces GPS

5.4 Méthode : "Image Processing"

5.4.1 Principe

Cette méthode est basée essentiellement sur le traitement d'image. Le but est, à partir d'images satellite d'une ville, appliquer des traitements sur les images afin d'extraire le réseau piéton.

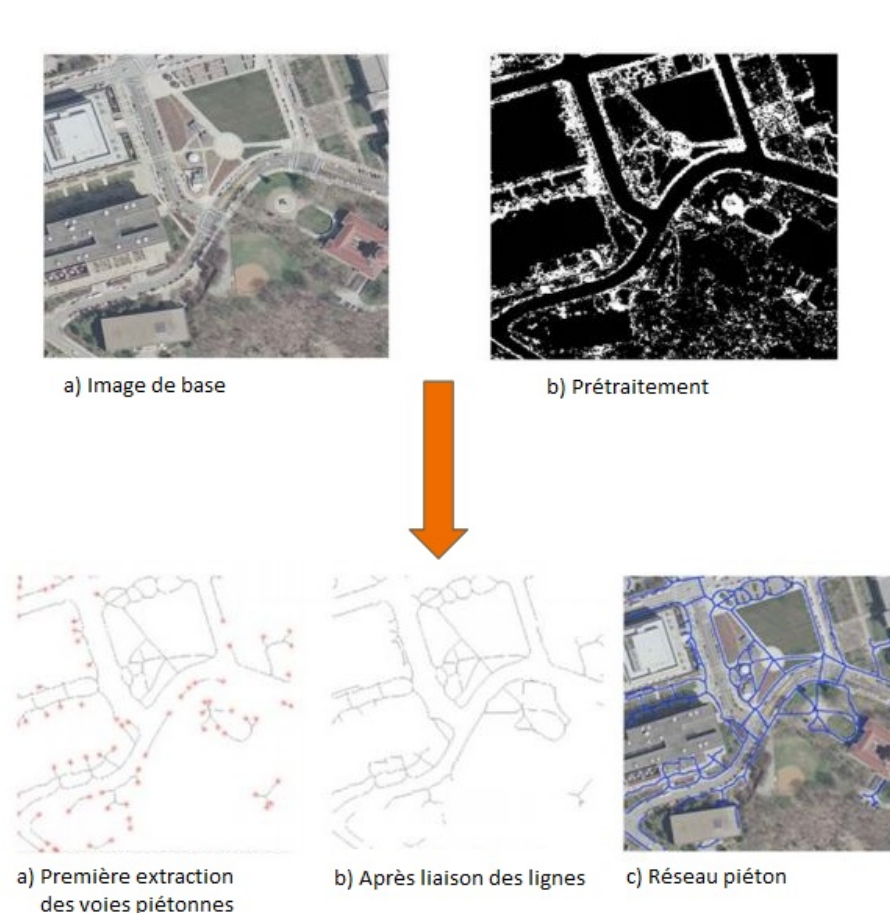


Figure 15 – Étapes de l'algorithme de construction du réseau piéton à partir d'image

Le principal avantage de cette méthode est qu'il permet d'obtenir le réseau piéton le plus proche de la réalité. Comme il se base sur des images réelles, les trottoirs sont construits à l'endroit exact où ils se situent dans la réalité.

Cette méthode est plutôt contraignante sur les données en entrées. En effet, il est nécessaire d'avoir un grand nombre d'images, afin d'avoir une vue globale de la ville où on souhaite construire le réseau piéton.

3

Solution retenue et adaptée

La méthode la plus adaptée à notre problème et celle du premier algorithme qui consiste à créer un buffer autour de chaque route afin de déterminer les voies piétonnes. En effet, sachant que les types de données qui sont utilisées par GéoVélo proviennent d'Open Street Map, il s'agit de segment et de noeud. Ce qui est en adéquation avec cette méthode.

Cependant, cet algorithme doit être adapté aux technologies utilisées par GéoVélo. De plus, il ne gère pas le fait qu'il ne peut exister qu'un unique trottoir au bord d'une route.

Nous nous sommes donc basé sur la démarche de cet algorithme tout en optimisant et ajoutant les traitements qui étaient essentiels pour le projet.

1 Données en entrées

Pour ce projet, il était nécessaire d'avoir le réseau routier de la ville de Lyon. Le site [Géofabrik](#) fournit des fichiers au format .osm.pbf pour plusieurs villes de France, et notamment Lyon. Ces fichiers contiennent toutes les informations disponibles sur Open Street Map. Tel que les routes, les batiments, les points d'eau...

C'est à partir de ce fichier que nous allons extraire le réseau routier de la ville de Lyon. Pour rendre ces données exploitables, il faut utiliser [Osmosis](#). Cet outil va permettre d'extraire les informations géographiques contenues dans se fichier et de les stocker dans une base de données géographique [PostGIS](#). Une fois ces données stockées dans la base, il sera ensuite possible d'effectuer des requêtes sur celle-ci afin d'extraire des informations selon des critères définis, mais aussi insérer/modifier des données.

Notre base de données sera approximativement de la forme suivante :

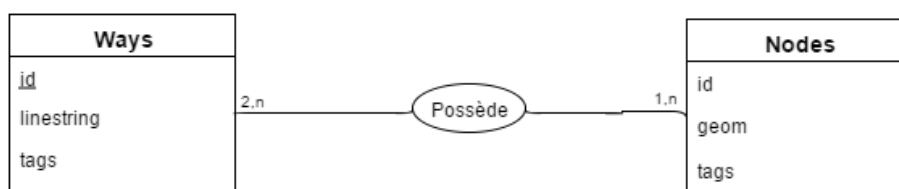


Figure 1 – Base de données géographique de la solution

Elle comporte donc trois tables :

- Ways : table qui stockera les voies (arcs).
- Nodes : table qui contiendra les noeuds.
- Ways_Nodes : table qui fera les liens entre les arcs et les noeuds qu'ils contiennent.

Les champs *linestring* et *geom* sont des champs de type *geometry* (typique des **bases de données géographiques**).

2 Algorithme adapté

Afin de générer le réseau piéton, nous nous sommes basé sur le premier algorithme que consiste à "bufferiser" les routes. Cependant, nous avons adapté cet algorithme aux outils utilisés pour ce projet ainsi qu'aux données en entrées.

Le script est développé en PostGreSQL utilisant une base de donnée PostGIS (voir **les détails sur ces technologies**)

Voici les différentes étapes de notre algorithme :

Étape 1 :

On récupère tous les **chemins** qu'un piéton est susceptible d'emprunter. Pour cela, on effectue une requête sur notre base de données préalablement remplie avec les données de Lyon trouvées sur Géofabrik. Le filtre sur les chemins est appliqué sur l'attribut *tags*. L'attribut *tags* peut prendre plusieurs valeurs (voir le **site officiel** pour plus d'informations).

Dans notre cas, nous récupérerons toutes les voies dont l'attribut *tags* correspond à l'une de ces valeurs, car elles correspondent aux voies qu'un piéton peut emprunter :

- footway
- pedestrian
- steps
- living_street
- crossing
- residential
- secondary
- tertiary
- primary
- unclassified
- tertiary_link
- secondary_link

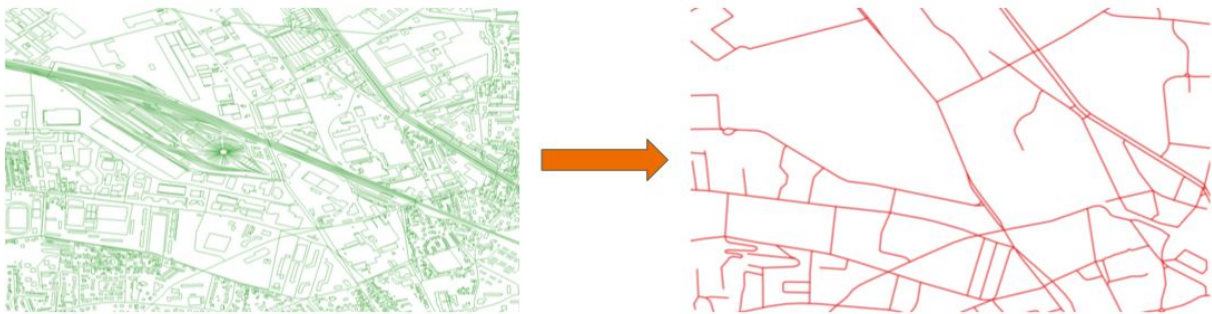


Figure 2 – Etape 1 : Récupération des routes et voies piétonnes

Étape 2 :

Une fois que les chemins praticables par les piétons ont été récupérés, on va créer un polygone pour chaque route où il y a la présence de trottoirs. Pour les routes n'en possédant pas, il n'y aura pas de polygone créé.

Afin de respecter au mieux la réalité, lors de la création des polygones, la largeur des polygones doit être paramétrable. Selon le type de voie, la largeur de la chaussée n'est pas la même, il faut donc appliquer une largeur différente pour chaque type de voie.

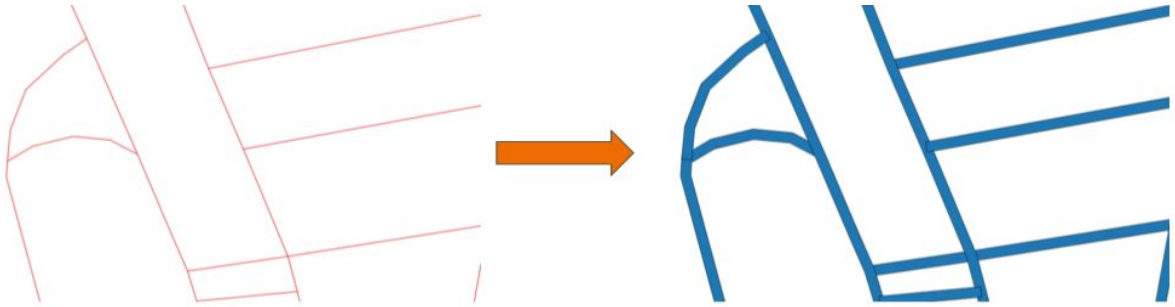


Figure 3 – Etape 2 : Création des polygones autour des routes

Étape 3 :

Les polygones étant créés, on va maintenant en récupérer les contours. Ces contours sont représentés par une seule et même ligne.

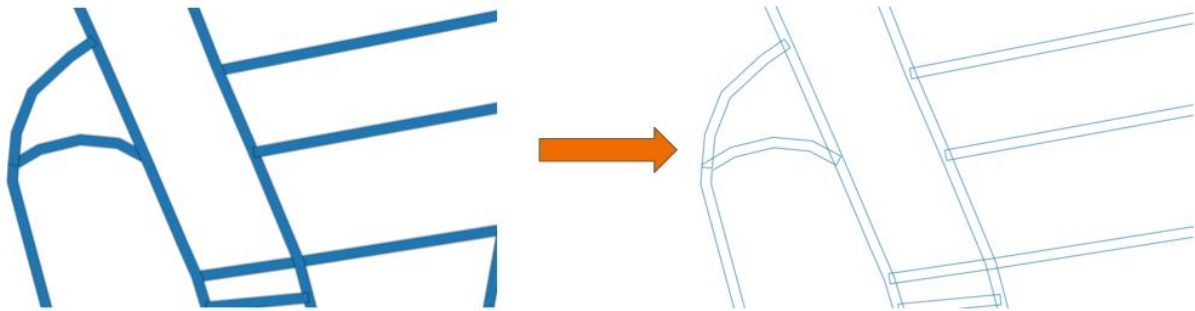


Figure 4 – Etape 3 : Récupération des contours des polygones

Ces contours représentent les trottoirs situés aux bords de la route. On constate, à ce stade, que les trottoirs se chevauchent les uns aux autres et qu'il existe un trottoir à chaque fin de route. Ce qui n'est pas vrai dans la réalité. Il faudra donc supprimer les segments qui chevauchent les autres trottoirs ainsi que les segments qui sont en fin de route car ils n'ont pas lieu d'exister.

Étape 4 :

Afin de supprimer les segments exprimés précédemment, on va déterminer les points d'intersection ainsi que les polygones d'intersection entre chaque polygone

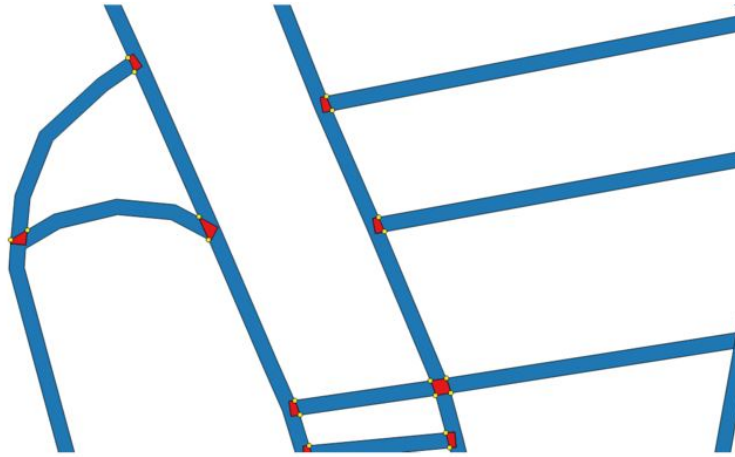


Figure 5 – Etape 4 : Détermination des intersections

Étape 5 :

Une fois les intersections déterminées, on va pouvoir diviser les contours de chaque polygone, au niveau de chaque intersection.

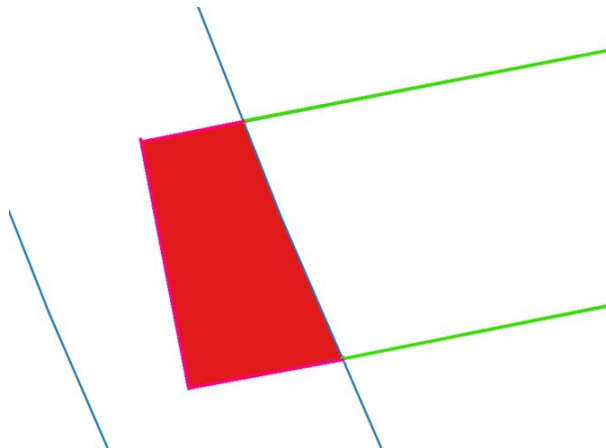


Figure 6 – Etape 5 : Split des contours des polygones

Chaque ligne est désormais divisée en plusieurs segments. Cette opération va être utile pour supprimer les segments indésirables (ici en rose).

Étape 6 :

Cette étape consiste à supprimer les segments résidus (voir étape 3). Pour ce faire, nous allons supprimer tous les segments qui appartiennent entièrement aux polygones d'intersections (polygones rouges) préalablement construits.

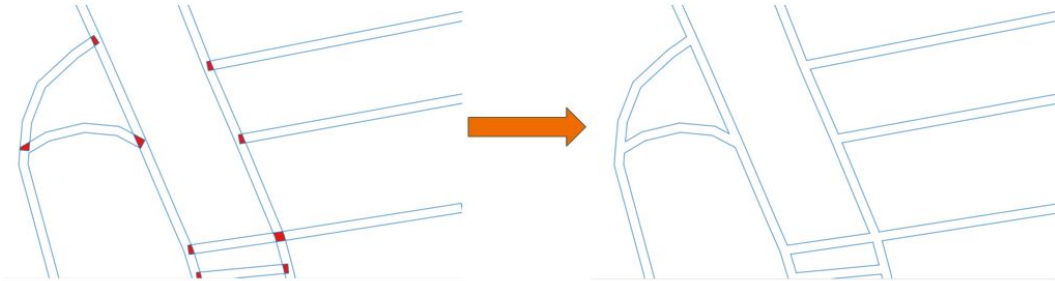


Figure 7 – Etape 6 :Suppression des segments résidus

Cela permet donc d'avoir les segments qui correspondent aux plus proche des trottoirs situé au bord des routes.

À la sortie, nous avons bien le réseau piéton qui est issu du réseau routier.

Dans cette explication, nous n'avons pas vu le traitement sur les voies qui sont déjà piétonnes à la base. En effet, lorsque l'on extrait les voies à l'étape 1, nous avons déjà des voies qui sont des voies piétonnes et non routières. Dans ce cas, il n'y a aucun traitement à effectuer. Il suffit juste de stocker directement la voie dans notre base.

Voici la méthode présentée de manière algorithmique :

```

Data : Données géographique du zone (routes, bâtiments, parcs...)
Result : Réseau piéton de la zone
data <- Extraction des données nécessaires ; // routes, voies piétonnes...
dataBuffurisee <- Bufferisation(data) ; //on crée un polygone pour chaque voie
for i=0 ; i<size(data) ; i++ do
  for j=0 ; j<size(data) ; j++ do
    if data[i] intersekte data[j] and i !=j then
      | On stocke le point d'intersection ;
    end
    if dataBuffurisee[i] intersekte dataBuffurisee[j] and i !=j then
      | On stocke le polygone d'intersection ;
    end
  end
end
for i=0 ; i<size(data) ; i++ do
  | sousVoies <- On divise data[i] à chaque point d'intersection qu'elle possède ;
  | data <- sousVoies (en supprimant la voie initiale) ;
end
for chaque polygone d'intersection do
  | data <- SupprimeVoieDansPolygone(data,polygone) ;
end
return data ;

```

Algorithme 1 : Pseudo-algorithme de génération de réseau piéton

Un script effectuant toutes ces étapes a été développé et est fonctionnel.

3 Critiques et améliorations

Nous avons vu précédemment que la largeur de la route n'était pas la même pour toute. Et cela se répercute donc directement sur l'écartement entre les deux trottoirs d'une route. Actuellement, la largeur de la route n'est pas prise en compte, elle a été fixée et est la même pour toute. Il reste à définir la largeur de chaque type de voie et d'ajouter cette partie dans l'algorithme.

Le temps d'exécution est très important. En effet, pour lancer l'algorithme sur l'instance complète de la ville de Lyon, sur un ordinateur avec 4Go de RAM, cela dure environ 12h. Cependant, l'algorithme sera lancé sur un serveur (que GéoVélo possède) et qui contient au moins 20 fois plus de RAM. Aucun test n'a encore été fait sur celui-ci. Mais le temps devrait être fortement réduit.

4

Outils et technologies utilisées

1 Open Street Map (OSM)



Figure 1 – Open Street Map

OSM est un projet international, créé en 2004, dont le but est de créer une carte libre du monde entier. Cette carte contient diverses données du monde tel que les routes, bâtiment, voies ferrées, trottoirs, feux rouges... Chaque personne peut contribuer à l'alimentation de cette carte. La communauté d'OSM tiens à jour et corrige les erreurs, pour rendre cette carte la plus fiable et la plus complète possible. Toutes ces données collectées sont libres d'accès et réutilisable par n'importe qui.

Chaque information est représentée sous forme d'objets. Par exemple, une autoroute, une rivière... est un objet. Et chaque objet possède des attributs. Par exemple, pour un objet de type route, on peut savoir si elle possède ou non des trottoirs, s'il y a la présence d'arbres ou non...

Toutes ces informations sont renseignées par des utilisateurs lambda ou par des organismes spécialisés. Il faut noter que toutes les informations sur un objet ne sont pas toujours totalement renseigné.

2 GeoFabrik

Le site [Géofabrik](#) extrait, sélectionne et traite des données géographiques et les met à disposition, gratuitement. On peut y télécharger des cartes, des fichiers de polygone, des tuiles pour mettre sur une carte. GeoFabrik tient toujours à jour ses données provenant de OSM.

Il est possible de télécharger des données sur des zones prédéfinies. Par exemple sur un pays, une région, une ville... Pour notre projet, nous avons téléchargé les données de la ville de Lyon. Ces données peuvent être téléchargées sous différents formats :

- .osm.pbf
- .shp.zip
- .osm.bz2

Pour ce projet, c'est le format .osm.pbf qui a été utilisé.

3 PostGreSQL et PostGIS



Figure 2 – PostgreSQL / PostGIS

PostgreSQL est un système de gestion de bases de données (SGBD) très performant. **PostGIS** ajoute le support d'objets géographiques à la base de données PostgreSQL. En effet, PostGIS "spatialise" le serveur PostgreSQL, ce qui permet de l'utiliser comme une base de données SIG (Système d'informations géographiques).

Les informations géographiques dans PostGIS sont représenté par le type **geometry**. Le type geometry est représenté par une chaîne de caractères. Voici les principaux types de géométries qui sont disponibles dans PostGIS :

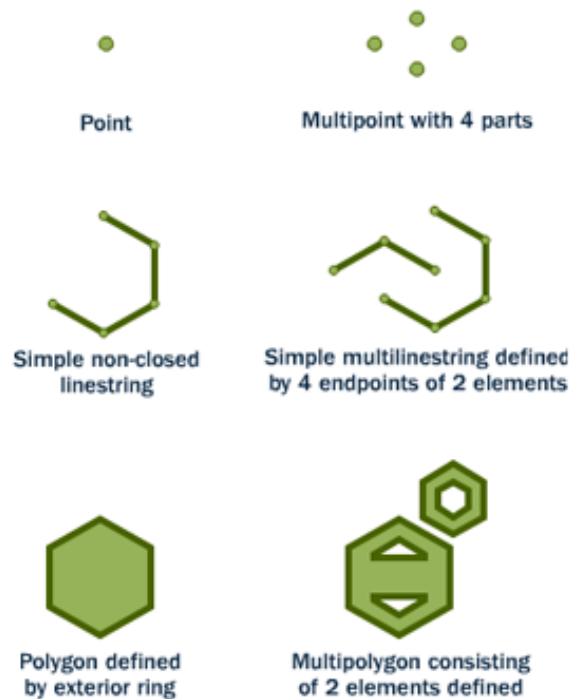


Figure 3 – Types de géométries

Il est possible d'y appliquer diverses fonctions (voir [documentation](#)) afin d'effectuer des opérations sur ces géométries. Ce sont ces fonctions que nous allons utiliser pour créer notre réseau piéton.

Et voici à quoi ressemble la structure d'une base géographique avec PostGIS :

- nodes
- relation_members
- relations
- schema_info
- spatial_ref_sys
- users
- way_node
- ways

Les tables qui vont le plus nous intéressées sont les tables nodes , ways et way_node. Car ce sont ces trois tables qui représentent le réseau (points, lignes, relations entre les deux). Et nous allons créer le réseau piéton à partir de ces trois types de données.

4 OSMOSIS

Osmosis est une application Java en ligne de commande qui permet de traiter des données provenant de OSM. Cet outil est composé de plusieurs fonctionnalités qui peuvent être chaînées les unes à la suite des autres. Par exemple, on peut lire/écrire des bases de données géographiques ou fichiers, filtré les données sources et sorties... Cet outil nous servira pour extraire les données issues d'OSM concernant la ville de Lyon (via le site GéoFabrik) afin de récupérer que le réseau routier (et non toutes les informations géographiques de Lyon). Ces informations seront stockées dans une base de données.

5 QGIS



Figure 4 – QGIS

QGIS est un système d'informations géographiques gratuit et Open Source. Il permet de créer, modifier, visualiser, analyser et publier des informations géospatiales. Il est possible de connecter QGIS à une base de données PostGis. C'est cette fonctionnalité que nous allons utiliser pour visualiser les données que nous avons dans notre base de données de façon graphique.

Dans ce projet, QGIS est très utile pour visualiser les résultats que nous obtenons. Les résultats issus de l'algorithme sont stockés dans une base de données, et QGIS est capable de lire une base de données géospatiale et de les présenter sous forme de carte.

Deuxième partie

Développement

5

Méthodologie

1 Méthode de travail

Tout au long du projet, des réunions ont été organisées entre l'encadrant et moi-même. Ces réunions étaient à fréquence d'une par semaine environ. Le but de ces réunions était d'échanger sur différents points :

- Présentation de l'avancement des tâches planifiées à la réunion précédente.
- Échange sur les difficultés rencontrées si présentes.
- Planification des prochaines tâches et objectifs à atteindre.

Quelques réunions à GéoVélo ont été également planifiées afin d'échanger directement avec les personnes de la société sur mon avancé. Et d'avoir certains conseils de leur part.

2 Planning

Le projet a été découpé en différents sprints. Pour plus d'informations concernant les différents sprints, veuillez référer au cahier de spécification [cahier de spécification](#).

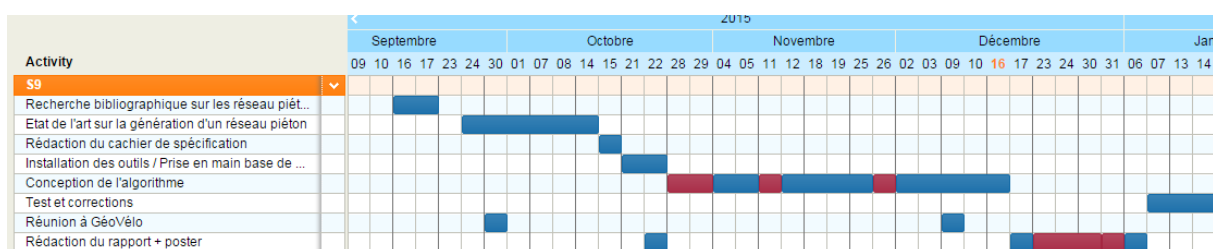


Figure 1 – Planning du semestre 9

Le planning a été globalement respecté. Les tâches "création de l'IHM" et "Ajout de fonctionnalités" ont été regroupés en une seule nommée "Création de l'IHM/WebServeur". Mais globalement, ces tâches sont similaires.

Une tâche n'était pas planifiée, celle de la modification de l'algorithme de génération du réseau piéton. Cette tâche s'est vue ajouter suite à une réunion avec la société GéoVélo. Qui a eu le souhait d'ajouter des modifications permettant d'obtenir d'amples données nécessaire au calcul d'itinéraire piéton.

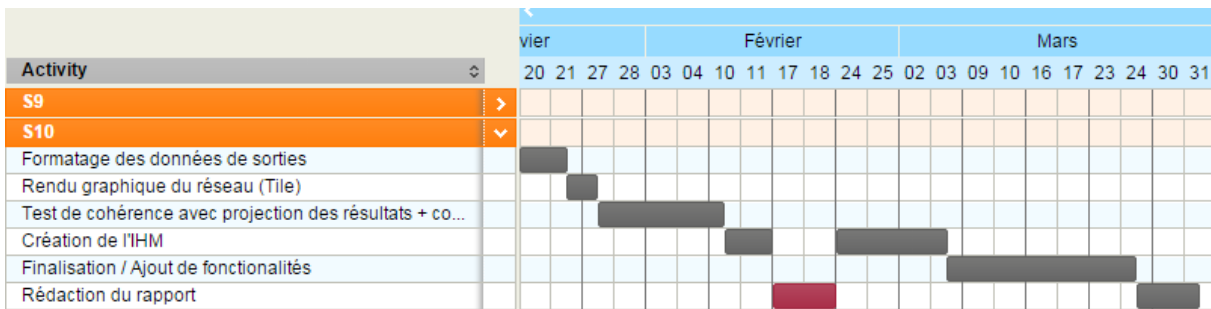


Figure 2 – Planning prévisionnel du semestre 10

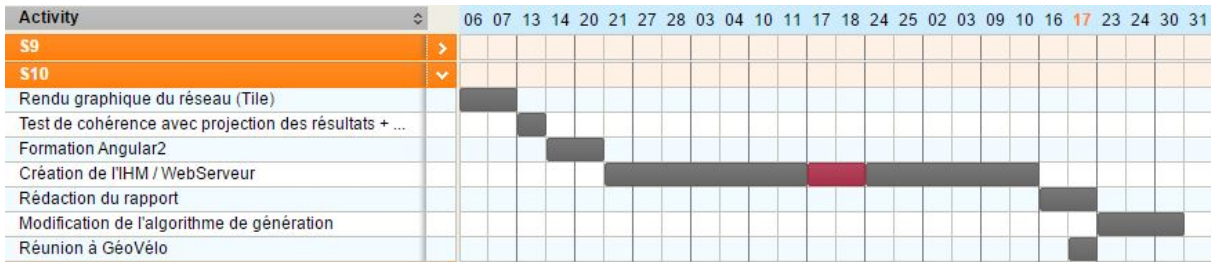


Figure 3 – Planning réel du semestre 10

3 Outils



Figure 4 – WebStorm

Les développements ont été effectués via l'IDE WebStorm. Qui est spécialement conçu pour les développements web, tel qu'Angular et NodeJS (technologies utilisées pour ce projet).

PgAdminIII a été utilisé en ce qui concerne la base de données.

4 Qualité du code



Figure 5 – JSHint

La qualité du code a été assurée grâce à un outil intégré dans WebStorm. Il s'agit de JSHint. Cet outil permet d'analyser le code produit et de détecter toutes erreurs de syntaxes non bloquantes, les variables inutilisées, commentaires de fonction mal structurés etc.. Cela permet de respecter les règles de codage Javascript.

De plus, chaque fonction a été commentée pour expliquer sa fonctionnalité, ses paramètres ainsi que ce qu'elle retourne.

5 Versionning

Le système de versionning utilisé pour ce projet est Subversion (SVN). Et plus précisément celui de l'école qui se nomme Redmine. Le projet est accessible à l'adresse suivante redmine.polytech.geoPieton. Pour y accéder, la personne se connectant doit être ajoutée au projet. Car le projet possède le statut de privé.

Des commits ont été effectués lorsqu'une grande étape du projet était terminée. (regroupant le site web et le serveur web).

6

Mise en oeuvre

La base de données étant créée et alimentée dans la première partie, le reste du travail consiste à développer l'interface-web qui permet de visualiser ces données sur une carte interactive afin d'interagir avec eux. Cette partie va donc traiter les différents travaux qui ont été réalisés (avec les outils utilisés) afin d'obtenir cette interface.

1 Structure du système

Le système qui a été développé comporte différentes parties. Voici sa structuration :

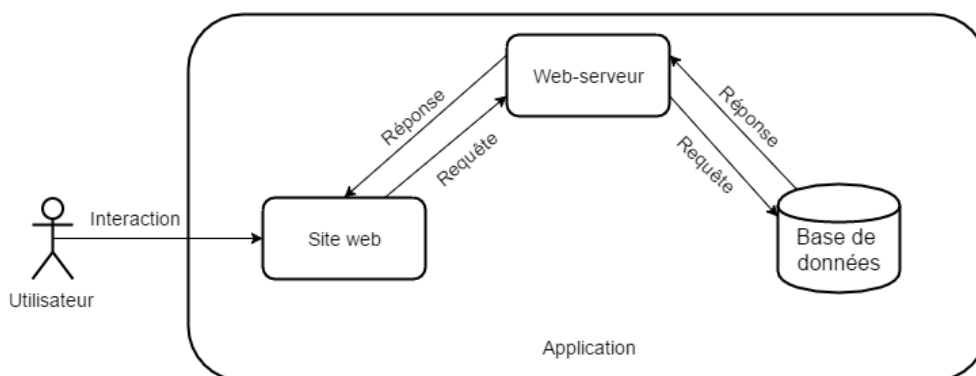


Figure 1 – StructureDuSystème

Son fonctionnement est celui du client/serveur. Le client est l'interface web, et le serveur le web-serveur.

La base de données est celle développée dans la première partie du projet. Il s'agit d'une base PostgreSQL, avec l'extension PostGIS qui permet de gérer des données géographiques. Elle contient toutes les données construites à partir du réseau routier et représentant le réseau piéton de la ville de Lyon, grâce au script développé. C'est dans cette base que nous allons chercher les données pour les afficher sur notre interface web.

L'interface graphique permet d'afficher les données contenues dans la base, ainsi que d'interagir avec celles-ci (modification). Les données étant du type géographique, et pour les besoins de ce projet, elles seront affichées principalement sur une carte.

Le web-serveur permet de faire le lien entre l'interface web et la base de données. Lorsque l'interface a besoin d'informations contenues dans la base, elle va effectuer une requête à notre web serveur, qui va lui-même effectuer la requête dans notre base de données, et retourner les résultats à notre interface web.

L'utilisateur interagit directement avec notre système depuis l'interface web. Les différentes actions que peut effectuer l'utilisateur sont présentés en **annexe** dans un diagramme de cas d'utilisation.

Les données entre le site web et le web-serveur sont transmises via le protocole HTTP. Les données sont formatées sous le format JSON. Le web-serveur est directement connecté à la base de données. Il peut donc directement effectuer les requêtes PGSql et d'en récupérer le résultat.

2 Base de données

Structure

Les différentes technologies utilisées pour la réalisation de la base de données ont déjà été évoqués **ici**

La structure est similaire à celle évoquée dans la partie recherche de ce projet (disponible **ici**). Seul un champ *idOSM* a été ajouté aux deux tables. Ce champ correspond à l'identifiant unique de l'objet dans OpenStreetMap.

L'attribut 'tags'

Comme cela a été évoqué dans la première partie de ce rapport, les données sont issues d'OpenStreetMap. Notre application repose essentiellement sur les champs *geometry* et *tags*.

Pour nos voies piétonnes que nous avons créées. Nous possédons tous les *tags* d'OSM qui était associés à chaque objet. Cependant, dans notre application, seul les *tags* suivants peuvent être modifié depuis l'interface avec les valeurs correspondantes :

- name
- highway->Residential
- highway->Service
- highway->Track
- highway->Footway
- highway->Path
- highway->Teriary
- highway->Primary
- highway->Secondary
- highway->Cycleway
- highway->Living-street
- highway->Motorway
- highway->Steps
- highway->Pedestrian
- highway->Unclassified

3 Interface web

3.1 Technologies

Plusieurs technologies ont été utilisées pour développement de l'interface web.

Angular2

Le site web [WWW1] a été développé sous le langage Angular2. Ce langage a été choisi car la société GéoVélo a son site web développé sous AngularJS. Elle avait le souhait de garder cette technologie pour ce projet. Seule la version change. Angular2 est la dernière version d'Angular. Cependant cette version est encore en Beta. La documentation n'est donc pas complète et la communauté plutôt faible.

Mapbox

Mapbox [WWW3] est un outil cartographique en ligne qui fournit des cartes pour les sites web. On peut y créer nos propres cartes, et les utiliser dans nos sites webs. Les cartes sont stockées sur un compte personnel et l'utilisation des cartes se fait via un appel Http. C'est également dans Mapbox que l'on importe nos tuiles (couche) que l'on souhaite ajouter à nos cartes.

Leaflet

Leaflet [WWW2] est une librairie Javascript pour les cartes interactives. Elle met à disposition tout un ensemble de fonctionnalités qui permettent d'interagir avec des cartes. Et notamment celle de Mapbox. C'est grâce à cette librairie qu'il est possible de charger les cartes de Mapbox, et d'y dessiner/ajouter tout un panel de composant. Comme dessiner des lignes sur la carte, récupérer la latitude et longitude où le clique a été effectué etc.

TileMill

TileMill [WWW4] est un outil qui permet de créer des tuiles (sous forme de PNG), afin de les afficher sur une carte. Une tuile est simplement une couche que l'on va appliquer au-dessus de la carte. Dans notre projet, la tuile créée et celle du réseau piéton. La tuile est générée directement grâce aux données que nous possédons dans notre base de données géographiques développée et alimentée au premier semestre.

Material Design Lite

Le design de l'application a été fait grâce à Material Design Lite. Material Design Lite est un outil permettant de créer des interfaces facilement et rapidement. Il s'agit d'un framework CSS développé par Google.

3.2 La tuile : représentation du réseau piéton

La première étape a été de créer la tuile qui aura pour but de représenter le réseau piéton sur la carte. Une tuile peut être considérée comme une couche qui vient s'ajouter par-dessus la carte.

La tuile a été générée grâce à deux outils qui sont QGIS et TileMill. Pour rappel, QGIS permet de visualiser des données géographiques d'une base de données. Grâce à ce logiciel, un fichier au format MBTiles a été généré. Ce fichier contient toutes les informations de notre réseau piéton.

Ce fichier a été ensuite importé dans le logiciel TileMill. C'est depuis ce logiciel que la tuile a été créée. Il est possible d'appliquer tout un ensemble de style sur notre tuile (couleur, taille etc...). Une fois importée et désignée, la tuile est générée au format PNG.

Voici à quoi ressemble l'interface de création de la tuile :

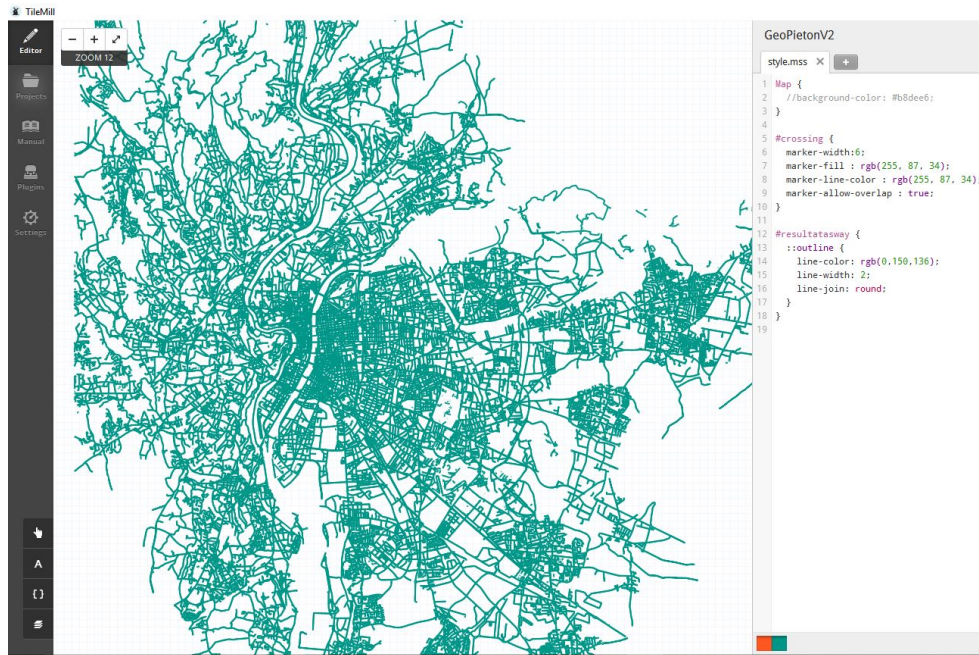


Figure 2 – Création de la tuile depuis TileMill

Une fois le PNG généré, il est importé sur Mapbox. Une fois importé, il est possible d'utiliser la tuile sur les cartes proposées par Mapbox (que nous utilisons).

3.3 L'interface

Tout d'abord, rappelons un des objectifs du projet qui était de pouvoir visualiser le réseau piéton de la ville de Lyon à travers une interface web. L'interface est donc un site web, permettant de visualiser les données de la base (construites au premier semestre) et d'interagir avec elles.

Le design a été fait grâce au framework CSS Matéria lDesign Lite.

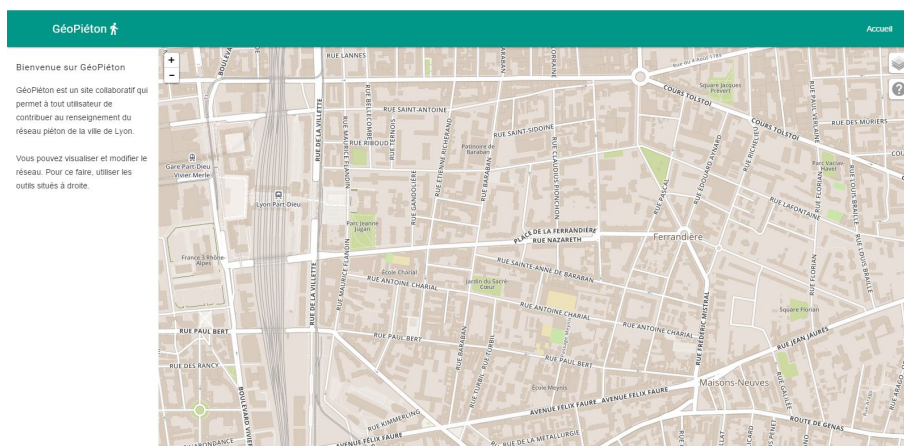


Figure 3 – Page d'accueil

Depuis cette interface, il est possible d'effectuer d'autres opérations que la visualisation du réseau piéton. Voici la liste des différentes fonctionnalités que cette interface dispose :

- Afficher/Cacher le réseau piéton.
- Afficher les voies dans un rayon de 50 mètres du clique effectué sur la carte.
- Sélectionner l'une de ces voies afin de voir ses caractéristiques.
- Modifier les caractéristiques d'une voie.

La **page d'accueil** du site est très basique, on y trouve la carte de Lyon, ainsi qu'un panneau latéral qui permet d'afficher différentes informations. Sur la droite on y trouve deux outils. Le premier permet d'afficher ou de cacher le réseau piéton (**voir affichage**). Le deuxième outil permet d'effectuer des requêtes sur la carte. Cette requête permet d'afficher les voies qui se trouvent dans un rayon de 50 mètres du clique (**voir affichage**). Une fois les voies affichées, il est possible d'en sélectionner une afin de voir ses propriétés, mais également de les modifier (**voir affichage**).

À savoir que le site ne présente aucun rafraîchissement de page, dû à la technologie utilisée qui est Angular2.

3.4 Structuration du site

Le site est structuré suivant l'architecture des projets développés sous Angular2.

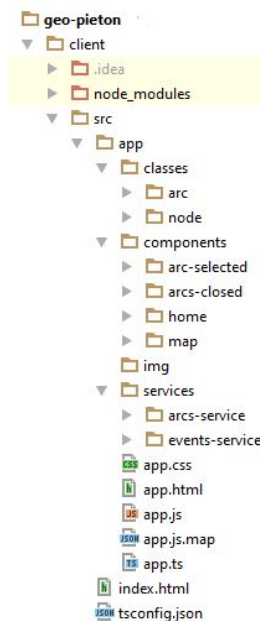


Figure 4 – Structuration du site web

Voici à quoi correspondent les dossiers et fichiers principaux :

- node_modules : contient toutes les librairies JS utilisées.
- arc : contient la classe qui représente une voie.
- node : contient la classe qui représente un noeud.
- components : les "components" Angular sont similaires aux contrôleurs.
- services : contient les classes qui vont interagir avec le web-serveur.
- tsconfig : contient l'ensemble des dépendances du projet (librairies externes).

4 Web serveur

4.1 Technologies

Le web serveur a été développé en NodeJS. NodeJS et Angular s'associent facilement. NodeJS, contrairement au Javascript, permet d'utiliser le JS du côté serveur. À la manière du PHP.

Les principaux avantages de NodeJS sont sa performance et son adaptabilité sur les applications qui doivent répondre à de nombreuses requêtes rapidement et efficacement, en temps réel. Comme l'application est développée sous Angular, il n'y a aucun rafraîchissement de pages, les requêtes sur la base de données doit être effectuées rapidement pour ne pas gêner l'expérience utilisateur.

4.2 Rôle

Dans cette application, le web-serveur a pour rôle de faire le lien entre l'interface graphique et la base de données. Lorsque l'application web a besoin d'effectuer une requête dans la base de données, cela ne se fait pas directement. Le site web (client) va effectuer une requête au serveur. Une fois la requête reçue, le serveur va effectuer la requête PostgreSQL dans la base de données. Les requêtes entre le client et le serveur se font via le protocole HTTP.

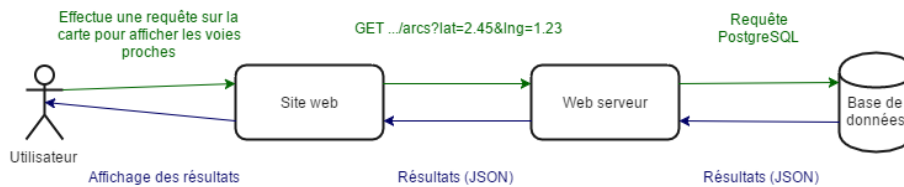


Figure 5 – Rôle du web-serveur

4.3 Structuration du web-serveur

La structuration est similaire à celle de l'interface web :

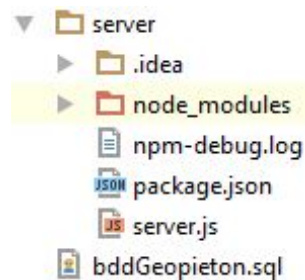


Figure 6 – Structuration du serveur

Dans ce projet, le web serveur permet de répondre aux requêtes suivantes et renvoie les données sous format JSON :

- GET `.../api/v1/arcs?lat=param1&lng=param1` : Renvoie toutes les voies (arcs) qui se trouvent dans un rayon de 50 mètres du point de coordonnée (latitude=param1 et longitude=param2) ;
- GET `.../api/v1/arc?id=param` : Renvoie les informations de la voie (id, nom, type) dont l'id de la base de données correspond au paramètre fourni.
- GET `.../api/v1/arc/update` : Modifie la voie avec les données fournis dans le formulaire (id, nom, type)

7

Tests

1 Tests fonctionnels

Des tests fonctionnels ont été effectués sur l'interface web. Voici la liste exhaustive des tests effectués sur l'interface web :

Test 1 : Afficher / Cacher la tuile (réseau piéton)

Description : Depuis l'interface web, quel que soit la page où l'utilisateur se trouve, il est possible d'afficher ou de cacher la tuile du réseau piéton. Cela se fait grâce à l'outil de gestion de tuile, situé à droite de l'écran.

- La tuile n'est pas affichée.
- Je coche la checkbox qui permet d'afficher la tuile.
- La tuile s'affiche bien sur la carte.
- Je décoche la checkbox.
- La tuile disparaît bien.

Test 2 : Test du bouton de requête

Description : Depuis l'interface web, quel que soit la page où l'utilisateur se trouve, il est possible d'effectuer une requête sur la carte permettant d'afficher la liste de toutes les voies (dans le panneau de gauche) situées à 50 mètres du clique. Cela se fait grâce à l'outil de requête symbolisé par l'icône '?' à droite de l'écran.

- Le bouton pour effectuer les requêtes est désactivé.
- Je clique sur la carte, rien ne se passe.
- Je clique sur le bouton de requête, le bouton est activé.
- Je clique sur la carte.
- La requête se lance.

Test 3 : Affichage des voies sur la carte au survol de leur nom dans le panneau de gauche

Description : Après une requête sur la carte, l'ensemble du nom des voies est affiché dans le panneau de gauche. Au survol de ces noms, la voie se dessine sur la carte.

- La liste des voies est affichées sur le panneau de gauche.
- Je survole chacune des voies.
- Pour chaque voie, la voie est bien affichée en orange sur la carte et au bon endroit.

Test 4 : Affichage des propriétés de la voies sélectionnées

Description : Après une requête sur la carte, l'ensemble du nom des voies est affiché dans le panneau de gauche. Lorsque l'on clique sur un nom, les propriétés de cette voie sont affichées et le nom des autres voies disparaît.

- La liste des voies est affichées sur le panneau de gauche.
- Je clique sur le nom d'une des voies.
- Les propriétés de la voie sont bien affichés et les autres nom de voies disparaît.

Test 5 : Modification des propriétés d'une voie

Description : Lorsque les propriétés d'une voie sont affichées, il est possible de les modifier via le formulaire.

- Les informations de la voie sont affichées.
- Je change le nom et le type de la voie.
- Je valide le formulaire.
- Les informations ont bien été modifiées en base de données.

2 Tests de performance

Temps d'exécution du script PostgreSQL

Le script de génération du réseau piéton a été exécuté sur l'ensemble des données géographiques de la ville de Lyon. Les données sont relativement importantes. Elles représentent un peu plus d'un million d'enregistrement dans les tables.

Pour effectuer le traitement, sur une machine de 2.5 Ghz et 4GO de RAM, le temps d'exécution a été de deux jours environ. Ce qui est important. Cependant, ce script est exécuté qu'une seule fois. Car une fois les données traitées, elles n'ont plus besoin d'être régénérées. L'importance du temps d'exécution est donc moindre dans ce projet.

Temps de réponse du web-serveur

Le temps de réponse du web-serveur est assez important dans cette application. En effet, lorsque l'utilisateur effectue une opération, le temps d'attente doit être le plus faible possible pour que l'expérience utilisateur ne soit pas impactée.

Des tests avec JMeter ont été effectués afin de mesurer le temps de réponse du web-serveur. Apache JMeter est un projet de logiciel libre permettant d'effectuer des tests de performance d'applications et de serveurs selon différents protocoles ainsi que des tests fonctionnels. Cependant, le web-serveur étant lancé depuis un ordinateur personnel, les performances en sont fortement réduites.

Pour ces tests, JMeter a été configuré de manière à effectuer une dizaine de fois la requête de façon instantanée.

Test 1

Test sur la requête `"/api/v1/arc?id=param"` en méthode GET. Qui renvoie l'ensemble des informations d'un arc dont l'id est passé en paramètre.

| | | | | | | | |
|-----|-----|-----|--------------|------|---|-----|------|
| 601 | ... | ... | HTTP Request | 2179 |  | 684 | 2179 |
| 602 | ... | ... | HTTP Request | 2124 |  | 684 | 2124 |
| 603 | ... | ... | HTTP Request | 2874 |  | 684 | 2874 |
| 604 | ... | ... | HTTP Request | 4675 |  | 684 | 4674 |
| 605 | ... | ... | HTTP Request | 5925 |  | 684 | 5925 |
| 606 | ... | ... | HTTP Request | 6598 |  | 684 | 6598 |

Figure 1 – Test 1 de performance du web-serveur

Test 2

Test sur la requête `"/api/v1/arcs?lat=param1&lng=param1"` en méthode GET. Qui renvoie l'ensemble des arcs situés à 50 mètre de la latitude et longitude passées en paramètre.





| | | | | | | |
|-----|-----|--------------|-------|---|------|-------|
| 627 | ... | HTTP Request | 5721 |  | 3209 | 5721 |
| 628 | ... | HTTP Request | 7202 |  | 3209 | 7202 |
| 629 | ... | HTTP Request | 10413 |  | 3209 | 10413 |
| 630 | ... | HTTP Request | 11758 |  | 3209 | 11758 |

Figure 2 – Test 2 de performance du web-serveur

Pour les deux tests, on constate que le taux de latence augmente rapidement et de manière constante. Cela est dû aux performances de l'ordinateur où a été lancé le web-serveur (en grande partie).

Quelques tests ont été également effectués directement depuis le navigateur. Les temps de réponse sont plus faible que ceux trouvés avec JMeter. Ce qui est normal car il n'y a qu'une seule requête qui est effectué à la fois.


| | | | | | | |
|---|-------------------------------------|-----|-----------|-----|-----------|-------|
|  | arc?id=15718026 localhost/api/v1 | GET | 200 OK | xhr | ht.. 5... | 55 ms |
| | | | | | Sc.. 2... | 53 ms |

Figure 3 – Test 3 de performance du web-serveur

8

Reproductibilité

1 Configuration de l'environnement

Afin de pouvoir lancer l'application web (site web) et le serveur web, il est nécessaire de préparer l'environnement.

Installation de NodeJS et npm

Pour installer NodeJS sur Windows, il suffit de télécharger l'exécutable sur le [site officiel](#) et de lancer l'installation. Une fois l'installation finie, NodeJS est installé sur l'environnement. A savoir que npm est inclus dans NodeJS. Npm est un gestionnaire de package. Mais il sert également à lancer et gérer des programmes node (tel qu'Angular).

Installation de la base de données

En ce qui concerne la base de données, il est nécessaire que PostgreSQL soit installé. L'installation sur Windows se fait également depuis un exécutable disponible sur [site officiel](#). Une fois l'installation effectuée, il est désormais possible de gérer des bases de données PostgreSQL.

Cependant, notre base de données contient des données de type géographique et PostgreSQL ne les supporte pas par défaut. Il faut donc installer une extension à PostgreSQL qui se nomme PostGIS (disponible [ici](#)). Une fois l'extension installée, il suffit de créer une base de données et d'importer le fichier bddGeopieton.backup qui va charger les données du réseau piéton de Lyon.

2 Lancement de l'application

Une fois l'environnement configuré, l'application peut être lancée. Pour que l'ensemble fonctionne, il faut lancer le client (qui est le site web) ainsi que le web-serveur.

Lancement du site web

1. Ouvrir une invite de commandes.
2. Placez vous dans le répertoire "client" de l'application avec le premier terminal.
3. Exécutez la commande "npm start". Cette commande permet de lancer l'application dans un onglet du navigateur.

Lancement du web-serveur

1. Ouvrir une invite de commandes.
2. Placez vous dans le répertoire "server" de l'application.
3. Lancer la commande "node server.js"

Compiler l'application web et le serveur

Pour lancer la compilation de l'application web, exécutez la commande "npm run tsc -p" dans le dossier "client". Cela va lancer la compilation. Une fois lancée, le compilateur va se ré-exécuter automatiquement lorsque celui aperçoit un changement dans un fichier source.

Pour compiler le serveur, il suffit d'arrêter celui-ci s'il est en cours d'exécution et de relancer la commande "node server.js".

Conclusion

Le domaine consacré aux algorithmes de génération de réseaux piétons est assez pauvre. En effet, de nos jours, ce sont les réseaux routiers qui sont en majorité utilisés pour le calcul d'itinéraire. Différentes recherches nous ont permis d'avoir une idée sur les différentes techniques qui ont déjà été réfléchies sur la génération de réseaux piétons.

Dans la première phase de ce projet, un algorithme a été mis en place permettant de construire le réseau piéton à partir du réseau routier, sous forme de base de données géographiques (PostGIS). Qui est basée sur la méthode de bufferisation des voies routières afin d'en extraire les voies piétonnes (trottoirs).

Dans la deuxième partie du projet, une application web a été développée permettant de visualiser le réseau piéton que nous avons construit lors de la première partie. Il est possible également de modifier les attributs des voies depuis cette interface.

L'ensemble forme donc un outil qui permet dans un premier temps de visualiser le réseau piéton de la ville de Lyon et de modifier les attributs des voies.

La prochaine étape de ce projet est d'y intégrer un outil permettant d'effectuer des itinéraires piéton sur ce réseau. C'est la société GéoVélo qui va se charger de cette intégration.

Pour conclure, ce projet m'a permis de découvrir un domaine nouveau pour moi qui est celui de la cartographie web. J'y ai découvert tout en ensemble de technologies et d'outils permettant de créer des applications cartographiques poussées.

Annexes

A

Cahier de spécification

Pour voir le cahier de spécification, veuillez vous référer au fichier "specification-geoPieton-brassart-maxime.pdf"

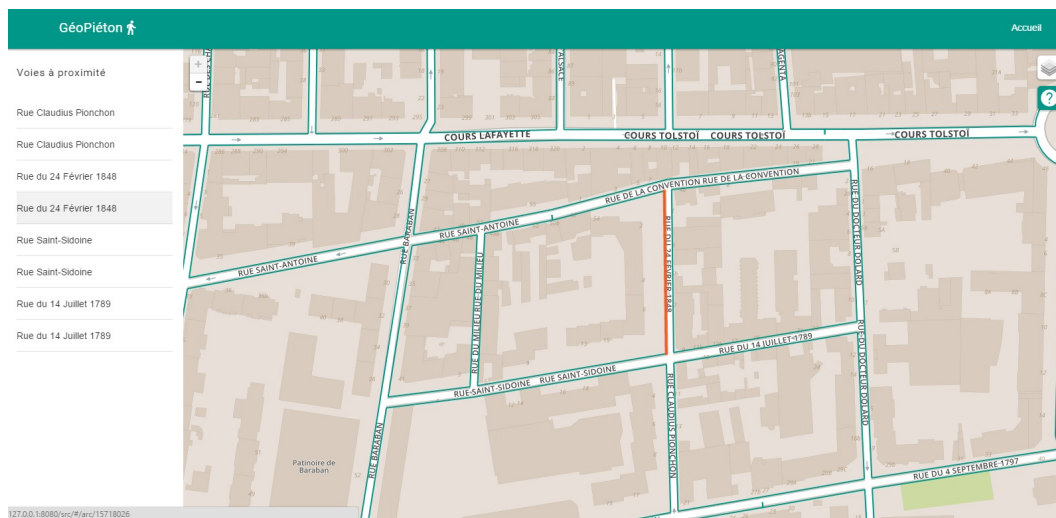


Figure 3 – Affichage des voies à proximité du clique

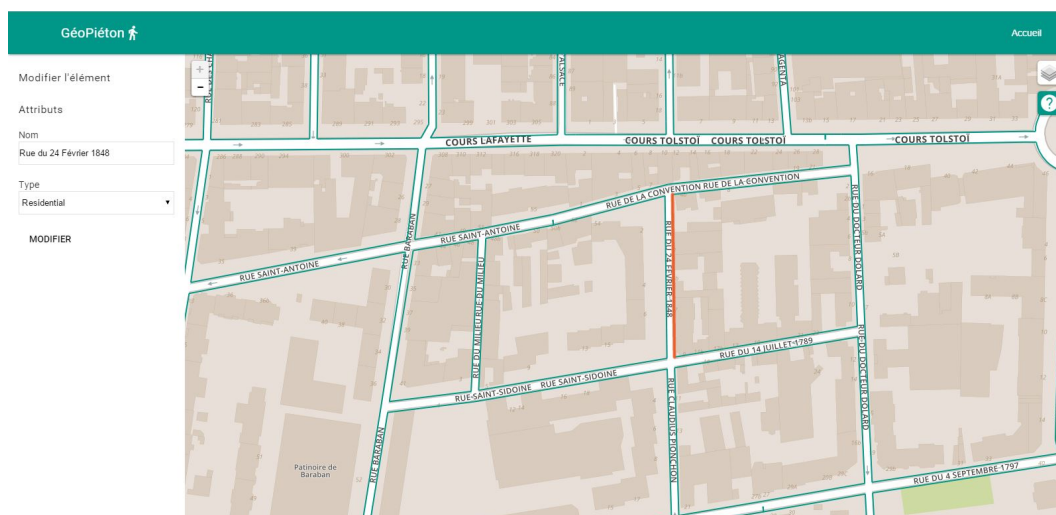


Figure 4 – Affichage des informations de la voie sélectionnée

C

Diagramme de cas d'utilisation

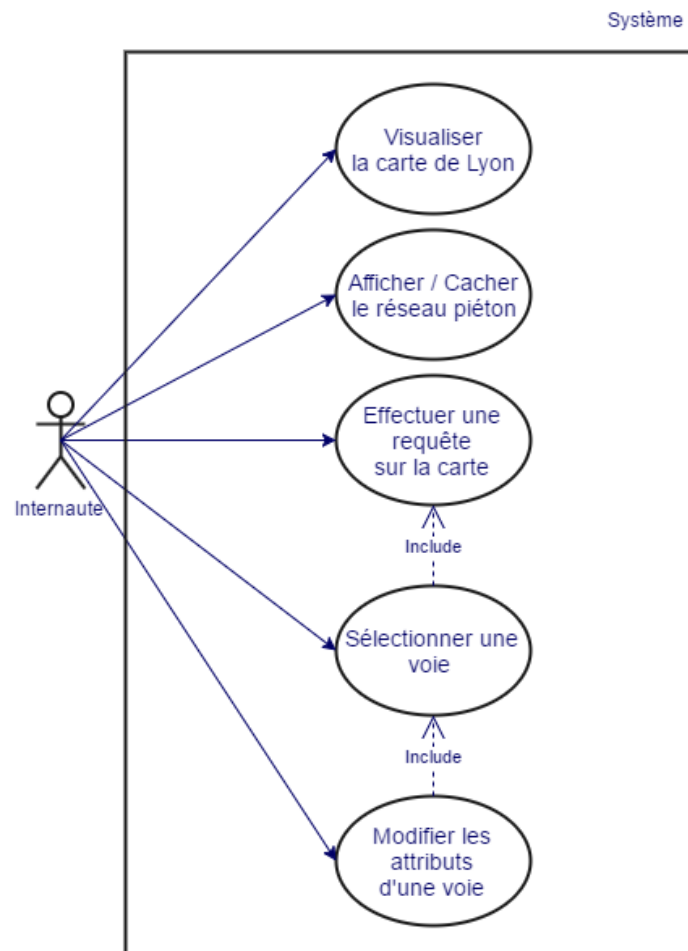


Figure 1 – Diagramme de cas d'utilisation

Comptes rendus hebdomadaires

Compte rendu n°1 du 17/09/2015

Étude des réseaux, et plus particulièrement les réseaux piétons. Recherche sur les algorithmes sur la génération d'un réseau piéton. Échange avec Antoine sur les différents documents que j'ai trouvés.

Compte rendu n°2 du 23/09/2015

Recherche sur les algorithmes sur la génération d'un réseau piéton. Un document intéressant a été trouvé. J'en ai fait part à Antoine. Renseignement sur les différents types de voies que peut emprunter un piéton. Parcours du Wiki de OSM afin de voir les différentes caractéristiques des objets qui vont nous intéresser.

Compte rendu n°3 du 30/09/2015

Réunion à GéoVélo. Explication détaillée du projet par GéoVélo. Présentation et échanges sur mes recherches. Prise de connaissance des outils que j'allais utiliser.

Compte rendu n°4 du 07/10/2015

Confirmation à Antoine que les outils ont été correctement installés et pris en main (PostGis...). Présentation d'un début d'algorithme sur la génération du réseau piéton à Antoine avec des premiers résultats. => Continuer les développements car réunion prévue avec Lyon le 04/11/2015

Compte rendu n°5 du 15/10/2015

Échange sur les derniers développements que j'ai effectués (Ajout des trottoirs, résolution des différents problèmes -> trottoirs qui s'entrecroisent sans raison, azimut)

Compte rendu n°6 du 22/10/2015

Réflexion sur certains cas problématiques avec Antoine. Concernant les croisements de routes. Envoi d'un mail à GéoVélo pour avoir un avis d'une personne ayant déjà manipulé PostGIS et qui pourrait nous aider.

Compte rendu n°7 du 06/11/2015

Nous avons réfléchi sur la problématique suivante : Comment supprimer les segments de trottoirs qui sont en trop ? Ceux qui se trouvent aux intersections et qui n'ont pas lieu d'être. Solution retenue : Créer pour chaque voie un polygone le représentant. Ainsi qu'un polygone représentant les intersections de voies (partie à supprimer).

Compte rendu n°8 du 13/11/2015

Implémentation de la génération des polygones évoquée lors de la réunion précédente. Démonstration des résultats que j'ai obtenus à Antoine. Objectif de la semaine suivante : Découper chaque trottoir en sous-segment pour pouvoir ensuite supprimer les segments des trottoirs appartenant aux différents polygones représentant les croisements.

Compte rendu n°9 du 27/11/2015

Implémentation d'un algorithme permettant de découper les voies à chaque intersection, dans le but de supprimer les segments de route indésirables.

Compte rendu n°10 du 03/12/2015

Prise de rendez-vous à GéoVélo pour montrer mon avancé. Antoine m'a demandé de regarder s'il était possible de créer un fichier PBF à partir des données de ma base de données représentant le réseau piéton.

Compte rendu n°11 du 09/12/2015

Réunion à GéoVélo pour montrer l'avancé de mes développements. Conclusion de la réunion : réussir à distinguer le trottoir de gauche et de droite.

Compte rendu n°12 du 16/12/2015

Rédaction du rapport. Le problème de la semaine dernière (distinction trottoir de gauche et droite n'est pas encore réglé).

Compte rendu n°13 du 23/12/2015

Prise en main de TileMill (outil permettant de créer la tuile qui va représenter le réseau piéton et qui sera affichée sur la carte de Lyon).

Compte rendu n°14 du 07/01/2016

Lancement de l'algorithme sur l'ensemble des données -> extraction complète du réseau piéton de Lyon. Création de la tuile (TileMill) sous forme de PGN -> représentation graphique du réseau piéton.

Compte rendu n°15 du 14/01/2016

Préparation soutenance (Power Point) + soutenance. Formation à AngularJS + WebServeur Node JS.

Compte rendu n°16 du 21/01/2016

Début de mise en place de l'interface web (Carte + Tuile + Webserveur + Design) Formation à AngularJS + WebServeur Node JS.

Compte rendu n°17 du 28/01/2016

Finalisation de l'intégration de la carte + tuile. Structuration du site (conceptuellement).

Compte rendu n°18 du 04/02/2016

Implémentation des fonctionnalités :

- Méthode retournant les arcs proches de la latitude et longitude données (webserver + base de données).
- Afficher les voies qui se trouvent dans un rayon déterminé du clique sur la carte (requête sur webserver).

— Afficher le nom de ces voies sur le panneau de gauche du site.

Compte rendu n°19 du 11/02/2016

Implémentation des fonctionnalités :

- Méthode retournant les informations sur une voie grâce à l'id (webservice + base de données).
- Afficher la voie sélectionnée sur la carte.
- Afficher toutes les informations de cette voie sur le panneau de gauche du site.
- Possibilité de modifier ces informations en "live" (début d'implémentation)

Compte rendu n°20 du 31/02/2016

Prise en compte de la remarque : Ne plus afficher l'ensemble des voies aux alentours du clique. Fin d'implémentation de la fonctionnalité de modification des attributs des voies (qui ont été fixés par l'encadrant).

Compte rendu n°21 du 10/03/2016

Changement du design global (couleur tuile, positionnement des éléments...). Ajout du bouton pour effectuer les requêtes sur la carte. Recherche d'une méthode à ajouter dans le script de génération du réseau afin d'ajouter les passages piétons qui ne sont pas créés actuellement (aux intersections). Mise au propre du code grâce à l'inspecteur de code de WebStorm.

Compte rendu n°22 du 17/03/2016

Début du rapport. Réunion à GéoVélo.

Compte rendu de la réunion : Ajouter la génération des passages piétons au script, ainsi que l'indication du sens des voies.



Webographie

- [WWW1] *site officiel Angular2*. URL : <https://angular.io/>.
- [WWW2] *site officiel Leafletjs*. URL : <http://leafletjs.com/>.
- [WWW3] *site officiel Mapbox*. URL : <https://www.mapbox.com/>.
- [WWW4] *site officiel TileMill*. URL : <https://www.mapbox.com/tilemill/>.
- [WWW5] *University of Maryland Pedestrian Map System*. URL : <https://terpnav.umd.edu/>.

Bibliographie

- [1] Miquel Ginard BALLESTER, Maurici Ruiz PÉREZ et John STUIVER. « Automatic Pedestrian Network Generation ». In : *AGILE* (avr. 2011).
- [2] Piyawan KASEMSUPPAKORN. « METHODOLOGY AND ALGORITHMS FOR PEDESTRIAN NETWORK CONSTRUCTION ». Thèse de doct. B.Sc. in Statistics, Chulalongkorn University et M.Sc. in Info Sys Mgt, The National Institute of Development Administration (Thailand), 2011.
- [3] Mohammadamin TAJGARDOON et Hassan A. KARIMI. « Simulating and visualizing sidewalk accessibility for wayfinding of people with disabilities ». In : *International Journal of Cartography* (2015), p. 79–93.

GéoPiéton : Génération d'un réseau piéton à partir d'un réseau routier

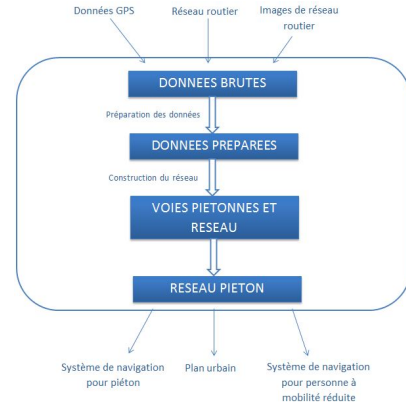
Maxime Brassart

Encadrement : Antoine Giret

En collaboration avec
GéoVélo

Description du projet

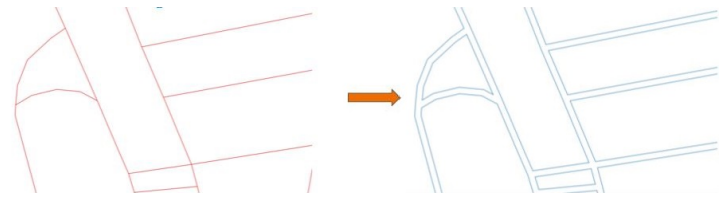
Le but de ce projet est de développer un algorithme qui permet de construire le réseau piéton de la ville de Lyon, à partir de son réseau routier. Les données sont issues de bases de données géographiques.



Principe

Algorithme de construction du réseau piéton

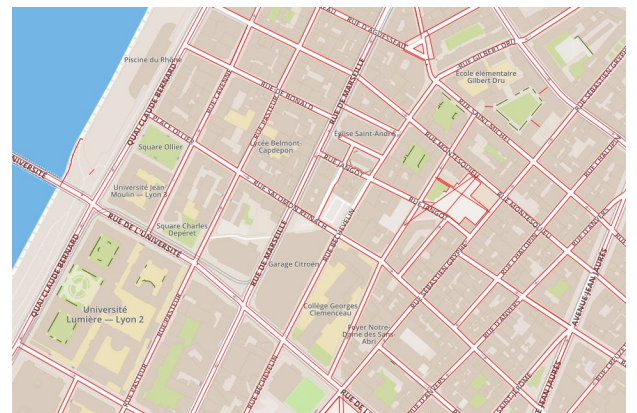
À partir du réseau routier d'une ville, il est possible de construire son réseau piéton. Diverses méthodes existent. L'une de ces méthodes se nomme la **bufferisation**. Cette méthode consiste à construire des polygones autour de chaque route où il y a présence de trottoirs afin d'en extraire les voies piétonnes.



Avant / Après exécution de l'algorithme

Représentation du réseau piéton

Une fois le réseau piéton construit, il est affiché sur la carte de Lyon à travers une interface web. Cette interface permettra d'effectuer des calculs d'itinéraires piétons.



Rendu graphique d'une partie du réseau piéton sur une carte

GéoPiéton : Génération d'un réseau piéton à partir d'un réseau routier

Résumé

Projet de fin d'études supervisé par Antoine Giret qui a pour but de développer un algorithme permettant de construire le réseau piéton d'une ville (Lyon) à partir de son réseau routier. Ce réseau doit être stocké dans une base de données géographiques. L'interface web permet de visualiser ce réseau, mais également de le modifier. Il s'agit d'une application de cartographie web.

Mots-clés

Polytech'Tours, Projet Recherche et développement, GéoPiéton, Génération de réseau piéton, Algorithme réseau piéton, Itinéraire piéton, Données géographiques, Cartographie web

Abstract

This project is supervised by Antoine Giret. The goal of this project is to develop an algorithm which build the pedestrian network of a city (Lyon) from its road network. This network should be stored in a geographic database. It's possible to view this network and modify too. This is a web-mapping application

Keywords

Polytech'Tours, Research and development project, GéoPiéton, Pedestrian Network Generation, Pedestrian Network Algorithm, Itinerary pedestrian, Geographic data, Web mapping

Entreprise

GéoVélo

Tuteurs entreprise

Antoine GIRET

Étudiants

Maxime BRASSART (DI5)

Tuteurs académiques

Antoine GIRET