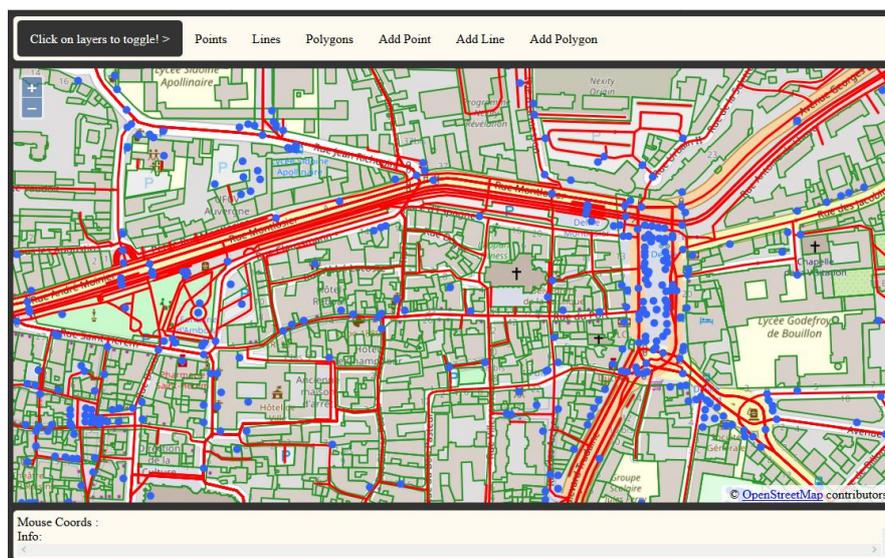


Rapport de stage
Cycle : ING 3 DDMEG

Prototype d'une plateforme de construction de carte tactile
Projet ACTIVmap



Ridley CAMPBELL

12/07/2021

Non confidentiel Confidentiel IGN Confidentiel Industrie jusqu'au

Commanditaire :

Institut Géographique National de l'Information Géographique et Forestière

73 Avenue de Paris, Saint-Mandé 94160 France

Encadrement du stage :

Guillaume TOUYA, Directeur de recherche, maître de stage

Responsable pédagogique du cycle :

Cécile HUET : ENSG, référente ENSG principale

Stage du 01/02/2021 au 17/07/2021

Nombre de pages : 51 dont 13 d'annexes

Version : Finale

Je dédicace ce rapport à toute personne persévérant pour traverser des périodes difficiles. Malgré toutes les peurs, les anxiétés, les défaites, ce travail marque la fin de mes études. Qu'il soit la preuve d'une victoire résonante et triomphante.

Remerciements

Je voudrais remercier les personnes suivantes :

Guillaume Touya : pour ses conseils et son respect, continus depuis les débuts du stage. J'aurai appris énormément de ses leçons, et m'en rappellerai bien après ces 5 petits mois !

Cécile Huet et Delphine Gagnon-Genès : pour leur soutien précieux, professionnel et bienveillant, aussi bien lors du stage que pendant mes études à l'ENSG

Emmanuel Fritsch : pour son soutien administratif au long du stage, et pour avoir répondu à l'appel de d'accueil de Saint-Mandé pour me permettre de rentrer au premier jour du stage !

Tous les membres d'ACTIVmap et du couloir du 3^{ème} étage du bâtiment K : pour avoir contribué à un environnement de travail professionnel et chaleureux

Et toutes les autres personnes, chercheur.euses, doctorant.es ou professeur.es, qui m'ont aidé de près ou de loin durant ce projet de fin d'études, ou prêté main forte pour relire ce rapport.

Résumé

Le projet **ACTIVmap** (*Assistance à la Conception des carTes pour déficients Visuels*), cherche à développer des cartes libres accessibles aux personnes atteintes de déficiences visuelles. Par le biais d'une plateforme en ligne, un utilisateur pourra parcourir une carte épurée et adaptée, puis la modifier interactivement pour un trajet ou une utilisation précise. Un intérêt particulier est porté à la traversée de carrefours, parfois complexes, notamment par des informations sonores, pour rajouter des détails sans perdre la dimension épurée de la carte. Cette carte deviendra ensuite utilisable sous forme tactile, soit par une impression 3D ou en relief, soit par un support tactile adapté, sur tablette ou téléphone portable.

Mon stage porte sur la conception et le développement de la plateforme en ligne d'ACTIVmap, de l'outil interactif de cette plateforme, et de la base de données servant la carte. Ce stage est relativement court par rapport à la durée du projet entier (5 mois/ 4 ans), et arrive dans la première année d'ACTIVmap. Mes objectifs étaient donc de développer une base informatique servant à la suite du projet, plutôt que de livrer une plateforme aux fonctionnalités finies. Enfin, j'avais aussi une mission de flexibilité et d'écoute, pour que la plateforme naissante soit cohérente avec les visions et les besoins des différents experts travaillant dans l'équipe ACTIVmap.

Abstract

The goal of the ACTIVmap project is to develop free and accessible online maps for visually impaired or blind people. ACTIVmap will be based on an online platform, displaying a symbolically simplified map, which can then be modified interactively, adding or removing details, to sculpt it for its intended use. Particular attention has been paid to crossroads and how to help navigate them, notably through audio prompts and information, which help convey extra details without overloading the map. Once modified, the online map can be exported for tactile use, whether through physical printing (3D or heated paper) or on adapted tactile tools (on phones or tablets).

My goal was to design and develop the ACTIVmap online interface, build the interactive map editing tools and structure the cartographical database serving the map. The internship is relatively short compared to the scope of the whole project (5 months/ 4 years), and begins within ACTIVmap's first year. My objectives were thus not to deliver a finished interface and online platform, but more so to build up a development base, a prototype, which will be useful for the future of the project. Finally, my mission was also to listen and communicate, to incorporate all the views and the needs of the various experts working on the projet into my prototype!

Sommaire

Introduction	8
Partie 1 : Etat de l'art et objectifs.....	9
I. L'équipe et le projet.....	9
II. Mes objectifs et mes contraintes	12
a) Contraintes et objectifs liés à la plateforme.....	12
b) Contraintes et objectifs liés à l'interface	13
c) Contraintes pratiques et physiques	15
III. Etat de l'art	16
Partie 2 : Recherche et Développement.....	18
I. Problèmes et solutions	18
II. Développement de l'interface	20
a) Structure de la page	20
b) Instanciation des couches (« <i>layers</i> »).....	21
c) Fonctionnalités (« <i>functions</i> »)	22
III. Développement de la BDD cartographique.....	24
IV. Le projet CartAGen	29
Conclusion.....	33

INTRODUCTION

Mon stage de fin d'études a commencé le 1^{er} Février 2021 au sein de l'équipe **GéoVIS** (Visualisation, Interaction et Immersion) du laboratoire **LaSTIG** (Laboratoire de Sciences et Technologies de l'Information Géographique pour la Ville et les territoires Numériques) de l'**IGN** (Institut National de l'Information Géographique et Forestière). Mon recrutement, mon installation et la supervision de mes travaux ont été encadrés par Guillaume Touya, directeur de recherche au LaSTIG, qui codirige aussi le projet ACTIVmap sur lequel porte mon stage.

ACTIVmap a pour objectif de produire des cartes tactiles ou imprimables en relief, à partir de données **OSM** (OpenStreetMap), destinées à des utilisateurs déficients visuels. Le projet prend en compte la production de cartes tactiles de A à Z, et réunit donc plusieurs organismes et laboratoires, dont le LaSTIG, chacun apportant son expertise sur un certain domaine. Mon stage porte plus précisément sur le développement de la plateforme en ligne d'ACTIVmap, ainsi que la création de l'architecture de la base de données cartographique qui va servir cette plateforme.

La problématique à laquelle j'ai dû répondre est donc celle d'un soutien technique et logiciel à la construction et au développement du projet ACTIVmap. J'ai dû développer un outil libre centré autour d'une carte en ligne, servie depuis une base de données adaptée. Cette carte et la symbologie de ses objets pourront être modifiés, affinés interactivement, puis visualisés en 3D avant d'être exportés sous un format compatible à une impression en relief ou à un outil spécialisé. Les fonctionnalités finales de la plateforme ne sont pas gravées dans le marbre : elle servira d'abord aux membres du projet à faire des essais, et à terme elle doit pouvoir évoluer, être simplifiée et devenir accessible à des utilisateurs pas forcément adeptes des **SIG** (Systèmes d'Information Géographiques). Toutes ces données seront servies depuis une **BDD** (Base De Données) cartographique dont l'architecture reflètera la symbologie et les objets particuliers de la carte épurée, et il me revient aussi de la développer et de l'installer.

Pour ce faire, j'ai eu accès aux bureaux de l'IGN à Saint-Mandé, et à un ordinateur portable pour la durée du projet. La crise sanitaire a bien entendu affectée le stage, dont une grande partie a eu lieu en télétravail. Néanmoins, j'ai pu progresser, parfois lentement mais toujours sûrement, grâce notamment à des entretiens vidéo réguliers avec mon référent, et profitant de plus en plus des bureaux de l'IGN au cours du stage. Ces points réguliers, ainsi que la tenue journalière d'un journal de bord, m'ont aidé à m'organiser et structurer mes efforts au long de ce stage.

PARTIE 1 : ETAT DE L'ART ET OBJECTIFS

I. L'équipe et le projet

Qu'est-ce qu'ACTIVmap ?

C'est un projet de recherche publique financé par l'**ANR** (Agence Nationale de Recherche). D'après le site officiel du projet : « ACTIVmap [...] est un PRCE (*Projet de Recherche Collaborative – Entreprise*) mené par le **LIMOS** (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes), conjointement avec l'équipe GeoVIS de l'UMR LASTIG (IGN), l'équipe ELIPSE (Etude de l'Interaction Personne Système) de l'Institut de Recherche en Informatique de Toulouse, et l'entreprise FeelObject. »¹. Cette définition souligne une propriété importante du projet ; son envergure, humaine comme thématique, et donc sa complexité. Il regroupe en effet des acteurs d'à travers la France, mais les restrictions sanitaires n'ont cette année jamais permis à ces différents groupes de se rejoindre physiquement pour échanger directement. Si le projet ACTIVmap commence officiellement en Mars 2020, les restrictions ont fortement ralenti ses débuts, et ce n'est que vers la fin 2020 qu'ACTIVmap a repris plus « normalement ». J'ai donc intégré la grande équipe ACTIVmap par le biais de GéoVIS, à Paris. L'équipe du LIMOS et l'entreprise FeelObject travaillent à Clermont-Ferrand, l'équipe ELIPSE est basée à Toulouse.

La première partie de mon stage, tout comme la première partie de ce rapport, était de plonger dans ce projet et de comprendre le rôle de chaque membre, pour pouvoir m'appropriier le mien.

Le but d'ACTIVmap, est de produire des cartes adaptées et adaptables aux besoins de déficients visuels et personnes aveugles. Cette problématique vaste requiert la coordination de travaux très différents et très spécialisés. En arrivant dans ACTIVmap, j'ai pu découvrir les travaux de deux étudiants doctorants du projet, Jérémy Kalsron et Markie Jiang, qui portaient

¹ [Site officiel du projet ACTIVmap : <https://activmap.limos.fr/> au 01/07/2021]

respectivement sur l'étude en profondeur de la traversée de carrefours pour une personne aveugle, et sur une première symbologie cartographique adaptée pour représenter cette traversée. Par exemple, une personne déficiente visuelle peut approcher des carrefours adjacents comme un seul et même carrefour, s'il est possible d'entendre les véhicules du carrefour plus loin depuis un passage du carrefour plus proche. Il faut donc potentiellement adapter les représentations cartographiques pour pouvoir identifier au toucher chaque élément d'un carrefour, allant de la direction de conduite aux sas vélos sur la route.

Le projet traite aussi de la question de comment transposer ces informations tactilement, et là encore différentes solutions sont explorées, notamment par l'entreprise startup FeelObject, faisant aussi partie d'ACTIVmap. FeelObject propose notamment « Virtuoz », un plan tactile applicable sur une tablette ou un téléphone ; une solution toute aussi pertinente que des cartes en relief plus classiques, imprimé en papier thermoformé, ou par impression 3D par exemple. Bien entendu, ces options différentes impliquent des propriétés, des prix et des portées différentes.

Le passage de données géographiques d'une carte visuelle à une représentation sur une carte tactile est complexe. On peut comparer les deux figures ci-dessous (*fig.1* et *fig.2*) pour apprécier la différence entre ces deux méthodes de représentations.



Figures 1 et 2 : représentation 'visuelle' et 'tactile' de la Place Delille de Clermont-Ferrand.²

² POC du projet ACTIVmap. Carte visuelle provenant d'OSM, carte 'tactile' par Markie Jiang (LaSTiG)

Le passage entre ces deux cartes demande plusieurs opérations, et la plateforme ACTIVmap servira à les recréer interactivement. Après avoir sélectionné une zone des données de base, une première phase consiste à simplifier et enrichir les données, en enlevant tous les détails superflus (comme le nom et le détail de bâtiments) et en rajoutant ou mettant en avant des informations supplémentaires (comme les passages piétons et les îlots directionnels). Les travaux sur la représentation et la traversée de carrefours complexes prennent toute leur importance ici, guidant l'importance donnée aux différents éléments de l'équipement routier.

Il faut ensuite appliquer des algorithmes de généralisation cartographique, pour simplifier par exemple la représentation des routes et des trottoirs. Tous les traits et routes de la carte ont une épaisseur minimum pour être identifiables et pouvoir être suivies au toucher. On peut donc omettre des routes plus fines, ou des traits trop rapprochés. Il faut parfois schématiser, styliser plutôt que de simplement généraliser les données. Au-delà de choisir l'épaisseur des traits, il faut aussi repenser la symbolisation des objets et des textes par rapport à une carte visuelle, d'où l'intégration d'information sonore à notre carte.

Tous ces traitements seront faits en partie interactivement sur la plateforme, avec l'aide de l'utilisateur, et en partie semi-automatiquement par des scripts prédéfinis. Ces exemples ne sont en réalité que peu représentatifs de toute la portée de ce projet, et de la complexité des différentes tâches. Je me concentrerai par la suite plutôt sur les parties du projet me concernant plus directement, et les tâches qui m'ont été confiées.

J'ai donc intégré le projet avec un rôle de développement logiciel et de conception d'architecture cartographique. J'expliquerai dans la prochaine partie de ce rapport mes objectifs et ce qui était attendu de moi en terme de livrables et de développement. Mais en fait, il n'y avait pas à proprement parler de *livrables* attendus. Les objectifs à long-terme pour l'interface et le projet ne correspondent simplement pas au cadre temporel d'un stage de 5 mois comme le mien. Mon référent Guillaume n'attendait donc pas de moi de produire une interface finie, mais plutôt de commencer une base logicielle solide pour la suite d'ACTIVmap.

Enfin, l'organisation pratique du stage s'est faite essentiellement par des réunions vidéo. J'ai déjà expliqué mon cadre de travail dans l'introduction, qui était largement en télétravail et ponctué de quelques jours par semaine à Saint-Mandé quand cela était possible. Des réunions intermittentes avec tous les membres du projet ont aussi eu lieu en ligne, pour discuter de l'architecture de la plateforme et de la vision de chacun du projet. J'ai aussi pu assister à différents évènements et conférences scientifiques, organisés en interne par l'IGN ou pour les Journées de

la Recherche notamment, qui m'ont permis de mieux comprendre les travaux de mes collègues et de chercheurs à travers la France. Ces réunions ont toutes été importantes pour comprendre mon rôle, puisque la plateforme que je développais devait pouvoir accueillir les parties de tous les acteurs du projet, avec une continuité logique et logicielle.

II. Mes objectifs et mes contraintes

Mon rôle dans le projet ACTIVmap a été de développer un prototype de plateforme en ligne, hébergeant une carte modifiable selon nos besoins et liée à une base de données à l'architecture appropriée. Un objectif secondaire a été de travailler sur l'intégration d'outils de généralisation dans la plateforme, par le biais de l'outil CartAGen, expliqué en dernière partie de ce rapport. J'ai séparé ci-après les contraintes qui m'étaient posées et les règles que j'ai eu à suivre en arrivant dans ce projet, en séparant thématiquement les contraintes liées à la plateforme, au développement, et enfin aux autres conditions plus globales.

a) Contraintes et objectifs liés à la plateforme

Une première condition du développement de la plateforme était d'être basée sur des logiciels libres. J'expliquerai dans la prochaine partie l'état de l'art effectué en début de projet, et quels logiciels libres ont été choisis. Le fond cartographique de base serait tiré des données libres OSM, choisi pour sa liberté d'accès internationale et sa granularité sémantique cohérente avec ce projet, comme détaillé dans l'état de l'art plus loin dans ce rapport.

Ensuite, la flexibilité de ma plateforme était une condition importante, car une fois qu'un premier prototype serait prêt, il servirait à repenser l'architecture logicielle du projet. En effet, comme mentionné précédemment, ce projet est toujours relativement à ses débuts, et pour la majorité de mon stage j'ai travaillé seul sur le développement du prototype. Mais les autres membres d'ACTIVmap allaient bien entendu aussi utiliser le prototype pour tester leurs parties du projet ou confirmer des hypothèses. Le travail de généralisation cartographique et l'équilibrage de la symbologie pour une carte, n'est pas toujours une science exacte, et dépend en partie de conventions établies sur la lisibilité « confortable » d'une carte. Mais ces règles ne se traduisent

pas pour les cartes tactiles, et dans beaucoup de cas il n'y a simplement pas de conventions établies. C'est pourquoi avoir un prototype serait important, pour imprimer une première carte à montrer aux chercheurs et aux utilisateurs, ne serait-ce que pour décider de la largeur optimale pour les traits sur les cartes.

Enfin, à terme cette plateforme va aussi devoir servir à des utilisateurs non adeptes du SIG, pour générer et imprimer la carte dont ils ont besoin pour accompagner des personnes déficientes visuelles. Donc mon développement devait dans un premier temps permettre un grand contrôle sur tous les paramètres de la carte et de la symbologie des couches, tout en gardant l'optique de devenir une interface épurée et « vulgarisée » sur le long-terme.

b) Contraintes et objectifs liés à l'interface

Pour la carte en elle-même, mes contraintes étaient d'aller au plus simple. L'essentiel de mes efforts portaient sur le développement du squelette logiciel d'ACTIVmap, donc certains choix ont été simples, comme celui des données de base. Il se trouve que le LIMOS, acteur principal d'ACTIVmap, est centré sur Clermont-Ferrand en France, et que la ville regroupe quelques carrefours routiers plutôt complexes. C'est donc là que la base de données ACTIVmap a commencée à être construite, et c'est là que je me suis restreint pour mes travaux. Une petite zone de quelques km² autour du vieux centre de Clermont-Ferrand a constitué mon jeu de données de références. C'était simplement un jeu de données assez petit pour ne pas perdre de temps sur le chargement, mais assez grand et représentatif pour une construction cohérente de ma base de données.

Cela dit, le choix de se concentrer sur Clermont-Ferrand reflète aussi un choix plus important sur la portée du projet. Au-delà du fait que Clermont ait des carrefours complexes intéressants, permettant de tester au maximum les algorithmes de traversée de carrefours, il était utile au projet de se restreindre géographiquement pour se concentrer sur la précision de ses fonctionnalités. Ce choix servira aussi aux membres du LIMOS (basé à Clermont) de faire des expériences terrain avec des personnes déficientes visuelles, notamment pour le développement d'algorithmes de traversée de carrefours. En clair, le prototype serait restreint sur Clermont-Ferrand, s'appuyant sur des données de bases déjà traitées à la main.

Les fonctionnalités attendues sur cette carte étaient variées. L'aspect interactif de la carte concerne surtout une modification de la symbologie et des représentations. Concrètement, on veut pouvoir enlever et ajouter des objets à notre couche de base, et altérer les propriétés de leur symbologie. Les objets de la carte sont ponctuels, surfaciques ou linéaires, et il faut pouvoir accéder aux propriétés de chaque classe d'objet, pour modifier leur style. On veut pouvoir intégrer les modifications ainsi apportées à la BDD de référence, mais aussi pouvoir revenir à la carte de base avant les modifications.

Cette carte comportera aussi des « objets sonores », et des nouveaux types d'objets créés dans le cadre des travaux d'ACTIVmap sur les carrefours routiers. Bien entendu, ces objets ne sont pas initialement présents dans les données OSM. Il faut donc aussi pouvoir créer de toutes nouvelles classes, pas simplement rajouter des objets à des couches existantes. Par exemple, des nouvelles représentations des ilots directionnels, et l'intégration de données audio, seront à intégrer dans la symbologie et dans l'architecture de la BDD.

Il faut aussi prévoir l'intégration de tuilage vectoriel à la carte, même si elle n'est pas implémentée tout de suite. Le tuilage vectoriel permet de n'afficher qu'une seule « tuile » de données à la fois, servant simplement à alléger les temps de chargement et d'affichage, ce qui est particulièrement pertinent pour ACTIVmap. A terme, le projet devra aussi supporter des représentations multi-échelle. Différentes représentations des objets à différentes échelles, notamment pour les affichages des routes, seront donc à prévoir dans la conception de la carte. Ces deux dernières fonctionnalités sont plutôt « ambitieuses », et seront développées plus tard dans le cycle de vie du projet ACTIVmap : mais elles sont à intégrer à l'état de l'art, et aux choix de développement, pour ne pas se retrouver face à des problèmes de compatibilité de représentation.

On peut donc imaginer l'interface ACTIVmap comme une sorte de SIG en ligne, permettant un grand contrôle sur les données à afficher et leur représentation. Cela dit, mon objectif principal pour ce stage restait d'aller au plus simple, pour atteindre le but d'un premier prototype simple et fonctionnel. L'ensemble des fonctionnalités décrites ci-dessus n'ont pas été développées pendant mon stage, mais la base logicielle a été construite en les prenant en compte.

c) Contraintes pratiques et physiques

La contrainte physique la plus limitante durant ce stage a bien entendu été de devoir travailler en distanciel, compte tenu de la crise sanitaire. Pour la plupart du stage, j'ai eu un accès limité à l'IGN, pouvant venir au plus 2 jours par semaine aux bâtiments de Saint-Mandé : si ce n'est pas vraiment une contrainte liée au stage, c'est un facteur externe qui a affecté l'environnement de travail et l'accès à l'information de mon stage. J'expliquerai en plus de détails les conséquences de la crise sanitaire sur mon stage en début de Partie 2.

Ensuite, je devais effectuer mon stage sur un ordinateur portable IGN. Ceci n'est pas vraiment une contrainte, en lui-même, mais implique devoir passer par le VPN de l'IGN pour sécuriser les données, et la structure de l'IGN pour accéder aux réseaux internet internes dans les bureaux. La mise en place de mon compte IGN et le fait toujours devoir passer par ce VPN ont causé quelques soucis à de rares occasions, comme je détaille au début de la Partie 2 de ce rapport. En réalité, j'ai surtout bénéficié d'avoir accès à l'IGN, notamment grâce aux accès aux diverses colloques et activités organisées en ligne.

Enfin, comme mentionné précédemment, lors de mon arrivée dans ce projet, j'étais la seule personne travaillant sur le code de la plateforme. Je n'avais donc aucune contrainte liée au développement en commun, et pour la plupart de mon stage, je suis resté sur mon environnement local. Ceci permettait de contourner d'autres contraintes, notamment à cause du fait de devoir passer par le VPN de l'IGN pour travailler. Les membres du LIMOS n'ayant pas d'accès à ce VPN, je suis resté sur un géoserver en local pour la plupart de mon stage. Et ce n'est qu'en début du mois de Juillet qu'a été mis en place un Github commun au projet, suivant l'arrivée de plus de développeurs. Bien entendu, je n'avais pas de contraintes sur cet aspect du développement, j'ai tout de même du préparer mon code à être intégré à un projet en commun. J'ai pu intégrer mon travail à la branche principale du projet en fin de stage.

III. Etat de l'art

Le travail a commencé par un état de l'art des logiciels à utiliser pour le développement de la plateforme à venir. Compte tenu des différentes contraintes, et comme expliqué précédemment il fallait choisir des outils logiciels libres et flexibles.

PostgreSQL et l'application PostGIS ont plutôt rapidement été validés comme choix de support de base de données cartographiques. Ensemble, ils forment un outil libre, bien documenté, et surtout facile à relier à un serveur web. Si d'autres options comme MySQL ou SQLite existent aussi, PostgreSQL a été choisi pour sa solidité et sa compatibilité, notamment avec des services comme Géoserver. Ce duo Géoserver/postGIS est un choix classique pour le développement web libre, dont l'utilisation est flexible et très bien documentée.

Pour la partie server d'ACTIVmap, Géoserver a en effet été préféré à d'autres options comme Wampserver, QGISserver ou ArcGIS Online, notamment grâce aux liens simples à une BDD cartographique, et une fonctionnalité intégrée permettant de rajouter facilement du tuilage vectoriel plus tard dans le projet. L'aspect bien documenté de ce choix libre a aussi beaucoup compté.

Enfin, le choix le plus difficile a été celui de la bibliothèque JavaScript pour la plateforme. Les principaux 'candidats' considérés étaient Openlayers, Leaflet et MapLibre, et ils ont tous leurs avantages. MapLibre est une branche libre créée sur la base de MapBox, qui n'est plus un outil libre. Si MapBox présente des fonctionnalités intéressantes, notamment au niveau de la représentation en 2.5D ou en tuilage vectoriel, MapLibre était potentiellement moins mis à jour ou moins documenté que les deux autres choix. Leaflet et OpenLayers sont comparables, et OpenLayers a finalement été choisi pour sa documentation plus exhaustive et son support plus intéressant du tuilage vectoriel.

En fin de compte, le trio Postgres/OpenLayers/Géoserver a été retenu pour notre plateforme en début de projet. Cependant, ces choix n'étaient pas forcément finaux, et allaient de

toute façon être remis en question au cours du stage. Mon rôle dans ACTIVmap était aussi de participer à l'élaboration globale du fonctionnement de la plateforme. Mes objectifs initiaux n'étaient pas forcément clairs, et les fonctionnalités exactes dont on aurait besoin n'étaient pas totalement définies. C'est pour cela que je me suis reposé sur des logiciels bien documentés et fonctionnant bien ensemble, plutôt que de chercher des avantages sur certaines fonctionnalités en particulier. La création d'une première page web avec ces trois programmes n'était pas vraiment la fin de la tâche, mais plutôt l'occasion de montrer aux autres membres du projet comment fonctionnait l'interface pour ensuite la repenser, et repenser tout le projet autour.

Mon rôle a autant été de développer cette base informatique que de participer à l'organisation globale de la plateforme d'ACTIVmap. La figure suivante montre justement une représentation de cette architecture globale de la plateforme. C'est le produit de réunions avec les autres membres du projet, et découle de mes choix de l'état de l'art.

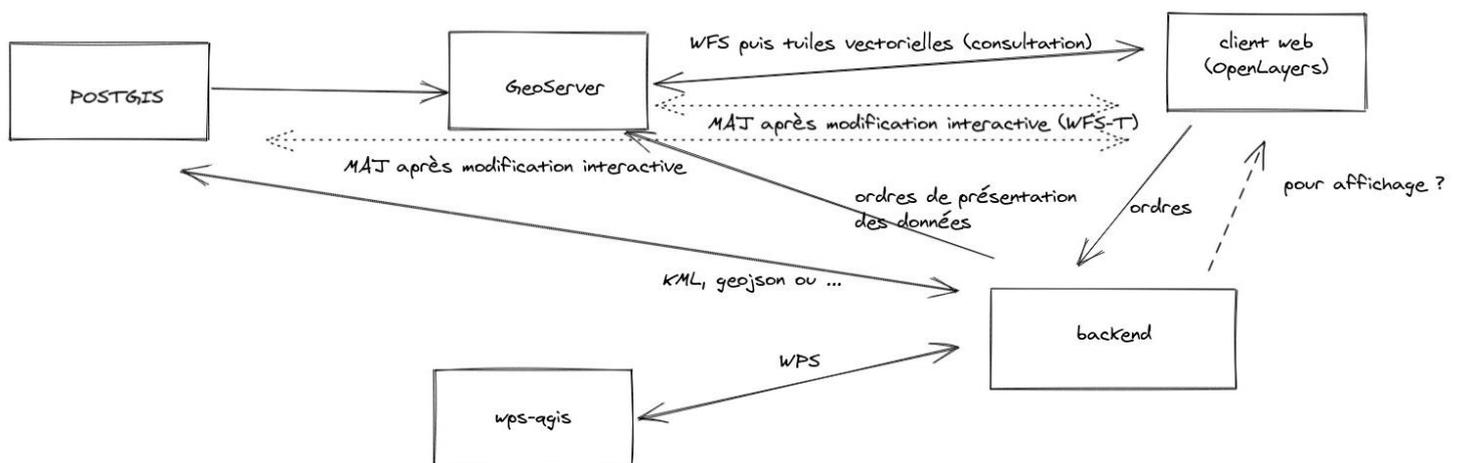


Figure 3 : Schéma du fonctionnement global et logiciel de la plateforme ACTIVmap.

On retrouve ci-dessus tous les éléments de l'interface décrits précédemment, et les interdépendances entre les différents outils. L'élément WPS-QGIS vers le bas du schéma correspond à l'inclusion des algorithmes de généralisation sous forme de scripts, qui ont été développés sur un QGISserver par d'autres membres du projet. Ce schéma détaille simultanément le résultat de mon état de l'art et ses implications logicielles et fonctionnelles.

PARTIE 2 : RECHERCHE ET DEVELOPPEMENT

Cette deuxième partie de mon rapport décrit tout ce que j'ai produit, les fonctionnalités implémentées, et les difficultés rencontrées. Je commencerai par détailler les différentes difficultés rencontrées au cours du stage et du développement et comment elles ont été surmontées, pour donner du contexte au code que j'ai produit.

I. Problèmes et solutions

L'élément principal de contretemps durant mon stage a été un problème de droits d'accès. J'ai travaillé depuis le début de mon stage sur un ordinateur portable fourni par l'IGN, aussi bien pour assurer la sécurité des données (l'IGN protège les données internes, et mes sessions de travail devaient passer à travers le VPN de l'IGN en utilisant mes identifiants) que pour me permettre de gérer des programmes et des données relativement lourds (installation de SIG, importation de grand volume de données d'OSM notamment). Mon problème : mon compte IGN n'avait pas de droits d'administrateur, y compris sur mon ordinateur. Sans ces droits, je ne pouvais pas lancer des fichiers certains .exe pour installer des programmes, ni accéder à des dossiers protégés sur mes disques C et D. Sans compte administrateur, je ne pouvais rien donc installer moi-même, et le PC de l'IGN était fourni vide. J'étais bloqué sans programmes pour coder et sans pouvoir télécharger des bibliothèques de programmation.

La demande pour avoir accès à un compte admin a été faite rapidement, mais je n'y ai eu accès qu'en début Avril, donc après 2 mois de stage. En attendant, je suis rentré en contact avec le service informatique de l'IGN, pour qu'ils installent à distance des programmes sur mon PC. Ces contacts étaient très utiles, me permettant de sortir de la phase d'état de l'art et installer des programmes pour commencer à produire du code. Mais ils étaient aussi limités, puisque l'accès à certains dossiers était toujours bloqué, et je devais faire de mon mieux pour prévoir le plus d'actions possibles à chaque intervention pour avancer le plus possible. Par exemple, en amont d'un contact avec le service info, je préparais des programmes à installer, mais c'est parfois

uniquement le jour d'après que je me rendais compte qu'il me manquait une bibliothèque à installer, ou qu'il me fallait une autre version du programme.

Je mentionnerai ici que je craignais particulièrement un problème de ce type, qui, couplé à une situation parfois incertaine ou inconfortable en confinement ou en télétravail, a des répercussions démotivantes et décourageantes. Ce sont justement les raisons pour lesquelles j'avais personnellement eu beaucoup de mal lors de mon stage précédent, pendant l'été 2020. Heureusement, j'ai réussi à surmonter cette étape, notamment grâce à mes contacts réguliers avec Guillaume, et le soutien de mes référentes à l'ENSG, Cécile et Delphine. Je pense que la pression psychologique et les problèmes de motivation sont pertinents à mentionner dans ce type de rapport, compte tenu de leur ampleur cette année, et la difficulté à y faire face, pour les étudiants comme pour les chercheurs. Pour ma part, j'ai été réconforté et motivé par la communication avec des encadrants bienveillants, à l'écoute et respectueux, comprenant la situation et valorisant les efforts et l'apprentissage plutôt que la productivité. Ces difficultés auront été une occasion pour moi d'en apprendre plus sur comment travailler et communiquer, plutôt que de simplement acquérir des nouvelles connaissances informatiques.

Les autres obstacles rencontrés lors de mon stage sont plus 'classiques', je les mentionnerai ici plutôt que dans les parties décrivant le code que j'ai rédigé. Le premier obstacle rencontré était lié à l'écriture de requêtes **WFS** (Web Feature Service) et **WMS** (Web Map Service) vers le géoserver du projet. Je donnerai plus de détails sur ces requêtes dans la prochaine partie, mais pour résumer, elles permettent de servir des couches du géoserver sur ma carte web. C'est aussi par des requêtes WFS complexes que sont possibles des modifications directes de la BDD PostGIS depuis l'interface web. Le problème est survenu lors des premiers essais de requêtes WFS, qui renvoyaient un problème lié aux autorisations **CORS** (Cross-Origin Request Sharing).

En fait, c'est un problème purement informatique : les paramètres du géoserver bloquaient les requêtes provenant de certaines sources. C'est sans doute la partie de mon stage que je maîtrisais le moins, et j'ai dû tester différentes solutions avant de trouver la bonne. J'ai pu aussi demander de l'aide autour de moi, et éliminer une à une les sources possibles de cette erreur. Parmi les sources possibles étaient le fait que je travaillais en passant par le réseau de l'IGN, ou bien que mon géoserver était hébergé en local sur ma machine, ou enfin un problème lié à la syntaxe de la requête WFS. Enfin, le fichier de paramétrage a été trouvé, et avec l'aide de mon nouveau compte admin, modifié pour autoriser ces requêtes.

II. Développement de l'interface

Comme expliqué précédemment, la plateforme ACTIVmap est centrée autour d'une carte interactive. Je présenterai dans cette partie les fonctionnalités que j'ai développées au cours du stage, et celles qui sont toujours en cours de développement. Je ferai référence directement aux fonctions du code en annexe, écrit en HTML JavaScript et CSS et utilisant surtout la bibliothèque OpenLayers 6. Mon interface finale propose un fond de carte provenant de données OSM, avec des fonctionnalités permettant de charger des couches depuis la BDD cartographique. Une première fonction interactive permet d'afficher les informations d'un objet de la carte lorsqu'on clique dessus, et une deuxième fonction, toujours en cours de développement, permettra d'ajouter et de supprimer un nouvel objet de façon permanente (écriture sur la BDD).

a) Structure de la page

Le fichier *index.html* (Annexe 1), inclut dans les annexes, décrit la structure de la page. L'interface contient actuellement un volet d'outils en haut de la page, ce sont les éléments de la classe « *topnav* » : les boutons permettant d'afficher les différentes couches, et les boutons permettant de modifier la carte en rajoutant des objets à ces couches. La classe « *map* » correspond à la carte centrale sur la page, affichant les différentes couches. Enfin, une barre inférieure (la classe « *bottom-text* » affiche des informations, ici la position en coordonnées du curseur de l'utilisateur « *mouse-position* » et des informations sur les couches sélectionnées « *info* ».

La figure ci-dessous montre une vue de la page de l'interface zoomé autour de la place Delille, avec les différentes divisions de page présentées ci-dessus. La carte contient actuellement 3 couches de données, points, lignes et polygones, chargés respectivement en bleu, en rouge et en vert sur la carte.

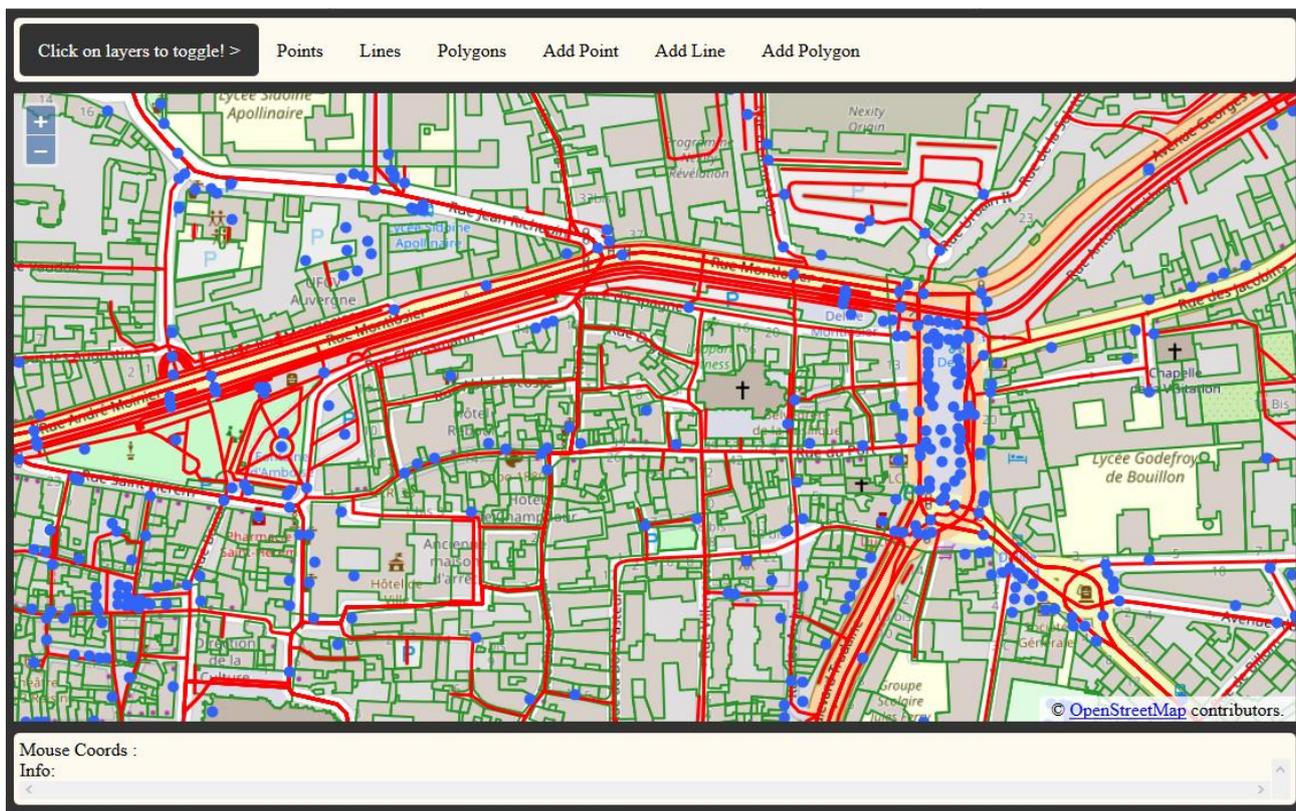


Figure 4 : prototype de l'interface Web ACTIVmap. Carte orientée Nord.

Le choix de n'afficher que 3 couches a été fait pour simplifier et accélérer le premier prototype, et pouvoir se concentrer sur d'autres blocages, plutôt que de rajouter tous les objets de la BDD cartographique, dont je détaillerai l'architecture en sous-partie III.

Les prochaines sous-parties b) et c) détailleront le fichier **main.js** ([Annexe 2](#)), qui contient le cœur du code en lui-même. Le fichier est séparé en plusieurs parties par des lignes commentées, « *layers* » et « *functions* », décrites respectivement dans b) et c). Je passerai aussi l'explication de **style.css** ([Annexe 3](#)), qui ne contient que des choix cosmétiques, qui ne seront pas maintenus sur la suite du projet.

b) Instanciation des couches (« layers »)

Pour la bibliothèque Openlayers, une carte est un conteneur, auquel on associe une liste de couches à afficher et une vue (l'emprise de départ). On peut aussi définir une liste d'interactions et de fonctions prédéfinies. **Les lignes 3 à 127** du fichier **main.js** ([Annexe 2](#)) comprend donc l'instanciation ma carte, et de tous ces éléments : je les décrirai ici.

Le fond de carte de base provient d'OpenStreetMap, et est rajoutée sur notre page par une simple requête WMS, c'est la couche « *raster* » instanciée à **la ligne 6**. La bibliothèque OpenLayers propose une fonction pour charger directement un fond OSM, c'est *ol.source.OSM()*. L'inclusion de ce fond de carte raster sert à contextualiser visuellement les différentes couches vecteur chargées dessus. Même s'il n'a pas vocation à rester sur l'interface, puisqu'on veut

Pour charger des couches depuis Géoserver, on différencie deux types de requêtes : WMS ou WFS. D'abord, les requêtes WMS : on charge une 'image' de la couche, ce qui est plus léger, mais ne permet pas d'accéder directement à ses propriétés, ni de les modifier. **Les lignes 13 à 23** donnent un exemple d'instanciation de couche WMS, dont je me suis servi en début de projet.

La deuxième possibilité pour charger une couche est une requête WFS, qui permet de charger aussi tous ses attributs et de les modifier. **Les lignes 26 à 73**instancient les trois couches simples visibles sur la page de l'interface : « *points* », « *lines* » et « *polygons* ». On rajoute de l'information vectorielle à notre carte, donc la syntaxe est différente, mais il y a plus d'options possibles. La requête renvoie les données sous forme de GeoJSON. Les couches vectorielles doivent être instanciées avec un style graphique correspondant au type de données, suivant des conventions classiques de **l'OGC**.

Des **lignes 74 à 102**, j'instancie les objets de type Interaction, des fonctionnalités interactives d'Openlayers permettant notamment de sélectionner un objet d'une couche pour le modifier, ou bien rajouter un nouvel objet. Je me sers de ces objets plus tard, dans la fonction de rajout d'un point.

Enfin la carte est initialisée des **lignes 104 à 128**, et on lui passe en argument les couches, les interactions et la vue initiale décrites ci-dessus. **La ligne 117** précise la cible de l'objet carte ainsi créé dans *index.html*, c'est le cadre principal au centre de l'interface graphique.

c) Fonctionnalités (« *functions* »)

Les **lignes 129 à 260** décrivent trois fonctions développées pour l'interface graphique.

D'abord, la fonction plus importante de transaction WFS des **lignes 131 à 169**. Elle sert à rajouter interactivement un point qui sera enregistré dans la BDD permanente, et elle représente la base de l'outil ACTIVmap. Pour ceci, il faut rédiger une requête WFS spéciale, et poster la requête vers le géoserver en précisant les propriétés du nouveau point.

La ligne la plus importante de la requête est la **ligne 146**, à l'appel de la fonction *WriteTransaction*. Les 3 premiers arguments de la fonction indiquent si l'on rajoute, supprime, ou modifie un point. Le dernier argument contient les propriétés du répertoire où ajouter le point, c'est la variable *formatGML* définie plus haut. Le reste des lignes **132 à 153** servent à définir la syntaxe de notre requête, en précisant le type de géométrie de notre point, ou formattant la transaction pour la prochaine étape. Enfin, cette requête est postée vers le serveur par une requête ajax, des lignes **154 à 165**.

Les **lignes 170 à 190** font aussi partie de cette requête WFS, mais plus indirectement. Elles utilisent les objets *Interaction* d'Openlayers pour rajouter un point à la carte, en définissant ses caractéristiques, ses géométries, et la couche où le rajouter.

Cette requête WFS-T ne fonctionne pas totalement. Ce code rajoute bien un point à la carte, et la requête ne renvoie aucune erreur, mais le point n'est pas rajouté à la couche correspondante de géoserver, ni dans le tableau correspondant de la BDD. J'explore actuellement les sources d'erreurs qui causent ce dysfonctionnement, qui est sûrement dû à la complexité de la BDD dans laquelle je veux stocker les modifications.

J'ai d'ailleurs inclus en annexe le fichier *transaction_test.xml* ([Annexe 4](#)), qui est un autre essai d'écriture de transaction WFS. L'interface de géoserver contient une zone de « démos », avec des requêtes-test dont j'ai pu me servir pour comprendre la syntaxe à utiliser. J'inclus ce fichier pour montrer ma progression actuelle sur la syntaxe de la requête.

Il reste deux autres fonctions à mon code. D'abord, une fonction permettant de rajouter ou d'enlever couches à la carte (*toggle-layer*). Des **lignes 198 à 231**, des fonctions très simples attendent un clic sur un des boutons de la bannière supérieure de l'interface, et rajoutent ou enlèvent la couche correspondante, selon sa présence ou non sur la carte. Cette fonction n'est pas du tout optimisée, et a surtout servi à faciliter les phases de débogage pour vérifier que les trois couches s'affichaient bien, mais sans avoir à toujours afficher tous les détails.

Enfin, la dernière fonction des lignes **233 à 256** sert à requêter et afficher les propriétés d'une couche vecteur chargée par requête WMS. Vu que les propriétés de ces couches ne sont pas accessibles, j'ai développé cette fonction aux débuts du projet, avant de basculer vers le WFS. J'ai tout de même choisi de la garder dans mon code, car elle pourrait s'avérer utile à l'avenir.

Je conclurai cette partie en notant que si ce code est peut-être un peu simple, c'est que sa rédaction n'était pas le seul objectif de mon stage. Il fallait aussi passer du temps sur la conception de l'architecture de la base, et du schéma de fonctionnement globale de la plateforme. C'est pourquoi j'ai passé du temps à résoudre les problèmes liés aux requêtes WFS ou à la connexion au géoserver, plutôt que d'améliorer le rendu visuel ou les fonctions secondaires de ma carte.

III. Développement de la BDD cartographique

Un deuxième objectif de mon stage était de reconstruire l'architecture base de données cartographique destinée à fournir le géoserver de la plateforme. Comme mentionné précédemment, les données de base proviennent du fond cartographique d'OpenStreetMap, à cause de leur liberté d'accès et de l'inclusion d'objets particulièrement pertinents pour ACTIVmap, comme les passages piétons et les équipements routiers. Mais si certains objets OSM nous intéressent, beaucoup des détails du fond de carte OSM ne sont pas utiles pour nos besoins. Une carte tactile doit être épurée de détails pour être utilisable, et il y a une limite au nombre de symbolologie différentes qui peuvent être utilisées sur un support tactile sans causer de confusion.

Mais avant de comprendre comment reconstruire l'architecture de données, je devais commencer par charger un fond de carte et analyser les données de base. Un autre membre du projet avait déjà importé les couches OSM de toute la France dans une première version de la BDD ACTIVmap, je pouvais donc lui en demander des extraits pour les intégrer dans ma propre base PostGIS locale.

Les données utilisées pour la construction du prototype étaient donc un extrait de quelques km² autour du vieux centre de Clermont-Ferrand (*fig. 5 ci-dessous*). Cet extrait de la BDD ACTIVmap est d'environ 3500m par 3500m, mais sa position et ses dimensions exactes ne sont pas très importantes. Le plus important pour moi était d'avoir accès à une variété des types d'objets de la BDD, sans avoir un volume de données trop important qui aurait ralenti le temps d'affichage et de calcul.

La classification initiale des données était plutôt complexe, séparée en 4 couches (points, lignes, routes, et polygones) chacune décrite par de nombreux champs d'attributs. On voit ici mon extrait de la couche 'lignes', et son emprise sur la ville de Clermont-Ferrand.

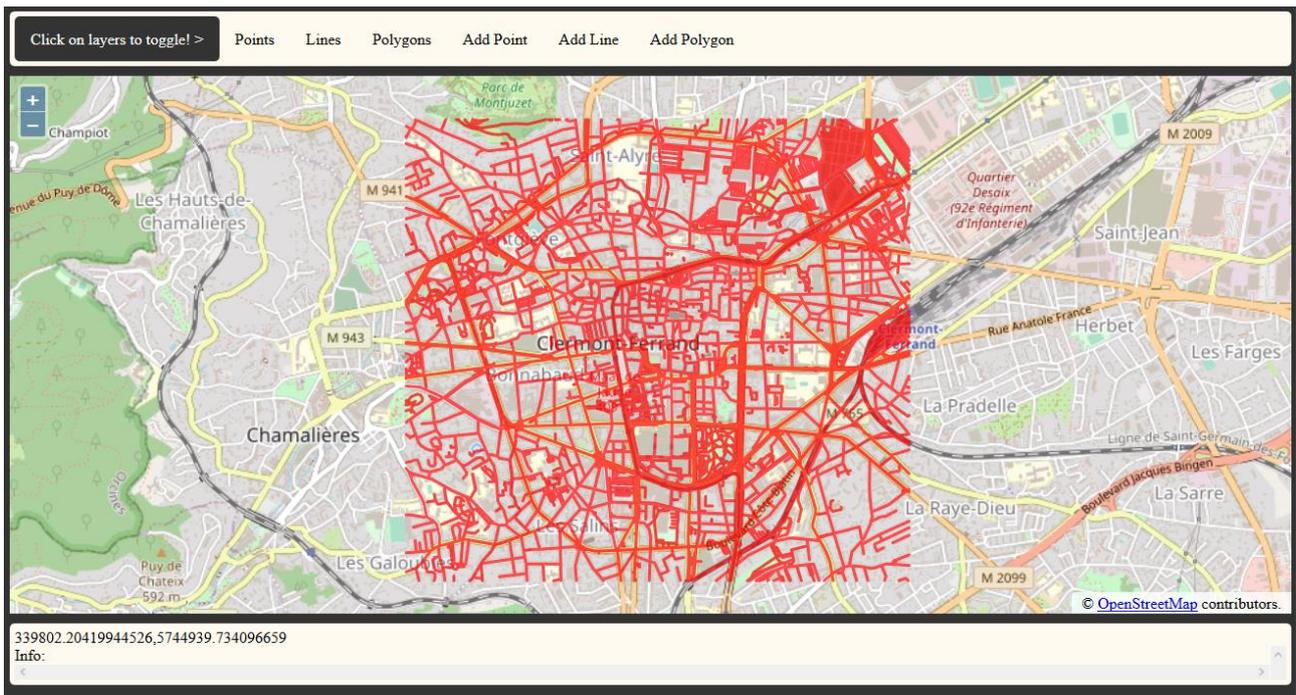


Figure 5 : extrait de la BDD ACTIVmap autour du vieux centre de Clermont-Ferrand. Carte orientée Nord.

Les données d'OSM sont récoltées et publiées collaborativement par ses utilisateurs, ce qui a des avantages et des défauts. Pour notre projet, les avantages étaient une grande granularité sémantique, donc beaucoup de détails qui nous intéressaient. Cependant il y avait logiquement aussi beaucoup d'objets qui ne nous intéressaient pas, ou alors qui étaient parfois mal classifiés, ou classifiés de manière hétérogène. J'entends par classification hétérogène des mêmes types d'objets décrits par des champs différents, selon les choix des utilisateurs qui les ont cartographiés. Au final, ces divergences sémantiques sont relativement rares et sont normales pour une plateforme collaborative. Un objet sur OSM peut être décrit avec des centaines de champs possibles, et il y a donc forcément des zones de superposition entre certains champs.

Mes données de base étaient séparées en environ 70 champs sémantiques. Pour comparer, le prototype de carte tactile vu précédemment dans ce rapport (fig.2) comporte moins d'une sixaine de types d'objets distinguables. Même si l'étendue de ces deux cartes ne sont pas les mêmes, la différence est claire. Ma nouvelle architecture de la base de données devait être bien plus simple et compacte.

J'ai commencé par analyser les 70 champs de mes données initiales. Je ne détaillerai pas dans ce rapport l'intégralité de ce processus, mais plutôt la logique que j'ai utilisée pour choisir les champs. La première chose à faire était d'identifier les champs les plus utiles pour notre carte tactile, et ceux dont on pouvait se passer. Heureusement, OpenStreetMap est très bien documentée, et j'ai eu accès à la description exhaustive³ de chaque champ possible de mes données, pour comprendre les propriétés de chaque point, ligne et polygone.

La figure suivante (fig.6) montre un extrait de cette première analyse des données. La colonne A renseigne le nom du champ, et la colonne C me servait à le décrire brièvement. J'ai ensuite regroupé les champs par groupe « thématique », visibles sur la figure par un champ de couleur et un titre de groupe en gras dans la colonne G.

	A	B	C	D	E	F	G	H	I
1	name	value count	description	value type (string)	DROP / fuse	new name			
2	fid	11071	table reference ID			drop			
3	osm_id	11071	osm reference ID	Long	fuse fid	drop	new ID. [no need for tags?]		
4	tags	11071			Yes	drop			
5	power	93	power terminal			keep			
6	highway	56	describes roads & footpaths			KEEP			
7	boundary	39	admin, protected, or other boundary			drop			
8	place	13	suburb, city [CAPITAL]			drop			
9	admin_level	7	admin boundary scale indicator			drop			
10	waterway	7	various waterways [river]			keep			
11	population	5	city population [CAPITAL]			drop			
12	military	1				drop			
13	tower:type	1				drop			
14	natural	17	tree, volcano, cave entrance			keep	street objects/features		
15	barrier	6	street objects			keep			
16	man_made	4	street objects			keep			
17	landuse	110				drop	LANDUSE properties		
18	wetland	0				drop			
19	water	1	water areas [pool]			drop			
20	public_transport	9	public transport features			keep	Public Transport		
21	railway	9	partly [public_transport]			keep			
22	route	0	bus routes			keep			
23	leisure	105	leisure & sports facilities			drop	(public) Activities/Usages	fuse these	
24	religion	25	place of worship			drop			
25	historic	20	historic places/landmarks			drop			
26	sport	20				drop			
27	denomination	18	place of worship faith/religious			drop			

Figure 6 : Classification thématique des champs OSM des données de base

³ <https://wiki.openstreetmap.org/wiki/Features> : Liste & détails des champs possibles sur OpenStreetmap

Regrouper grossièrement les champs de cette manière m'a d'abord permis de réduire le volume de données. J'ai pu ainsi réduire le nombre de champs de 70 à environ 10-20 par couche. Cette étape simplificatrice m'a permis de mieux comprendre les différents champs, pour l'étape suivante, qui était la nouvelle répartition des objets dans des nouvelles classes.

Pour cette nouvelle répartition, je devais donc sélectionner les objets qui nous intéressaient, les réorganiser, et supprimer le reste. A ce stade, je devais me poser beaucoup de questions sur ce qui constituait une information à garder ou à supprimer. L'interface finale devait être aussi épurée que possible, mais en même temps, il était souhaitable de garder le plus de détails possibles, quitte à les garder comme de la métadonnée accessible. Par exemple, on peut imaginer que les noms de bâtiments importants comme la mairie soient à omettre de la carte finale. Mais ces informations seront sans doute importantes lors de la phase de construction de la carte, pour que l'utilisateur d'ACTIVmap puisse se repérer. Ce type de données ne pouvait pas être effacé complètement de la base, et devait pouvoir être accessible d'une façon ou d'une autre.

Cependant, comme pour mon interface graphique, cette classification de la BDD n'avait pas pour but d'être parfaite, ni exhaustive, et j'ai donc choisi d'ignorer certains objets ou champs 'problématiques' dans ma classification. Mon objectif était de proposer une 'première ébauche' qui serait ensuite complétée et adaptée et enrichie au fil du projet ACTIVmap, et des retours des utilisateurs déficients visuels. Je n'avais simplement pas les moyens de savoir quels champs seraient utiles dans certaines situations, donc j'ai simplement fait au mieux. Je devais trouver un «juste milieu entre la carte surpeuplée initiale (*fig.1*) et la carte épurée finale (*fig 2*). Le schéma suivant (*fig.7*) montre le résultat de cette nouvelle répartition.

Ce schéma contient beaucoup d'informations. En haut à droite, on peut y trouver 10 des nouveaux champs (« *properties* ») créés lors de l'étape précédente (*fig.6*). La partie gauche détaille ma répartition des objets de la carte (« *map features* ») en 5 nouvelles classes (« *categories* »). Enfin, en bas à droite, la correspondance entre chacune des 5 classes (numérotées de 1 à 5) et les propriétés. Les propriétés notées à côté des chiffres correspondent aux propriétés déterminantes de cet objet. Ces propriétés 'déterminantes' permettent d'identifier le type des objets, et sera utile notamment pour l'écriture des requêtes SQL transposant cette architecture sur la BDD PostGIS.

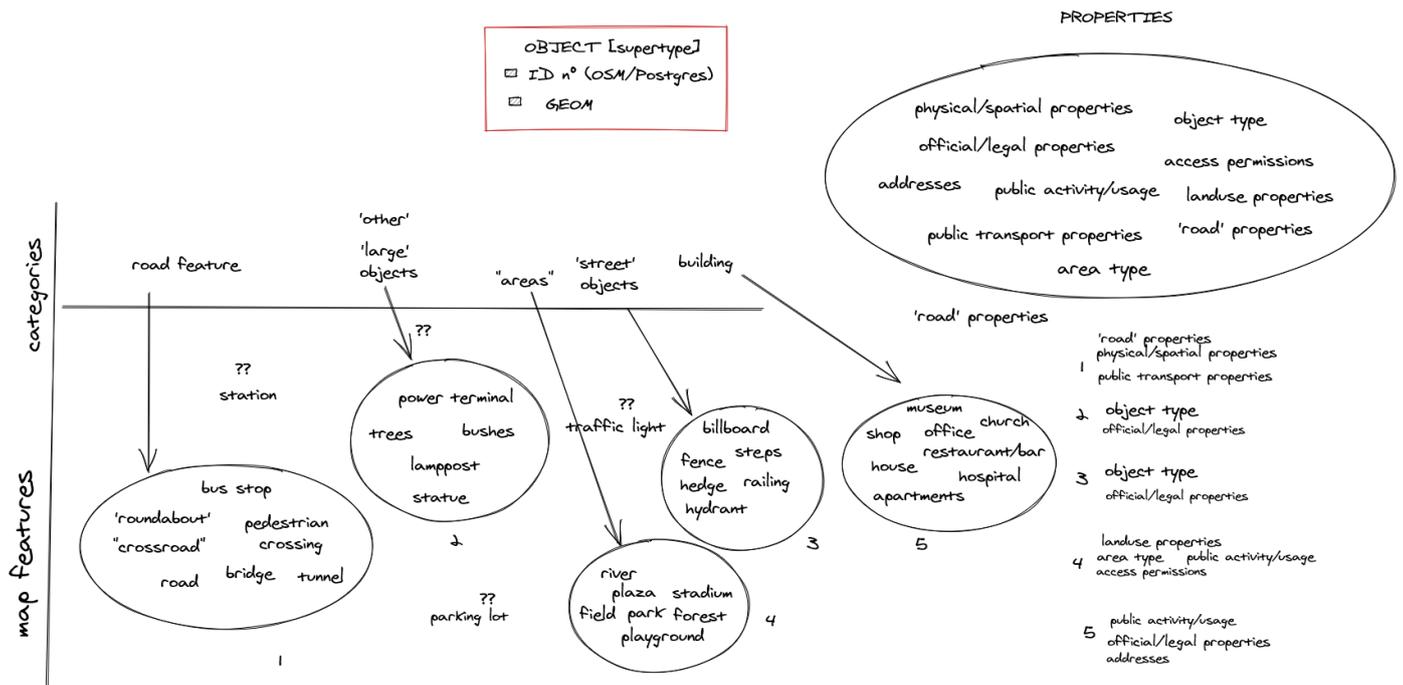


Figure 7 : schématique de la nouvelle architecture de la Base de Données ACTIVmap

La conception de cette architecture est donc complète, même si elle n'est pas parfaite. On peut remarquer quelques objets qui m'ont posé problème, se promenant entre les 'bulles' des catégories auxquelles ils pourraient potentiellement correspondre. Ces objets plus difficiles à catégoriser sont pour l'instant rangés dans la catégorie 'autres'.

Il restait maintenant à traduire toutes ces étapes en requête SQL. A l'heure actuelle, l'architecture entière n'a pas encore été traduite, il ne manque plus que la dernière étape. Les requêtes SQL en question sont attachées dans la dernière annexe de ce rapport (Annexe 5). Les 4 classes et 70 champs initiaux ont été réduits à 3 classes (par type de géométrie), avec environ 20 champs chacun. J'ai initialement gardé cette séparation par classe géométrique, PostGIS ne permettant pas différentes géométries dans une même table. La prochaine étape à faire pour cette BDD est donc d'implémenter la séparation en 5 catégories décrites en figure 7, chaque catégorie séparée en 3 sous-tables (selon les géométries ponctuelles, linéaires et surfaciques).

Ce travail sur la BDD s'est donc arrêté ici, et on y distingue actuellement les 3 catégories géométriques sur la capture d'écran de la page 1, l'interface de mon prototype d'ACTIVmap. Comme pour le reste de mes travaux, il reste du travail à faire, mais l'architecture de la nouvelle BDD est prête pour la suite.

IV. Le projet CartAGen

Pendant les deux dernières semaines de Juin, je n'ai plus consacré mon temps à la plateforme ACTIVmap mais, plutôt à un autre projet parallèle. Le but était de travailler avec Jakub Wabiński, un chercheur de l'université de l'Université Militaire de Technologie (WAT) à Varsovie, en Pologne, pour le former à l'utilisation de l'outil CartAGen. Bien entendu, c'était aussi l'occasion de me former sur cet outil, pour la suite du projet ACTIVmap.

Dans le cadre de sa thèse, Jakub travaille sur des questions de généralisation cartographique sur des cartes tactiles. Il s'intéresse en particulier à l'enrichissement et la démocratisation des cartes tactiles, souvent trop chères ou trop limitées. Ses objectifs s'alignent avec ceux d'ACTIVmap, et Jakub vise notamment à produire des cartes généralisées à petite échelle pour des écoliers déficients visuels. Travaillant donc sur des thématiques similaires, Guillaume Touya voulait le former à l'outil CartAGen, une plateforme Java libre qu'il a co-développée et qui propose une large variété de processus de généralisation cartographique.

Les objectifs de ce « mini-projet » étaient d'installer CartAGen sur l'ordinateur de Jakub et de lui permettre de tester différents algorithmes de généralisation sur ses propres données. Guillaume travaillant à distance, j'ai passé la semaine avec Jakub à l'IGN pour atteindre avec lui ces différents objectifs.

La première étape était de prendre en main CartAGen de mon côté avant que Jakub n'arrive en France. L'installation de la plateforme Java, nécessitant des manipulations précises, est certainement la partie qui a nécessité le plus de temps, même avec la supervision de Guillaume. Il fallait notamment télécharger et intégrer des certificats web, et les intégrer en ligne de commande, pour contourner les processus de sécurités de Maven bloquant certaines dépendances du projet.

Après quelques jours de préparation en amont, l'installation a pu se faire sur mon PC comme sur celui de Jakub, et nous avons pu consacrer plusieurs jours à tester différents algorithmes de généralisation. Ci-dessous (*fig. 8 et 9*) sont deux captures d'écran de l'interface CartAGen, et un exemple de simplification par lissage effectué sur des données shapefile à petite échelle de la Pologne fournies par Jakub. Ces données étaient relativement simples, et ont été choisies par

Jakub comme couches « classiques » à ajouter en fond pour des cartes à petite échelle : lignes administratives, cours d'eau importants, lacs, villes principales et objets thématiques.

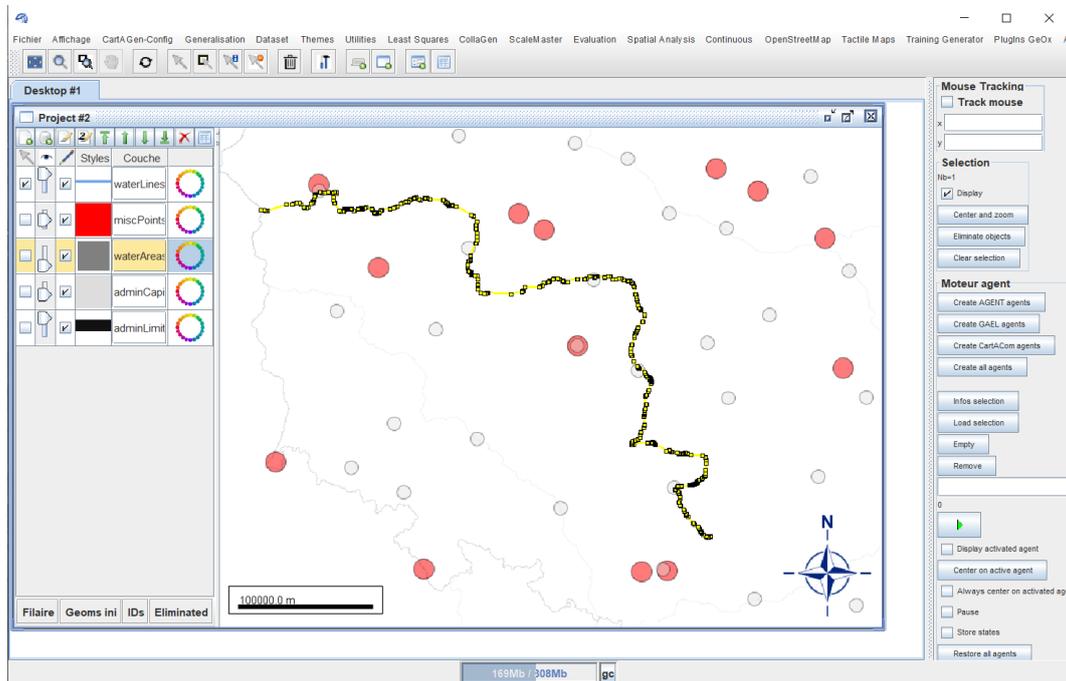


Figure 8 : vue de l'interface CartAGen, avec 5 couches chargées et un objet fleuve sélectionné (en jaune)

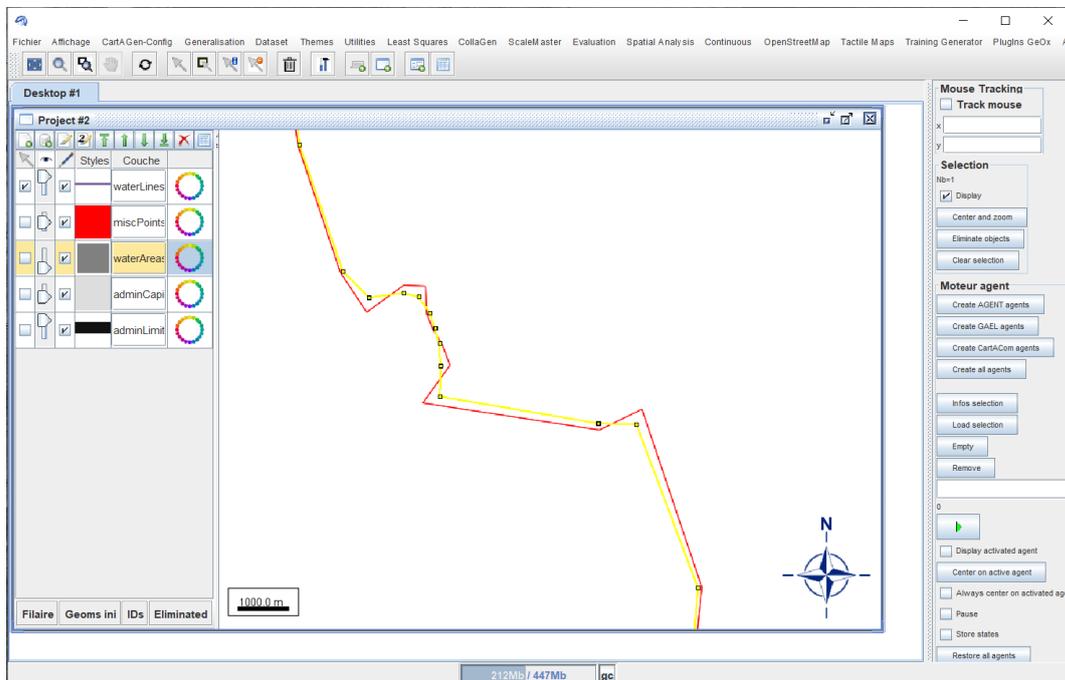


Figure 9 : Zoom sur un segment du fleuve traité par un algorithme de lissage (gaussian smoothing)

Ces captures d'écrans ne cherchent pas à montrer des résultats particulièrement intéressants en eux-mêmes, mais plutôt des exemples fonctionnels des capacités de la plateforme CartAGen. Le but étant plutôt d'explorer les algorithmes applicables et des variables paramétrables, cette semaine a été un franc succès, grâce à l'aide de Guillaume en distanciel et au temps consacré en amont à tester le guide d'installation de la plateforme.

En plus d'élargir ma culture scientifique et informatique, mes travaux avec Jakub s'inscrivent dans le projet ACTIVmap. Des algorithmes de généralisation cartographique seront en effet incorporés aux fonctionnalités de la plateforme au cours du projet. Jusque-là, c'était plutôt une question sur laquelle travaillaient d'autres étudiants doctorants du projet. C'est notamment par les scripts QGIS server que seront intégrés ces algorithmes. Si je continue à travailler sur ce projet, j'aurai donc l'occasion de travailler sur ce type de fonctions en plus de détails. Donc c'est d'autant plus que j'aurai appris de ce stage.

J'ai également apprécié l'occasion de pouvoir travailler en équipe plus directement avec quelqu'un sur un projet informatique, même sur une période courte d'une semaine. Je me suis rendu compte que je n'en avais pas eu l'occasion depuis presque deux ans, et j'ai pu à nouveau constater les avantages de coder à plusieurs, dans la même salle. Les problèmes rencontrés lors du développement ou de l'installation du logiciel étaient bien plus rapides à résoudre en s'y mettant à plusieurs, et j'ai apprécié cette semaine, d'une certaine façon plus « normale » que le reste du stage.

En conclusion, ce court projet parallèle d'environ 10 jours autour de la généralisation cartographique m'aura permis de travailler sur une partie d'ACTIVmap que je ne connaissais pas, et de découvrir l'interface CartAGen. J'ai pu en tirer des nouvelles connaissances utiles à la suite du projet ACTIVmap, découvrir une nouvelle plateforme Java, et enfin travailler et échanger avec un étudiant doctorant étranger (une occasion rare cette année !)

CONCLUSION

Voici donc les résultats de mes travaux depuis le 2 Février 2021. Le projet ACTIVmap m'intéresse depuis que j'en ai appris l'existence, surtout par sa dimension utile et sa portée humaine, son objectif de faciliter la vie d'une classe souvent défavorisée de la population. Je retracerai dans cette conclusion les grandes lignes des résultats à la fin du stage, ainsi que les perspectives d'avenir pour mon travail et ce projet.

Tout d'abord, le prototype de plateforme ACTIVmap est construit. J'ai réussi à mettre en place la structure principale, d'une page web codée avec OpenLayers, affichant des couches depuis un Géoserver, servi par une BDD cartographique locale sur PostGIS. J'ai aussi construit une première architecture de cette base de données, en partant du fond de carte OSM et restructurant les classes pour nos besoins. Je dirais donc que les fondations de base du projet ont été posées, c'est l'interface visible sur la figure 1 en page de couverture.

Les pistes d'amélioration sont d'abord le rajout de toute la panoplie d'outils qui rendront l'interface interactive. La première étape est de compléter la fonctionnalité de rajout d'un point par transaction WFS, qui n'est pas loin d'être finie. Le code de cette requête servira de guide pour développer les autres outils, et je continuerai à travailler sur cette requête WFS-transactionnelle sur les derniers jours qu'il me reste pour ce stage en espérant naturellement pouvoir la compléter pour la soutenance de ce rapport. Les étapes suivantes seront d'enrichir la plateforme, d'abord par l'intégration de l'architecture de la base de données, et des nouveaux types d'objet qui en découleront. Ensuite pourront être intégrés les scripts depuis QGISserveur, rajoutant des algorithmes plus complexes de traitement et de généralisation cartographique.

Je finirai par dire que suis satisfait de mon travail durant ces 5 mois. Si je n'arrive pas à résoudre ces problèmes d'ici la fin de mon stage, je suis confiant que les autres membres du projet pourront se servir des bases que j'ai posées avec mon travail pour le faire. Même si tout ne marche pas, le plus important reste d'avoir « débroussaillé » les choix techniques et l'architecture de la BDD. Les prochaines étapes de progression et d'amélioration du projet sont claires. J'ai justement pu rajouter mon code à un dépôt Github commun au projet. J'ai hâte de pouvoir continuer mon développement sur cette plateforme, ayant l'occasion de poursuivre mon travail avec l'IGN dès septembre. Le projet ACTIVmap m'a intéressé depuis le début, et c'est avec joie que je continuerai d'y investir de mon énergie !

BIBLIOGRAPHIE

Site officiel du projet ACTIVmap. Visité le 01/07/2021 <https://activmap.limos.fr/>

Présentation des contrastes entre Leaflet et Openlayers. Visité le 01/07/2021
<http://ivansanchez.github.io/leaflet-vs-openlayers-slides/#/>

Wiki des référence des champs de carte OpenStreetMap. Visité le 01/07/2021
https://wiki.openstreetmap.org/wiki/Map_features

Documentation Géoserver sur les requêtes WFS. Visité le 01/07/2021
<https://docs.geoserver.org/master/en/user/services/wfs/reference.html>

Documentation OpenLayers sur l'écriture de requêtes WFS. Visité le 01/07/2021
https://openlayers.org/en/latest/apidoc/module-ol_format_WFS-WFS.html

Page de forum détaillant l'écriture d'une transaction WFS. Visité le 01/07/2021
<https://gis.stackexchange.com/questions/214850/editing-wfs-layer>

ANNEXES

Annexe 1 : index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6.5.0/css/ol.css"
type="text/css">
    <link href="style.css" rel="stylesheet">
    <title>ACTIVmap</title>
  </head>

  <body>
    <div class="topnav">
      <a class="active" id="reset">Click on layers to toggle! > </a>
      <a id="points">Points</a>
      <a id="lines">Lines</a>
      <a id="polygons">Polygons (slow)</a>
      <a id="add_point">Add Point</a>
      <a id="add_line">Add Line</a>
      <a id="add_polygon">Add Polygon</a>
    </div>
    <div class="map" id="map" ></div>
    <div class="bottom-text">
      <div id="mouse-position"></div>
      <div id="info">Info:&nbsp;</div>
    </div>
    <!-- include link to OL library, JQuery library & js script -->
    <script
src="https://cdn.jsdelivr.net/gh/openlayers/openlayers.github.io@master/en/v6.5.0/build/ol.js"></script
>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
    <script src="main.js"></script>

  </body>
</html>
```

Annexe 2 : main .js

```
//////////////////////////////////// LAYERS //////////////////////////////////////

// base map layer
var raster = new ol.layer.Tile({
  source: new ol.source.OSM(),
});

//////////////////////////////////// WMS LAYERS //////////////////////////////////////

// geoserver WMS layer example
// var wms_src_lines = new ol.source.TileWMS({
//   url: 'http://localhost:8080/geoserver/AM_workspace/wms', // link to db [local geoserveur]
//   params: {'LAYERS': 'AM_workspace:lines', 'TILED': false},
//   serverType: 'geoserver',
// });

// var wms_lines = new ol.layer.Tile({
//   extent: [341520, 5742997, 345218, 5746411], // clermont ! check layer bounds in geoserver for
//   this.
//   source: wms_src_lines
// });

//////////////////////////////////// WFS LAYERS //////////////////////////////////////

///// POINTS /////
var wfs_src_points = new ol.source.Vector({
  format: new ol.format.GeoJSON({dataProjection: 'EPSG:3857'}),
  url:
'http://localhost:8080/geoserver/AM_workspace/ows?service=WFS&version=1.0.0&request=GetFeatur
e&typeName=AM_workspace%3Aam_points&maxFeatures=1000000&outputFormat=application%2Fj
son'
});

var wfs_points = new ol.layer.Vector({
  source: wfs_src_points,
  style: new ol.style.Style({
    image: new ol.style.Circle({
      radius: 5, // default radius is 0 so must define!!
      fill: new ol.style.Fill({color: 'rgba(50, 100, 250, 1.0)'} // red
    }),
  }),
});
```

```

/// LINES ////
var wfs_src_lines = new ol.source.Vector({
  format: new ol.format.GeoJSON(),
  url:
'http://localhost:8080/geoserver/AM_workspace/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=AM_workspace%3Aam_lines&maxFeatures=1000000&outputFormat=application%2Fjson'
});

var wfs_lines = new ol.layer.Vector({
  source: wfs_src_lines,
  style: new ol.style.Style({
    stroke: new ol.style.Stroke({
      color: 'rgba(255, 0, 0, 1.0)', // red
      width: 3,
    }),
  }),
});

///// POLYGON /////
var wfs_src_polygons = new ol.source.Vector({
  format: new ol.format.GeoJSON(),
  url:
'http://localhost:8080/geoserver/AM_workspace/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=AM_workspace%3Aam_polygon&maxFeatures=1000000&outputFormat=application%2Fjson'
});

var wfs_polygons = new ol.layer.Vector({
  source: wfs_src_polygons,
  style: new ol.style.Style({
    stroke: new ol.style.Stroke({color: 'rgba(40, 150, 40, 1.0)', width : 2}),
  }),
});

////////////////////// INTERACTIONS ////////////////////////

var interaction;

var interactionSelectPointerMove = new ol.interaction.Select({
  condition: ol.events.condition.pointerMove
});

var interactionSelect = new ol.interaction.Select({
  style: new ol.style.Style({
    stroke: new ol.style.Stroke({
      color: '#FF2828'
    })
  })
});

```

```

var interactionSnap = new ol.interaction.Snap({
  source: wfs_src_points,
});

//display mouse position over map.
var mousePositionControl = new ol.control.MousePosition({
  className: 'custom-mouse-position', // show mouse position
  target: document.getElementById('mouse-position'),
  undefinedHTML: 'Mouse Coords : ',
});

////////////////////////////////////// MAP STARTUP ////////////////////////////////////////

var map_view = new ol.View({
  center : [343434, 5744750], // [clermont]
  zoom : 16,
  // center: ol.proj.fromLonLat([3.0913,45.78063]), // [place delille]
  // zoom: 16,
});

var map = new ol.Map({
  controls: ol.control.defaults().extend([mousePositionControl]), // add mouse position
  target: 'map',
  layers : [raster, wfs_lines],
  interactions: [
    interactionSelectPointerMove,
    new ol.interaction.MouseWheelZoom(),
    new ol.interaction.DragPan()
  ],
  view : map_view,
});

```

```

////////////////////////////////////// FUNCTIONS ////////////////////////////////////////

```

```

////////////////////////////////////// WFS-T TRANSACTION ////////////////////////////////////////

```

```

// define
var formatWFS = new ol.format.WFS();

var formatGML = new ol.format.GML3({
  featureNS: 'AM_workspace',
  featureType: 'am_points',
  srsName: 'EPSG:3857'
});
var dirty = {};

```

```

var transactWFS = function (mode, f) {
  console.log('f:',f);
  console.log("start!");
  var node = formatWFS.writeTransaction([f],null,null,formatGML);
  var xs = new XMLSerializer();
  var payload = xs.serializeToString(node);
  console.log('payload',payload);
  var geo = f.getGeometry();
  var geometry = geo.getCoordinates();

  $.ajax({
    type: 'POST',
    url:'http://localhost:8080/geoserver/AM_workspace/ows',
    dataType: 'xml',
    processData: false,
    contentType: 'text/xml',
    data: payload,
    success: function(response){
      console.log('response : ',response);
    }
  }).done();
};

// on click button, call 'add' interaction & send feature info to the request

document.getElementById('add_point').addEventListener("click", function (){
  map.removeInteraction(interaction);

  interactionSelect.getFeatures().clear();
  map.removeInteraction(interactionSelect);
  interaction = new ol.interaction.Draw({
    type: 'Point',
    source: wfs_src_points,
    geometryName:'geom'
  });
  map.addInteraction(interaction);
  interaction.on('drawend', function (e) {
    var feature = e.feature;
    feature.set('geom', feature.getGeometry());
    transactWFS('insert', e.feature);
    console.log('insert');
  });
  map.addInteraction(interaction);
});

```

```

//////////////////////////////// toggle function [ to be improved ] //////////////////////////////////
toggle_points = 0;
toggle_lines = 1;
toggle_polygons = 0;

document.getElementById('points').addEventListener("click", function (){
    if (toggle_points == 1){
        map.removeLayer(wfs_points);
        toggle_points = 0;
    }else{
        map.addLayer(wfs_points);
        toggle_points = 1;
    }
});

document.getElementById('lines').addEventListener("click",function(){
    if (toggle_lines == 1){
        map.removeLayer(wfs_lines);
        toggle_lines = 0;
    }else{
        map.addLayer(wfs_lines);
        toggle_lines = 1;
    }
});

document.getElementById('polygons').addEventListener("click",function(){
    if (toggle_polygons == 1){
        map.removeLayer(wfs_polygons);
        toggle_polygons = 0;
    }else{
        map.addLayer(wfs_polygons);
        toggle_polygons = 1;
    }
});
//////////////////////////////// display features on click //////////////////////////////////

/*
// display feature info on click. [ WMS ]
map.on('singleclick', function (evt) {
    document.getElementById('info').innerHTML = "";
    var viewResolution = map_view.getResolution();
    var url = imgsrc_points.getFeatureInfoUrl( // select WMS layer source to fetch info
        evt.coordinate, viewResolution, 'EPSG:3857',
        {'INFO_FORMAT': 'text/html'}
    );
    if (url) {
        console.log("fetching..");
        fetch(url)
    }
});

```

```
.then(function (response) { return response.text(); })
.then(function (html) {
  document.getElementById('info').innerHTML = ('Info :'+html);
  //console.log('html:',html);
});
//
}
});
*/
```

Annexe 3 : style.css

```
html{
  background-color: #333;}

.map {
  width: 100%;
  height: 550px;
  border-radius: 5px;
}

#info{
  overflow: scroll;
}

/* Add a black background color to the top navigation */
.topnav {
  background-color: floralwhite;
  overflow: hidden;
  padding : 5px;
  border-radius: 5px;
  margin-bottom: 10px;
  color: black;
}

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color: #black;
  text-align: center;
  padding: 14px 16px;
  border-radius: 5px;
}

/* Change the color of links on hover */
.topnav a:hover {
  background-color: #333;
  color: white;
}

/* Add a color to the active/current link */
.topnav a.active {
  background-color: #333;
  color: white;
}
```

```
.bottom-text{  
  background-color: floralwhite;  
  overflow: hidden;  
  padding : 5px;  
  border-radius: 5px;  
  margin-top: 10px;  
}
```

Annexe 4 : transaction_test.xml

```
<wfs:Transaction service = "WFS" version="1.0.0"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wfs="http://www.opengis.net/wfs/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:ActivMap="http://ActivMap"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
numberMatched="unknown" numberReturned="0" timeStamp="2021-06-24T09:50:55.125Z"
xsi:schemaLocation= "http://www.opengis.net/wfs/2.0
http://localhost:8080/geoserver/schemas/wfs/2.0/wfs.xsd http://ActivMap
http://localhost:8080/geoserver/AM_workspace/wfs?service=WFS&version=1.0.0&request=
DescribeFeatureType&typeName=AM_workspace:am_points http://www.opengis.net/gml/3.2
http://localhost:8080/geoserver/schemas/gml/3.2.1/gml.xsd">
```

```
    <wfs:Insert>
      <ActivMap:am_points xmlns="http://localhost:8080/geoserver/web/AM_workspace">
        <ActivMap:geom>
          <ActivMap:Point xmlns="http://www.opengis.net/gml"
srsName="EPSG:3857">
              <gml:pos srsDimension="2">
                342508.39536060917 5744913.384133379
              </gml:pos>
            </ActivMap:Point>
          </ActivMap:geom>
          <ActivMap:GID>99999</ActivMap:GID>
          <ActivMap:amenity>test</ActivMap:amenity>
          <ActivMap:barrier>test</ActivMap:barrier>
          <ActivMap:bridge>test</ActivMap:bridge>
          <ActivMap:boundary>test</ActivMap:boundary>
          <ActivMap:covered>test</ActivMap:covered>
          <ActivMap:cutting>test</ActivMap:cutting>
          <ActivMap:highway>test</ActivMap:highway>
            <ActivMap:junction>test</ActivMap:junction>
          <ActivMap:layer>test</ActivMap:layer>
          <ActivMap:man_made>test</ActivMap:man_made>
          <ActivMap:natural>test</ActivMap:natural>
          <ActivMap:oneway>test</ActivMap:oneway>
          <ActivMap:power>test</ActivMap:power>
          <ActivMap:public_tra>test</ActivMap:public_tra>
          <ActivMap:railway>test</ActivMap:railway>
          <ActivMap:route>test</ActivMap:route>
          <ActivMap:surface>test</ActivMap:surface>
          <ActivMap:tunnel>test</ActivMap:tunnel>
          <ActivMap:waterway>test</ActivMap:waterway>
          <ActivMap:wood>test</ActivMap:wood>
```

```
<ActivMap:z_order>0</ActivMap:z_order>
  </ActivMap:am_points>
</wfs:Insert>
</wfs:Transaction>
```

// parameters used for geoserver demo :

// URL : <http://localhost:8080/geoserver/wfs>

// schemalocation : "http://www.opengis.net/wfs http://schemas.opengis.net/wfs/1.0.0/WFS-transaction.xsd http://www.opengis.net/wfs http://localhost:8080/geoserver/AM_workspace/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=AM_workspace%3Aam_points&maxFeatures=1000000"

Annexe 5 : requêtes SQL de l'architecture de la BDD

```
/* ***** CREATE TABLES *****  
  
CREATE TABLE public.am_points AS  
TABLE public.points;  
CREATE TABLE public.am_lines AS  
TABLE public.lines;  
CREATE TABLE public.am_polygon AS  
TABLE public.polygon;  
  
DROP TABLE public.adressless_points;  
CREATE TABLE public.adressless_points AS  
SELECT *  
FROM public.points  
WHERE (addr_house IS NULL) and  
      (addr_hou_1 IS NULL) and  
      (addr_inter IS NULL);  
*/  
  
/* ***** DROP ALL UNWANTED COLUMNS *****  
*/  
ALTER TABLE public.am_points  
  
DROP COLUMN OSM_ID , DROP COLUMN ACCESS, DROP COLUMN ADDR_HOUSE,  
DROP COLUMN ADDR_HOU_1, DROP COLUMN ADDR_INTER, DROP COLUMN  
ADMIN_LEVE, DROP COLUMN AERIALWAY, DROP COLUMN AEROWAY, DROP COLUMN  
AREA,  
DROP COLUMN BICYCLE, DROP COLUMN BRAND, DROP COLUMN BUILDING,  
DROP COLUMN CONSTRUCTI, DROP COLUMN CULVERT,  
DROP COLUMN DENOMINATI, DROP COLUMN DISUSED, DROP COLUMN  
EMBANKMENT, DROP COLUMN FOOT, DROP COLUMN GENERATOR_,  
  
DROP COLUMN HARBOUR, DROP COLUMN HISTORIC, DROP COLUMN HORSE,  
DROP COLUMN INTERMITTE, DROP COLUMN LANDUSE,  
DROP COLUMN LEISURE, DROP COLUMN LOCK, DROP COLUMN MILITARY,  
DROP COLUMN MOTORCAR, DROP COLUMN NAME,  
DROP COLUMN OFFICE, DROP COLUMN OPERATOR, DROP COLUMN PLACE,  
DROP COLUMN POPULATION, DROP COLUMN POWER_SOUR,  
DROP COLUMN REF, DROP COLUMN RELIGION, DROP COLUMN SERVICE,  
DROP COLUMN SHOP, DROP COLUMN SPORT,  
DROP COLUMN TOLL, DROP COLUMN TOURISM, DROP COLUMN TOWER_TYPE,  
DROP COLUMN WATER,  
DROP COLUMN WIDTH, DROP COLUMN WETLAND, DROP COLUMN TAGS;
```

```

/*
*/

/* NON-UNIVERSAL COLUMNS
ALTER TABLE public.am_points      DROP COLUMN CAPITAL,      DROP COLUMN ELE;
ALTER TABLE public.am_lines DROP COLUMN TRACKTYPE, DROP COLUMN WAY_AREA;
ALTER TABLE public.am_polygon      DROP COLUMN TRACKTYPE, DROP COLUMN
WAY_AREA;
*/

/* COLUMNS TO ADD/DROP :
tables to add
GEOM,AMENITY,BARRIER,BRIDGE,COVERED,CUTTING,HIGHWAY,JUNCTION,LAYER,MAN_MADE,NATURAL,ONEWAY,POWER,PUBLIC_TRA,RAILWAY,ROUTE,SURFACE,
TUNNEL,WATERWAY, WOOD, Z_ORDER

tables to drop
FID,OSM_ID,ACCESS,ADDR_HOUSE,ADDR_HOU_1,ADDR_INTER,ADMIN_LEVE,AERIALWAY,AE
ROWAY,AREA,BICYCLE,BRAND,BUILDING,CAPITAL,CONSTRUCTI,CULVERT,DENOMINATI,DIS
USED,ELE,EMBANKMENT,FOOT,GENERATOR_,HARBOUR,HISTORIC,HORSE,INTERMITTE,LAN
DUSE,LEISURE,LOCK,MILITARY,MOTORCAR,NAME,OFFICE,OPERATOR,PLACE,POPULATION,
POWER_SOUR,REF,RELIGION,SERVICE,SHOP,SPORT,TOLL,TOURISM,TOWER_TYPE,TRACKT
YPE,WATER,WIDTH,WETLAND,WAY_AREA,TAGS
*/

/* ***** REMOVE ALL EMPTY POINTS ***** */
/*
-- deleting empty rows for points
SELECT * FROM public.am_polygon
/*DELETE FROM am_lines */
WHERE AMENITY is null
and BARRIER is null
and BRIDGE is null
and COVERED is null
and CUTTING is null
and HIGHWAY is null
and JUNCTION is null
and LAYER is null
and MAN_MADE is null
and am_polygon.NATURAL is null -- NATURAL is keyword so specify table here
and ONEWAY is null
and am_polygon.POWER is null -- same for POWER
and PUBLIC_TRA is null
and RAILWAY is null
and ROUTE is null
and SURFACE is null
and TUNNEL is null
and WATERWAY is null
and WOOD is null
and Z_ORDER is null ;*/

```

Glossaire et sigles utiles

IGN	Institut National de l'Information Géographique et Forestière
LaSTiG	Laboratoire de Sciences et Technologies de l'Information Géographique pour la Ville et les territoires Numériques
GéoVIS	Visualisation, Interaction et Immersion
ACTIVmap	<i>Assistance à la Conception des carTes pour déflcients Visuels</i>
LIMOS	Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
BDD	Base De Données
OSM	OpenStreetMap
SIG	Système d'Information Géographique (QGIS, ArcGIS...)
CORS	Cross-Origin Request Sharing
OGC	Open Geospatial Consortium

Liste des annexes