
RAPPORT D'ANALYSE

PROJET DE DÉVELOPPEMENT INFORMATIQUE



Cartographie semi-automatique d'itinéraires de randonnée

Réalisé par Correia Clara, Esteoulle Alexandre, Peseux Lorine, Zobel Thibaud

Commandité par Duchêne Cécile et Van-Damme Marie-Dominique

12 Mai 2024

Table des matières

1	Contexte	3
1.1	Introduction	3
1.2	Aspects sociaux-financiers	3
1.3	Glossaire	4
2	Objectifs de l'étude - Reformulation du besoin	4
2.1	Objectifs de l'étude	4
2.2	Contraintes	5
2.3	Le recueil du besoin - les Acteurs	5
3	Analyse Fonctionnelle	6
3.1	Reformulation des fonctionnalités	6
3.2	Méthode envisagée/ réalisée	7
4	Etude Technique	8
4.1	Gestion du décalage	10
4.1.1	Par la Symbologie	10
4.1.2	Par la géométrie	10
4.2	Gestion de l'agencement manuel	11
4.2.1	Selection de l'objet	11
4.2.2	Inversions des itinéraires	11
4.3	Le Plugin	11
5	Suivi du Projet	11
5.1	Analyse des risques	11
5.2	Organisation	11
5.3	Logos	14
6	Conclusion	15
6.1	Améliorations	15
7	Annexes	16
7.1	Outils "décalage" QGIS	16
7.2	Ancien Diagramme de Classe	17
7.3	Schéma explicatif des sens des itinéraires lors du décalage	18

1 Contexte

1.1 Introduction

Notre projet de développement informatique porte sur la création d'un plugin QGIS. Il est commandité par Cécile Duchêne et Marie-Dominique Van Damme, toutes deux chercheuses au LASTIG (Laboratoire en sciences et technologies de l'information géographique), et par extension, par l'IGN et l'ENSG.

Nos commanditaires souhaitent que nous développions un nouvel outil technologique interactif sous la forme d'un plugin à intégrer dans QGIS, afin d'ajouter des fonctionnalités innovantes à QGIS. Les deux fonctionnalités principales demandées sont les suivantes : la représentation d'itinéraires de randonnée les uns à côté des autres, avec un affichage clair et lisible à la manière d'une carte d'itinéraire de bus, et l'interactivité du plugin afin qu'il soit possible de gérer l'affichage en changeant l'agencement des portions d'itinéraires.

L'enjeu derrière la demande de nos commanditaires réside principalement dans l'amélioration de la lisibilité et de la clarté des cartes de randonnées.



FIGURE 1 – Carte montrant les itinéraires touristiques pour les motos (CRDT Auvergne / Actuel)

Le cadre de notre projet s'étend cependant à une plus grande échelle, puisqu'il est voulu comme une solution destinée à un public large tel que les communautés de communes voulant mettre en avant des randonnées, ou bien les utilisateurs de Qgis tout simplement. Ainsi, notre plugin se doit d'être simple d'utilisation et interactif.

1.2 Aspects sociaux-financiers

Pour les aspects sociaux de notre projet, nous avons décidé de nous répartir les rôles de la manière suivante :

- Thibaud, chef de projet, pôle gestion interactivité
- Clara, responsable des rapports, pôle gestion interactivité
- Lorine, responsable des rendus graphiques, pôle placement automatique
- Alexandre, responsable QGIS, pôle placement automatique

À noter que nous avons formé des binômes afin de rentabiliser le temps en accomplissant plusieurs tâches simultanément. Cela inclut la rédaction de ce rapport en parallèle avec le développement de notre projet, ainsi que l'exploration des différentes pistes de développement envisagées tout au long de ce projet.

Nous sommes également en possession de quelques travaux effectués sur ce sujet avant nous, tel que l'article de recherche de Teulade-Denantes et al. [1].

Nous avons également pris en compte le rapport du projet de recherche portant sur les portions d'intérêt d'un itinéraire [2], à partir duquel nous avons extrait un jeu de données fourni pour un scénario d'itinéraire.

Nous avons décidé de le simplifier afin de faciliter nos tests de développement. L'aspect financier de notre projet est principalement représenté par un coût humain, avec notre équipe de quatre personnes travaillant environ 80 heures. Nous utilisons également du matériel équipé du logiciel QGIS et de Visual Studio Code pour la programmation.

1.3 Glossaire

- **Plugin QGIS** : module additionnel qui étend les fonctionnalités d'un logiciel (ici QGIS), extension qui permet d'ajouter des outils ou des fonctionnalités dans QGIS, adaptant ainsi le logiciel aux besoins particuliers des utilisateurs (sans modification du code source du logiciel)
- **Routes (Roads en anglais)** : ce sont des objets géographiques issu d'un réseau routier support (la BD TOPO de l'IGN par exemple) dont la géométrie est une polyligne. On nomme tronçon de route un segment de cette polyligne (cf figure 6).
- **Itinéraires (Routes en anglais)** : les itinéraires de randonnées sont les trajectoires possibles pour un randonneurs. Il peut y avoir plusieurs itinéraires sur une même route. Ces itinéraires sont constitués d'un ensemble de segments de routes.
- **Tronçons d'itinéraires** : Segment crée à chaque intersection. Une ensembles de tronçons d'itinéraire fournira un itinéraire.
- **RVA** : Nom de notre plugin Routes Visualization Arrangement (Visualisation et agencement d'itinéraire)
- **Niveaux** : Nom attribué à la représentation des différents emplacements des itinéraires autour d'une route représenté par le niveau 0 et ayant un sens positif et négatif.

2 Objectifs de l'étude - Reformulation du besoin

2.1 Objectifs de l'étude

Notre projet vise à offrir un nouvel outil de visualisation dans le logiciel QGIS, permettant à l'utilisateur de visualiser de manière plus fluide les différents itinéraires se chevauchant. Ces itinéraires seront fournis par celui utilisant le plugin. Il pourra ensuite les agencer à sa guise, grâce à une interaction graphique en cliquant avec la souris. On peut donc résumer ces objectifs en quelques points :

- Création d'un plugin visant à simplifier automatiquement la visualisation des itinéraires de randonnée.
- Possibilité de déplacer l'agencement des itinéraires grâce à une interaction entre le plugin et l'utilisateur.

Au final, nos tronçons d'itinéraire devraient ressembler à la configuration de la figure 2.

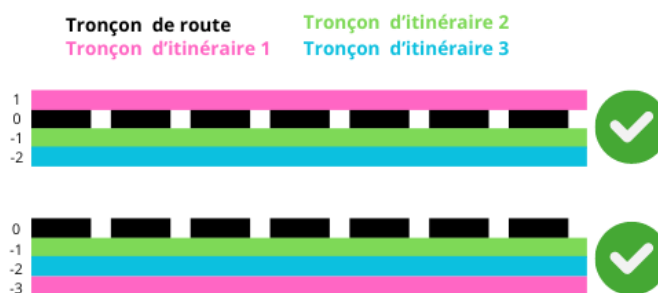


FIGURE 2 – Représentation des "niveaux" pour 3 itinéraires

Si ces objectifs sont atteints, la direction finale de notre projet serait la présentation officielle de ce nouvel outil aux développeurs de QGIS courant 2025.

2.2 Contraintes

Ce projet possède plusieurs contraintes :

- La contrainte majeure de ce projet réside dans l'échelle du temps. En effet, nous ne disposons que d'une dizaine de séances qui seront rythmées par des réunions avec les commanditaires et des séances de coaching. La date butoir pour le rendu des livrables de ce projet est fixée au 15 mai 2024.
- Parmi les livrables attendus, nous avons tout d'abord la première version de ce Rapport d'Analyse prévu pour le 11 mars, suivi de sa mise à jour pour le 15 mai. La soutenance aura lieu le 14 Mai. Nous devons également rendre le code commenté de notre plugin, accompagné de son ReadMe, ainsi que le produit informatique du projet, le tout pour le 15 mai. Cette date marquera également le jour du GéoDev où nous effectuerons une démonstration de nos résultats.
- Ce projet servant également de support pour les cours de gestion de projets, nous devons effectuer des présentations lors du Copil de lancement le 28 février et du Copil de clôture le 13 mai. Parmi les livrables attendus, il faudra fournir des comptes rendus des différentes réunions tenues avec nos commanditaires, ainsi que les supports actualisés à chaque séance de coaching.
- Devant implémenter un plugin QGIS, nous sommes également contraints d'utiliser un langage adapté à ce logiciel, soit du python orienté objet, avec l'API pyQGIS et PyQt pour la bibliothèque de visualisation dans QGIS .
- L'utilisateur devra savoir à quoi sert ce plugin, car il devra importer lui-même au moins deux couches shapefile : une couche qui représentera les tronçons de routes (et donc notre "état initial") qui serait obtenue grâce aux travaux issus du projet de recherches qui précède notre travail ([2]) et une couche qui représentera les itinéraires. L'utilisateur veillera à ce que la couche d'itinéraires soit parfaitement alignée avec la couche de routes. Les objets des deux couches devront posséder les mêmes coordonnées initiales, une contrainte que nous imposons.

2.3 Le recueil du besoin - les Acteurs

Les principaux acteurs de notre projet seront les utilisateurs du monde du SIG, et plus particulièrement ceux de QGIS, comme mentionné précédemment.

En effet, notre plugin a pour objectif d'être intuitif dans son utilisation. Cependant, il reste assez spécialisé en raison du cadre spécifique des itinéraires de randonnée. Le plug-in pourrait s'avérer utile pour les communautés de communes ou les espaces touristiques cherchant à promouvoir leur région pour une meilleure compréhension des trajets de randonnée, par exemple.

L'utilisateur devra savoir à quoi sert ce plugin, car il devra importer lui-même au moins deux couches shapefile : une couche qui représentera les routes (et donc notre "état initial") et une couche qui représentera les itinéraires. Ces itinéraires devront avoir les mêmes coordonnées que les routes par lesquelles ils passent (contraintes que nous imposons).

Voici un exemple de diagramme de cas d'utilisation (Figure 3) afin de mieux comprendre comment l'interaction entre l'utilisateur et le plugin fonctionnera :

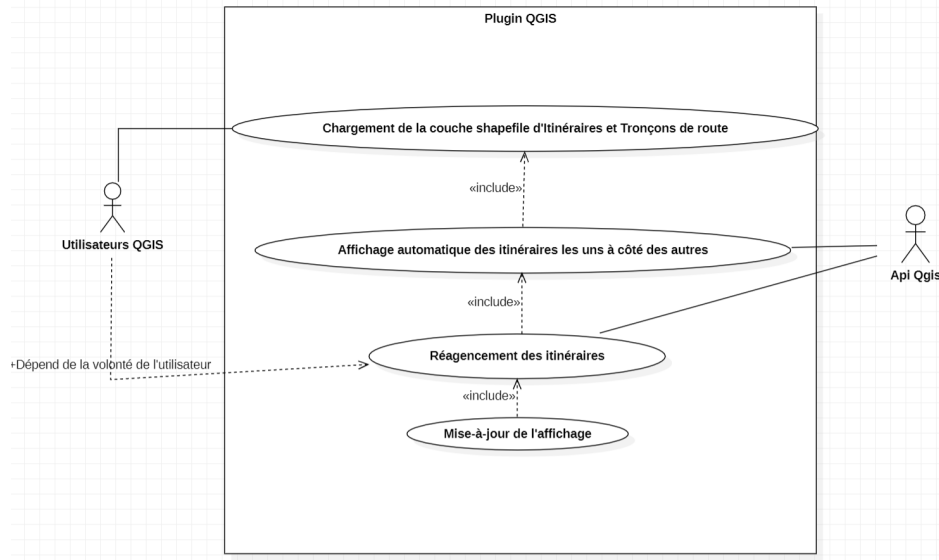


FIGURE 3 – Diagramme d'utilisation

3 Analyse Fonctionnelle

3.1 Reformulation des fonctionnalités

L'objectif principal pour les commanditaires étant d'obtenir un outil de génération de placement de ces itinéraires et de déplacement de leur position de manière semi-automatique. Les deux fonctionnalités principales pour notre plugin sont les suivantes :

- Importation de couches shapefiles (routes et itinéraires) avec génération d'itinéraires alignés sur les segments de routes (à la manière des plans de bus) dans un ordre aléatoire.
- Interactivité du modèle, offrant à l'utilisateur la possibilité de déplacer les itinéraires et de modifier leur ordre.

Voici un diagramme d'activité (figure 4) qui résume ces fonctionnalités principales :

Pour les fonctionnalités secondaires, nous avons, au début, prévu d'utiliser la notion de strokes (ensemble de tronçons d'itinéraire) expliqué à la page 60 de l'article [1], offrant ainsi la possibilité de déplacer ces strokes, plutôt que des tronçons d'itinéraires. Cependant nous avons juger cela comme chronophage et donc nous l'avons relayé en possible amélioration pour une future version de ce plugin.

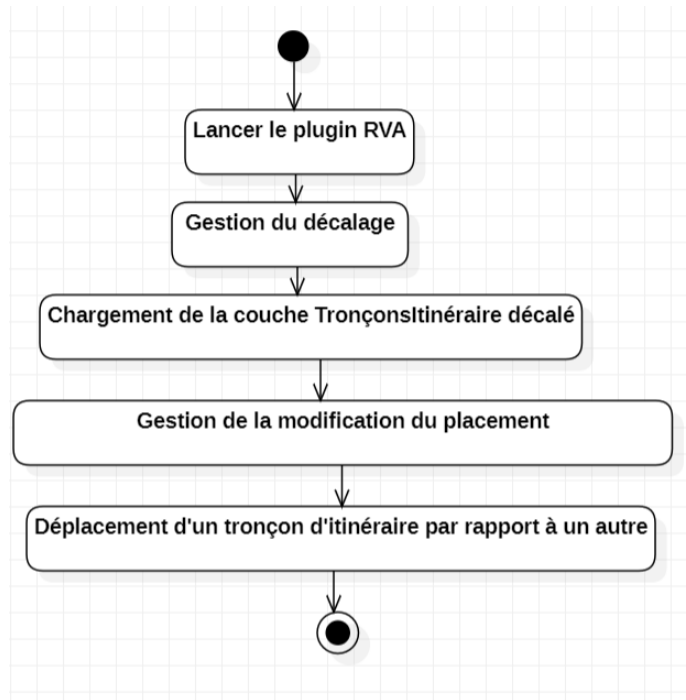


FIGURE 4 – Diagramme d'Activité

3.2 Méthode envisagée/ réalisée

En ce qui concerne la stratégie choisie, dans un premier temps il fallait que nos données fournies (la couche route et itinéraires) soit découpées en ce que l'on qualifie de tronçons (cf 1.3), puis nous partons de l'hypothèse que les tronçons d'itinéraires et les tronçons de routes se superposent dans les couches originelles, c'est une contrainte que l'on impose à notre utilisateur pour le bon fonctionnement du plugin. Ainsi, nous prévoyons l'affectation des tronçons d'itinéraires aux tronçons de route par le test 'géométrie de la route contient géométrie de l'itinéraire', (cela pourrait se traduire par une intersection entre ces deux couches). Sur la première version de ce rapport nous étions résolu à utiliser la notion de strokes et la majorité de nos recherches était alors orientées sur cet aspect qui à finalement été jugé trop secondaires vis-à-vis de la demande principale de nos commanditaires. De ce fait nous avons dû revoir notre analyse et donc notre approche. Ces changements ont impliqué de modifier les diagrammes UML proposés également. Voici le diagramme de Classe mis-à-jour (figure (5)) Nous avons décidé de conserver l'ancien mais sous forme d'annexe (7.2)

Pour plus de détails :

Nous avons tout d'abord créé nos deux classes principales du projet. D'une part la classe TronçonRoute qui possède un identifiant et une géométrie, auxquelles seront rattachés des tronçons d'itinéraires. Ces tronçons d'itinéraires auront les fonctionnalités qui seront nécessaires à notre projet car c'est sur ceux-ci que notre plugin travaillera. Notamment lors de la première étape du décalage ou il faudra qu'il utilise l'ensemble des tronçons d'itinéraire d'un même tronçon de route pour les décaler un à un. Il faudra également que les tronçons d'itinéraire décalés soient gérables par le clic afin de modifier leurs attributs pour que ceux-ci soit inversés.

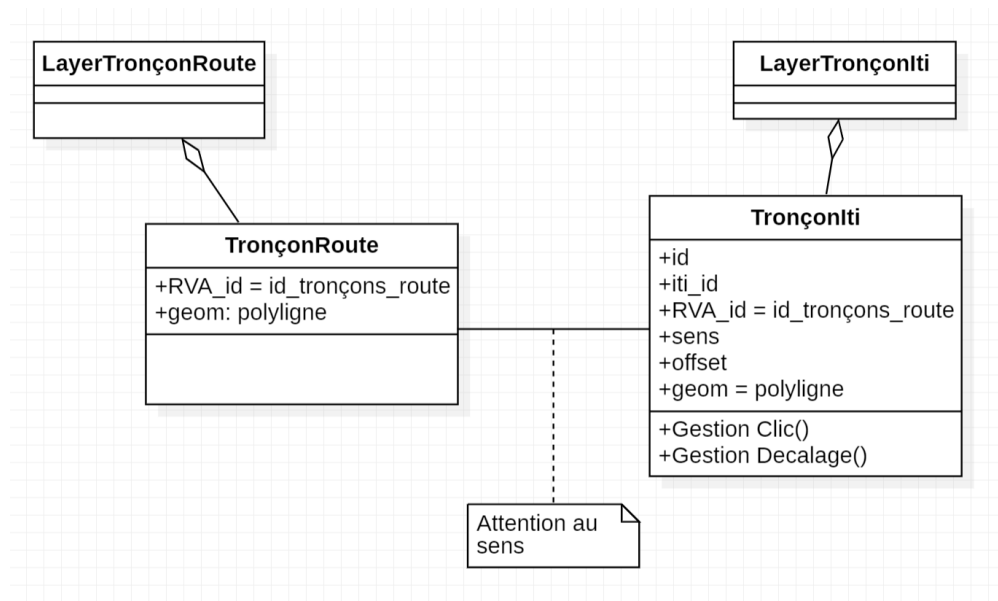


FIGURE 5 – Diagramme de Classes

Afin de mieux comprendre cette notion de tronçon, voici la figure6 d'un scénario simple :

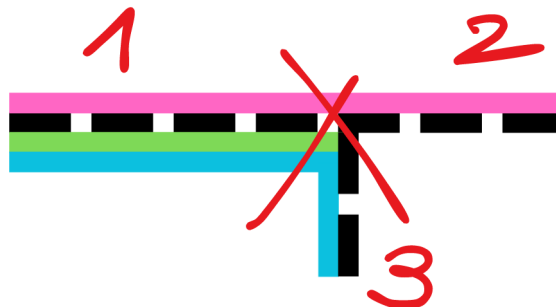


FIGURE 6 – Représentations de tronçons

Nous nous assurerons d'abord que la génération du décalage se passe correctement avec un décalage calculé avec un certain offset pour attribuer une place par niveau (Ce référencer au 1.3). Puis, le déplacement des itinéraires en fonction d'un clic contrôlé par l'utilisateur devra être bien fonctionnel.

La gestion de ce clic s'effectuera avec les coordonnées de la souris et une affectation au tronçon d'itinéraire le plus proche.

4 Etude Technique

Concernant l'étude technique de notre projet, qui consiste à développer un plugin pour QGIS, nous avons effectué une analyse approfondie des classes existantes afin d'accéder aux outils nécessaires, tels que ceux disponibles à travers la classe **QgisMapTool**, par exemple.

Nous avons également jugé nécessaire d'imposer des conditions concernant les données fournies par l'utili-

sateur. Comme spécifié dans les contraintes, il est impératif que les couches fournies correspondent respectivement à une couche de tronçons de routes et à une couche de tronçons d'itinéraires. Par ailleurs, nous avons élaboré un diagramme d'état qui reprend les principes du diagramme d'activité, en mettant davantage l'accent sur les aspects techniques. (figure 7) :

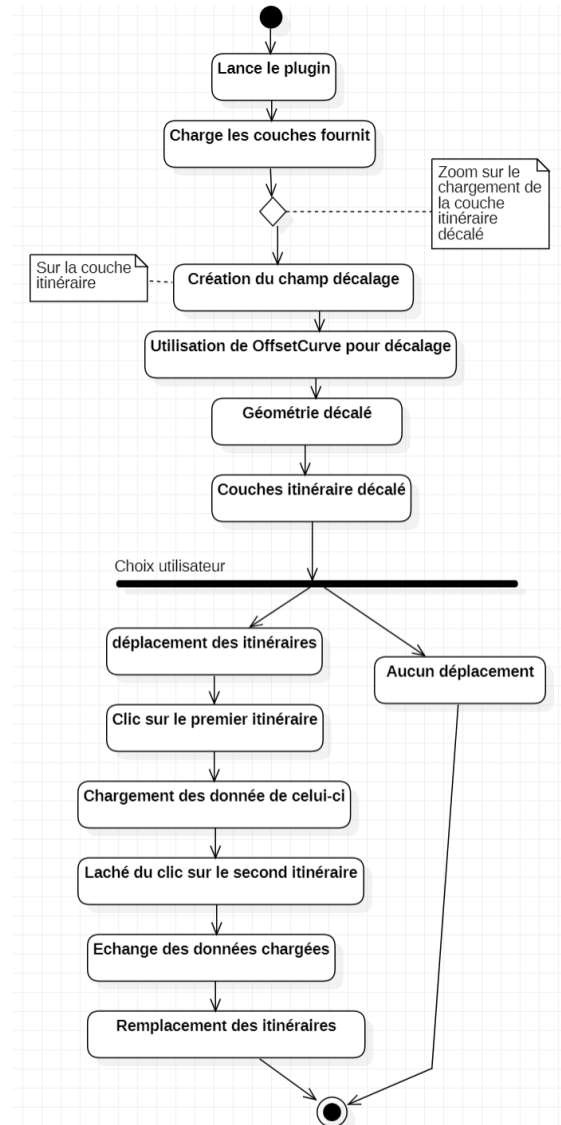


FIGURE 7 – Diagramme d'Etat

Nous avons dans un premier temps traité les données fournies par l'utilisateur avec la création de certains champs.

Puis nous avons fait en sorte de placer automatiquement les tronçons d'itinéraires les uns à la suite des autres.

4.1 Gestion du décalage

4.1.1 Par la Symbologie

Concernant la première partie de notre projet, nous avons dans un premier temps cherché si le décalage des itinéraires était possible avec les outils existants du logiciel Qgis. Après avoir examiné les modifications possibles pour le symbole de ligne, nous avons trouvé l'outil "décalage" dans l'onglet "Symbologie" d'une couche sur QGIS (Annexe 7.1).

Cet outil permet de décaler la symbologie tout en conservant la géométrie initiale. Pour nous assurer que le simple changement de décalage suffisait, nous l'avons d'abord testé manuellement. Ensuite, nous avons cherché à automatiser le même processus en écrivant un code.

Nous avons écrit un code qui décalait tous les itinéraires du même montant, et ce code a fonctionné. Ensuite, nous avons dû réfléchir à la manière de faire en sorte que le décalage soit différent pour chacun des itinéraires. Après analyse, nous avons compris que le décalage de chaque itinéraire devait être calculé en fonction de la largeur initiale de la route et de la largeur des itinéraires.

⇒ Nous avons décidé de faire partir du principe que tous les itinéraires avaient la même largeur.

Ainsi les décalages sont calculés de la manière suivante : soit "Lr" la largeur du trait de la route et "Li" la largeur des traits des itinéraires, sur un même tronçon de route, le premier tronçon d'itinéraire sera décalé de $(Lr+Li)/2$, le deuxième de $(Lr+2Li)/2$, le troisième de $(Lr+3Li)/2$ etc. Ainsi soit "x" le numéro de l'itinéraire (c'est-à-dire le niveau dans lequel l'itinéraire a été placé sur ce tronçon de route), les itinéraires seraient alors positionnés comme suit :

$$- (Lr+xLi)/2$$

Cela permet de déterminer le champ décalage sur lequel reposera le décalage des tronçons d'itinéraire.

Cependant, il y avait plusieurs problèmes avec cette méthode de décalage de la symbologie :

- Un problème était lié au style de symbole original de la couche d'itinéraires. Pour représenter de façon lisible les itinéraires il faut "catégoriser" la symbologie. Or, l'automatisation de ce procédé nous a posé des problèmes.
- Un autre problème était lié à la fonction *setOffset()* de Qgis qui semblait ne pas fonctionner.
- Le problème le plus important concernait la gestion des clics. Une fois le décalage par symbologie effectué, il était impossible de récupérer l'objet car seule sa symbologie avait changé, mais pas sa géométrie. Ce problème nous a donc poussé à abandonner la méthode basée sur la symbologie et à chercher à décaler la géométrie directement.

4.1.2 Par la géométrie

L'autre stratégie était celle qui déplaçait directement les géométries des tronçons d'itinéraires. On utilise dans un premier temps deux nouveaux champs créés : *sens* et *décalage*. Le champ *sens* va permettre de décaler les tronçons d'itinéraire selon leur sens (positivement si ils ont le même sens que le tronçon de route ou négativement sinon) (Annexe 7.3). Le champ *décalage*, calculé par la formule expliquée précédemment, va nous servir à savoir à quelle distance de la route on décale le tronçon d'itinéraire.

En utilisant ensuite l'outil *OffsetCurve* de l'API Qgis, on peut décaler directement la géométrie des tronçons d'itinéraire comme on le souhaite grâce aux champs énoncés précédemment.

4.2 Gestion de l’agencement manuel

4.2.1 Selection de l’objet

Une fois les objets décalés, il fallait implémenter un moyen de les faire sélectionner par l’utilisateur et de les déplacer. Pour ce faire, nous avons d’abord cherché une méthode basée sur la symbologie où, en récupérant les coordonnées de la souris lors du clic, nous identifierions l’objet le plus proche. Cependant, un clic sélectionne la géométrie et non la symbologie. Nous avons donc dû adapter l’idée en utilisant la géométrie de l’objet, ce qui s’est avéré fonctionner.

4.2.2 Inversions des itinéraires

Finalement, une fois l’objet correctement sélectionné, nous examinons de la même manière le moment où le clic est relâché, puis nous récupérons le tronçon d’itinéraire le plus proche pour l’inverser avec le premier. Cela permet d’intervertir les données et nous nous assurons que l’algorithme boucle pour combler les éventuels trous.

4.3 Le Plugin

Finalement afin de créer un plugin sur Qgis nous avons opté pour l’utilisation d’un plugin créé spécifiquement pour cela : L’extension *plugin builder* permet de créer le dossier source d’un plugin QGIS. Nous avons alors adapté le code source vide avec nos fonctions d’actions déclenché par des boutons issu du plugin RVA.

Pour son fonctionnement, nous avons décidé de permettre à l’utilisateur de sélectionner lui-même ces couches dans ses fichiers et de cliquer sur un bouton pour visualiser les couches décalées. Ensuite, en cliquant sur un nouveau bouton, il pourra alors réagencer ce placement à sa guise. Il est à noter que pour déplacer lui-même ces itinéraires, il devra impérativement avoir chargé les couches décalées au préalable.

5 Suivi du Projet

5.1 Analyse des risques

Tout au long de notre projet, nous avons dû élaborer des matrices de risques qui suivaient l’évolution de notre travail. Nous avons bien identifié différents aspects de notre projet qui présentaient des risques, notamment le timing, qui, comme nous le craignions, s’est avéré être un risque majeur. Nos changements de stratégie, dus à des impasses techniques, ainsi que le démarrage compliqué, en raison de notre familiarisation avec les outils utilisés (notamment l’API de QGIS, avec ses nombreuses classes et une documentation assez limitée sur certains aspects), ont été des défis significatifs. Ainsi voici la dernière forme de celle-ci (CF la figure 8). Nous avons également réalisé une matrice SWOT (figure 9).

5.2 Organisation

Afin de centraliser nos recherches, nous avons convenu d’utiliser Microsoft Teams pour partager des documents résumant nos travaux et pour prendre des notes régulières sur nos recherches et nos avancées lors de la phase d’analyse et de développement, tant entre nous qu’avec nos commanditaires. Pour le partage de code, nous avons également utilisé la plateforme offerte par Teams, permettant d’échanger des bouts de code, car ceux-ci n’étaient ni très longs ni directement reliés les uns aux autres.

Voici un diagramme de Pert (on y retrouve la succession logique des étapes de développement) :

Le court délai du projet nous impose une parallélisation des travaux.






Nature du risque	Probabilité	Gravité	Conséquence	Solution corrective	Solution préventive	Evolution
Code non fonctionnel	Forte	Forte	Absence du livrable attendu	Identification du problème avec des commentaires et des test. Recherche en lignes sur le type d'erreur rencontrée	Réalisation de test et contrôle du bon fonctionnement du code au fur et à mesure de la création de celui-ci	
Manque de cohésion du groupe	Faible	Moyenne	Mauvaise avancée du projet	Discussions et compréhension du soucis.	Activité de groupes, comme partager l'heure de repas	
Mauvaise évaluation du temps pour certaines tâches	Moyenne	Moyenne	Risque de problème au niveau du livrable	Adaptation du digramme Gantt	Communications sur notre niveau d'avancement	
Désordre dans le partage du code	Moyenne	Faible	Perte de temps et d'effort	Centralisation du travail via Git	Création d'un partage Git, sauvegarde sur clé	
Perte de données	Faible	Forte	Perte de temps	Création de backup via un partage git, teams.	Création d'un partage Git, sauvegarde sur clé	

FIGURE 8 – Matrice de risque

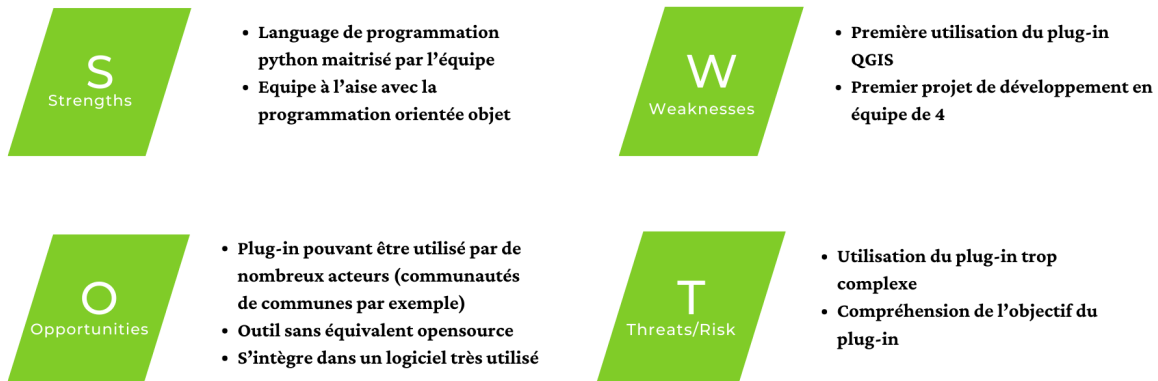


FIGURE 9 – Matrice SWOT

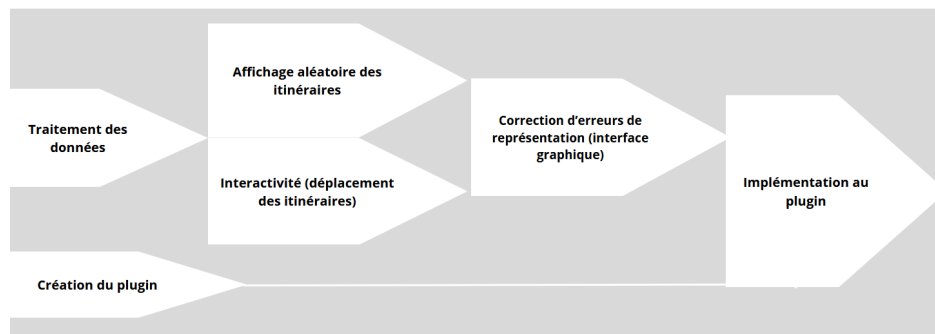


FIGURE 10 – Etapes de développement du Plugin

Pour cela, nous avons envisagé d’avoir sur le dépôt git un dossier de script python de ‘cas techniques’ en plus du dossier plug-in, mais cela c’est avéré plutôt inutile car nos code effectuait des tâches majoritairement indépendantes, de ce fait nous n’avons eu a regroupé ces codes qu’au moment de l’implémentation dans le plugin. Un regroupement qui s’est plutôt bien passé. Nous avons attribué ces différentes étapes aux membres de l’équipe. Par exemple Alexandre et Lorine s’occupait du traitement du placement automatique alors que Clara et Thibaud s’occupait de la gestion des clics et de la création du plugin en parallèle. Suivant les remarques précédentes, nous avons mis à jour le planning prévisionnel qui aura grandement évolué du au retard pris à cause des impasses de certaines pistes comme expliqué plus haut :

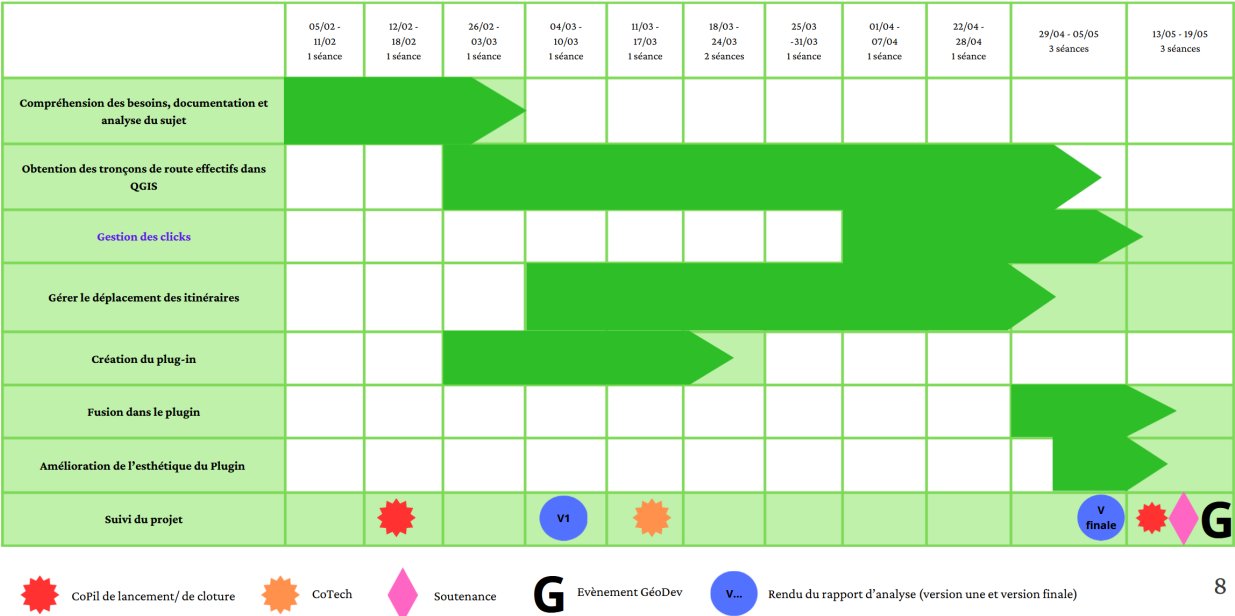
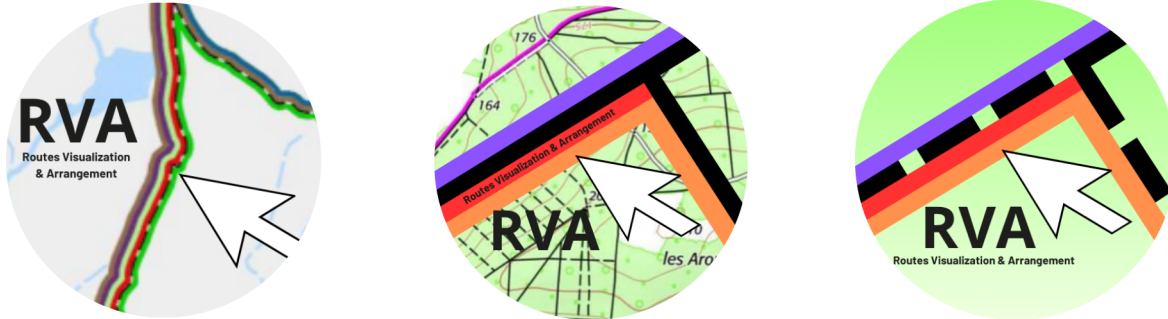


FIGURE 11 – Diagramme de Gantt

5.3 Logos

Routes Visualization and Arrangement (RVA) est le nom que nous avons retenu pour notre plugin car nous voulions qu'il soit assez explicite dans son application. Nous avons décidé de le mettre en anglais, dans le cas où il devait devenir international. En français, cela donnerait : "Visualisation et Agencement d'itinéraires", ce qui est le but de notre plugin. En ce qui concerne notre logo, nous avions une image à disposition qui représentait différents itinéraires. Nous nous en sommes inspirés et l'idée d'ajouter un clic de souris pour indiquer le but du plugin nous a semblé être une bonne idée. Un simple fond de carte étant trop chargé sur le plan esthétique, nous avons opté pour un fond de la couleur de la charte graphique de QGIS. Voici les quelques prototypes proposés et celui retenu est le troisième.



6 Conclusion

Pour conclure, ce rapport d'analyse mis à jour marque un point clé dans notre progression vers la finalisation du projet. Les analyses et stratégies initiales ont évolué au fil du temps, reflétant notre adaptation aux défis rencontrés et intégrant de nouvelles solutions plus efficaces. Bien que certaines approches décrites initialement aient divergé, les ajustements effectués ont été essentiels pour mieux répondre aux exigences techniques et aux attentes du projet.

6.1 Améliorations

Plusieurs pistes d'améliorations pourraient être alors envisagées. En effet nous avons éloignée la notion de strokes en raison de manque de temps, mais ceux-ci pourraient représenter l'évolution naturel de notre plugin. Ils permettraient de déplacer un itinéraire directement au lieu de le découper par tronçons. Et améliorerait l'aspect visuel dans les virages notamment.

- Implémenter la notion de strokes : étape 3 du schéma figure 12.
- Prendre en compte le fait que les itinéraires n'ont pas forcément tous la même largeur.
- Gestion du zoom

Ce schéma est issu de notre première analyse :

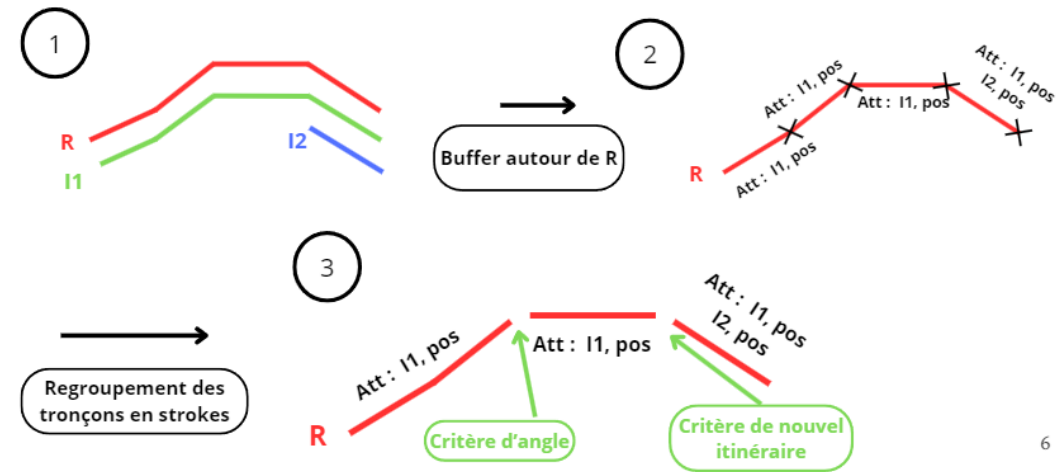


FIGURE 12 – Construction des données

Références

- [1] Teulade-Denantes J., Maudet A., Duchêne C. *Routes visualization :Automated placement of multiple route symbols along a physical network infrastructure*. The 11th Journal of spatial information science, 2015, p53-79.
- [2] Lecordix B., Palos J., Plounevez M. *Un modèle pour décrire les portions d'intérêt des itinéraires : Où trouver les itinéraires qui passent par une portion de virage mais pas dans un parc naturel ?*. 2024.

7 Annexes

7.1 Outils "décalage" QGIS

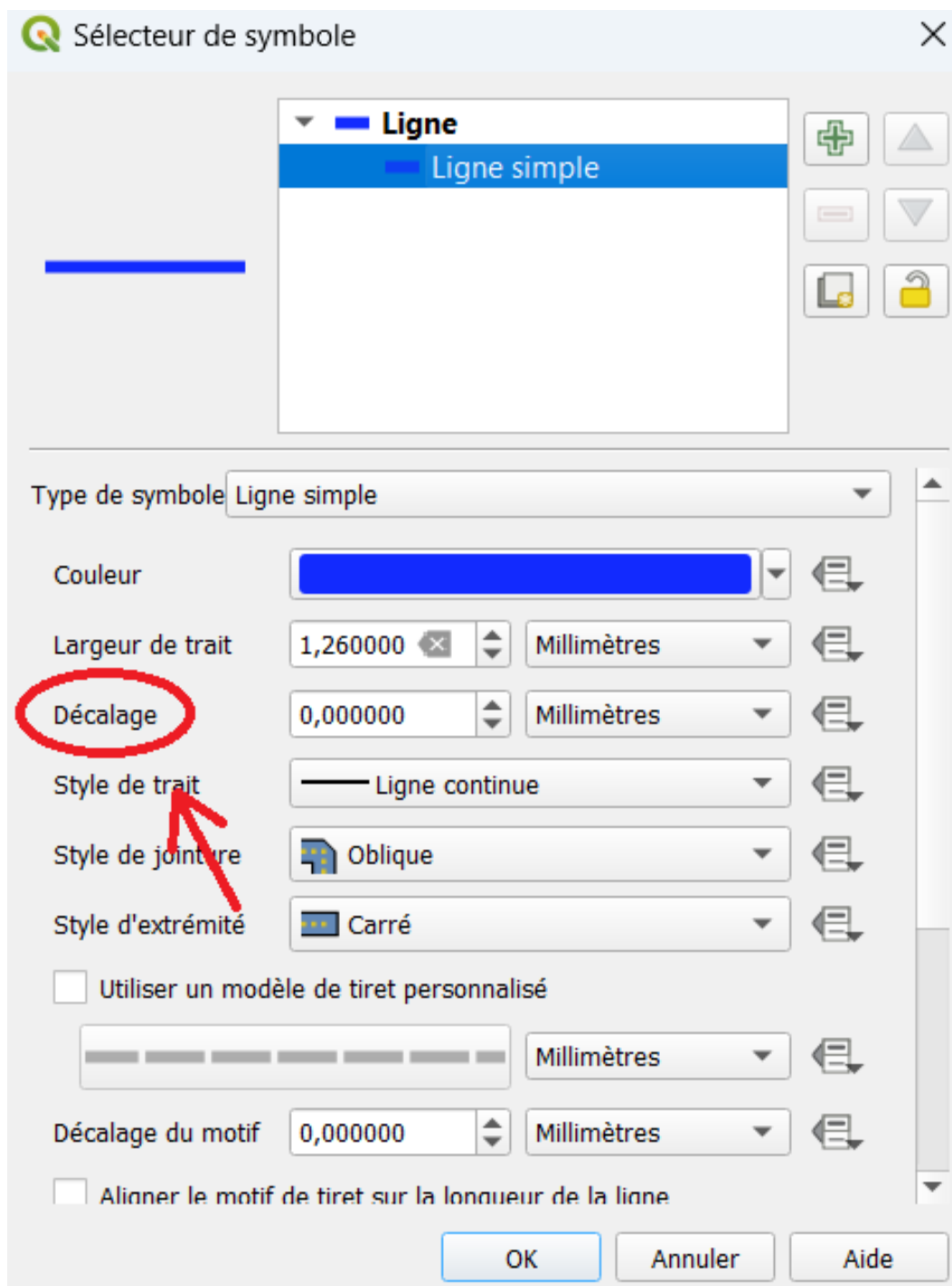


FIGURE 13 – Outil décalage dans la partie "symbolologie" de QGIS.

7.2 Ancien Diagramme de Classe

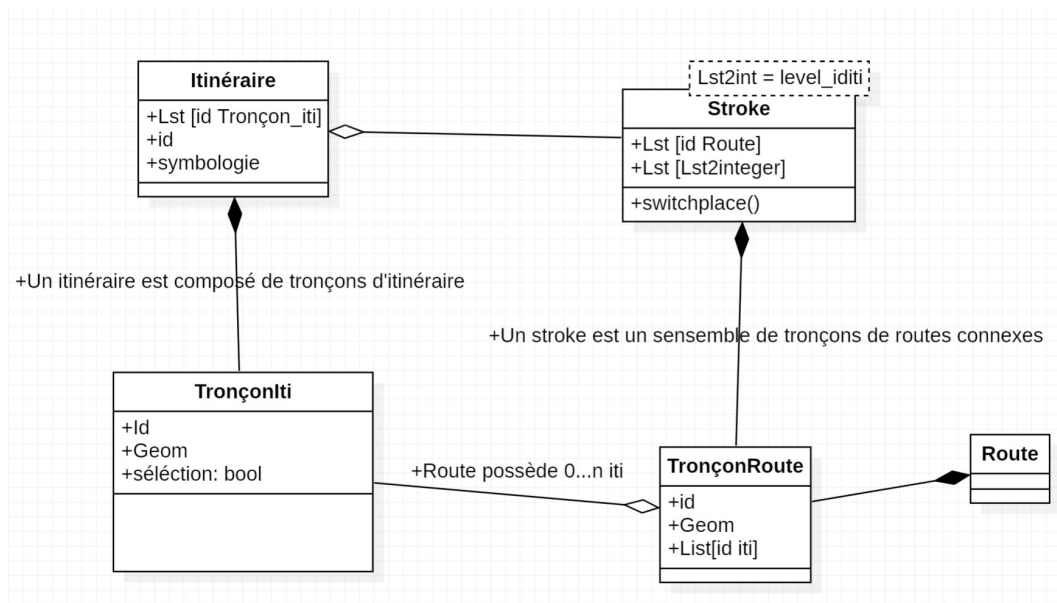


FIGURE 14 – Ancien Diagramme de Classes

7.3 Schéma explicatif des sens des itinéraires lors du décalage

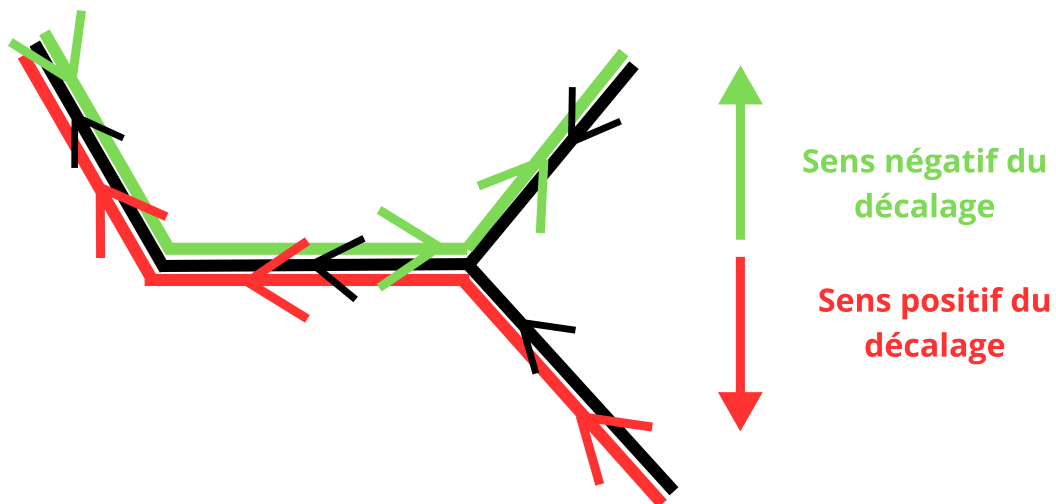


FIGURE 15 – Schéma explicatif des sens des itinéraires lors du décalage