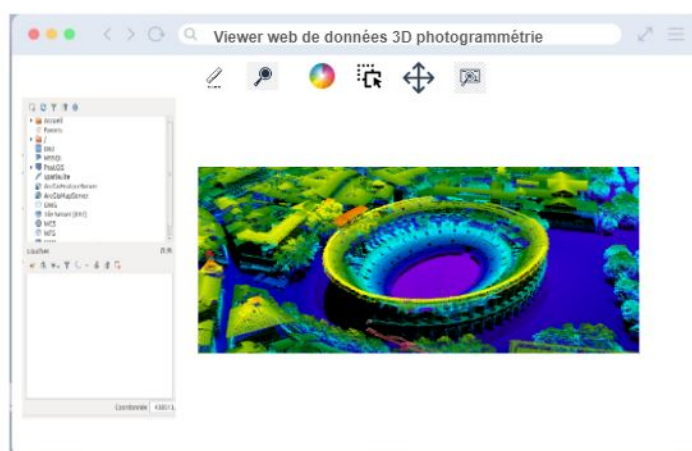


Rapport d'analyse - Projet Informatique

Cycle des ingénieurs diplômés de l'ENSG 3^{ème} année

Spécialité PPMD

Développement d'un viewer web de données 3D photogrammétrie



Gautier TABORDET

Novembre 2023

Ecole Nationale des Sciences Géographiques
6-8 Avenue Blaise Pascal - Cité Descartes - 77420 Champs-sur-Marne
Téléphone 01 64 15 31 00 Télécopie 01 64 15 31 07

Contents

1	Introduction	2
2	Contexte et objectifs	3
2.1	Contexte	3
2.2	Objectifs de l'étude	3
2.3	Contraintes	4
3	Modélisation	5
3.1	Diagramme d'utilisation	5
3.2	Diagramme de classe	7
3.3	Diagramme d'activité	8
3.4	Diagramme d'état-transition	10
4	Etude technique - Choix des logiciels et langages	11
4.1	Langage de programmation	11
4.2	Différents outils de développement	11
4.3	Choix de la solution technique	12
4.4	Agencement du viewer web	13
5	Gestion de projet	14
5.1	Outils et méthodes de travail	14
5.2	Planning provisoire	15
5.3	Matrice des risques	16
6	Synthèse	18
7	Table des figures	19
8	Bibliographie	20

1 Introduction

Au cours de ma troisième année du cycle ingénieur à l'École Nationale des Sciences Géographiques (ENSG), filière Photogrammétrie, Positionnement et Mesure de Déformations (PPMD), j'ai pour mission de mener à bien un projet informatique de Novembre 2023 à Février 2024. Au total, 72 heures sont consacrées à ce projet, divisé en 2 parties : une partie Analyse (27 heures) et une partie Développement (45 heures).

Ce projet individuel s'intitule "Développement d'un viewer web de données 3D photogrammétrie" et est commandité par Adrien Gressin, professeur de photogrammétrie et de télédétection à l'institut INSIT.

2 Contexte et objectifs

2.1 Contexte

Le contexte de ce projet repose sur la croissance exponentielle des données 3D disponibles, présentées dans une variété de formats tels que les nuages de points, les maillages (mesh) ou encore les tuiles 3D. Malgré cette abondance de données, la visualisation de ces éléments demeure souvent un défi. C'est dans ce contexte qu'émerge l'idée du projet, qui vise à développer un visualiseur web dédié aux données 3D issues de la photogrammétrie. L'objectif central est de créer un outil accessible via une interface web facilitant la visualisation de ces données complexes. Le projet aspire également à intégrer des fonctionnalités avancées tout en restant simple d'utilisation, offrant ainsi aux utilisateurs des options personnalisées pour explorer et interagir avec ces représentations 3D de manière efficace et intuitive.

2.2 Objectifs de l'étude

Dans le cadre de ce projet informatique, 3 objectifs sont principalement définis :

- **Rajouter des données par "drag and drop"**. L'utilisateur pourra donc facilement intégrer de nouvelles données sur l'interface graphique, et de manière intuitive.
- **Tourner autour du modèle**. Il est crucial de pouvoir se déplacer autour d'un modèle lorsqu'on explore des données 3D, car cela permet aux utilisateurs de visualiser tous les aspects d'un objet ou d'une scène sous différents angles, ce qui améliore considérablement la compréhension spatiale et la perception des détails.
- **Faire des mesures simples**. Un des objectifs majeurs de ce viewer web est de pouvoir réaliser des mesures sur le modèle 3D entre 2 points.

Si le temps le permet, d'autres options peuvent être implémenter pour améliorer l'utilisation du viewer web. Parmi elles, on retrouve :

- gérer les couches, la superposition. L'utilisateur doit pouvoir mettre une couche en avant par rapport à une autre, ou bien superposer 2 couches dans un environnement.
- gérer la transparence de certaines couches ou certaines classes dans une couche. L'utilisateur doit pouvoir faire varier l'option de transparence sur une couche ou une classe, afin de rendre plus ou moins visible celle-ci.
- gérer la couleur du modèle. L'utilisateur doit pouvoir sélectionner la couleur du nuage de point. Une palette de couleur peut lui être proposée.
- gérer les classes d'un nuage de point. L'utilisateur doit pouvoir sélectionner les classes qu'il souhaite sur un modèle afin de n'afficher qu'elles.

2.3 Contraintes

- Données d'entrée

Dans un premier temps, il est conseillé d'utiliser des données 3D sous forme de nuage de point, définies comme telles : XYZ RVB. On retrouve donc la position du point X,Y,Z dans le nuage, ainsi que la couleur associée au point R,V,B dans le nuage. A cela, nous pouvons ajouter un élément C qui correspondrait à la classe à laquelle appartient le point.

Pour obtenir ce type de format de données, voici un processus simple :

- Télécharger un jeu de données LidarHD IGN : Géoservice IGN - LidarHD
- Ouvrir la dalle dans CloudCompare
- Exporter le fichier au format XYZ
- Modifier les colonnes du fichier de sortie, de façon à avoir XYZ RGB C

Dans un second temps, le viewer web devrait être capable de prendre en entrée des données 3D photogrammétrie sous le format **.las**. Ce format de fichier LAS (Lidar Data Exchange File) est un format de données largement utilisé pour stocker des nuages de points lidar, qui sont des ensembles de données tridimensionnelles générées à partir de la télédétection laser. Les fichiers LAS contiennent des informations telles que les coordonnées XYZ des points lidar, l'intensité du signal laser réfléchi, la classification du point (sol, végétation, bâtiment, etc.), et d'autres attributs. Ce format est souvent utilisé pour stocker des données lidar collectées à partir d'avions, de drones ou de scanners terrestres.

- Problème de projection

Parmi les interactions à implémenter sur le viewer, l'une d'entre elles serait de pouvoir superposer des couches entre elles. Cette option soulève un problème important, celui de la projection des couches. Toutes les couches doivent être dans la même projection pour ne pas avoir des problèmes de distorsion, de décalage ou des incohérences dans la superposition des informations. Ainsi, un des enjeux majeurs sera de choisir la projection, ou bien de convertir automatiquement la projection d'un modèle dans une projection imposée.

De plus, l'outil de mesure de distance pourrait apporter des valeurs erronées si la projection n'est pas la même entre 2 modèles 3D.

Une des solutions possibles serait de convertir, dès le chargement d'une donnée, le modèle 3D dans une projection imposée. Ainsi, tous les modèles auront la même projection. Cette projection imposée pourrait être modifiée par l'utilisateur par la suite.

3 Modélisation

Avant de débiter le code, il faut modéliser l'ensemble du viewer web sous forme de diagrammes UML, permettant ainsi d'accélérer la compréhension des différentes étapes à suivre lors de l'implémentation

3.1 Diagramme d'utilisation

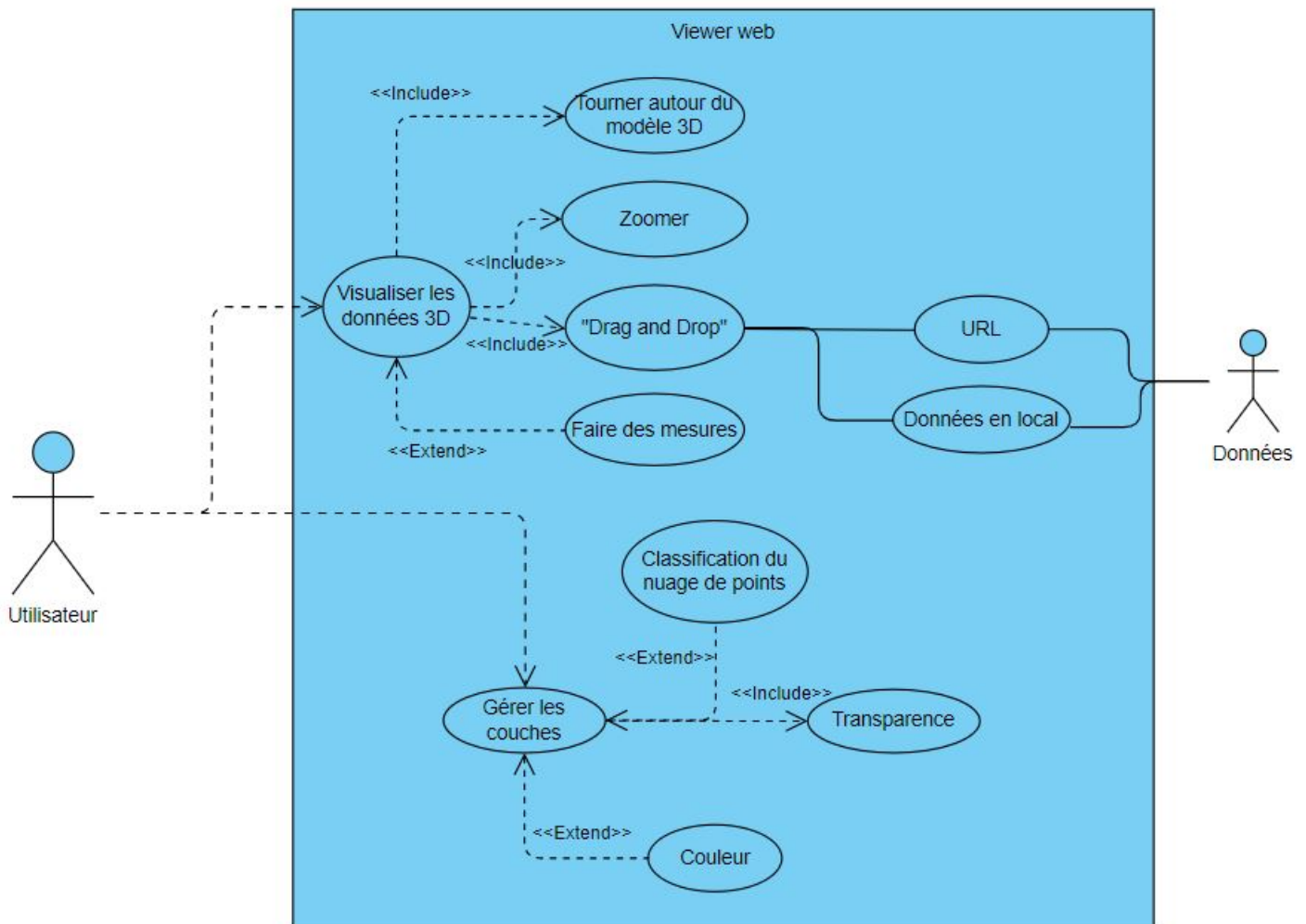


Figure 1: Diagramme d'utilisation du viewer web

Ce diagramme recense l'ensemble des fonctionnalités dont disposera le viewer web.

Tout d'abord, l'utilisateur aura la possibilité de visualiser ses données en 3D. Une fenêtre sera consacrée uniquement à l'affichage de sa donnée 3D. Pour l'ajouter au viewer, il pourra glisser son fichier par "drag and drop" directement dans la fenêtre. Les données pourront provenir d'une URL, pour les récupérer en ligne, ou alors d'un stockage local. Les deux solutions d'import seront implémentées. Une fois ses données ajoutées, il aura la possibilité de tourner autour, de se déplacer, et de zoomer sur le modèle 3D. Cela lui permettra une meilleure représentation visuelle de la 3-dimensions. Egalement, un bouton sera disponible afin de réaliser des mesures sur le modèles 3D.

A cela s'ajoute des options secondaires, comme gérer les couches grâce à une jauge de transparence, ou une sélection de couleur. De même, il sera possible de classer le modèle 3D en attribuant une couleur différente pour chaque classe.

3.2 Diagramme de classe

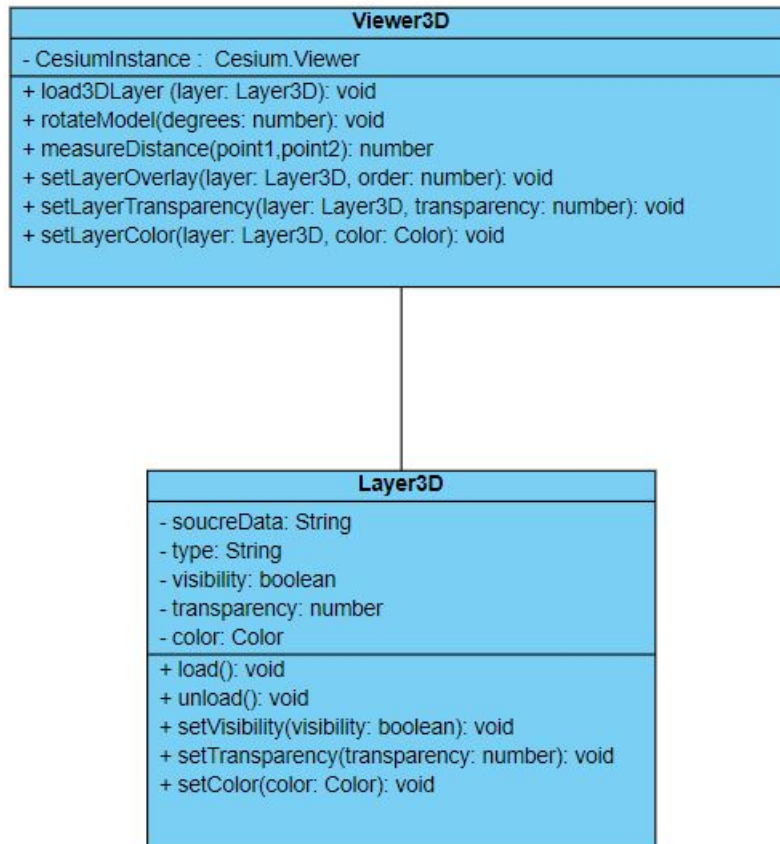


Figure 2: Diagramme de classe du viewer web

Le viewer web (représenté par la classe **Viewer3D**) peut contenir et gérer plusieurs couches 3D (représentées par la classe **Layer**). Les méthodes de la classe **3DViewer** manipulent et interagissent avec les objets **Layer**, notamment pour charger des données, effectuer des rotations, mesurer des distances, etc.

Dans un premier temps, le viewer web peut être défini à partir de 2 classes uniquement. Le nombre de classe peut-être amené à augmenter, selon l'ajout de nouvelles options au viewer web.

3.3 Diagramme d'activité

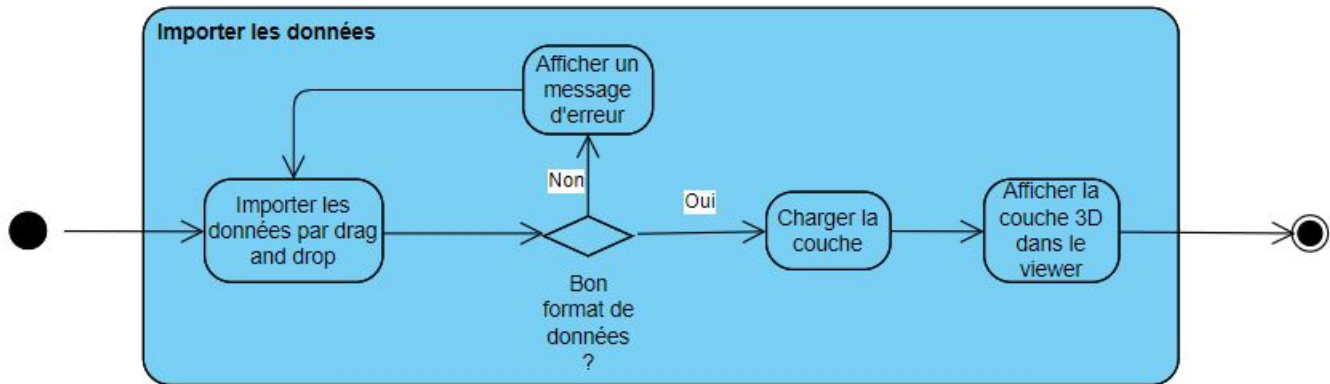


Figure 3: Diagramme d'activité du viewer web - Import de données

Lors de l'import des éléments 3D, il est essentiel de vérifier si le type des données est accepté par le viewer. Pour cela, le viewer va lire l'extension du fichier contenant les données et le comparer à la liste des types de données qu'il accepte. Si le format est le bon, alors on peut charger la couche et l'afficher dans le viewer web. Sinon, on renvoie un message d'erreur en indiquant les formats acceptés par le viewer.

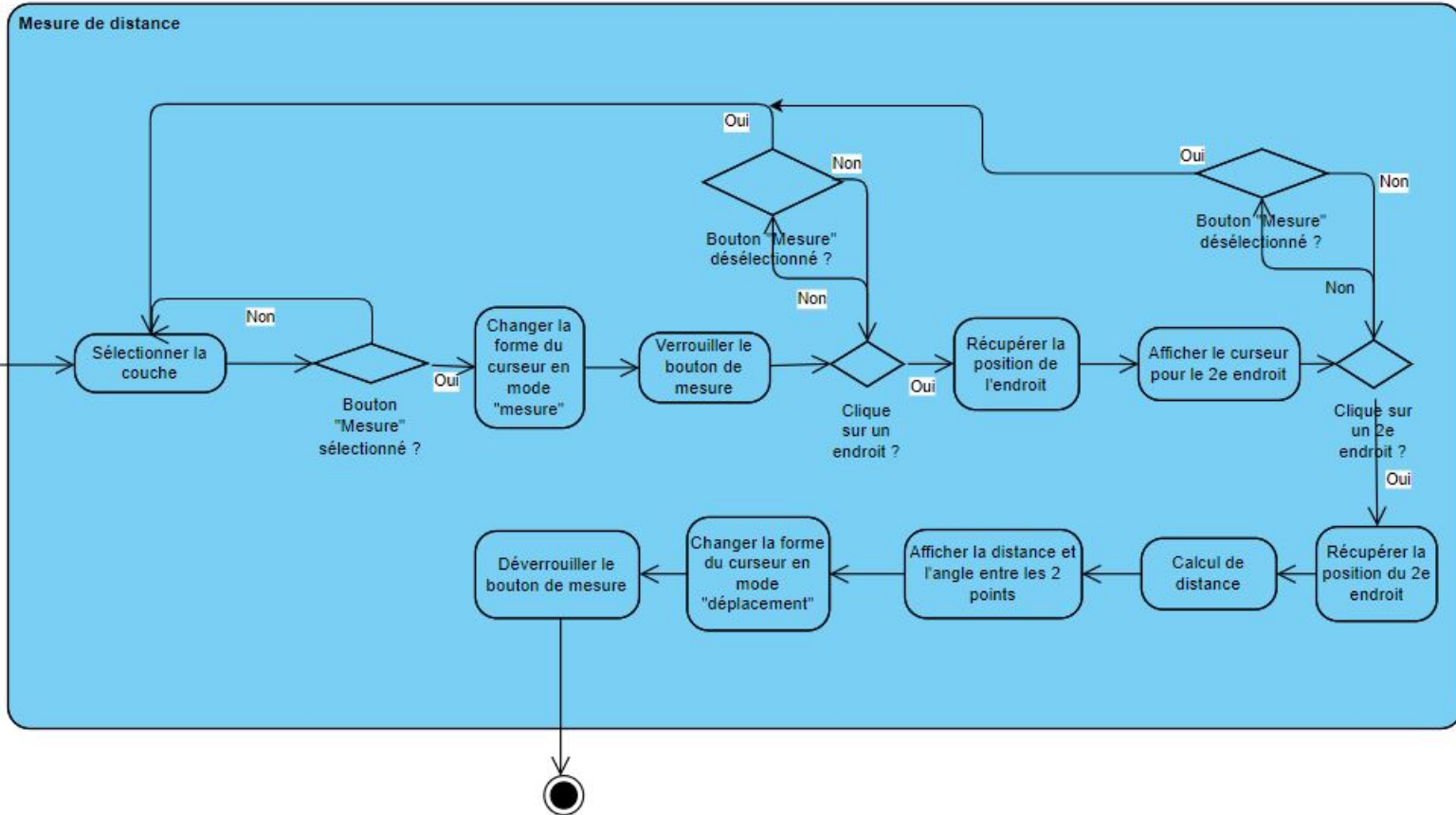


Figure 4: Diagramme d'activité du viewer web - Mesure de distance

Une fois qu'une couche 3D a été insérée dans le viewer, il est possible de réaliser des mesures afin d'obtenir la distance entre 2 points, ainsi que l'angle les séparant. Pour cela, l'utilisateur doit d'abord sélectionner la couche en entrée, puis il doit cliquer sur le bouton "Mesure" qui sera disponible un peu plus haut.

Une fois ce bouton sélectionner, le curseur (en mode "déplacement" jusqu'ici [c'est-à-dire une croix classique pour se déplacer instinctivement dans un modèle 3D]) basculera en mode "mesure", c'est à dire une petite cible afin de pouvoir pointer le premier point nécessaire à notre mesure. Le bouton de mesure en haut deviendra grisé, ce qui signifie que l'utilisateur peut commencer ses mesures.

Une fois cela fait, l'utilisateur a la possibilité de cliquer à un endroit pour l'associer au premier point de la mesure. Il peut continuer de faire tourner le modèle pour sélectionner un point qui ne serait pas visible sous ce côté. Une fois

ce premier point choisi, on proposera donc à l'utilisateur de choisir un deuxième point. Si le deuxième point est bien sélectionné, alors un calcul de distance et d'angle s'actionne, en prenant en compte la projection actuelle des données 3D photogrammétriques.

Dans un coin de l'interface s'affichera donc les 2 valeurs calculées, pour informer l'utilisateur. En parallèle, le bouton de mesure redeviendra normal (il ne sera plus grisé), et le curseur re-basculera en mode "déplacement".

Pendant la sélection des points, l'utilisateur peut annuler le mode "mesure" à tout moment en re-cliquant sur le bouton de mesure en haut de page. Ainsi, il retournera dans le mode de visualisation et de déplacement classique.

3.4 Diagramme d'état-transition

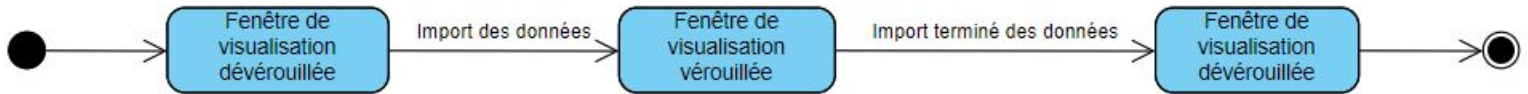


Figure 5: Diagramme d'état-transition du viewer web

Ce diagramme permet de justifier la présence d'une page de chargement qui bloquera l'ensemble du viewer web lors de l'import de données dans le viewer. Cela permettra de s'assurer du bon import de la donnée, et d'éviter tout problème de chargement.

4 Etude technique - Choix des logiciels et langages

4.1 Langage de programmation

Pour créer ce visualiseur web, il est crucial de choisir les langages de programmation appropriés afin de simplifier la visualisation des données en 3D. Nous avons opté pour l'utilisation d'HTML et de JavaScript, principalement en raison de leur compatibilité et de leur portabilité élevées. Cela garantit une expérience utilisateur cohérente sur différents appareils et navigateurs. La nature interactive de JavaScript joue un rôle essentiel en permettant aux utilisateurs de manipuler et d'explorer les données de manière dynamique.

Pour faciliter le rendu 3D, nous avons recours à des bibliothèques spécialisées telles que Three.js. Ces bibliothèques simplifient le processus de rendu 3D, tandis que la rapidité de développement, les mises à jour simples et l'intégration fluide avec d'autres technologies web contribuent à la popularité de cette approche. De plus, la flexibilité du langage JavaScript permet une intégration transparente d'outils de visualisation 3D existants tels que Potree ou CesiumJS, élargissant ainsi les possibilités de développement et offrant des fonctionnalités avancées pour la visualisation des données en 3D.

Enfin, il n'est pas utile d'implémenter en PHP pour ce projet (pourtant souvent en lien avec HTML et JS) car il n'y a pas de base de données derrière ce viewer web. Ainsi, les communications entre serveur et client se feront uniquement à l'aide d'événements via JavaScript.

4.2 Différents outils de développement

Pour développer un viewer web, 2 choix principaux s'offrent à nous :

- Potree

Potree est une bibliothèque JavaScript open-source spécialisée dans la visualisation de grands ensembles de données 3D, tels que les nuages de points générés par des technologies de photogrammétrie ou de télédétection. Conçu pour être utilisé dans des viewers web, Potree offre une représentation efficace des données 3D en utilisant des structures hiérarchiques octree. Cela permet de charger dynamiquement des parties de la scène en fonction du niveau de zoom de l'utilisateur, optimisant ainsi les performances et la gestion de grandes quantités de données. Potree propose des fonctionnalités avancées telles que la coloration des points, la classification et la mesure de distances, en faisant un outil puissant pour la visualisation interactive de nuages de points 3D dans un environnement web.

- CesiumJS

CesiumJS est une bibliothèque JavaScript open-source spécialisée dans la création d'applications de cartographie 3D interactives sur le web. Elle permet de visualiser des données géospatiales en trois dimensions, notamment des modèles 3D, des images satellites, des cartes topographiques, et d'autres informations géographiques. CesiumJS exploite les capacités des navigateurs modernes pour offrir des expériences immersives, permettant aux utilisateurs de naviguer dans des environnements 3D complexes. La bibliothèque prend en charge des fonctionnalités telles que la gestion du temps, la visualisation de trajectoires, l'interaction avec des entités géographiques, et elle peut être intégrée avec d'autres technologies comme WebGL. CesiumJS est largement utilisé dans des domaines tels que la géospatialisation, la visualisation de données géographiques, et la création d'applications basées sur la localisation.

4.3 Choix de la solution technique

Dans un premier temps, il serait préférable d'utiliser la bibliothèque CesiumJS pour l'implémentation de ce viewer web 3D. CesiumJS possède une documentation très détaillée, et de nombreux projets développés avec CesiumJS existent et sont disponibles pour se renseigner sur l'utilisation de certaines fonctions. Si des erreurs persistent, ou que l'implémentation semble compliquée au début, il est tout à fait envisageable d'implémenter le viewer web avec Potree. Comme les deux bibliothèques possèdent des fonctions permettant de réaliser ce projet, cela dépendra principalement de la complexité des fonctions à implémenter, et de comment les bibliothèques peuvent nous aider à trouver des solutions.

De plus, afin de visualiser notre viewer web pendant son développement, l'utilisation de MAMP est préconisée. Si des problèmes surviennent lors de l'implémentation, et qu'il n'est plus possible de visualiser le viewer web (à cause de l'environnement de serveur local), il est tout à fait possible de basculer à WAMP ou Vite.js. Pour rappel, le choix de l'environnement de serveur local ne changera pas l'implémentation du code, c'est uniquement un moyen d'afficher notre projet sur un serveur en local pour le visualiser.

Après une rapide recherche (qui sera à approfondir pendant la partie Développement), voici quelques fonctions de CesiumJS qui peuvent être utilisées pour l'implémentation du viewer web :

- Evenement : viewerDragDropMixin : *Globa - Cesium Documentation*
Cette fonction permet d'inclure l'option "Drag and Drop". Attention, dans notre cas, cette option doit être liée à l'import des données.
- Outil de mesure : Cesium.Measure : *Mesure - Cesium Documentation*
- Se déplacer d'un point de vision A vers un point de vision B : *moving 3D model using lat, long, ...*

Ces éléments de documentation peuvent être une aide, un point de départ dans l'implémentation du viewer.

4.4 Agencement du viewer web

Afin de mieux se rendre compte du viewer web final, voici une rapide illustration de ce à quoi devrait ressembler le viewer web :

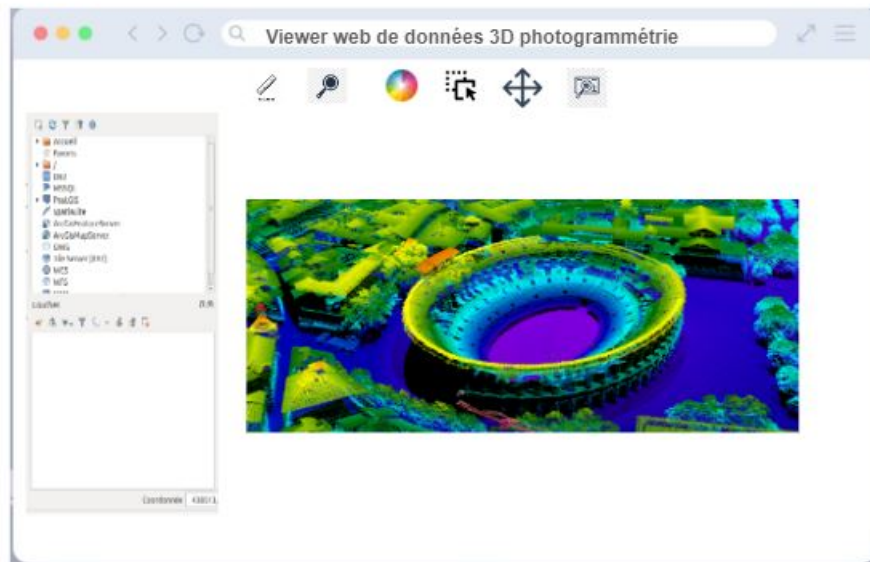


Figure 6: Montage photo pour visualiser le viewer web

On retrouve principalement une grande fenêtre centrale destinée à l’affichage des données 3D. C’est sur cette fenêtre que l’utilisateur pourra ”drag and drop” ses données pour les importer dans le viewer.

A gauche de la fenêtre centrale se trouve un gestionnaire de couches. La liste de toutes les couches importées sera visible ici, et il sera possible d’ordonner les couches les unes par rapport aux autres en les faisant glisser, ou grâce deux boutons flèches juste au dessus de ce gestionnaire.

Enfin, une dernière partie essentielle dans cette interface sera l’ensemble des outils disponibles. On y retrouve par exemple la mesure de distance, la transparence d’une couche, la classification dans une couche, mais également le changement de couleur d’une couche ou encore la flèche pour revenir à un mode de déplacement normal autour du modèle 3D.

D’autres parties pourront être ajoutées, pour rendre l’interface plus esthétique et plus ergonomique.

5 Gestion de projet

5.1 Outils et méthodes de travail

Pour le bon déroulement de ce projet informatique, il est important d'avoir des échanges avec le commanditaire pour vérifier si le projet va dans la bonne direction. Pour cela, des réunions seront régulièrement organisées via Teams. De plus, un compte-rendu de chaque réunion sera publié sur le dépôt en ligne, et sera visible à la fois par l'étudiant, le commanditaire et les professeurs référents.

Comme énoncé juste avant, l'ensemble du projet sera suivi sur GIT, une plateforme de dépôt pour accéder au code développé et recenser toutes les activités en lien avec ce projet informatique.

5.2 Planning provisoire

Ce projet se divise en 2 grandes parties : une partie Analyse (novembre) et une partie Développement (Décembre - Février). Il est donc essentiel d'organiser les différentes étapes à suivre pour le bon déroulement de ce projet.

Cependant, il est difficile d'estimer exactement le nombre d'heures passées à chaque étape d'implémentation. Voici donc une première visualisation provisoire du planning de ce projet informatique :

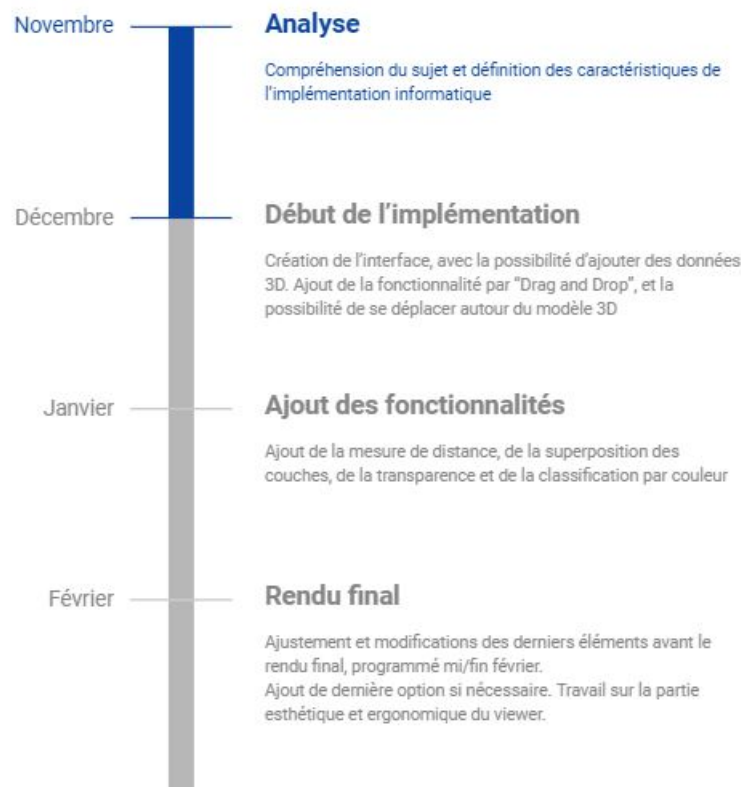


Figure 7: Planning provisoire du projet

On retrouve donc la partie Analyse pendant le mois de Novembre, puis le début de la phase d'implémentation en Décembre. Cette phase commencera principalement par la mise en ordre du lien entre client et serveur, d'un premier affichage graphique, et d'ajout de données locales dans le viewer. Egalement, si le temps le permet, l'implémentation du "drag and drop" pour ajouter des données débutera à ce moment là.

En Janvier débutera l'implémentation de la mesure de distance, ainsi que les différentes options annexes, comme la transparence ou la superposition des couches.

Enfin, en février, les dernières modifications auront lieu. Le projet arrivant à sa fin, il faudra améliorer le côté "esthétique" et "ergonomique" de l'interface, en ajustant l'emplacement des boutons et les couleurs de l'interface.

Le projet se terminera à la fin du mois de février.

5.3 Matrice des risques

Comme tout projet, des risques peuvent être présents et il est important de les recenser. Egalement, définir les solutions pour résoudre ces problèmes peut être utile pour le bon déroulement du projet. Ainsi, tout est répertorié dans la matrice des risques ci-dessous :

Risques	Mécompréhension avec le commanditaire	Le débogage échoue sur quelques points	Tous les objectifs supplémentaires n'ont pas été atteints	Prise en main difficile de la librairie CesiumJS ou Potree
Probabilité	Faible	Faible	Faible	Moyenne
Impact	Léger	Modéré	Modéré	Important
Conséquences	Fausse route	Délai	Aucun	Délai
Solution préventive	Réunions régulières	Bien commenter le code	Bien répartir le travail sur la durée	Documentation sur la bibliothèque
Solution corrective	Trace écrite des réunions	Refaire le code	Aucun	Aucun
Etat	Inactif	Actif	Actif	Actif

Figure 8: Matrice des risques du projet informatique

Comme l'implémentation du projet n'a pas encore démarrée, la majorité des risques sont encore "Actif" car ils dépendent principalement de la manière dont l'implémentation du code évolue.

On retrouve en premier lieu la mécompréhension avec le commanditaire. Ce risque a un impact faible, car des réunions sont régulièrement organisées afin d'assurer un suivi dans le projet. Les problèmes de délai et de débogage présentent un impact modéré, car ceux-ci peuvent compromettre le rendu en temps et en heure du projet.

Enfin, un risque que l'on estime à important est celui de la prise en main des bibliothèques CesiumJS ou Potree. En effet, il est primordial de connaître l'utilisation des fonctions dans ces bibliothèques, afin de perdre le moins de temps possible dans la documentation.

6 Synthèse

Ce présent rapport permet de donner un premier aperçu de l'ensemble de ce projet informatique. L'objectif est donc de créer une interface, ergonomique et facile d'utilisation, pour manipuler des données 3D issues de photogrammétrie.

L'ensemble des diagrammes et solutions préconisées sont amenés à changer pendant le développement de ce viewer web.

7 Table des figures

List of Figures

1	Diagramme d'utilisation du viewer web	5
2	Diagramme de classe du viewer web	7
3	Diagramme d'activité du viewer web - Import de données	8
4	Diagramme d'activité du viewer web - Mesure de distance	9
5	Diagramme d'état-transition du viewer web	10
6	Montage photo pour visualiser le viewer web	13
7	Planning provisoire du projet	15
8	Matrice des risques du projet informatique	16

8 Bibliographie

- Télécharger des dalles Surface 3D en .las : <https://www.swisstopo.admin.ch/fr/geodata/height/surface3d>.
- Exemple Potree utilisé sur Cesium : http://potree.org/potree/examples/cesium_retz.html
- GitHub Potree : <http://potree.org>
- Exemple viewer données Lidar du SITN : <https://sitn.ne.ch/lidar/>
- Télécharger un jeu de données : swissurface3d 2019 2540-1181 2056 5728.las.zip
- Nouveau lien de téléchargement de données : swissurface3d 2019 2548-1160 2056 5728.las.zip
- Modèle 3D : <https://drive.switch.ch/index.php/s/KuWINw6ctOWlJNw>
- Géoservice IGN - LidarHD : <https://geoservices.ign.fr/lidarhd>
- Premiers tests pour utiliser CesiumJS : Visualize a Proposed Building in a 3D City