

Rapport d'Analyse - Projet Informatique

TABORDET Gautier

November 2023

Développement d'un viewer web de données 3D photogrammétrie

Contents

1	Introduction	3
2	Contexte et objectifs	3
2.1	Contexte	3
2.2	Objectif de l'étude	3
2.3	Contraintes	4
3	Modélisation	4
3.1	Diagramme d'utilisation	5
3.2	Diagramme de classe	6
3.3	Diagramme de séquence	7
3.4	Diagramme d'activité	7
3.5	Diagramme d'état-transition	7
4	Etude technique - Choix des logiciels et langages	7
4.1	Langage de programmation	7
4.2	Choix des outils de développement	7
5	Gestion de projet	8
6	Synthèse	8
7	Table des figures	8
8	Annexe	8

1 Introduction

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) et les tuiles 3D de Cesium. 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, englobant des types variés tels que les nuages de points, les maillages, et les tuiles 3D. L'objectif central est de créer un outil convivial, accessible via une interface web, facilitant la visualisation de ces données complexes. Le projet ambitionne d'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 Objectif 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 implémentées 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.

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.

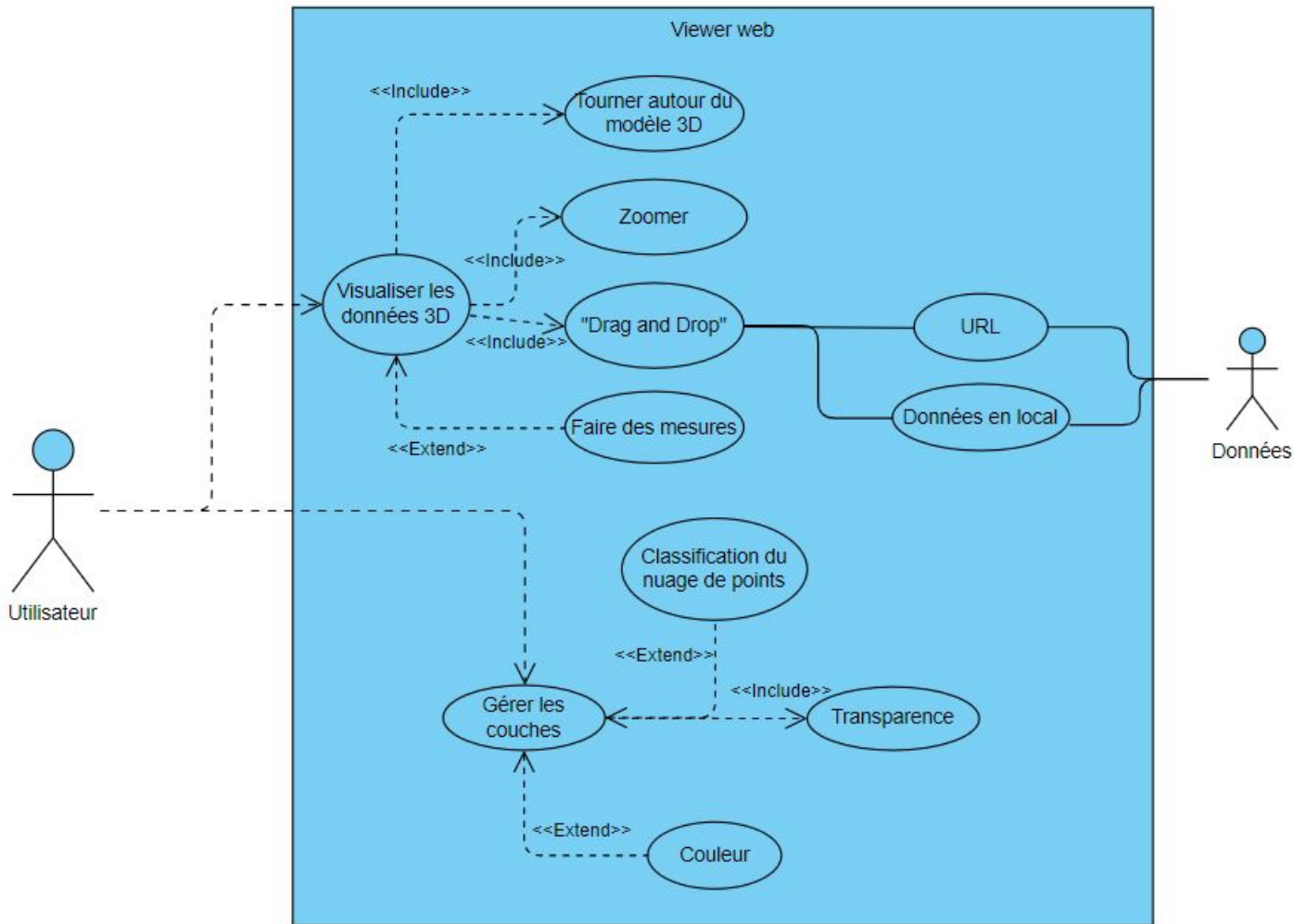
- Problème de projection

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.

3 Modélisation

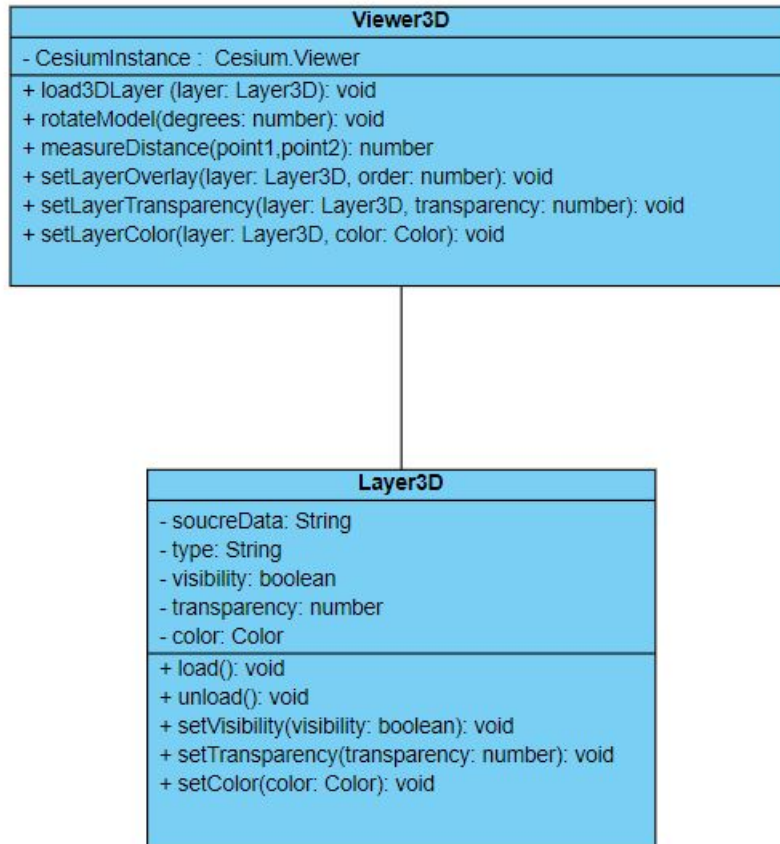
Avant de débiter le code, il faut modéliser l'ensemble du viewer web sous forme de diagrammes UML, permettant ainsi d'accélérer la compréhension des différentes étapes à suivre lors de l'implémentation

3.1 Diagramme d'utilisation



Faire la description du diagramme

3.2 Diagramme de classe



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 de séquence

3.4 Diagramme d'activité

3.5 Diagramme d'état-transition

4 Etude technique - Choix des logiciels et langages

4.1 Langage de programmation

Afin de réaliser ce viewer web, il est important d'utiliser les bons langages de programmation pour faciliter la visualisation des données 3D. Le choix s'est donc porté vers l'utilisation de HTML et JavaScript pour implémenter ce viewer web, principalement car leur compatibilité et portabilité est particulièrement élevée. Cela permet une expérience utilisateur cohérente sur divers appareils et navigateurs. La nature interactive de JavaScript est cruciale pour permettre aux utilisateurs de manipuler et explorer les données de manière dynamique. Des bibliothèques spécialisées comme Three.js simplifient le rendu 3D, tandis que la rapidité de développement, les mises à jour faciles, et l'intégration aisée avec d'autres technologies web contribuent à la popularité de cette approche. De plus, le langage JavaScript permet l'intégration transparente d'outils de visualisation 3D préexistants tels que Potree ou CesiumJS, élargissant les possibilités de développement et offrant des fonctionnalités avancées pour la visualisation de données 3D. La communauté active et l'écosystème étendu de services complémentaires renforcent encore la pertinence d'HTML et JavaScript pour la création de viewers web interactifs.

4.2 Choix des outils de développement

Pour développer un viewer web, 2 choix 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.

5 Gestion de projet

6 Synthèse

7 Table des figures

8 Annexe