



Université de Toulouse

MASTER 2 GÉOMATIQUE

« Sciences Géomatiques en environneMent et Aménagement »  
(SIGMA)

<http://sigma.univ-toulouse.fr>

Rapport de stage

# Convergence BIM – SIG dans le monde open-source

Florent LAIDIN

Magellium Artal Group

Maître de stage : Guillaume OLIVE

Enseignant-référent : Nicolas LAGARRIGUE

Mars / Août 2023



## Remerciements

Je souhaiterais remercier en premier lieu Guillaume Olive, en tant que tuteur de stage. Merci d'avoir encadré mon stage durant ces 6 mois. Tu as toujours été attentif à ma bonne intégration au sein de l'équipe. Et, comme tu l'as dit si souvent durant nos différents points, j'espère que les résultats de ce stage sont « plus mieux que si c'était moins bien ».

Merci également à toute l'unité SIG de Magellium Toulouse pour leur accueil, leur bonne humeur et les nombreux conseils qu'ils ont pu m'apporter durant ce stage.

Enfin, merci à Nicolas Lagarrigue, en tant qu'enseignant référent de ce stage. Il s'est toujours montré disponible pour répondre aux questions que j'ai pu avoir. J'ai pu bénéficier de ses différents conseils et de son suivi régulier lors de ces 6 mois.

# Résumé

Ce document traite de la convergence BIM-SIG dans un contexte open-source, en développant des solutions novatrices pour intégrer efficacement les données BIM (Building Information Modeling) dans les SIG (Systèmes d'Information Géographique). Le projet se divise en deux missions majeures.

La première mission consiste en la création d'une extension QGIS appelée « BimGeoref ». Cette extension permet le géoréférencement des objets BIM en utilisant des transformations géométriques en trois dimensions.

La deuxième mission se focalise sur le développement de la plateforme web 3DGS. Les efforts sont concentrés à la fois sur l'expansion des fonctionnalités existantes, notamment la gestion des projets et les animations 3D, ainsi que sur l'exploration de nouvelles caractéristiques comme l'intégration d'élévations et de données Lidar.

Ce travail offre une opportunité unique de pousser la convergence BIM-SIG dans un environnement open-source. Alors même que les plateformes de visualisation 3D se développent de plus en plus, les missions de développement de l'extension QGIS BimGeoref et de la plateforme web 3DGS ont permis des avancées majeures sur cette thématique.

**Mots-clés** : BIM, SIG, extension QGIS, 3D, Lidar, Web-SIG, maquettes numériques, open-source

## Abstract

This paper deals with BIM-GIS convergence in an open-source context, developing innovative solutions for efficiently integrating BIM (Building Information Modeling) data into GIS (Geographic Information Systems). The project is divided into two major missions.

The first involves the creation of a QGIS plug-in called "BimGeoref". This extension enables BIM objects to be georeferenced using three-dimensional geometric transformations.

The second mission focuses on the development of the 3DGS web platform. Efforts are focused both on expanding existing functionalities, notably project management and 3D animations, as well as exploring new features such as the integration of elevations and Lidar data.

This work offers a unique opportunity to push forward BIM-GIS convergence in an open-source environment. At a time when 3D visualization platforms are becoming more and more widespread, the development missions for the QGIS BimGeoref extension and the 3DGS web platform have enabled major advances in this area.

**Keywords** : BIM, GIS, QGIS plug-in, 3D, Lidar, Web-GIS, digital model, open-source

# Glossaire

<b>BIM :</b>	Building Information Modeling : sigle anglais pour Modélisation des Informations du Bâti.
<b>SIG :</b>	Système d'Information Géographique.
<b>JSON :</b>	JavaScript Object Notation.
<b>MNT :</b>	Modèle Numérique de terrain.
<b>IFC :</b>	Industry Foundation Class : format libre du BIM le plus utilisé.
<b>IGN :</b>	Institut National Géographique
<b>IHM :</b>	Interface Homme-Machine
<b>LAS – LAZ :</b>	Formats de fichier pouvant contenir des nuages de point. Le LAZ est le format compressé du LAS.
<b>3D :</b>	Trois dimensions.
<b>Git :</b>	Système de contrôle des versions d'un code informatique.
<b>SCR :</b>	Système de Coordonnées de Référence

# Table des matières

---

<b>Introduction</b> .....	<b>1</b>
<b>1. Contexte du stage</b> .....	<b>2</b>
1.1. Présentation de la structure d'accueil .....	2
1.2. Building Information Modeling (BIM) .....	3
1.2.1. Définition .....	3
1.2.2. Les formats existants .....	4
1.3. Système d'Information Géographique (SIG) .....	5
1.4. La convergence BIM-SIG.....	6
1.4.1. La convergence dans le monde propriétaire et open-source.....	6
1.4.2. Le BIM-SIG à Magellium.....	7
1.5. Travaux précédents.....	7
1.5.1. Extension BimToGis .....	7
1.5.2. Plateforme web de visualisation : 3DGS.....	8
1.5.3. Bilan et objectifs.....	9
1.6. Organisation du stage .....	10
<b>2. Géoréférencement des objets BIM : création de l'extension QGIS BimGeoref</b> .....	<b>12</b>
2.1. Conception du plug-in .....	12
2.2. Mise en place du géoréférencement .....	13
2.2.1. Transformations géométriques en trois dimensions .....	13
2.2.2. Calcul des paramètres .....	14
2.2.3. Sauvegarde des couches aux formats ESRI Shapefile et Geopackage .....	16
2.3. Prise en compte de l'expérience utilisateur .....	17
2.3.1. Fonctionnalités améliorant le confort d'utilisation.....	17
2.3.2. Sécurités du plug-in .....	17
2.4. Discussion .....	18
2.4.1. Bilan .....	18
2.4.2. Limites et perspectives d'évolution .....	19
<b>3. Plateforme web 3DGS</b> .....	<b>20</b>
3.1. Poursuite des développements existants .....	20
3.1.1. Gestion des projets.....	20
3.1.2. Animations sur la scène 3D.....	21
3.2. Exploration de nouvelles fonctionnalités.....	24
3.2.1. Mise en place d'une élévation au sein de l'application .....	24
3.2.2. Intégration de données Lidar .....	27
3.3. Discussion .....	31
3.3.1. Bilan .....	31
3.3.2. Limites et perspectives d'évolution .....	32
<b>Conclusion</b> .....	<b>35</b>
<b>Références</b> .....	<b>36</b>
<b>Table des figures</b> .....	<b>38</b>
<b>Annexes</b> .....	<b>39</b>

# Introduction

---

La convergence entre le BIM (Building Information Modeling) et les SIG (Systèmes d'Information Géographique) dans le monde open-source représente une avancée majeure dans le domaine de la gestion de l'information spatiale et de la construction. Le BIM désigne un dispositif de gestion des informations relatives à un projet de construction tout au long de son cycle de vie. Les SIG, quant à eux, sont des systèmes informatiques utilisés pour collecter, stocker, analyser et visualiser des données géographiques.

Ces deux domaines, bien que distincts, sont étroitement liés par leur capacité à intégrer des données géographiques avec des informations de construction et de modélisation. En combinant les fonctionnalités et les données du BIM et des SIG, la convergence permet d'obtenir une vision globale et cohérente des projets de construction et de l'environnement dans lequel ils s'inscrivent.

La convergence entre le BIM et les SIG dans le monde propriétaire a déjà fait ses preuves depuis plusieurs années, offrant des avantages significatifs en termes de planification, de gestion et d'exploitation des bâtiments ou infrastructures. Les logiciels BIM et les SIG propriétaires ont développé des fonctionnalités spécifiques pour intégrer et exploiter conjointement les données issues de ces deux domaines, facilitant ainsi la collaboration entre les acteurs du projet et améliorant la prise de décision.

Cependant, dans le contexte du monde open-source, la convergence entre le BIM et les SIG représente un potentiel peu exploité. La société Magellium a donc entrepris un projet de Recherche et Développement en 2020 portant sur cette thématique novatrice qu'est la 3D, afin de voir ce qui peut être fait, avec quels outils et les résultats qui en découlent.

Ce présent rapport de stage (réalisé de mars à août 2023) reprend les travaux précédents sur ce sujet et a pour objectif de poursuivre les recherches, d'apporter de nouveaux éléments sur cette question plus que d'actualité dans le monde de la géomatique. L'ensemble des outils utilisés pour ce travail appartient au monde de l'open-source. Ce terme, désigne que les logiciels open-source ont le droit d'être copiés, leur code source peut être récupéré et repris dans différents travaux (Perens B., 1999).

La suite de ce document se scinde en trois axes principaux. Nous reviendrons tout d'abord sur le contexte plus détaillé du stage avec une présentation des travaux précédents et l'organisation de ce stage. Ensuite, on s'intéressera à la première mission du stage qui consistait à développer un plug-in QGIS permettant de géoréférencer des projets BIM. La seconde mission de stage porte cette fois-ci sur le développement d'une plateforme web permettant de visualiser et manipuler des maquettes numériques en 3D. Pour chacune de ces missions, des parties dédiées aux apports de mon travail et des perspectives d'évolution seront présentes.

# 1. Contexte du stage

---

## 1.1. Présentation de la structure d'accueil

Magellium est une entreprise d'informatique créée en 2003, spécialisée dans les domaines de l'imagerie et de la géomatique qui comptabilise aujourd'hui plus de 300 salariés.

Les équipes techniques se répartissent entre deux sites de production à Toulouse (siège social) et Courbevoie (Paris).

Les secteurs d'activité dans lesquels intervient Magellium sont assez nombreux. On retrouve notamment le secteur de la défense et sécurité, les transports, l'énergie mais aussi l'environnement. L'entreprise est divisée en plusieurs unités dont les missions se recoupent parfois entre elles :

- L'unité **EO** (Earth Observation) s'appuie sur les imageries satellites et aériennes pour réaliser du traitement d'images et ainsi effectuer de la reconnaissance de formes, du calcul d'indices et d'autres missions.
- L'unité **SIG** (Système d'Information Géographique) se concentre majoritairement sur des solutions métiers permettant de mieux visualiser et modifier de la donnée cartographique. Cette unité livre également des outils conçus selon les besoins des clients pour produire de l'information spatiale.
- L'unité **IA** (Imagerie et Applications) quant à elle se concentre sur les systèmes de vision et détection par ordinateur, le traitement des données dans la robotique, la défense et l'industrie.
- L'unité **SAM** (Spatial, Aéronautique et Météo) intervient dans le développement de logiciels principalement axés sur les activités de simulation, d'intégration et de validation d'algorithmes.
- L'unité **C-Napps** (Cloud Native applications) participe à la création d'applications embarquées pour visualiser des données spatiales. Cette unité développe et maintient également des applications SIG concernant l'agriculture de précision notamment.

Mon stage s'est déroulé au sein du service SIG dans le site Toulousain.



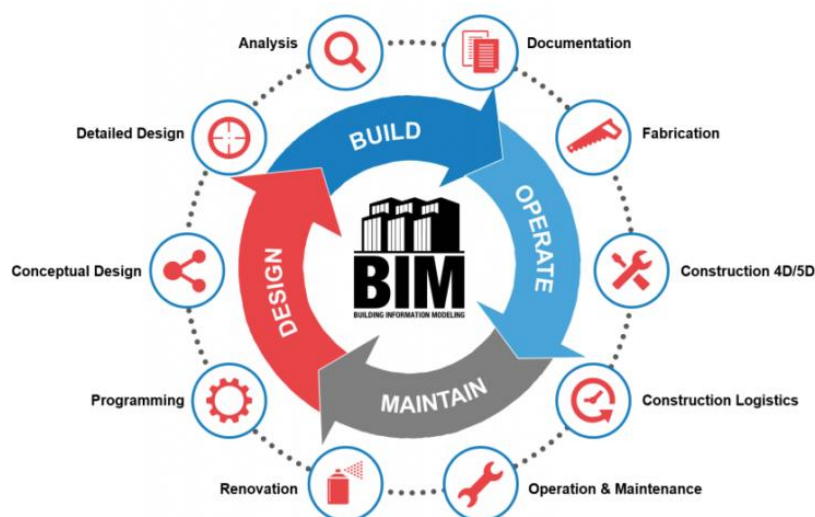
Figure 1 : Logo de l'entreprise Magellium Artal Group

## 1.2. Building Information Modeling (BIM)

Avant d'aborder le contenu du stage, il est important de revenir sur les éléments centraux de celui-ci. En premier lieu, nous pouvons nous intéresser à une définition du BIM et des différents formats existants.

### 1.2.1. Définition

Le BIM désigne « un dispositif de gestion des informations relatives à la conception, à la construction et à l'exploitation du bâtiment du point de vue de leur modélisation, leur processus et de leur management » (Chaudet B. *et al*, 2016). Ainsi, l'ensemble des données relatives à la construction d'un ouvrage, que ce soit son site ou l'ouvrage en lui-même, constituent le BIM. Ici, il faut rappeler que le BIM traite plusieurs types d'ouvrages et pas seulement le bâtiment en tant que tel. La figure ci-dessous vient illustrer ceci, le BIM fait le lien entre toutes les informations nécessaires dans la construction d'un ouvrage.



Source : villiers94.fr

Figure 2: Approche intégrée du BIM sur la gestion des projets de construction

En utilisant le BIM, toutes les données pertinentes, telles que les plans, les modèles 3D, les spécifications techniques, les calendriers, les coûts, les matériaux, et même les éléments relatifs au site, peuvent être encapsulées dans un seul fichier. Cela facilite la collaboration et la communication entre les différents acteurs du projet, qu'il s'agisse des architectes, des entrepreneurs ou des propriétaires.

Grâce à cette centralisation des informations, les modifications et les mises à jour apportées au projet peuvent être facilement suivies et intégrées. Par exemple, lorsqu'une modification est apportée à la conception d'un bâtiment, elle est automatiquement reflétée dans tous les documents et les vues associés, évitant ainsi les erreurs et les incohérences.



En résumé, le BIM offre une approche intégrée de la gestion des projets de construction (Chaudet *et al., op. cit.*), en rassemblant toutes les informations et en facilitant leur mise à jour, leur partage. Cela permet d'améliorer la productivité, de réduire les risques d'erreurs, d'optimiser les processus de construction et de favoriser une meilleure coordination entre les parties prenantes. Le BIM est donc devenu un outil essentiel dans l'industrie de la construction, offrant des avantages significatifs tant sur le plan de l'efficacité que sur celui de la qualité des ouvrages réalisés.

### 1.2.2. Les formats existants

En plus de sa capacité à centraliser les informations liées à un projet de construction, le BIM favorise également la collaboration et l'échange d'informations entre les différents acteurs impliqués. Dans ce sens, il convient d'aborder les différents formats de fichiers du BIM.

Au sein du monde propriétaire, les formats sont spécifiques à chaque logiciel et sont souvent optimisés pour fonctionner en interopérabilité avec d'autres logiciels du même éditeur. Par exemple, Autodesk Revit utilise le format de fichier RVT, tandis que Graphisoft ArchiCAD utilise le format de fichier PLN. Les formats propriétaires peuvent offrir des fonctionnalités avancées spécifiques au logiciel, mais ils peuvent également poser des défis d'interopérabilité lorsqu'il s'agit de travailler avec d'autres logiciels ou partenaires utilisant des formats différents. Ce sont tout de même les formats les plus utilisées à l'heure actuelle.

En contraste avec les formats propriétaires, il existe aujourd'hui des formats de fichier BIM ouvert dont le plus répandu est l'IFC (Industry Foundation Classes). Ce format de fichier permet d'échanger et partager des données BIM entre différents logiciels et permet une meilleure interopérabilité entre fichiers. Ce format fait donc partie de ce qu'on appelle aujourd'hui l'« open-bim ».

L'IFC, tout comme les formats propriétaires, contient l'ensemble des informations relatives au projet, les éléments sont décrits de manière hiérarchique en partant du projet jusqu'aux éléments précis composants les bâtiments avec toutes leurs caractéristiques (matériaux, géométries, revêtements, dimensions...). Cette structure hiérarchique est visible en annexe 1.

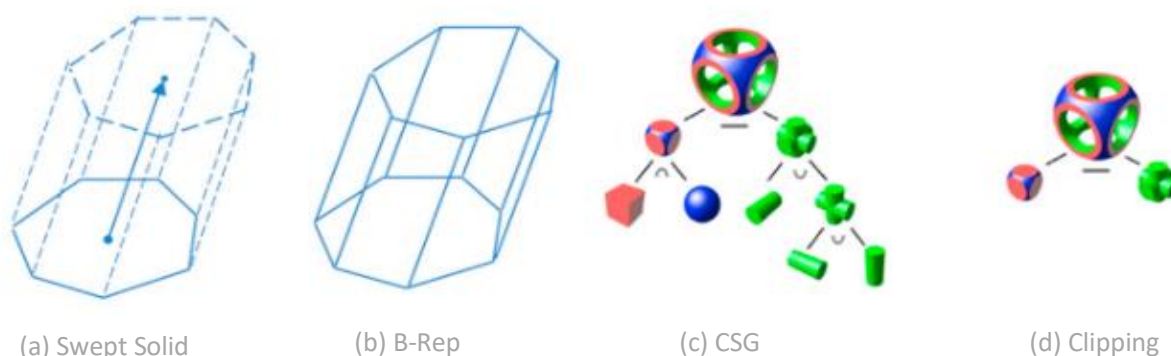


Figure 3: Types de représentations géométriques en 3D utilisées par le format IFC (Junxiang Zhu et al., 2020)

Enfin, le format IFC utilise différentes méthodes de représentation des objets en 3D, comme visible sur la figure 3. Il convient de détailler chacune de ces méthodes afin de mieux comprendre le contexte de ce stage :

- ❖ La modélisation « B-Rep » ou « Boundary Representation » forme un solide en trois dimensions à partir de ses surfaces extérieures.
- ❖ La modélisation « Swept-Solid » définit le solide à partir d'une surface (2D) et un vecteur dit d'« extrusion » qui détermine la hauteur et le volume du solide.
- ❖ « Constructed Solid Geometry » (CSG) est une méthode assez complexe de représentation géométrique qui implique des opérations entre différents objets simples (sphères, pyramides...) pour former une géométrie plus complexe.
- ❖ Les « Clipping » se forment aussi avec des opérations booléennes entre un demi-espace et un Swept Solid (ou un autre objet Clipping).

### 1.3. Système d'Information Géographique (SIG)

Un système d'information géographique (SIG) est un outil informatique permettant l'analyse et la visualisation de données spatiales. Cet outil englobe à la fois le stockage de la donnée, ses outils d'analyse et également la visualisation cartographique. L'utilisation des SIG est aujourd'hui présente dans tous les domaines. Ils offrent de nombreuses possibilités dans la collaboration et le partage des données, quel que soit l'objectif tant qu'il contient une composante liée à la localisation.

Alors que ces derniers étaient exclusivement dédiés à des traitements sur un espace en deux dimensions, on constate aujourd'hui l'arrivée de la 3D au sein des SIG à la fois dans le monde open-source et propriétaire. On retrouve notamment depuis plusieurs années la possibilité de gérer une scène 3D dans ArcGIS Pro (suite ESRI) avec également l'arrivée d'un nouveau format SIG : le format ESRI Multipatch, contenant une dimension Z.

De la même manière, le SIG open-source QGIS s'est vu implémenté dès la version 2.14 (2016) d'une fonctionnalité « QGIS 3D ». Cette fonctionnalité apporte la possibilité d'afficher une nouvelle vue cartographique pour les objets en 3 dimensions. Les formats SIG incluant une troisième dimension sont également lisibles dans QGIS avec notamment le Polygon Z et Multipolygon Z.

Concernant la visualisation des objets et couches en 3D dans QGIS, une extension a également été développée dès 2014, il s'agit de Qgis2ThreeJS. Cette dernière est aujourd'hui très répandue et utilisée lors de la manipulation d'objets 3D au sein de QGIS avec de nombreux outils pour modifier le rendu 3D comme l'exagération d'un relief, la colorisation mais également des possibilités d'export et d'import selon différents formats (image, page web).

## 1.4. La convergence BIM-SIG

Les systèmes d'information géographique et le BIM sont des solutions établies depuis maintenant de très nombreuses années et ont prouvé leur efficacité et utilité. De ce fait, la convergence entre ces deux outils ouvre la voie vers de nouvelles possibilités, à la fois en termes de traitements, mais également en termes de précision.

Pour prendre un exemple concret, l'utilisation de données BIM dans un SIG peut permettre de connaître avec précision le nombre de bâtiments présentant certains types de matériaux à l'échelle d'une commune, d'un département ou autres territoires. La combinaison de ces dispositifs apporte des plus-values des deux côtés (figure 4). Du point de vue des SIG, l'ajout de maquettes BIM améliorerait grandement leur pouvoir de communication. Concernant le BIM, les données SIG permettent d'exploiter de façon plus poussée les données contenues dans les maquettes.

La convergence BIM-SIG	
Building Information Modeling (BIM)	Système d'Information Géographique (SIG)
Etude des maquettes dans leur contexte géographique	Précision des calculs (données plus précises)
Suivi du cycle de vie du projet en lien avec la vie du territoire	Amélioration de la 3D dans les SIG
Planification de maintenance, prévention face aux risques naturels, industriels	Pouvoir de communication accru
<b>Interopérabilité , Collaboration, Gestion</b>	

Figure 4 : Synthèse comparative des avantages de la convergence BIM-SIG

### 1.4.1. La convergence dans le monde propriétaire et open-source

Dans le monde propriétaire, cette convergence voit le jour en 2017 avec un rapprochement entre les deux plus grands acteurs dans leurs domaines respectifs : Autodesk pour le BIM et ESRI pour les SIG. Cette coopération a permis des avancées majeures pour visualiser des données 3D dans les SIG et accroît les possibilités de traitement. Il est désormais possible de lire et interagir avec des maquettes au format Revit (format d'Autodesk) dans les solutions ESRI.

Dans le contexte actuel d'ouverture de l'information et du partage des données, nous pouvons qualifier cette convergence BIM-SIG dans le monde open-source d'enjeu majeur. Le chargement de formats « open-bim » dans des SIG libres est un défi de taille au vu de la complexité de ces derniers. Des recherches ont tout de même vu le jour depuis les années 2019-2020 avec les travaux de Juxiang Zhu. Ce dernier travaille sur l'intégration des fichiers IFC dans les SIG avec des algorithmes permettant de convertir les différents types de géométrie du format en question ([1.2.2](#)).

## 1.4.2. Le BIM-SIG à Magellium

Dans ce contexte, on comprend bien tout l'intérêt et l'évidence d'intégrer des maquettes numériques dans les SIG. Si cela est déjà possible et opérationnel parmi les solutions propriétaires, ce n'est pas encore le cas en open-source. Des travaux ont tout de même pu montrer la voie à suivre pour prendre en compte l'IFC dans des SIG libres.

C'est dans ce cadre que Magellium a mis en place un projet de recherche et développement sur ces questions de modélisations 3D qui sont plus que d'actualité. L'objectif étant de s'intéresser aux différentes possibilités techniques pour combiner le BIM et le SIG. À partir de là, différents axes de recherche sont disponibles avec à la fois un travail sur ces questions de conversion du format IFC mais également des enjeux autour de solutions innovantes telles que le Web-SIG, avec des outils libres. Ce projet de recherche, débuté en 2020, a permis de nombreuses avancées qu'il convient de détailler.

## 1.5. Travaux précédents

Ce travail intervient après déjà trois stages sur la thématique. Il est donc nécessaire de détailler les avancées conséquentes faites sur le sujet.

### 1.5.1. Extension BimToGis

Un premier stage réalisé en 2020 a permis la création d'une extension QGIS nommée « BimToGis ». Cette extension a pour objectif de charger en entrée un projet BIM (format IFC) et de convertir ce fichier en différentes couches SIG aux formats Shapefile, Geopackage et PostgreSQL.

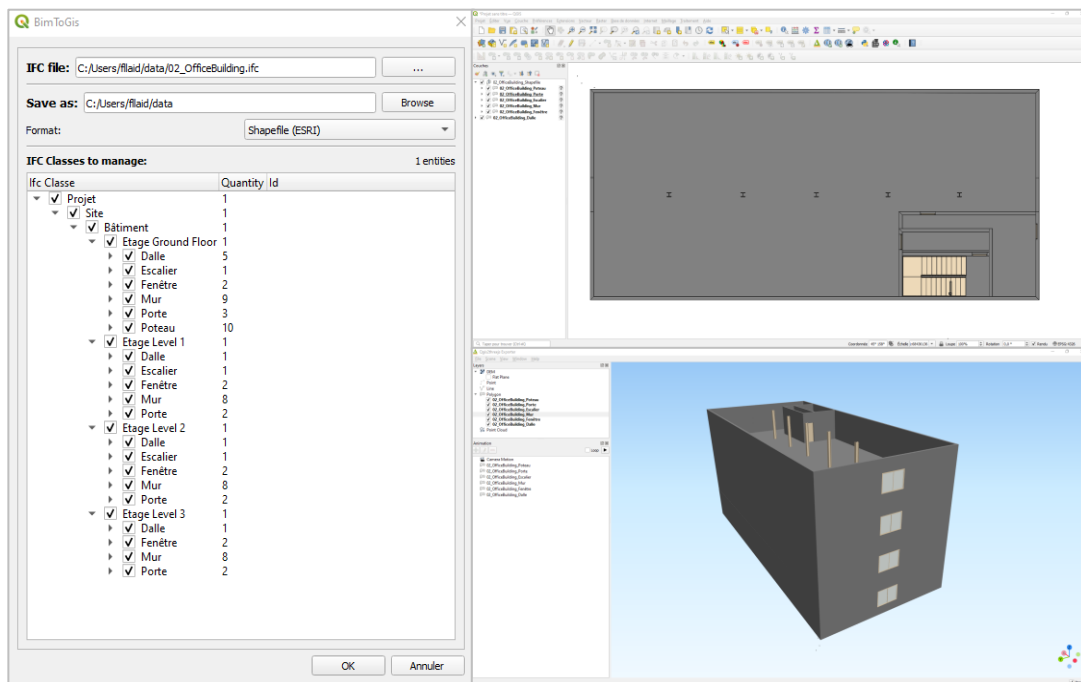


Figure 5 : Interface Homme-Machine du plug-in BimToGis et résultat obtenu dans QGIS et l'extension Qgis2Threejs

L'enjeu était ici de pouvoir lire le format IFC et de convertir chaque élément du projet en un format SIG. Comme évoqué précédemment (1.2.2), ce format utilise différentes méthodes de représentation 3D. Plusieurs algorithmes ont été créés dans l'extension afin de convertir chaque élément du projet BIM en une modélisation 3D utilisée dans les SIG (Juxiang Zhu *et al.*, 2019).

Les représentations géométriques les plus simples ont d'abord été traitées avant de prendre en compte les modélisations plus complexes. Il en résulte que le plug-in permet de convertir une grande partie des éléments d'un projet IFC chargé. Le résultat correspond à un ensemble de couches SIG décrivant le projet BIM, comme visible sur la figure 5.

Ce plug-in est à l'heure actuelle expérimental et n'est pas disponible dans le catalogue d'extension en ligne de QGIS. Au niveau des librairies python nécessaires pour lancer le plug-in, on retrouve les bibliothèques GDAL-OGR et PyShp pour les manipulations de géométrie ainsi que Sympy et Numpy pour les opérations mathématiques. Psycopg2 est également indispensable concernant la sauvegarde dans une base de données PostgreSQL. Enfin le parseur sémantique IFCOpenShell est nécessaire pour lire et analyser les fichiers IFC qui sont très complexes en raison de leur structure hiérarchique (annexe 1).

### 1.5.2. Plateforme web de visualisation : 3DGS

En parallèle à ces avancées, une plateforme web « 3DGS » a vu le jour en 2021 pour visualiser les données BIM dans un Web-SIG. Cette plateforme a pour principal objectif de mieux visualiser et interagir avec les maquettes numériques obtenues en sortie de BimToGis. Du point de vue de la réalisation technique, 3DGS a été développé en Javascript, CSS et HTML. La librairie pour le rendu 3D des projets BIM est ThreeJS. Cette dernière est très utilisée dans la manipulation d'objets 3D au sein de pages web et présente une documentation assez importante. Comme visible sur la figure 6, les données utilisées pour la plate-forme web sont stockées dans une base de données PostgreSQL. Celles-ci sont connectées à un Geoserver et appelées sous formes de flux au sein de l'application web.

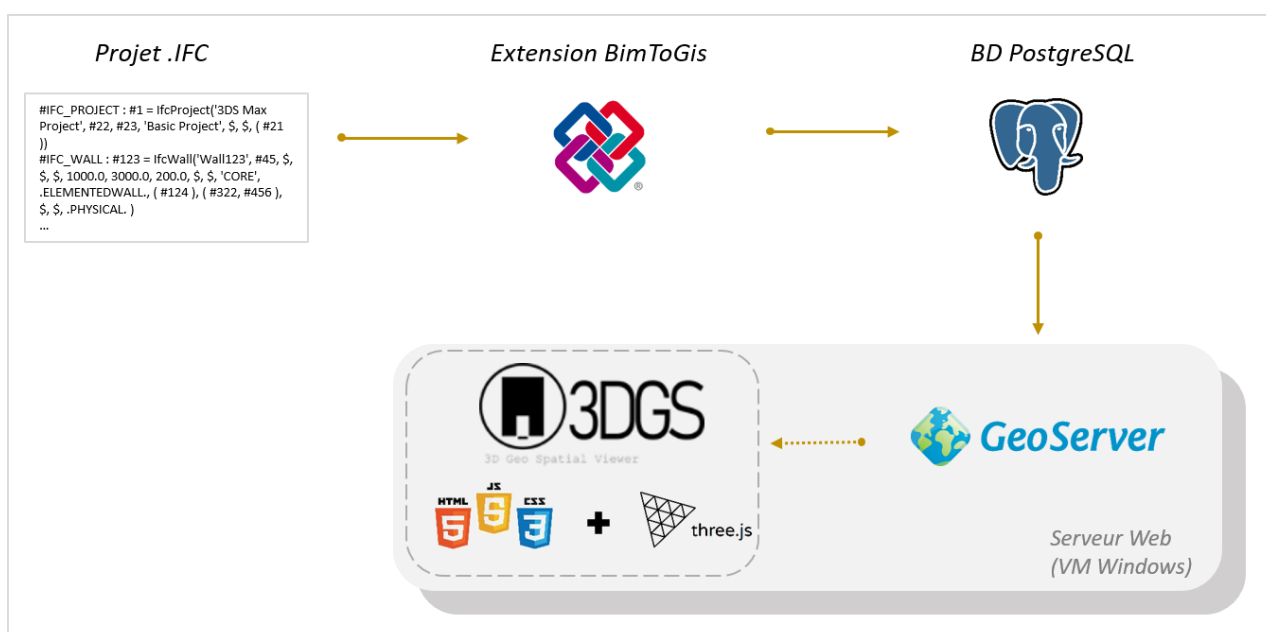


Figure 6 : Processus du traitement de données IFC vers la plateforme web 3DGS

Cette application web a fait l'objet de plusieurs développements consécutifs. Initialement conçue pour visualiser et interagir avec les maquettes numériques, 3DGS s'est vue dotée de nombreux outils. En effet, l'utilisateur a la possibilité d'afficher les maquettes selon certains niveaux ou étages, on retrouve des outils de découpes sur 3 axes des bâtiments, de requêtes sur les attributs des couches et aussi des statistiques sur ces dernières. Il existe des fonctionnalités propres aux SIG avec des options de symbologie des couches (figure 7), d'informations au clic et la mise en place de géosignets. L'entièreté du projet peut être sauvegardée et chargée depuis la mémoire locale (ou « LocalStorage » en anglais) des machines.

En résumé, la plateforme open-source 3DGS permet une visualisation 3D de maquettes et des outils d'analyse à la fois sur la composante SIG et BIM rendant compte des possibilités qu'implique la convergence entre ces deux domaines.

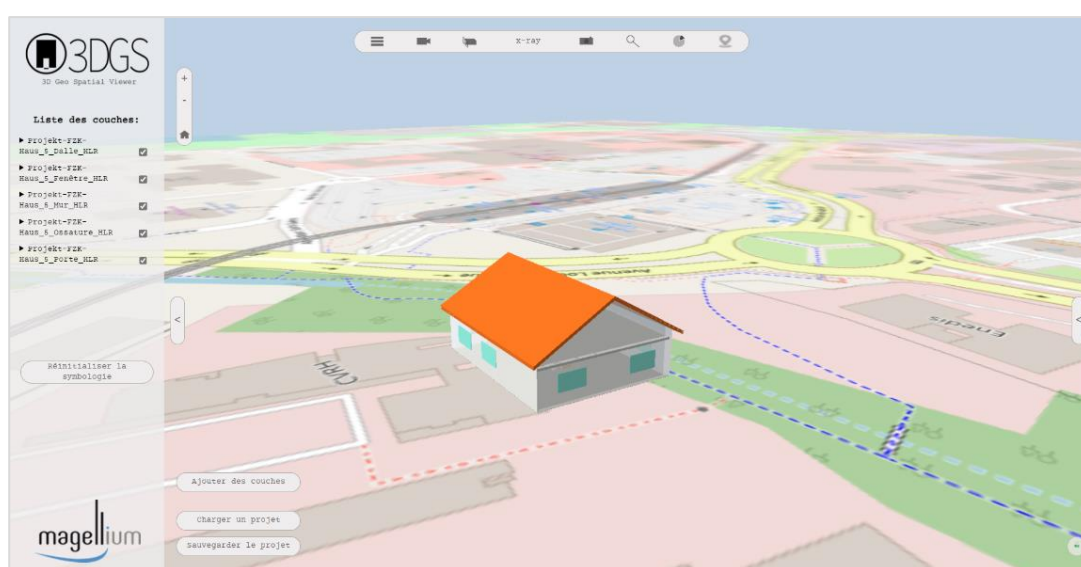


Figure 7 : Visualisation d'une maquette numérique sur l'application 3DGS.

### 1.5.3. Bilan et objectifs

Les travaux précédents se sont donc concentrés sur la conversion des projets BIM vers du format SIG en essayant d'être le plus précis possible. La plateforme web quant à elle a permis de montrer les possibilités techniques concernant la visualisation et la manipulation de maquettes numériques.

L'objectif du stage va être de poursuivre dans cette voie avec de nouveaux apports. Des améliorations sont en effet possibles sur la partie QGIS avec une extension qui à l'heure actuelle ne prend pas en compte la localisation souhaitée par l'utilisateur en sortie de traitement. Côté web, la plateforme 3DGS est encore récente et de nombreuses fonctionnalités peuvent y être implémentées pour améliorer l'expérience utilisateur dans sa manipulation de maquettes numériques, avec des outils open-source.

## 1.6. Organisation du stage

Avant d'en venir au contenu des missions réalisées durant mon stage, je peux donner des précisions sur le fonctionnement et l'organisation de ces 6 mois. En effet, ce stage de recherche et développement a été divisé en différentes phases (figure 8).

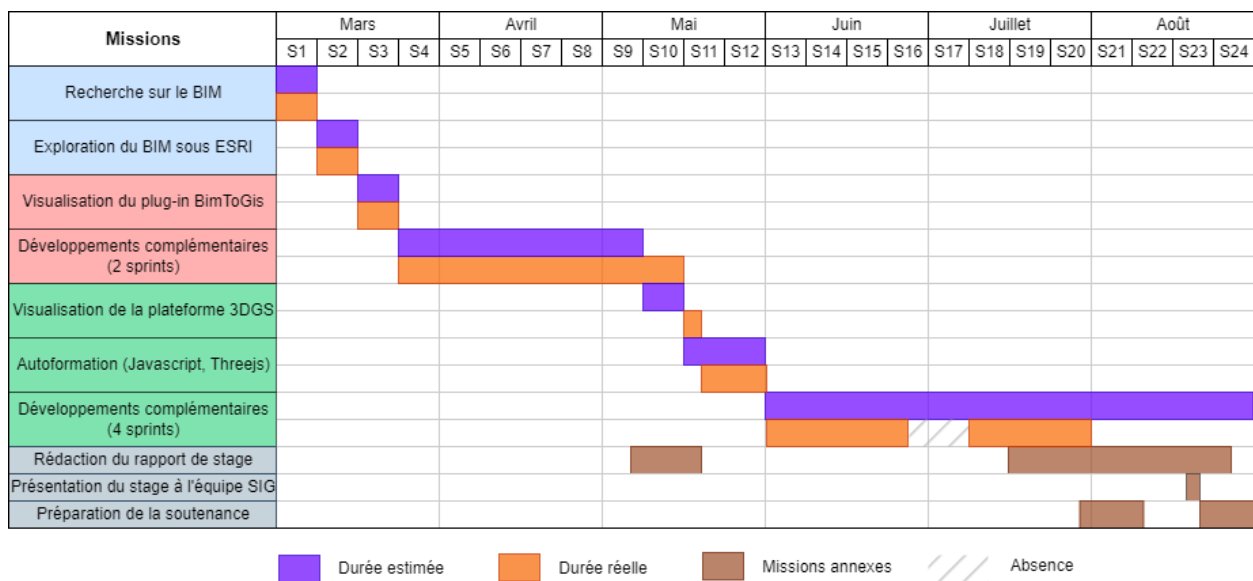


Figure 8 : Planning prévisionnel et durée réelle des grandes missions du stage

Le stage a été rythmé selon une méthode de gestion de projet dite « Agile ». Cela signifie que pour chaque phase de développement (sprint), des objectifs sur 3 semaines étaient clairement définis avec le maître de stage. Une fois le sprint terminé, une revue de sprint était systématiquement effectuée pour revenir sur le travail effectué, les réussites et les difficultés rencontrées.

Les premières semaines ont été entièrement consacrées à la découverte de l'entreprise et du sujet de stage. Étant donné le volume conséquent des travaux précédents, il m'a fallu un certain temps pour saisir les fonctionnalités mises en place et la méthodologie utilisée. Ces trois semaines ont donc servi de base pour m'approprier le sujet et me documenter ces questions du BIM et de la modélisation 3D dans les SIG.

Suite à cela, une première phase a été consacrée au développement d'une nouvelle extension QGIS. Le développement, initialement prévu à 2 sprints, a finalement pris 7 semaines. En effet, suite au deuxième sprint, des ajustements ont été nécessaires sur le rendu et la clarté des informations.

Dans un second temps, j'ai pu basculer sur une partie entièrement axée sur le web avec la plateforme 3DGS (1.5.2). Ici, un premier temps de prise de connaissance de l'application et ses fonctionnalités a été nécessaire avec en parallèle une remise à niveau sur le JavaScript et la librairie ThreeJS. Cela m'a permis d'être mieux préparé avant de débiter les développements complémentaires. Deux sprints se sont succédé dans l'été sur différentes thématiques. Initialement prévu à 4 sprints, il n'y eut que deux sprints à proprement parler. Des ajustements et développements additionnels se sont tout de même ajoutés sur la fin de l'été.

Des diagrammes de Gantt ont pu être réalisés pour chacun des sprints et sont visibles en annexe (annexe 2, 3 et 4). Ceux-ci permettent d'avoir une vision d'ensemble des missions réalisées dans le stage et du temps qui leur a été consacré.

À cela s'ajoutait un point qui était effectué tous les deux jours avec mon tuteur sur les avancées des missions. Ces points étaient l'occasion de tester les différentes fonctionnalités que j'ai pu mettre en place. En effet, la vision du développeur et celle de l'utilisateur diffèrent sur certains points. Ces réunions ont permis de rapidement prendre conscience des éléments manquants ou non fonctionnels.

J'ajouterais que, le stage étant principalement axé sur du développement, j'ai pu m'appuyer sur l'utilisation de Git comme outil de versionnement du code. Cet outil s'avère précieux tant sur sa capacité à sauvegarder les modifications que sur la possibilité de créer des branches lorsque l'on souhaite faire des tests qui peuvent influencer le fonctionnement global de l'application.



## 2. Géoréférencement des objets BIM : création de l'extension QGIS BimGeoref

---

Le détail des travaux précédents ([1.5](#)) a pu mettre en évidence la faisabilité technique de la conversion d'un format BIM vers un SIG et ce, avec des outils open-source. Cependant, les couches en sortie du plug-in n'ont pas de systèmes de coordonnées attribués et les dimensions ne sont parfois pas à la bonne échelle. La première partie du stage fut donc entièrement consacrée au géoréférencement des couches SIG en sortie de BimToGis.

À la suite d'un échange avec mon tuteur (Guillaume Olive), le choix de créer un nouveau plug-in plutôt que d'ajouter le géoréférencement dans BimToGis a été effectué. Ce plug-in est en effet déjà assez développé et le géoréférencement nécessite de nombreuses fonctionnalités supplémentaires dont l'intégration dans BimToGis aurait été plus complexe.

Cette partie reprend les grandes lignes du développement de l'extension avec la conception de l'IHM (Interface Homme-Machine) mais également les algorithmes mis en place. Nous verrons ensuite comment a été prise en compte l'expérience utilisateur. Le dernier point sera consacré sur un bilan et une discussion sur l'utilisation et l'efficacité de BimGeoref.

### 2.1. Conception du plug-in

Nous pouvons à présent détailler la conception de l'extension QGIS BimGeoref. Afin de concevoir une IHM adaptée aux besoins de l'utilisateur et au traitement du plug-in, il convient de bien comprendre quels sont les objectifs en fin de traitement. En effet, l'extension présente deux principales missions. La première est de pouvoir choisir la projection des couches SIG composant le projet BIM. Ensuite, à partir de cette projection l'extension doit permettre de géoréférencer les couches avec des nouvelles coordonnées.

Dans cette optique, le plug-in BimGeoref a pu être mis en place. Ici, j'ai pu m'appuyer sur le manuel de développement PyQGIS <sup>1</sup> pour créer les fichiers de dépendance nécessaires au bon fonctionnement de l'extension. Par la suite, l'IHM a pu être conçue, comme visible sur la figure 9. Celle-ci a pu être réalisée avec le concepteur d'interface graphique [Qt Designer](#). L'ensemble de l'interface est relié à l'algorithme de BimGeoref via le module python [PyQt](#).

On retrouve tout d'abord la partie sur le choix du Système de Coordonnées de Référence (SCR) qui modifie la projection du projet QGIS. La deuxième partie du plug-in QGIS concerne la sélection des points pour le géoréférencement. Un outil de sélection est disponible pour chacun des trois points, celui-ci permet de récupérer les coordonnées d'un sommet quelconque sur la couche active. Les points sont ensuite automatiquement rentrés dans les zones prévues à cet effet. L'utilisateur peut rentrer juste en dessous les nouvelles coordonnées du point. La dernière partie de l'extension concerne le choix du dossier de sauvegarde et le format SIG.

---

<sup>1</sup> PyQGIS Developer Cookbook : [https://docs.qgis.org/3.28/en/docs/pyqgis\\_developer\\_cookbook/index.html](https://docs.qgis.org/3.28/en/docs/pyqgis_developer_cookbook/index.html)

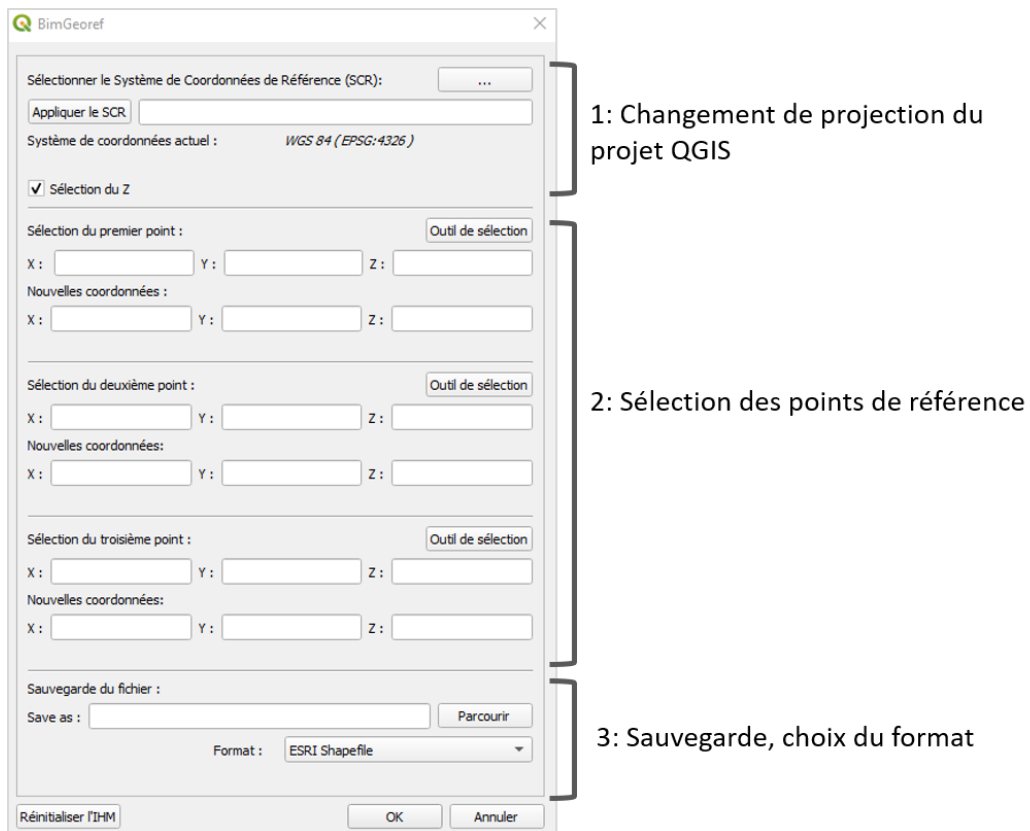


Figure 9 : Interface Homme-Machine (IHM) de l'extension BimGeoref

Il est possible de sauvegarder au format ESRI Shapefile et Geopackage. Au lancement du plug-in, l'algorithme de géoréférencement est alors exécuté pour modifier les différentes couches SIG au format choisi.

## 2.2. Mise en place du géoréférencement

### 2.2.1. Transformations géométriques en trois dimensions

Dans la mesure où les couches traitées possèdent trois dimensions (X Y Z), leur géoréférencement nécessite trois points d'appui. Ainsi, à partir de trois points connus et de leurs nouvelles coordonnées, il est possible de calculer les paramètres de la transformation qui a pu s'opérer.

Il existe différentes transformations géométriques pour les objets en 3 dimensions. Les plus connues et utilisées sont la transformation affine et la transformation de Helmert. Celles-ci diffèrent sur leur capacité à conserver la géométrie des objets traités. En effet, la transformation affine ne conserve pas les angles et longueurs (Laloux M., 2011). Elle dispose dans sa formule des changements d'échelle sur trois axes.

À l'inverse, la transformation de Helmert ne dispose que d'un seul changement d'échelle dans sa formule. L'ensemble des proportions est donc conservé lors de cette transformation (figure 10). C'est cette transformation qui a été choisie pour constituer l'algorithme de géoréférencement au sein de l'extension BimGeoref. Pour résumer, la transformation de Helmert est en réalité un sous-ensemble de transformation affine plus adaptée pour passer d'un système de référence à un autre.

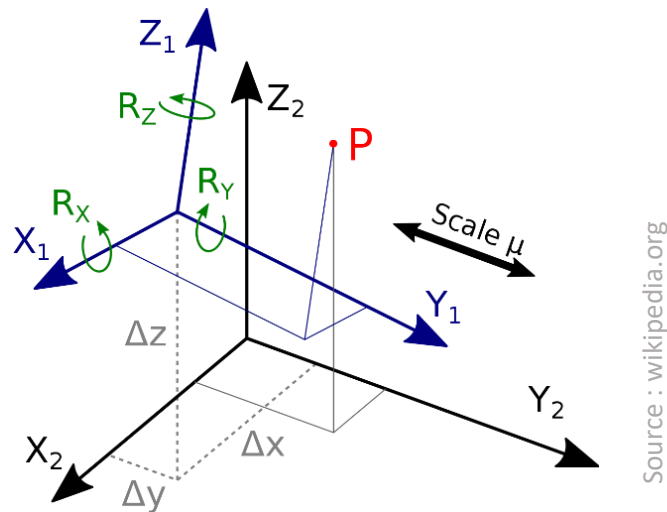


Figure 10 : Principe de la transformation de Helmert sur trois dimensions

$s$  : Facteur d'échelle  
 $R$  : Matrice de rotation (en X, Y et Z)  
 $T_x$  : Translation en X  
 $T_y$  : Translation en Y  
 $T_z$  : Translation en Z

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}^A = s * R * \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^B + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

Équation 1 : Formule de la transformation de Helmert pour passer d'un système de référence B à un système de référence A.

Comme visible sur l'équation ci-dessus, cette transformation correspond à la succession d'une rotation et translation sur trois axes, ainsi que d'une homothétie avec un seul facteur d'échelle « s ». À partir des points connus et de leurs nouvelles coordonnées on peut alors calculer les 7 inconnus de l'équation pour passer d'un système à un autre. Pour résoudre une telle équation, plusieurs méthodes sont possibles.

### 2.2.2. Calcul des paramètres

La première méthode est la méthode itérative de Jacobi. C'est une méthode d'approximation successive (Zavoti J., Tamas J., 2006). L'équation est d'abord réorganisée selon un système d'équations linéaires pour ensuite être initialisée avec des valeurs arbitraires. À partir de ces valeurs, on itère un certain nombre de fois jusqu'à obtenir les 7 inconnues de l'équation avec des valeurs d'erreurs les plus faibles possibles.

La seconde méthode permettant de résoudre une transformation de Helmert 3D est la méthode des bases de Gröbner. C'est cette méthode qui a été retenue pour l'algorithme de géoréférencement au sein de l'extension BimGeoref. Les bases de Gröbner permettent de réduire le système d'équations de Helmert en un ensemble d'équations plus simple à calculer appelée « forme normale de Gröbner ».

En comparaison avec la méthode de Jacobi, cette méthode de calcul est plus simple à implémenter en python puisque cela revient à définir plusieurs variables par un ensemble de calculs prédéfinis. Le résultat de l'algorithme varie selon les coordonnées rentrées.

Concrètement, la résolution de l'équation de Helmert (équation 1) avec la méthode des bases de Gröbner nécessite des étapes intermédiaires. Dans un premier temps, la matrice de rotation peut être paramétrée pour en obtenir une version plus simplifiée. Par la suite, un système d'équations non linéaires peut-être démontré. Pour résoudre ce système, Awange JL. et Grafarend EW. (2002) ont pu réduire les équations à un ensemble de calculs prédéfinis avec l'aide des bases de Gröbner.

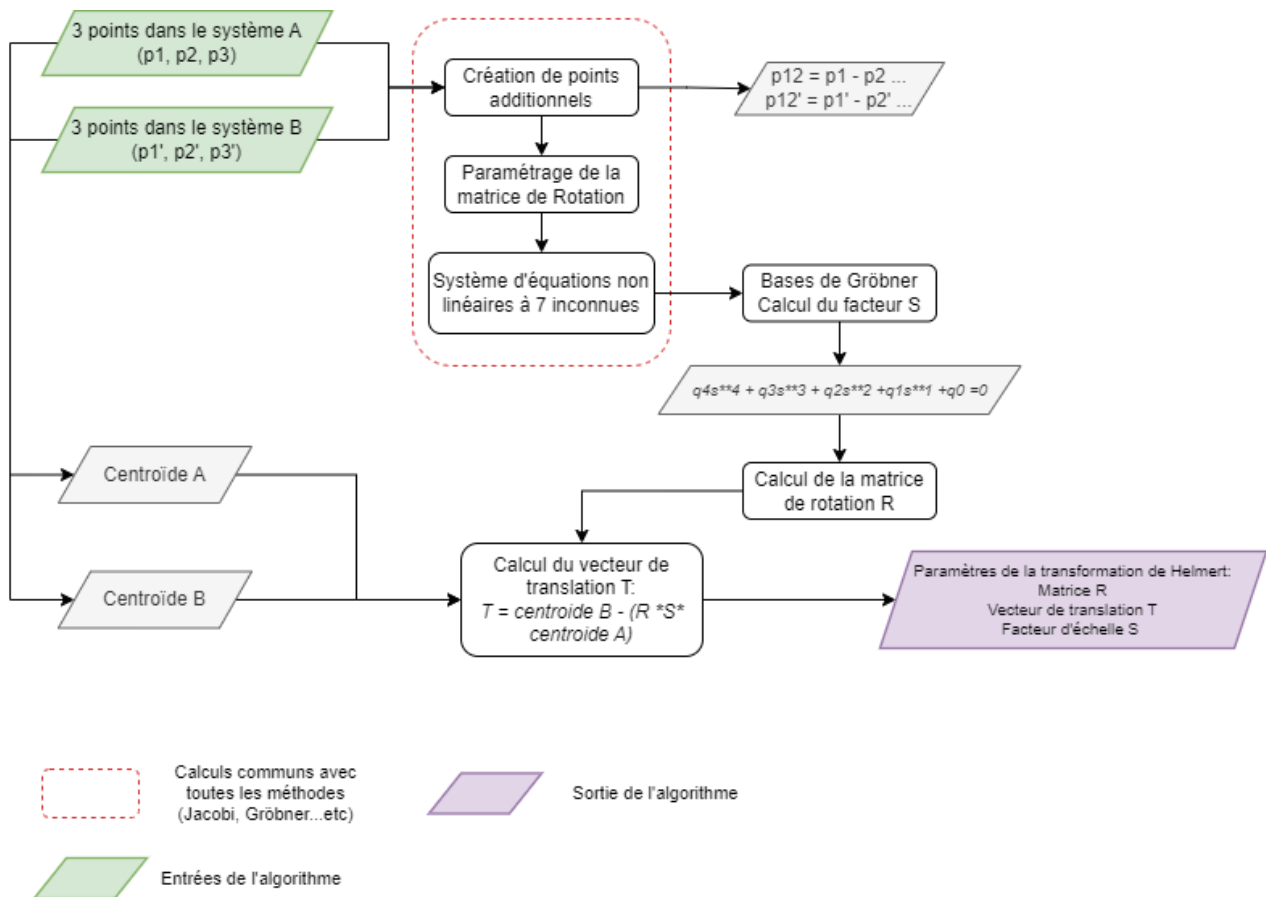


Figure 11 : Calcul des paramètres au sein de l'extension BimGeoref : utilisation des bases de Gröbner

Comme l'illustre la figure 11, le facteur d'échelle peut ainsi être obtenu avec la résolution d'une équation du quatrième degré. Les coefficients q4, q3, q2, q1 et q0 sont obtenus grâce à différents points additionnels. Ceux-ci sont correspondent à la soustraction des coordonnées entres elles et apportent ainsi plus de valeur connus dans les calculs suivants.

La racine minimale et positive de cette équation est alors le facteur d'échelle S. Les calculs détaillés sont visibles en annexe 5. La matrice de rotation est également calculée grâce à ces différents points. Enfin, le vecteur de translation est calculé en prenant en compte les centroïdes des points dans les deux systèmes de coordonnées ainsi que la matrice R et le facteur d'échelle. On obtient alors les différents paramètres de la transformation de Helmert.

### 2.2.3. Sauvegarde des couches aux formats ESRI Shapefile et Geopackage

Une fois les paramètres de la transformation obtenus, on peut l'appliquer à l'ensemble des couches du projet et ainsi modifier l'ensemble de la maquette. La sauvegarde des couches dans un nouveau système de coordonnées s'effectue par itération de l'ensemble des sommets sur lesquels on modifie directement les coordonnées.

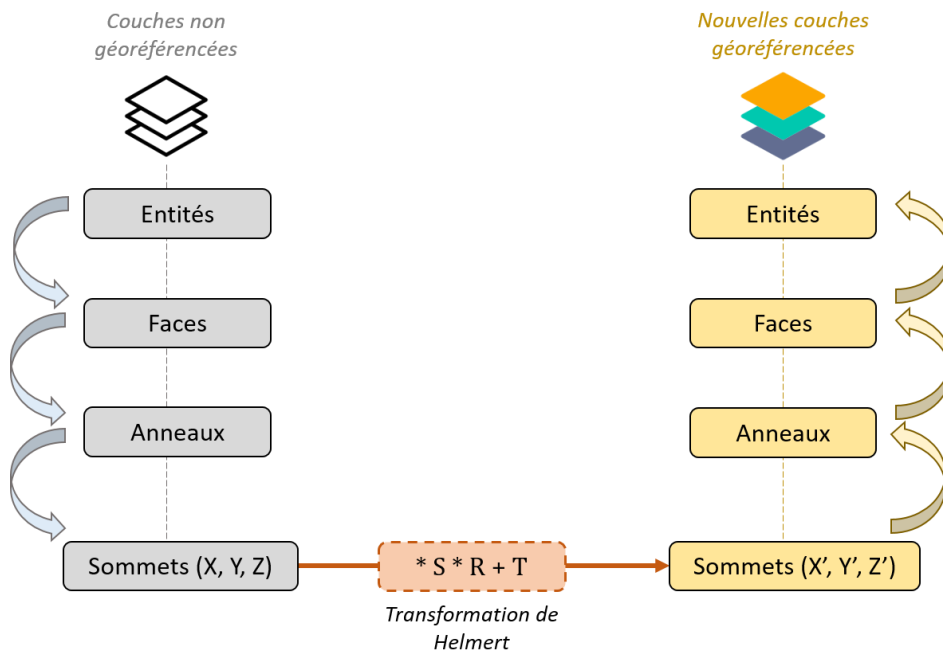


Figure 12 : Traitement des couches SIG dans l'extension BimGeoref

Comme illustré sur la figure 12, le traitement des couches QGIS pour leur géoréférencement se divise en deux parties. La première correspond à l'itération successive de la géométrie jusqu'à accéder aux coordonnées de chaque sommet. A partir de là, une nouvelle couche est recrée avec une modification des sommets auxquels on applique la transformation de Helmert. Les anneaux, les faces et les entités sont ensuite reconstruits de façon similaire à la couche de départ. Cela signifie que les champs et différentes données attributaires sont bien restitués sur les mêmes entités.

Il est important de préciser que selon le format de sauvegarde choisi, la librairie python utilisée pour ce traitement diffère. La sauvegarde en Geopackage nécessite l'utilisation de GDAL-OGR pour recréer des géométries de type Multipolygon Z. Concernant le Shapefile, c'est la librairie PyShp qui est utilisée pour recréer des géométries de type ESRI Multipatch.

La principale difficulté dans la création de ces géométries réside dans la détection et formation des ouvertures dans les entités qui correspondent généralement à des fenêtres. Des recherches pour implémenter au mieux la détection des anneaux extérieurs et intérieurs (« outer » et « inner rings ») ont été menées. Bien que les méthodes diffèrent entre le Shapefile et le Geopackage, les deux formats ont pu être traités et les ouvertures recrées correctement.

## 2.3. Prise en compte de l'expérience utilisateur

En parallèle, un travail sur les sécurités du plug-in et l'expérience utilisateur a été effectué. Contrairement au plug-in BimToGis, l'extension BimGeoref possède un nombre de valeurs et de boutons conséquent au sein de son Interface Homme Machine. Il a donc fallu ajouter un certain nombre de fonctionnalités et sécurités pour garantir une bonne saisie des points et le confort de l'utilisateur.

### 2.3.1. Fonctionnalités améliorant le confort d'utilisation

Le premier élément mis en place pour le confort d'utilisation concerne l'outil de saisie de points. Il correspond à un outil d'accroche des sommets qui vient accompagner l'utilisateur dans les points qu'il peut sélectionner. Cet outil reprend le fonctionnement de la digitalisation dans QGIS avec la possibilité de se raccrocher aux sommets les plus proches. Pour le cas de BimGeoref, une telle fonction garantit à l'utilisateur de cliquer sur le bon sommet.

La seconde fonctionnalité implémentée laisse le choix à l'utilisateur de prendre en compte les valeurs de hauteur de la couche. Cela correspond à la case à cocher « Sélection du Z ». En effet, le géoréférencement des couches 3D ne nécessite pas forcément de changer leur hauteur. Cet ajout permet donc de ne rentrer que les X et Y des 3 points et de lancer l'algorithme, le calcul du facteur d'échelle, des rotations et translation s'effectue de façon similaire.

Concrètement, si l'utilisateur décoche la sélection du Z, alors une constante (correspondant à la valeur Z minimale de la maquette) sera affectée pour l'ensemble des calculs. En revanche, si l'utilisateur souhaite faire une translation en Z pour surélever ou abaisser la maquette alors il doit laisser cette case cochée et rentrer les valeurs souhaitées.

Enfin, l'extension BimGeoref prend en compte la symbologie existante des couches de départ dans le traitement. Les modifications effectuées sur les maquettes 3D dans QGIS qu'elles soient une symbologie unique ou catégorisée selon un champ spécifique, sont restituées.

### 2.3.2. Sécurités du plug-in

À cela s'ajoutent un certain nombre de sécurités mises en place garantissant une bonne utilisation de l'extension. En effet, ce plug-in dispose de nombreuses zones dont il faut rentrer des valeurs. Il faut donc s'assurer de la bonne saisie des points. Ainsi, les coordonnées sélectionnées avec l'outil de saisie ne sont modifiables que par le biais d'une nouvelle saisie. Sinon, les coordonnées sont grisées.

De plus, l'algorithme de géoréférencement doit être lancé avec trois points dont les coordonnées sont différentes pour ne pas biaiser le calcul. Des vérifications ont été mises en place dans le plug-in pour vérifier l'ensemble des points. Si des points sélectionnés ou des nouvelles coordonnées sont identiques, alors le bouton de lancement du plug-in est désactivé. Un message d'erreur s'affiche alors dans QGIS invitant l'utilisateur à rentrer de nouvelles coordonnées.

## 2.4. Discussion

### 2.4.1. Bilan

- Les apports

Pour conclure, cette première partie du stage a permis la création d'un nouvel outil fonctionnel, mais encore au stade expérimental. L'apport à ce projet de recherche et développement est conséquent puisqu'un nouvel élément vient s'ajouter à l'extension BimToGis et leur complémentarité est totale. Les projets BIM au format IFC peuvent à présent être convertis sous la forme de différentes couches SIG (BimToGis) qui sont ensuite relocalisées et dimensionnées avec BimGeoref (figure 13). Dans le cadre de ce projet, ce travail permet d'avoir de nouveaux éléments pour manipuler de la donnée BIM au sein d'un SIG.

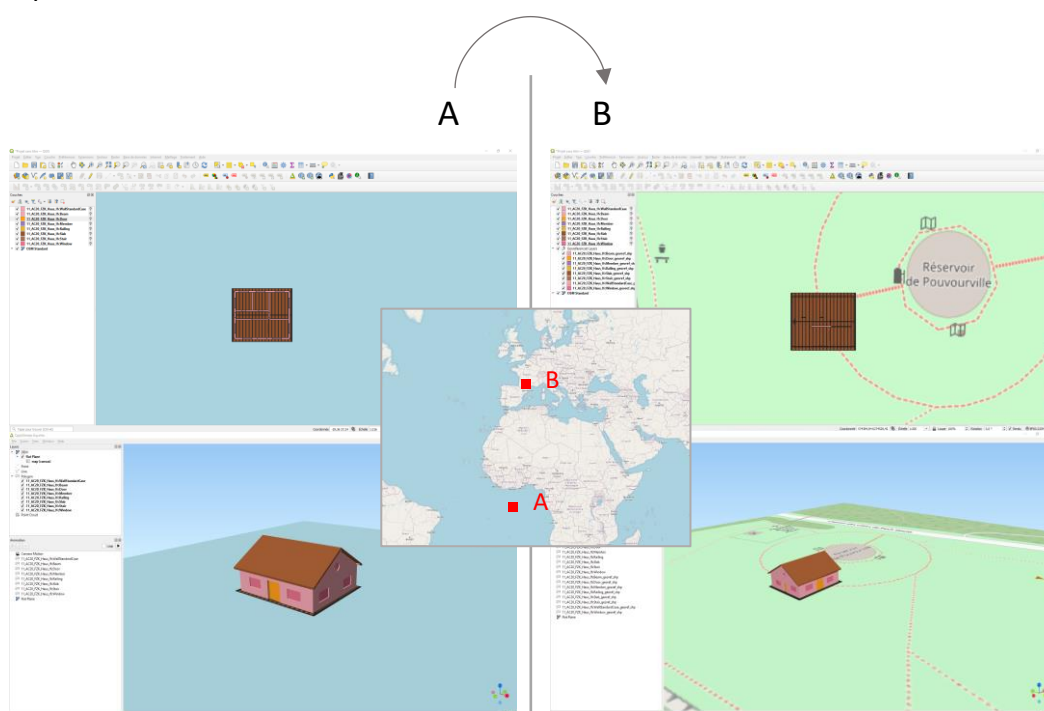


Figure 13 : Exemple d'application de BimGeoref : géoréférencement d'une maquette (initialement chargée en x:0, y:0) au niveau de Pech David. Exemple supplémentaire en annexe 6.

- Retour d'expérience

D'un point de vue plus personnel, les missions concernant la partie QGIS-Python ont été riches d'enseignements. J'ai pu découvrir le développement de plug-in QGIS au travers d'un objectif concret qui se situe au cœur de la géomatique. La création d'un outil, de son interface utilisateur jusqu'à son algorithme est quelque chose de très formateur pour comprendre les mécanismes en jeu.

Le développement de BimGeoref ne s'est pas fait sans difficultés. Certains points ont été particulièrement bloquants. C'est le cas de l'outil de saisie de points qui vient s'appliquer directement sur les couches QGIS. Activer et désactiver cet outil en permanence pour remplir respectivement les points voulus a pris un certain temps de développement. Le géoréférencement des couches a également été l'élément le plus long à mettre en place dans ce projet.

Il a fallu trouver les bonnes méthodes mathématiques pour calculer correctement les différents paramètres de la transformation géométrique et les appliquer ensuite sous python à l'ensemble des couches.

#### 2.4.2. Limites et perspectives d'évolution

D'un autre point de vue, le plug-in BimGeoref reste un outil expérimental. Dans le cas de transformations complexes, le calcul des paramètres de la transformation n'est pas toujours très précis, notamment sur le facteur d'échelle entre les deux systèmes de références où l'écart est important. Sur le même sujet, ces différents paramètres sont calculés actuellement avec la méthode des bases de Gröbner. C'est une solution qui s'avère efficace dans de nombreuses situations. En revanche, peu de tests ont été effectués sur des transformations complexes (plusieurs rotations, translations et changement d'échelle). Il faut donc se pencher sur la capacité de l'algorithme à prendre en compte ces éléments.

Comme perspective d'évolution de ce travail, plusieurs éléments peuvent être abordés. Le premier point serait d'ajouter des vérifications supplémentaires dans cette extension QGIS, notamment au niveau de l'algorithme au cœur du géoréférencement. Le calcul des paramètres de la transformation pourrait en effet être doté d'un calcul d'erreurs. Pour illustrer cela, on peut très bien imaginer qu'un utilisateur en rentrant les nouvelles coordonnées de chaque point se trompe parfois dans les valeurs. Or, si dans le calcul des paramètres on ajoute une évaluation de calcul avec un seuil d'erreur, cela permettrait d'avertir l'utilisateur et de bien vérifier les points saisis.

Enfin, le calcul des nouvelles coordonnées pour caler parfaitement les maquettes est parfois complexe. Des calculs prédéfinis pourraient donc être ajoutés à l'extension. Par exemple, on peut imaginer une liste déroulante avec plusieurs possibilités telles qu'une rotation en X de 90°, ou une réduction d'échelle de 50%. Si l'utilisateur rentre les 3 points et sélectionne une ces options, alors de nouvelles coordonnées sont automatiquement rentrées et calculées pour donner un aperçu à l'utilisateur. On se place ici dans une extension plus ouverte aux utilisateurs et qui les accompagne dans la manipulation des maquettes numériques.

En guise de conclusion de cette partie nous pouvons souligner le fait que les projets BIM en sortie du plug-in peuvent à présent être redimensionnés selon les coordonnées souhaitées. Grâce à BimGeoref, la chaîne de traitement des maquettes numériques dans QGIS s'enrichit : une première phase de conversion des maquettes (BimToGis) s'accompagne maintenant d'une seconde phase où l'on attribue cette fois-ci un Système de Coordonnées de Référence.

Il reste tout de même des axes d'amélioration concernant le développement de cet outil dont le contrôle et les tests du géoréférencement lorsque celui-ci est complexe.



## 3. Plateforme web 3DGS

---

La deuxième mission principale de ce stage correspond à l'amélioration et au développement de la plateforme web 3DGS. Comme décrite précédemment ([1.5.2](#)), cette plateforme permet de visualiser des maquettes numériques 3D, d'interagir avec elles et de faire différents traitements. Ce peut être des découpes, des requêtes, une vision des arrêtes et bien d'autres outils.

Ainsi, l'objectif est d'apporter de nouveaux éléments sur cette plate-forme. Une première partie concernera le développement et l'amélioration de fonctionnalités existantes. Ensuite, il s'agira d'explorer les nouvelles fonctionnalités mises en place sur l'application et les recherches menées pour les mettre en place.

### 3.1. Poursuite des développements existants

Cette première partie détaille le développement de fonctionnalités existantes avec dans un premier temps une amélioration de la sauvegarde des projets dans 3DGS et par la suite la mise en place d'un outil de création de parcours sur la scène 3D.

#### 3.1.1. Gestion des projets

Il est essentiel de mettre en place une gestion des projets au sein de 3DGS afin de garantir la fonctionnalité du Web-SIG, étant donné que cette application propose divers outils de traitement. En effet, si l'utilisateur ne peut pas sauvegarder les modifications effectuées sur les maquettes et les recharger par la suite, l'utilité de la plateforme en est mise à mal. Initialement, une gestion des projets a déjà été mise en place au sein de l'application. Il s'agit néanmoins d'une sauvegarde dans la mémoire locale (ou « local storage ») de l'appareil. Ainsi plusieurs projets peuvent être sauvegardés et chargés par la suite pour récupérer l'ensemble des informations dans la scène 3D.

Néanmoins, ce fonctionnement présente certaines limites. La première est le nombre de projets pouvant être sauvegardés dans la mémoire locale. Cela correspond à une dizaine de projets. Au-delà, la mémoire locale est saturée. La seconde limite est le partage potentiel de projets entre appareils qui n'est pas possible avec cette méthode.

L'objectif est donc de mettre en place un système de gestion de projets entre différents appareils. Cela passe par la création de fichiers externes. Étant donné qu'un système de sauvegarde/chargement de projets était déjà opérationnel, celui-ci a été repris pour le développement. Le format de fichier choisi est le JSON (JavaScript Object Notation).

- Le format JSON

Le format JSON est un format de données léger et largement utilisé pour l'échange de données entre applications. Il est basé sur une structure de données constituée de paires clé-valeur, où les données sont organisées en objets et tableaux. Les valeurs peuvent être des nombres, des chaînes de caractères, des booléens, des tableaux, d'autres objets JSON.

Il est devenu un format très utilisé dans la programmation web, car il permet aux applications de différents langages de communiquer facilement entre elles. Le JSON est également largement utilisé dans le stockage et la transmission de données structurées.

- **Intégration dans 3DGS**

Dans le cadre du développement de 3DGS, ce format permet de regrouper l'ensemble des informations du projet. Il s'agit notamment des paramètres de la scène 3D et des couches 3D. On retrouve ainsi la position et direction de la caméra mais également le nom des couches Geoserver chargées, leurs symbologie et les différentes requêtes sauvegardées (figure 14).

La mise en place d'une gestion des projets par fichiers .json passe dans un premier temps par la possibilité d'exporter le projet pour créer l'ensemble du fichier. Une fois le fichier exporté, celui-ci peut être chargé dans l'application. Dans ce cas, les différentes maquettes sont chargées progressivement et la caméra se positionne à l'endroit où elle était présente. Un exemple de fichier .json après l'export est visible en annexe 7.

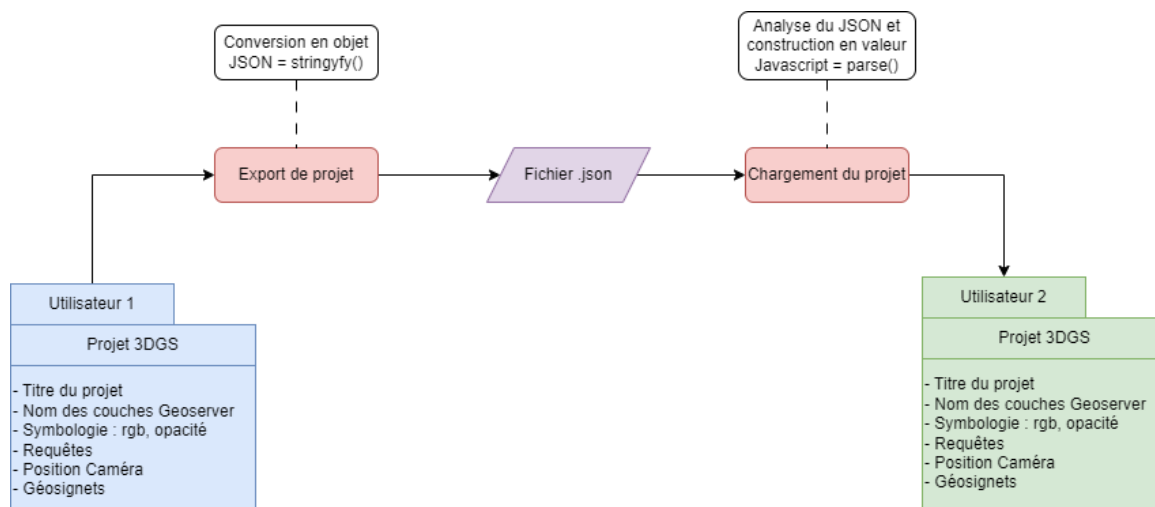


Figure 14 : Mise en place de la gestion par fichiers projets au sein de 3DGS

Cette fonctionnalité est en réalité le fil conducteur des autres missions concernant cette partie web. Avec cette gestion des projets par fichiers externes, l'ensemble des fonctionnalités ajoutées à l'application doivent être incorporées à la fois dans l'export et également dans l'import de fichiers .json.

### 3.1.2. Animations sur la scène 3D

Le second axe de développement concerne le widget « parcours » présent dans l'application. Ce dernier permet d'animer la scène 3D selon un parcours prédéfini. En lançant le parcours, la caméra suit un ensemble de points prédéfinis et fixe une direction. Jusqu'à présent les parcours devaient être rentrés manuellement dans le code source de l'application. Des recherches ont donc été entreprises lors de ce stage pour découvrir les possibilités avec la librairie ThreeJS en termes de création personnalisée de parcours.

Cette mission s'inscrit dans le cœur de 3DGS avec une fonctionnalité qui permettrait encore plus d'interagir, de visualiser et manipuler les maquettes numériques 3D. Cette approche accroît l'expérience utilisateur en la rendant plus immersive. Nous pouvons à présent détailler l'approche méthodologique retenue pour développer cet outil.

- Création automatique d'un parcours standard

Un parcours est désormais automatiquement créé au chargement d'une maquette, il s'agit d'une animation « 360 degrés » qui permet d'en faire le tour et de l'inspecter. Le parcours est calculé en fonction de la boîte englobante (ou « bounding box » en anglais) de la maquette.

Cette fonctionnalité permet de toujours avoir un parcours créé dans le widget. L'utilisateur peut alors directement le lancer et visualiser le résultat donnant finalement un aperçu très simple et rapide des possibilités de l'outil. Nous pouvons à présent détailler la mise en place technique de l'outil d'édition de parcours.



Figure 15 : Illustration du parcours automatique "360 degrés" pour deux maquettes de taille différente, visible en rouge.

- Outil d'édition

L'outil d'édition est activé lorsque l'on clique sur le bouton « Créer un parcours », visible en figure 15. S'ouvre alors le panneau d'édition (figure 16). Ce panneau présente différentes fonctionnalités qu'il convient de détailler.

On retrouve dans un premier temps un paramétrage général (figure 16, n°1) sur le nom du parcours, la vitesse de l'animation et si l'on souhaite centrer la caméra sur un endroit spécifique. La fonctionnalité centrale de l'outil réside dans la saisie des points (2). Lorsque la saisie est active, l'utilisateur peut cliquer sur les différents objets de la scène 3D, des sphères sont alors créées sur la scène et leurs coordonnées sont automatiquement rentrées dans un tableau (3). Il est alors possible de modifier les coordonnées de l'ensemble des sphères ou bien de modifier chaque sphère, voire de les supprimer. On a ici un outil qui permet de personnaliser entièrement le parcours et d'être précis.

Une partie sur la visualisation et l'aperçu du parcours a pu ensuite être implémentée (4). Lorsque le bouton de visualisation est sélectionné (5), le parcours est visible et dynamique, cela signifie que si l'on ajoute ou supprime des points le parcours sera directement modifié en fonction.

Il est également possible à tout moment lors de l'édition, de lancer un aperçu du parcours pour voir si la vitesse ou l'ordre des points correspond aux attentes de l'utilisateur. Celui-ci peut alors modifier le parcours en fonction de l'aperçu.

La dernière partie (5) englobe la possibilité de supprimer l'ensemble des points de la scène et du tableau et d'enregistrer le parcours. Une fois le parcours enregistré, celui-ci vient s'ajouter à la liste des parcours au niveau du widget. Sur ce dernier point, il est important de préciser que la sauvegarde de projet par fichier JSON (énoncé en [3.1.1](#)) prend en compte les nouveaux parcours créés avec l'ensemble de leurs paramètres.

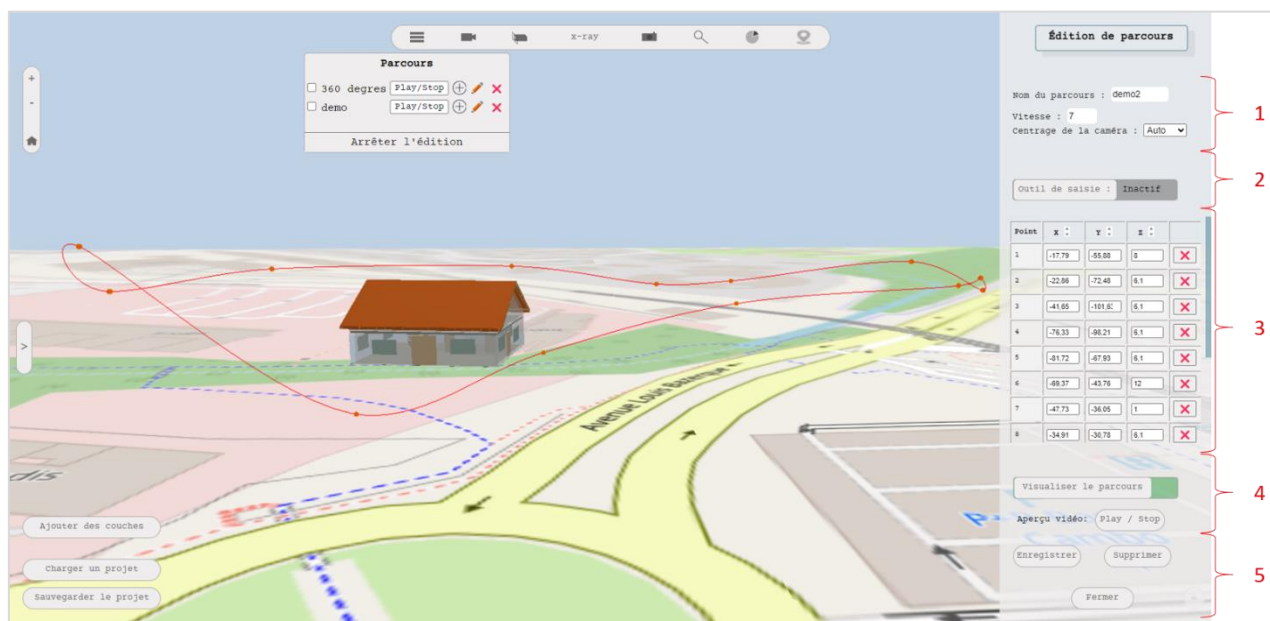


Figure 16 : Exemple d'édition de parcours sur la plateforme 3DGS

Au sein de ce widget la gestion des sphères créés au clic est un point central. À chaque création de sphère, un identifiant unique lui est attribué et cet identifiant est reporté dans le tableau. Afin de bien situer l'utilisateur dans son parcours, l'identifiant des sphères est incrémenté de 1 pour chaque saisie. Si l'utilisateur supprime une sphère en particulier, l'ensemble des sphères ayant un identifiant supérieur à cette sphère se voient décrétement de 1. En parallèle, les identifiants rentrés dans le tableau sont également actualisés. Ainsi, les identifiants des points de parcours sont toujours mis à jour et se suivent dans un ordre de 1 à n sans discontinuités, peu importe les suppressions ou les ajouts.

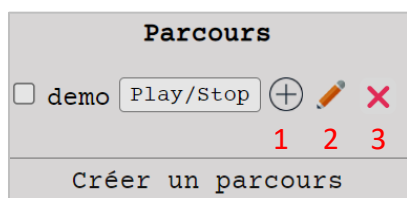
Par exemple, si un utilisateur dispose de 6 sphères dans son tableau et qu'il supprime la sphère n°4 alors les sphères 5 et 6 passeront respectivement à l'identifiant 4 et 5. La mise en relation des identifiants des sphères avec le tableau de manière dynamique a été un point central dans le développement de cet outil.

Afin que l'utilisateur retrouve à chaque fois la sphère correspondante entre le tableau et la scène 3D, une fonction de surbrillance au survol a été développée. Lorsque l'on passe la souris sur une ligne du tableau, la sphère avec l'identifiant correspondant passe de l'orange au jaune. Il est alors possible de modifier les coordonnées de cette sphère ou de la supprimer.

- Gestion des parcours existants

En complément de la mise en place du widget parcours, une gestion plus poussée des animations existantes a été implémentée. En effet, il n’y avait aucune possibilité de supprimer ou modifier les parcours existants sur l’application auparavant. À présent, l’utilisateur peut les modifier, dupliquer, ou bien les supprimer au travers de différentes icônes visibles ci-dessous (figure 17).

Pour chaque parcours, la duplication ou modification va permettre de retrouver l’ensemble des points et remplir le panneau d’édition avec toutes les informations. La différence entre une création simple et une modification réside dans la sauvegarde. Si l’utilisateur se trouve en modification, alors l’enregistrement écrasera le parcours existant et n’en créera pas de nouveau.



1 : Dupliquer le parcours

2 : Modifier le parcours

3 : Supprimer le parcours

Figure 17 : Gestion des parcours déjà enregistrés

## 3.2. Exploration de nouvelles fonctionnalités

À présent il va s’agir de détailler les travaux de recherche menés sur de nouvelles fonctionnalités. Dans un premier temps, la mise en place d’une élévation puis des recherches sur la possibilité d’incorporer du Lidar HD dans l’application.

### 3.2.1. Mise en place d’une élévation au sein de l’application

L’application 3DGS disposait jusqu’à maintenant d’un fond de carte OSM au niveau 0 pour charger les différentes maquettes. Un travail de recherche a donc été entrepris pour charger une élévation fidèle à la réalité terrain au sein de la plateforme web. Cela permettrait d’étudier et de visualiser les maquettes dans un contexte géographique qui leur est propre. En effet, analyser les maquettes numériques sur une plateforme où le relief n’est pas présent ne rend pas compte de la réalité terrain.

- Création d’une élévation

La génération d’un service d’élévation au sein de 3DGS nécessite d’avoir tout d’abord d’avoir une donnée initiale sur laquelle s’appuyer. La BD Alti de l’IGN à 25 mètres a donc été retenue pour ce travail avec l’utilisation d’une dalle MNT (Modèle Numérique de Terrain) sur le bassin toulousain. Le choix de cette donnée est principalement dû au fait qu’elle est open-source et très facile d’accès. Concernant la résolution à 25 mètres, il est plus judicieux de prendre une faible résolution pour faire des tests et vérifier le fonctionnement du code.

À partir de cette dalle MNT à 25 mètres, la génération 3D de l'élévation s'est effectuée en plusieurs étapes (figure 18). Dans un premier temps le raster d'élévation a pu être lu via la librairie Geotiff.js. L'ensemble des informations sur le MNT a donc été traité. Il s'agit notamment des coordonnées des pixels, leurs valeurs, la résolution et les dimensions de l'image. À partir de là, une géométrie plane fidèle aux dimensions du MNT a été générée avec ThreeJS.

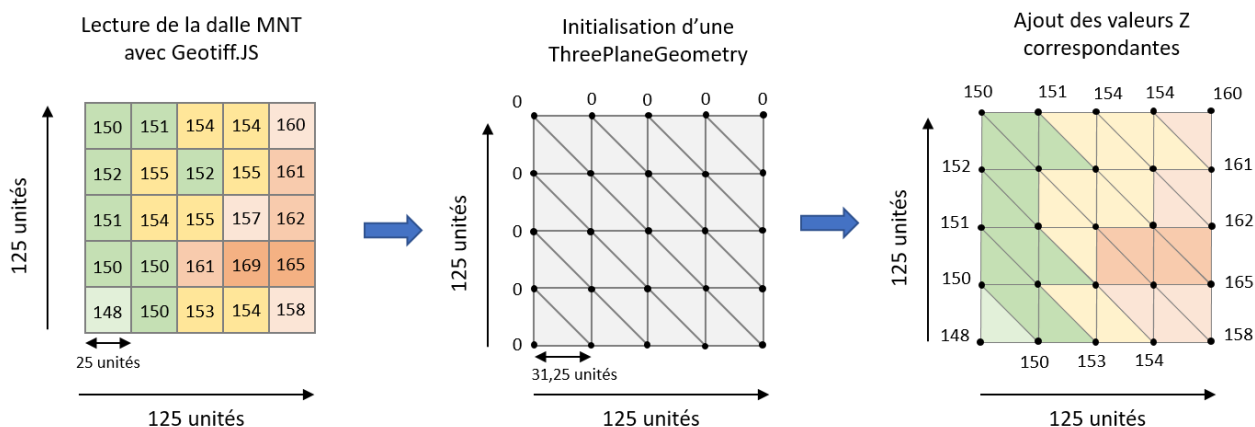


Figure 18 : Chargement d'une dalle MNT et conversion en ThreePlaneGeometry sur l'application

Cette géométrie, nommée « ThreePlaneGeometry », possède certaines spécificités. Elle est définie par un maillage de points, formant des triangles qui composent la surface du plan. Chaque triangle est défini par trois points. Ce qui signifie que plus il y a de points, plus la surface est détaillée. L'intérêt d'une telle structure est d'avoir une surface continue alors même que les données de base sont des pixels dont la résolution est de 25 mètres.

Dans le cas du MNT, l'élévation sur la scène 3D est générée en deux temps. Une surface plane uniforme de hauteur 0 est créée selon la largeur et la hauteur du MNT et selon un maillage de points correspondant au nombre de pixels sur le raster. Ainsi, la surface 3D contient le même nombre de points que le raster a de pixels. A partir de cette géométrie l'ensemble des points de la géométrie plane est itéré. Chaque sommet de cette ThreePlaneGeometry va récupérer la valeur du pixel qui lui correspond et donc la valeur d'élévation. On obtient alors une élévation fidèle aux données terrain (figure 19).

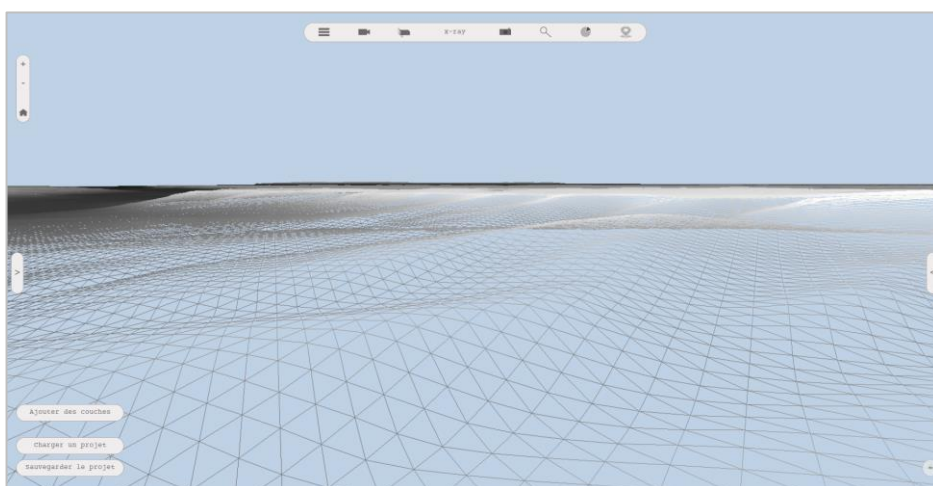


Figure 19 : Visualisation de l'élévation sous la forme d'une ThreePlaneGeometry

Bien évidemment, une certaine approximation réside dans la modélisation du MNT avec cette méthode mais les écarts de précision diminuent de manière croissante avec l'augmentation du nombre de points. Dans l'exemple sur la figure 18, l'écart entre les points est de 31.5 unités, ce qui est bien plus que 25 unités. Cependant, si l'on prend un MNT de 12 500 unités de largeur alors l'écart entre les points sera dans ces cas-là infiniment proche de 25 unités.

- Rendu visuel

Une fois l'élévation chargée, un travail sur son rendu visuel a été réalisé. Un contrôle dans le panneau des couches a été mis en place avec des palettes prédéfinies (figure 20), des couleurs uniques et une opacité. L'utilisateur peut alors choisir différentes options pour la symbologie du service altimétrique, l'ensemble des sommets est actualisé après chaque modification.

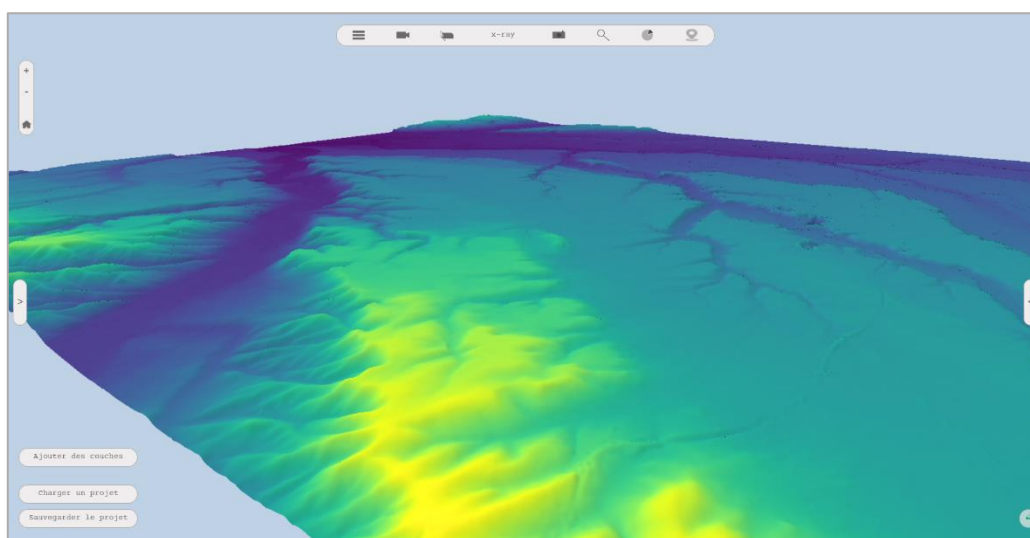


Figure 20 : Colorisation de l'élévation selon la palette Viridis.

Une exagération du relief a également été implémentée pour accentuer l'écart d'altitude entre certains espaces (annexe 8). La valeur d'élévation de chaque sommet est alors multipliée par la valeur souhaitée par l'utilisateur.

Enfin, des recherches sur la mise en place de textures telles qu'un fond OSM (OpenStreetMap) ont été entreprises mais sans succès, seule une image est chargée pour donner un aperçu de ce qui pourrait être visible. Le principal verrou pour implémenter le fond OSM est le rafraîchissement des images et de leur précision selon leur niveau de zoom. C'est une solution faisable mais qui n'a pas été abordée au vu de sa complexité.

- Ajustement automatique des maquettes sur l'élévation

L'autre enjeu sur l'incorporation d'une altimétrie est de pouvoir mettre à niveau les éléments 3D lors de leur chargement. En effet, les maquettes étaient jusqu'alors chargées sur le fond de plan OSM au niveau 0. À présent, lorsque l'utilisateur souhaite charger une élévation, les maquettes déjà chargées sont positionnées sur l'élévation. Pour réaliser cela, j'ai pu m'appuyer sur les fonctionnalités de ThreeJS pour estimer l'élévation moyenne au niveau de la maquette. Le principe se divise en plusieurs étapes.

Tout d’abord, la boîte englobante en X et Y est calculée autour de la maquette. À partir de cette même zone, un certain nombre de valeurs d’élévation sont récupérées. La moyenne de ces valeurs est ensuite calculée et la maquette est alors positionnée sur cette valeur.

Les valeurs d’élévation sont obtenues grâce au composant Raycaster de ThreeJS. Ce composant permet de détecter quels objets 3D sont intersectés par un rayon lancé depuis un point spécifique dans l'espace 3D. Dans le cadre de ce calcul d’élévation, les rayons sont paramétrés pour ne récupérer que les coordonnées du point qui s’arrête au niveau du modèle numérique de terrain.

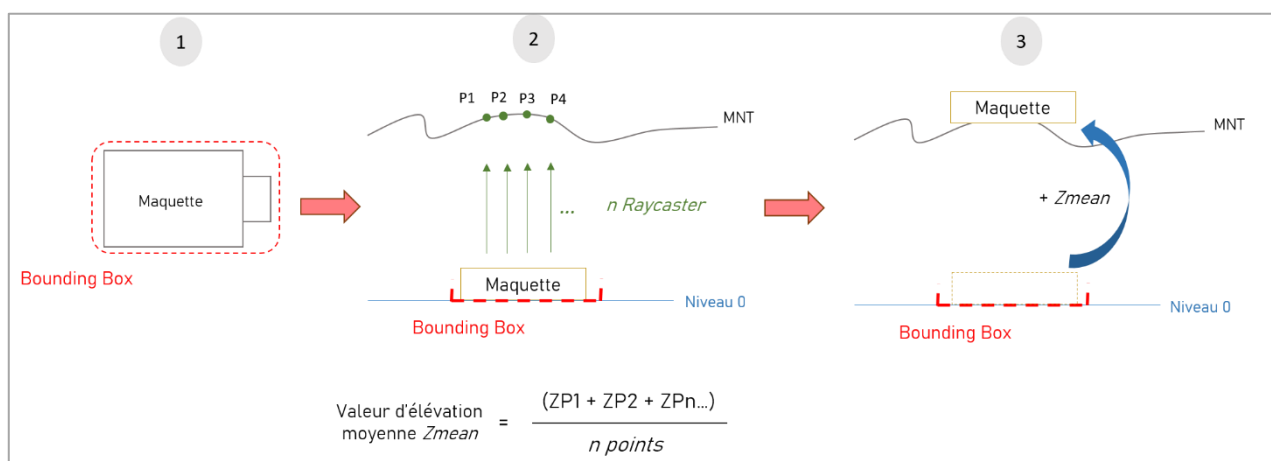


Figure 21 : Ajustement des maquettes sur l’élévation dans 3DGS

Le développement de cet outil permet désormais de positionner la maquette de la manière la plus précise sur l’élévation. L’ajustement de la maquette est appelé lorsqu’on ajoute une élévation ou bien lorsque l’on ajoute des éléments 3D et que l’élévation est déjà chargée. Une fonctionnalité supplémentaire permettant directement à l’utilisateur de caler la maquette au niveau 0 ou sur l’élévation serait très intéressante.

### 3.2.2. Intégration de données Lidar

Le dernier volet concernant le développement de la plate-forme web 3DGS est consacré aux recherches menées sur l’intégration de données Lidar. Dans le contexte actuel de l’arrivée de nombreuses données Lidar en libre accès, l’objectif est de voir ce qu’il est possible de faire avec ces données au sein de l’application et de faire une mise en perspective avec des maquettes BIM. En effet, depuis 2019, l’IGN a lancé une grande campagne pour couvrir l’ensemble du territoire métropolitain en Lidar haute définition. Cette étude part donc de cette donnée comme source.

- Données et bibliothèques existantes

Nous pouvons à présent aborder les outils open-source à disposition pour traiter des nuages de points sur des plateformes web. A l’heure actuelle peu d’outils sont efficaces pour traiter massivement ce type de données. La principale caractéristique des nuages de points est leur volume considérable. Une dalle Lidar HD de l’IGN fait en effet plus de 50 Mo en format compressé LAZ et plus de 500Mo en format décompressé LAS. Les données classées peuvent monter jusqu’à 1 Go avec un nombre de points parfois supérieur à 30 millions.



Il existe différentes bibliothèques JavaScript déjà développées permettant de lire des nuages de points :

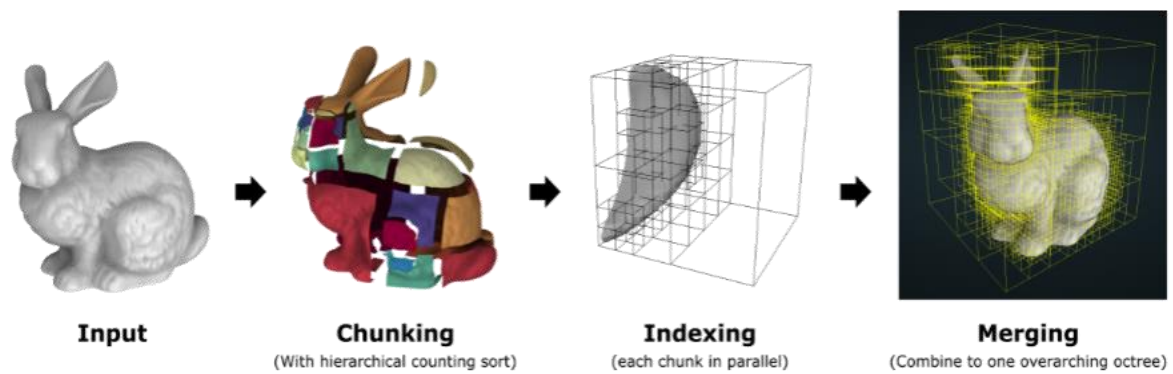
- ❖ ThreeJS est en mesure de charger des nuages de points dans une scène 3D bien que ce ne soit pas sa principale fonction. En effet, des ajouts ont été effectués pour charger des nuages de points et les lire tels que le « PCDLoader ». Cependant, les possibilités en termes d'interaction et de manipulation sont limitées.
- ❖ CesiumJS est avec ThreeJS la bibliothèque la plus efficace pour la visualisation d'objets au travers d'une scène 3D. Sa capacité à charger des nuages de points est nettement plus puissante que ThreeJS mais reste limitée lorsque le nombre de points dépasse la centaine de milliers.
- ❖ Plasio est une bibliothèque JavaScript qui permet de charger et visualiser spécifiquement des données Lidar. Elle permet de nombreuses fonctionnalités en termes de modification de l'apparence du nuage de points mais dispose de peu de fonctionnalités sur son traitement.
- ❖ PotreeJS est une bibliothèque open-source spécialement conçue pour le chargement de données Lidar. La bibliothèque dispose de son propre format basé sur l'indexation du nuage de points. Un convertisseur « PotreeConverter » est également disponible avec la bibliothèque pour convertir les données LAZ/LAS au format Potree. Afin de charger rapidement des centaines de milliers de points, voire des millions, PotreeJS est la bibliothèque la plus adaptée.

Dans le cadre de ce travail sur la visualisation de données Lidar haute définition, la bibliothèque PotreeJS a été retenue et ce, pour plusieurs raisons. Celle-ci englobe dans son code différents modules dont ThreeJS pour la scène 3D. Dans l'idée d'améliorer la plateforme 3DGS qui utilise déjà ThreeJS, on comprend bien la complémentarité qui pourrait se jouer. A cela s'ajoute le fait que Potree est une bibliothèque aujourd'hui assez bien documentée, avec de nombreux exemples et des outils déjà préconçus qui se prennent rapidement en main. De plus, PotreeJS est une bibliothèque très puissante pour lire des nuages de points denses, en raison de sa capacité à segmenter les données en différents sous-ensembles.

- L'indexation de points de PotreeJS

Comme détaillée précédemment, cette bibliothèque lit des nuages de points qui sont eux-mêmes dans un format particulier : le format Potree. Pour parvenir à ce format, les données lidar LAS/LAZ peuvent être converties via un convertisseur : PotreeConverter. C'est une application C++ permettant d'indexer massivement le nuage de points en différents sous-ensembles (Schütz, M. *et al.*, 2020).

Comme visible sur la figure 22, l'indexation du nuage de points de convertisseur s'effectue selon plusieurs étapes. Une fois les données de nuages de points chargées, le convertisseur divise l'espace en une structure hiérarchique appelée « octree ». L'octree partitionne l'espace en blocs de différentes résolutions, ce qui permet d'optimiser le chargement et la visualisation progressive des données. L'octree forme également la base de l'indexation spatiale pour une recherche rapide et efficace des points lors de la visualisation (Schütz, M. *op cit.*).



**Figure 1:** *Processing pipeline: An out-of-core hierarchical counting sort quickly generates chunks of suitable size which can then be indexed in parallel, and eventually merged into a single octree.*

Figure 22 : Principe de l'indexation du nuage de points au format Octree ; Source : Schütz, M., 2020

Le PotreeConverter crée des fichiers au format Potree, qui sont organisés en une structure de dossiers contenant les données spatiales, les niveaux de détail, les métadonnées et les fichiers d'index nécessaires pour la visualisation en 3D.

Une fois les données LiDAR converties au format Potree, celles-ci peuvent être chargées et visualisées sur le web via un « PotreeViewer ». Ce visionneur dispose de nombreux paramètres pour le traitement et l'affichage des nuages de points. On retrouve ainsi un dispositif de rendu implémenté directement dans le code source de Potree. Différentes scènes ThreeJS sont contenues dans le visionneur, avec notamment une scène globale où l'on peut ajouter des modèles 3D, différents types de géométries.

Une autre scène est spécifiquement présente pour contenir le nuage de points au format « PointCloudOctree ». Au sein de cet objet on retrouve les différents groupes de points indexés eux-mêmes en sous-groupes. Cela correspond aux résultats du convertisseur de Potree qui a pu découper les données.

Ici, il est important de préciser que l'affichage des données au travers d'un Potree Viewer rend plus complexe la prise en main du nuage de points directement sur 3DGS. Bien que la scène soit basée sur ThreeJS, les nombreuses modifications faites avec le visionneur Potree impliquent de modifier en profondeur le code. En effet, il est très difficile d'accéder aux données de la scène 3D sans passer par les fonctionnalités du visionneur Potree.

Le choix de manipuler le lidar sur une plateforme de test a été fait. L'objectif étant de voir ce qu'il est possible de faire et non pas de directement l'intégrer dans la plateforme.

- Premiers résultats

Des premiers tests ont donc été entrepris à partir d'un jeu de données Lidar HD classé de l'IGN. Le jeu de données a été découpé pour faire environ 400 000 points. Ce nombre est bien inférieur au volume total d'une dalle entière de l'IGN et se justifie par le fait qu'on souhaite dans un premier temps exploiter la librairie Potree et ses fonctionnalités mais pas forcément tester les limites de chargement, qui dépendent en grande partie du serveur sur lequel la plateforme est hébergée.

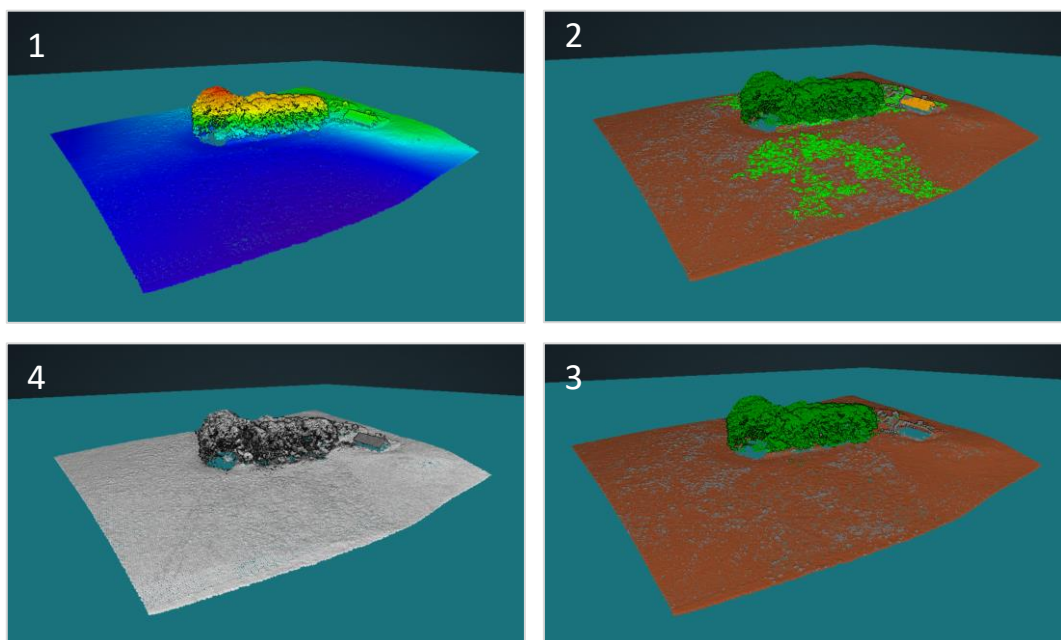


Figure 23 : Visualisation de nuages de points LiDAR sur le serveur de 3DGS : symbologie selon l'élévation (1), la classification (2), la sélection d'attributs de classe (3), et intensité (4)

Comme visible sur la figure 23, des premiers résultats sur la symbologie du nuage de points ont été obtenus. Les données peuvent être visualisées selon l'élévation des points, l'intensité, leur classification. Une sélection de certaines catégories de points (types de végétation, sol, bâtiment) est également possible. Il est important de préciser ici que l'ensemble des attributs contenus dans les fichiers lidar initiaux sont récupérés et peuvent être traités par Potree. Ainsi, on peut prendre en compte la colorisation vraie couleurs si les données RGB sont stockées dans le fichier lidar.

- **Chargement massif de points**

Enfin, différents tests ont été effectués en fin de stage, afin d'estimer le nombre maximum de points pouvant être pris en charge par PotreeJS. Ces tests ont été réalisés avec différents échantillons de données lidar haute définition classées, à la fois sur un serveur local et sur le serveur de test qui contient 3DGS.

Taille du fichier (.las)	Nombre de points	Statut de chargement : serveur de test	Statut de chargement : serveur local
210.266 Mo	6 151 773	Complet	Partiel
75.455 Mo	2 207 581	Complet	Partiel
34.29 Mo	1 003 086	Complet	Partiel
28.47 Mo	832 846	Complet	Partiel
13.70 Mo	400 562	Complet	Complet

Tableau 1 : Bilan des tests de chargement réalisés entre le serveur local et le serveur test

Le résultat de ce test est visible sur le tableau 1. Comme on peut le voir, le résultat est très distinct. Le chargement des nuages de points sur le serveur local est bien moins efficace que le serveur mis en place pour 3DGS. Cela peut être dû à plusieurs raisons liées à la performance du serveur local et de sa capacité à gérer la mémoire.

Toujours est-il que les performances du convertisseur Potree et de son visionneur ont pu être démontrées sur la plateforme de test. Le nombre de points maximum lu par Potree correspond à plus de 6 millions de points. Au vu du volume des fichiers convertis (plus de 200 Mo), des tests de plus grande envergure nécessiteraient la mise en place d'un back-end pour pouvoir stocker et lire de fichiers contenus dans une base de données externes.

### 3.3. Discussion

Nous pouvons à présent revenir sur les grands éléments de cette partie concernant 3DGS et discuter de la suite de ce travail.

#### 3.3.1. Bilan

- Les apports

Tout d'abord, les différents travaux durant le stage concernant la plateforme 3DGS se sont dissociés en deux sous-ensembles. J'entends par là qu'il y'eut dans un premier temps la continuité de missions déjà débutées auparavant et par la suite le lancement de recherches sur de nouvelles fonctionnalités.

Les apports au projet de convergence BIM-SIG dans l'open-source sont nombreux. On peut parler du fait qu'on peut désormais charger un modèle numérique de terrain fidèle aux données de la zone d'étude sur lequel les maquettes 3D peuvent être ajustées et étudiées dans un nouveau contexte. Ces données d'élévation peuvent être modifiées de la même manière que les autres couches 3D avec des colorisations en fonction de l'altitude, et la possibilité d'augmenter le contraste entre les zones hautes et basse.

En complément, un nouvel outil de parcours s'ajoute à la plateforme 3DGS. Cet outil permet à l'utilisateur de créer et manipuler de façon dynamique différentes animations autour des maquettes. Cela permet de faire une inspection des maquettes ou toutes autres données présentes dans l'application, de montrer certains éléments spécifiques aux bâtiments. On comprend bien ici l'intérêt de cet outil de parcours qui permet une immersion directe dans la maquette.

La plateforme a également gagné en termes de partage de données puisqu'il est désormais possible d'exporter l'entièreté des projets et de leurs modifications dans un seul fichier .json. Ce même fichier peut être diffusé à tout un ensemble d'utilisateurs qui pourront eux-mêmes bénéficier des modifications. Les travaux sur l'élévation et les parcours ont été intégrés en parallèle et sont contenus dans cette gestion des projets.

Ce sont autant des réalisations qui permettent à la plateforme de gagner en fonctionnalités, de poursuivre sa progression en termes de modélisation 3D. Les possibilités techniques sont enrichies avec des outils entièrement libres.

Enfin, il est également utile d'aborder les recherches faites sur l'intégration de données Lidar. Suite à une première phase sur les outils existants, le choix s'est finalement arrêté sur PotreeJS. Une plateforme annexe à 3DGS a été créée pour servir de test. Différents nuages de points ont été chargés et des premières analyses sur la visualisation (couleurs, sélection attributaire) ont été entrepris.

- Retour d'expérience

Je peux en venir à un retour d'expérience sur cette seconde phase du stage concernant cette fois-ci du développement web. Il est important de préciser que contrairement à la partie QGIS, il a fallu cette fois-ci reprendre un travail existant après deux stages sur la question. Un certain temps de compréhension de la plateforme, de ses outils, leur articulation dans le code source, a été nécessaire.

Les apports en termes de compétences techniques ont été nombreux, notamment sur la manipulation du langage JavaScript et plus généralement sur le développement web. Les missions impliquaient une certaine liberté sur les outils et les méthodes à utiliser pour y parvenir. C'est quelque chose que j'ai pu apprécier, notamment puisque cela impliquait de créer les éléments en partant de la recherche de solutions techniques jusqu'à la leur réalisation, leurs tests et validations.

### 3.3.2. Limites et perspectives d'évolution

- Missions réalisées

Des éléments ont bien évidemment été bloquants. Nous pouvons à présent les détailler et soumettre des perspectives d'évolution quant au travail réalisé.

Dans un premier temps, il convient de revenir sur la génération de l'élévation. En effet, l'objectif initial était de charger l'élévation selon un flux WMS de la BD Alti de l'IGN. Néanmoins la lecture du flux ne permettait pas de lire la valeur des pixels et de régénérer l'élévation. Un choix a donc été fait de générer une élévation à partir d'un raster sur le serveur pour tout de même poser des bases dans la construction du service altimétrique.

Cette fonctionnalité est donc encore loin d'être finalisée. On peut par exemple imaginer l'incorporation de l'élévation directement depuis un Geoserver et non pas sur le serveur pour diminuer la quantité de données stockées sur celui-ci.

Puis, la mise en place de dalles OSM sur l'élévation a été un point bloquant pour cette mission, la possibilité de charger différentes textures sur le modèle numérique avec ThreeJS s'avère être complexe.

Pour ce qui est de l'outil de parcours et de la gestion des projets par fichiers .json, les éléments mis en place sont opérationnels et remplissent les objectifs souhaités. Il n'y a donc pour l'heure pas de réel axe d'évolution à reprendre si ce n'est l'aspect visuel et graphique de l'outil de parcours, qui pourrait être amélioré.

En revanche les perspectives d'amélioration sont bien évidemment nombreuses concernant le traitement de données lidar haute définition et leur mise en relation avec le contexte du stage du BIM-SIG. En effet, une plateforme de test a été mise en place pour visualiser ces données sur le web et voir les différentes interactions possibles avec le nuage de points. Les données sont chargées grâce à la librairie PotreeJS et visualisées au travers d'un « PotreeViewer ». Ce dernier permet ainsi d'avoir des possibilités en termes de symbologie, de catégorisation et autres modifications d'apparence.

Maintenant que cette plateforme est mise en place, il est donc possible de s'intéresser aux traitements réalisables sur le nuage de points pour le mettre en lien avec les données BIM. On peut imaginer plusieurs calculs en lien avec le nuage de points, avec par exemple des calculs de volumes pour certains bâtiments, la génération de modèles 3D à partir d'une sélection, mais également la création de modèle numérique de terrain encore plus précis sur certaines zones telles avec la présence de végétation.

Ces différentes perspectives permettraient de pouvoir générer des maquettes et de comparer avec les maquettes BIM à la fois en termes de précisions, de vitesse de chargement. À cela, on pourrait ajouter des possibilités en termes d'export de données classifiées à partir d'une sélection avec par exemple un export de données seulement rattachées à de la végétation haute, des bâtiments ou bien des routes.

Enfin, des améliorations pourraient être mises en place au niveau de l'import de données lidar. Comme détaillé précédemment ([3.2.2](#)), les données doivent nécessairement être au format Potree pour être chargées et visualisées. Elles doivent être converties en amont via le convertisseur « PotreeConverter ». Ainsi, on peut très bien imaginer mettre en place au sein du serveur un back-end tel que NodeJS pour pouvoir lancer le convertisseur en simultané dès que l'utilisateur charge un fichier de type .laz ou .las.

Mes recherches, effectuées sur un serveur local, ont montré la faisabilité technique d'une telle démarche (figure 24). Sa mise en place sur le serveur de 3DGS nécessiterait tout de même plus de temps avec la mise en place de reverse proxy lorsque les données sont appelées au sein du backend. De plus, l'utilisation d'un back-end permettrait de stocker les fichiers lidar volumineux dans une base de données externe et de gérer leur affichage par commande NodeJS

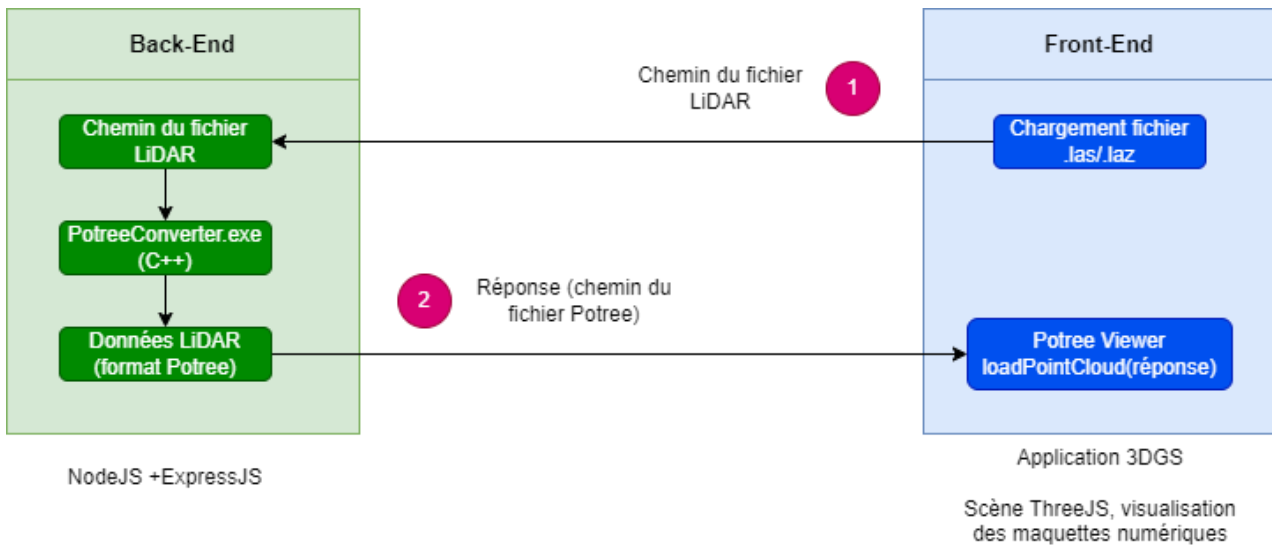


Figure 24 : Intégration d'un backend NodeJS : lecture et conversion de fichiers lidar au format Potree

- Perspectives générales sur 3DGS

Enfin, des grands axes de progression peuvent être abordés sur l'application 3DGS. On peut parler dans un premier temps de la gestion du système de coordonnées sur la plateforme. Il n'y a en effet pas de système de coordonnées défini dans l'application puisque les maquettes sont chargées au niveau 0 de la scène. Une étape dans l'élaboration de cette plateforme serait de charger à la fois les données OpenStreetMap et les maquettes à leurs coordonnées exactes. La faisabilité reste à évaluer avec ThreeJS.

Une autre piste d'exploration serait la mise en place d'un service d'authentification sécurisé avec des logins, mots de passe. Les utilisateurs devraient alors s'authentifier afin d'accéder à la plateforme. Cela impliquerait une liaison avec une base de données contenant les identifiants, il faudrait également pouvoir permettre de s'inscrire. La mise en place d'un back-end est donc, ici aussi, à envisager. Cela pourrait se faire avec la mise en place de NodeJS ou bien d'un langage côté serveur tel que php.

Avantages	Description
Utilisation d'applications externes	Permet de lancer des fichiers .exe directement lorsqu'un évènement est activé dans le Front-End, tel que PotreeConverter
Authentification sécurisée	Liaison avec une base de données contenant les logins/mots de passe des utilisateurs
Séparation des tâches	Division de l'application en deux fonctionnalités distinctes, pour plus de clarté

Tableau 2 : Synthèse des avantages de la mise en place d'un back-end dans le cas de 3DGS

# Conclusion

---

Pour conclure, ces 6 mois de stage à Magellium ont permis de poursuivre les avancées sur ce projet BIM-SIG avec des outils open-source.

Ce projet sur la liaison BIM-SIG est au cœur des nouvelles perspectives de la géomatique pour ces prochaines années. Outre le fait que cette convergence puisse permettre de nouveaux traitements et une meilleure précision, cela permettrait d'améliorer la capacité à communiquer et faire collaborer les projets de construction et leur implantation au sein de leur contexte géographique.

Les différentes missions que j'ai pu mener ont permis d'améliorer la chaîne de traitements existante, tant sur la partie QGIS que sur la partie web.

En effet, les maquettes numériques peuvent à présent être intégrées dans des logiciels SIG open-source et géoréférencées à l'emplacement souhaité par l'utilisateur. Ces mêmes couches peuvent ensuite être intégrées dans une base de données et desservies sur la plateforme 3DGS. Sur cet outil web, l'objectif était de développer et créer de nouvelles fonctionnalités pour interroger et manipuler les maquettes 3D. Sur ce point, des éléments ont pu être ajoutés.

L'utilisateur peut désormais créer des parcours immersifs dans l'application avec la possibilité de mettre en place un modèle numérique de terrain fidèle à la zone environnante. L'ensemble de ces données est désormais exportable par le biais de fichiers .json, améliorant la capacité de partage et de communication de cet outil.

Ce stage a également permis de montrer les possibilités d'un nouveau type de données : le lidar. C'est une technologie de plus en plus présente dans les plateformes 3D. La manipulation de nuages de points peut être complexe avec des outils open-source mais offre davantage de possibilités sur les traitements (génération de modèles numériques à haute précision, calculs de dimensions, de volumes, rendu vraies couleurs...).

Pour finir sur un point de vue plus personnel, ces 6 mois d'expérience à Magellium ont été riches d'apprentissages. Réaliser son stage dans un contexte de recherche et développement est très efficace pour bien comprendre, analyser, et résoudre les différents problèmes rencontrés. Bien que certains éléments soient au stade du prototype et pas encore finalisés, cela permet de créer des choses, de montrer les possibilités avec seulement des technologies libres.

Mais la plus grande force de ce stage, de mon point de vue, est sa diversité. J'ai été amené à travailler à la fois sur des solutions SIG avec la modification et création de plug-in, et à la fois sur des outils de développement web avec la mise en place d'une application qui doit être agréable à utiliser et la plus efficace possible. Dans les deux cas, l'expérience utilisateur est le fil conducteur des missions pour garantir la qualité des outils créés.



## Références

---

Awange Joseph L., Grafarend Erik W., « Linearized Least Squares and nonlinear Gauss-Jacobi combinatorial algorithm applied to the 7-parameter datum transformation problem », Fachbeiträge, 2002.

Chaudet Bruno, Patrascu Marcela et Bouillon Jean-Luc, « La maquette numérique dans le secteur du bâtiment », Revue française des sciences de l'information et de la communication [En ligne], 9 | 2016. URL : <http://journals.openedition.org/rfsic/2044> ; DOI: <https://doi.org/10.4000/rfsic.2044>

Junxiang Zhu et al. "A Critical Review of the Integration of Geographic Information System and Building Information Modelling at the Data Level". In : ISPRS International Journal of Geo-Information 7 (fév. 2018), p. 66. doi : 10.3390/ijgi7020066.

Junxiang Zhu et al. "Automatically Processing IFC Clipping Representation for BIM and GIS Integration at the Process Level". In : Applied Sciences 10 (mar. 2020), p. 2009. doi : 10.3390/app10062009.

Junxiang Zhu et al. "Integration of BIM and GIS : Geometry from IFC to shapefile using open-source technology". In : Automation in Construction 102 (juin 2019), p. 105-119. doi : 10.1016/j.autcon.2019.02.014.

Junxiang Zhu et al. "Integration of BIM and GIS : IFC geometry transformation to shapefile using enhanced open-source approach". In : Automation in Construction 106 (oct. 2019), p. 102859. doi : 10.1016/j.autcon.2019.102859.

Laloux M., « Les transformations affines (avec NumPy) ou la signification géométrique d'un fichier worldfile (.tfw, .jgw,...) pour le géoréférencement », Portail SIG, 2011.

Url : <https://portalsig.org/content/les-transformations-affines-avec-numpy-ou-la-signification-geometrique-d-un-fichier-worldfil.html>

Perens B., « The Open-Source Definition », In : Open Sources: Voices from the Open Source Revolution, Janvier 1999, 280 p.

Schütz, M., Ohrhallinger, S. and Wimmer, M. (2020), Fast Out-of-Core Octree Generation for Massive Point Clouds. *Computer Graphics Forum*, 39: 155-167. <https://doi.org/10.1111/cgf.14134>

Schütz M., Potree: Rendering Large Point Clouds in Web Browsers, diploma thesis, Faculty of informatics, Vienna University of Technology, 2016.

Watson G.A., Computing Helmert transformations, Department of Mathematics, University of Dundee, Dundee Scotland, 2005

Zavoti, J. & Jancsó, Tamas. (2006). The solution of the 7-parameter datum transformation problem with- and without the Gröbner basis. *Acta Geodaetica Et Geophysica Hungarica - ACTA GEOD GEOPHYS HUNG.* 41. 87-100. 10.1556/AGeod.41.2006.1.7.

## Table des figures

---

Figure 1 : Logo de l'entreprise Magellium Artal Group.....	2
Figure 2: Approche intégrée du BIM sur la gestion des projets de construction.....	3
Figure 3: Types de représentations géométriques en 3D utilisées par le format IFC (Junxiang Zhu et al., 2020) .....	4
Figure 4 : Synthèse comparative des avantages de la convergence BIM-SIG.....	6
Figure 5 : Interface Homme-Machine du plug-in BimToGis et résultat obtenu dans QGIS et l'extension Qgis2Threejs.....	8
Figure 6 : Processus du traitement de données IFC vers la plateforme web 3DGS.....	9
Figure 7 : Visualisation d'une maquette numérique sur l'application 3DGS.....	9
Figure 8 : Planning prévisionnel et durée réelle des grandes missions du stage.....	10
Figure 9 : Interface Homme-Machine (IHM) de l'extension BimGeoref.....	13
Figure 10 : Principe de la transformation de Helmert sur trois dimensions.....	14
Figure 11 : Calcul des paramètres au sein de l'extension BimGeoref : utilisation des bases de Gröbner.....	15
Figure 12 : Traitement des couches SIG dans l'extension BimGeoref.....	16
Figure 13 : Exemple d'application de BimGeoref : géoréférencement d'une maquette (initialement chargée en x:0, y:0) au niveau de Pech David. ....	18
Figure 14 : Mise en place de la gestion par fichiers projets au sein de 3DGS.....	21
Figure 15 : Illustration du parcours automatique "360 degrés" pour deux maquettes de taille différente...	22
Figure 16 : Exemple d'édition de parcours sur la plateforme 3DGS.....	23
Figure 17 : Gestion des parcours déjà enregistrés.....	24
Figure 18 : Chargement d'une dalle MNT et conversion en ThreePlaneGeometry sur l'application.....	25
Figure 19 : Visualisation de l'élévation sous la forme d'une ThreePlaneGeometry.....	25
Figure 20 : Colorisation de l'élévation selon la palette Viridis.....	26
Figure 21 : Ajustement des maquettes sur l'élévation dans 3DGS.....	27
Figure 22 : Principe de l'indexation du nuage de points au format Octree.....	29
Figure 23 : Visualisation de nuages de points LiDAR sur le serveur de 3DGS : symbologie selon l'élévation (1), la classification (2), la sélection d'attributs de classe (3), et intensité (4).....	30
Figure 24 : Intégration d'un backend NodeJS : lecture et conversion de fichiers lidar au format Potree.....	34

## Table des équations

---

Équation 1 : Formule de la transformation de Helmert pour passer d'un système de référence B à un système de référence A.....	14
---	----

## Table des tableaux

---

Tableau 1 : Bilan des tests de chargement réalisés entre le serveur local et le serveur test.....	30
Tableau 2 : Synthèse des avantages de la mise en place d'un backend dans le cas de 3DGS.....	34

# Annexes

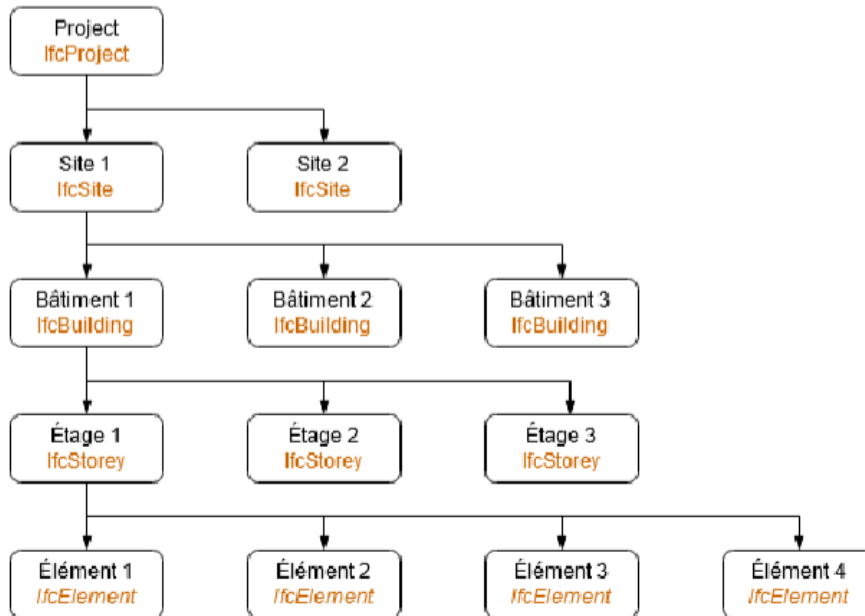
---

<b>Annexe I</b> : Structure hiérarchique des fichiers IFC.....	40
<b>Annexe II</b> : Diagramme de Gantt des deux sprints concernant le développement de BimGeoref.....	40
<b>Annexe III</b> : Diagramme de Gantt des six premières semaines de développement sur 3DGS.....	41
<b>Annexe IV</b> : Diagramme de Gantt du deuxième sprint sur 3DGS.....	41
<b>Annexe V</b> : Calculs permettant des coefficients pour obtenir le facteur d'échelle s.....	42
<b>Annexe VI</b> : Exemple d'application de l'extension QGIS BimGeoref.....	43
<b>Annexe VII</b> : Exemple de fichier JSON après sauvegarde d'un projet sur 3DGS.....	44
<b>Annexe VIII</b> : Exemples complémentaires sur la symbologie de l'élévation.....	45
<b>Annexe IX</b> : Comparaison du chargement de nuages de point sur un serveur local et le serveur de test 3DGS.....	46

## Annexe I

-

### Structure hiérarchique des fichiers IFC



## Annexe II

-

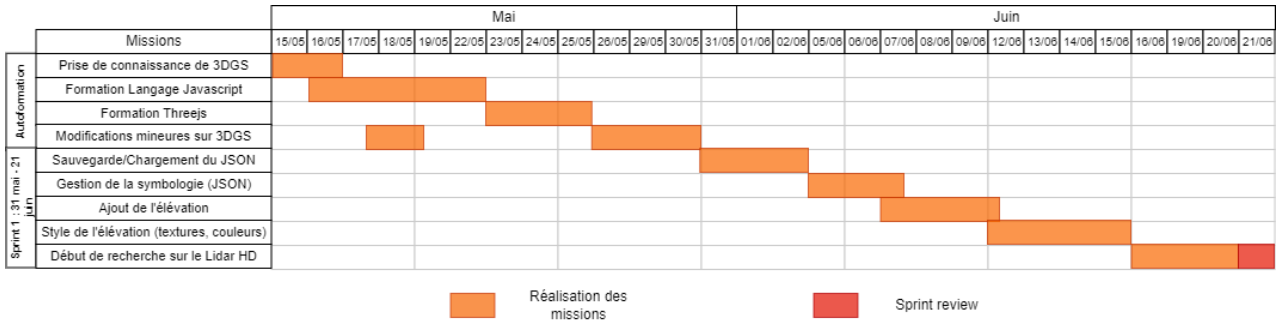
### Diagramme de Gantt des deux sprints concernant le développement de BimGeoref



### Annexe III

-

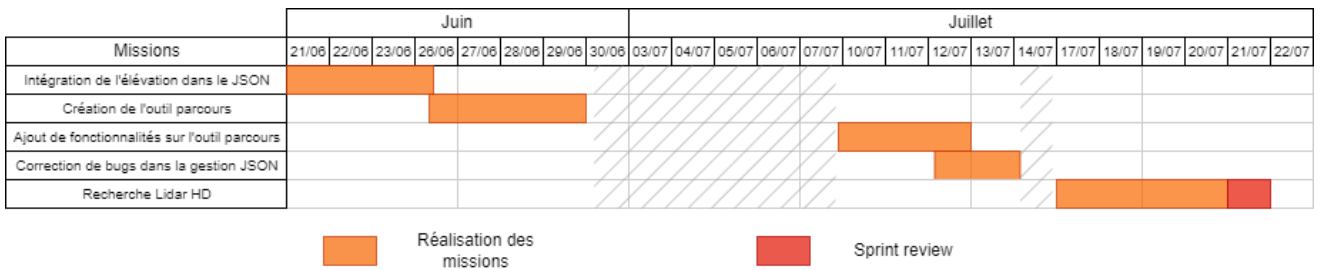
Diagramme de Gantt des six premières semaines de développement sur 3DGS



### Annexe IV

-

Diagramme de Gantt du deuxième sprint sur 3DGS



## Annexe V

-

### Calculs permettant des coefficients pour obtenir le facteur d'échelle s

*Source : Zavoti J., Jancso T., « The solution of 7-parameter datum transformation problem with- and without the Gröbner basis », 2006*

$$q4 = x13 * y12 * y12 * y23 + x12 * x12 * x13 * y23 - x12 * x12 * x23 * y13 \backslash \\ -x12 * y13 * z12 * z23 - x23 * y12 * y12 * y13 + x13 * y12 * z12 * z23 \backslash \\ -x23 * y12 * z12 * z13 + x12 * y23 * z12 * z13$$

$$q3 = c12 * x13 * y12 * z23 - b13 * x12 * z12 * z23 - c13 * x12 * y23 * z12 \backslash \\ +b23 * x13 * y12 * y12 - a23 * x12 * x12 * y13 + c23 * x12 * y13 * z12 \backslash \\ +c12 * x12 * y23 * z13 + a13 * y12 * y12 * y23 - b12 * x23 * z12 * z13 \backslash \\ -a23 * y12 * y12 * y13 + b12 * x13 * z12 * z23 - c12 * x12 * y13 * z23 \backslash \\ +a13 * x12 * x12 * y23 - c23 * x13 * y12 * z12 - a23 * y12 * z12 * z13 \backslash \\ +c13 * x23 * y12 * z12 - b13 * x23 * y12 * y12 - b13 * x12 * x12 * x23 \backslash \\ +b23 * x12 * z12 * z13 + a13 * y12 * z12 * z23 + b23 * x12 * x12 * x13 \backslash \\ +a12 * y23 * z12 * z13 - a12 * y13 * z12 * z23 - c12 * x23 * y12 * z13$$

$$q2 = a13 * b23 * x12 * x12 + b12 * b12 * x23 * y13 + b12 * c13 * x23 * z12 \backslash \\ -c12 * c23 * x13 * y12 + b13 * c23 * x12 * z12 - a23 * b12 * z12 * z13 \backslash \\ +a12 * a12 * x23 * y13 - b12 * b12 * x13 * y23 - a12 * a12 * x13 * y23 \backslash \\ -a23 * b13 * x12 * x12 + a13 * b23 * y12 * y12 - a23 * b13 * y12 * y12 \backslash \\ +a12 * b23 * z12 * z13 + a23 * c13 * y12 * z12 + a12 * c12 * y23 * z13 \backslash \\ -b12 * c12 * x23 * z13 - b23 * c13 * x12 * z12 - a12 * b13 * z12 * z23 \backslash \\ -a23 * c12 * y12 * z13 + a12 * c23 * y13 * z12 + b12 * c12 * x13 * z23 \backslash \\ -a12 * c12 * y13 * z23 - a13 * c23 * y12 * z12 - c12 * c13 * x12 * y23 \backslash \\ +c12 * c13 * x23 * y12 + c12 * c23 * x12 * y13 - b13 * c12 * x12 * z23 \backslash \\ +b23 * c12 * x12 * z13 + a13 * b12 * z12 * z23 - a12 * c13 * y23 * z12 \backslash \\ +a13 * c12 * y12 * z23 - b12 * c23 * x13 * z12$$

$$q1 = -a12 * b13 * c12 * z23 + b12 * c12 * c13 * x23 + b12 * b12 * b13 * x23 \backslash \\ -a13 * b12 * b12 * y23 - a12 * a12 * b23 * x13 + a23 * c12 * c13 * y12 \backslash \\ -a13 * c12 * c23 * y12 + a12 * b13 * c23 * z12 - a12 * a12 * a13 * y23 \backslash \\ -b23 * c12 * c13 * x12 + a12 * b23 * c12 * z13 - a23 * b12 * c12 * z13 \backslash \\ -b12 * b12 * b23 * x13 + a23 * b12 * c13 * z12 - a12 * c12 * c13 * y23 \backslash \\ +a12 * a12 * b13 * x23 + a12 * a12 * a23 * y13 - a12 * b23 * c13 * z12 \backslash \\ -a13 * b12 * c23 * z12 - b12 * c12 * c23 * x13 + b13 * c12 * c23 * x12 \backslash \\ +a12 * c12 * c23 * y13 + a23 * b12 * b12 * y13 + a13 * b12 * c12 * z23$$

$$q0 = a12 * b13 * c12 * c23 - a13 * b12 * b12 * b23 + a12 * a12 * a23 * b13 \backslash \\ -a12 * a12 * a13 * b23 - a12 * b23 * c12 * c13 + a23 * b12 * b12 * b13 \backslash \\ +a23 * b12 * c12 * c13 - a13 * b12 * c12 * c23$$

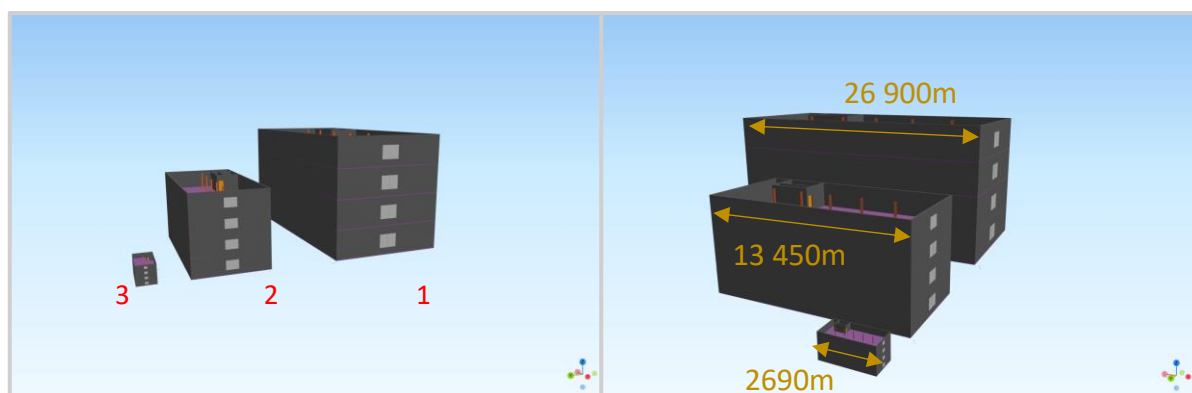
Avec :

x, y et z les points dans un système de référence et a, b et c les coordonnées des points dans le nouveau système de référence.

$$x12 = x1-x2 ; z23 = z2-z3 ; a12 = a1 - a2 \dots$$

## Annexe VI

### Exemple d'application de l'extension QGIS BimGeoref



**1** : maquette initiale en sortie de BimToGis

**2** : Réduction d'échelle par 0.5

	Point 1	Point 2	Point 3	Descriptif de la transformation
Points saisis	(767.14966 ; 7523.43077)	(27667.14966 ; 7523.32821)	(27667.14966 ; 19923.32821)	Pas de sélection du Z Translation de 20 000 en Y Facteur d'échelle : 0,5
Nouvelles coordonnées	(767.14966 ; 27523.43077)	(17667.14966 ; 27523.32821)	(17667.14966 ; 33723.32821)	

**3** : Réduction d'échelle par 0.2

	Point 1	Point 2	Point 3	Descriptif de la transformation
Points saisis	(767.14966 ; 7523.43077)	(27667.14966 ; 7523.32821)	(27667.14966 ; 19923.32821)	Pas de sélection du Z Translation de 30 000 en Y Facteur d'échelle: 0.2
Nouvelles coordonnées	(767.14966 ; 37523.43077)	(4767.14966 ; 37523.32821)	(4767,14966 ; 40003.32821)	



## Annexe VII

-

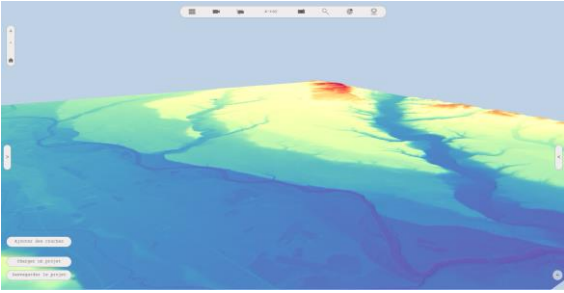
### Exemple de fichier JSON après sauvegarde d'un projet sur 3DGS

```
{
  "title": "PROJECT_demo",
  "cameraDirection": {
    "x": -39.95545,
    "y": 0.01,
    "z": -55.58487
  },
  "cameraPosition": {
    "x": -104.90284477539738,
    "y": 58.56201967206166,
    "z": -155.85199773492224
  },
  "cameraRotation": {
    "_x": -2.613050561465781,
    "_y": -0.5099968676286155,
    "_z": -2.86388457244972,
    "_order": "XYZ"
  },
  "layers": {
    "ws-3dgs:Projet-FZK-Haus_5_Garde-corps_HLR": {
      "Projet-FZK-Haus_5_Garde-corps_HLR_Mesh_0": {
        "color": {
          "r": 0.42745098039215684,
          "g": 0.14901960784313725,
          "b": 0.14901960784313725
        },
        "opacity": 1
      }
    },
    "ws-3dgs:Projet-FZK-Haus_5_Mur_HLR": {
      "Projet-FZK-Haus_5_Mur_HLR_Mesh_0": {
        "color": {
          "r": 1,
          "g": 1,
          "b": 1
        },
        "opacity": 1
      }
    }
  },
  "geosignets": {
    "Géosignet: 78": {
      "name": "78",
      "geoDir": "{\"x\":-39.95545,\"y\":0.01,\"z\":-55.58487}",
      "geoPos": "{\"x\":-104.90284477539738,\"y\":58.56201967206166,\"z\":-155.85199773492224}",
      "geoRot": "{\"_x\":-2.613050561465781,\"_y\":-0.5099968676286155,\"_z\":-2.86388457244972,\"_order\": \"XYZ\"}"
    }
  },
  "request": {},
  "description": "",
  "name": "demo",
  "elevation": {
    "name": "mnt_zone_etude.tif",
    "sympo": {
      "texture": "Spectral",
      "height_ex": 2,
      "visible": true,
      "opacity": 1
    }
  },
  "animations": {
    "360 degres": {
      "lookAtCenter": true,
      "centerPoint": {
        "x": -46,
        "y": 7,
        "z": -63
      },
      "path": [[-62,7,-78], [-62,7,-48], [-30,7,-48], [-30,7,-78]],
      "speed": 8
    }
  }
}
```

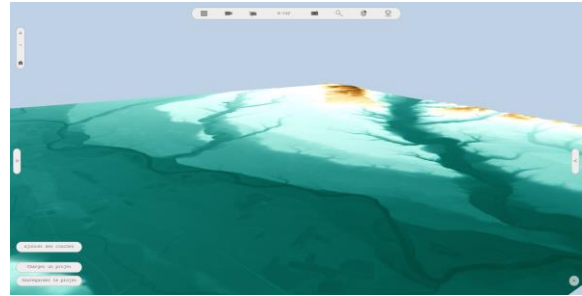
## Annexe VIII

-

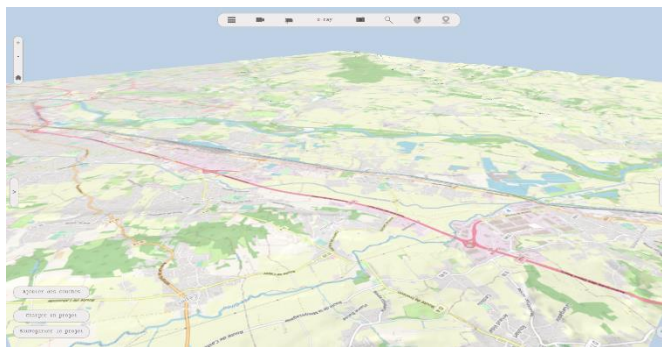
### Exemples complémentaires sur la symbologie de l'élévation



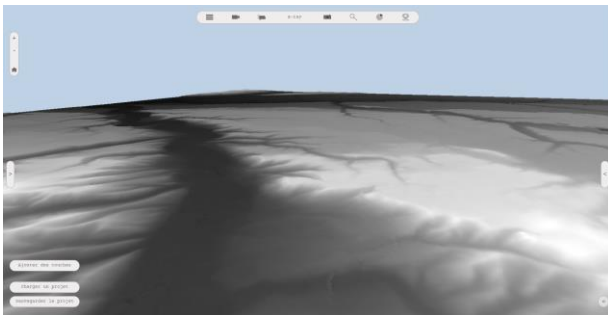
Palette Spectral



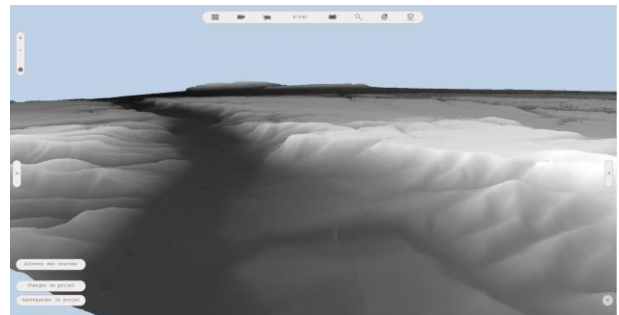
Palette BrownBlueGreen



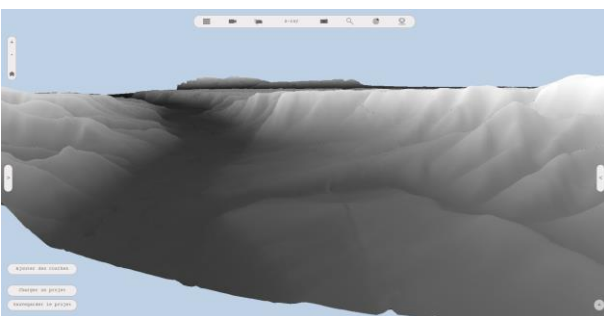
Texture OpenStreetMap



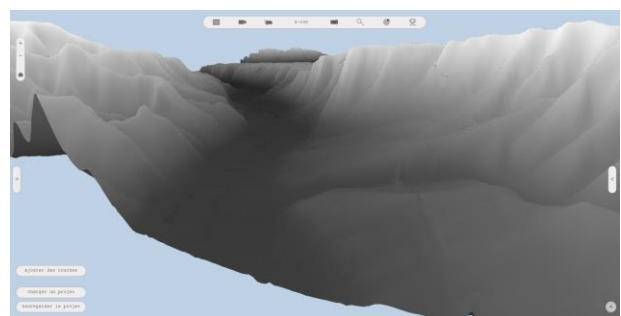
Facteur d'élévation : 1  
(élévation réelle)



Facteur d'élévation : 4



Facteur d'élévation : 8

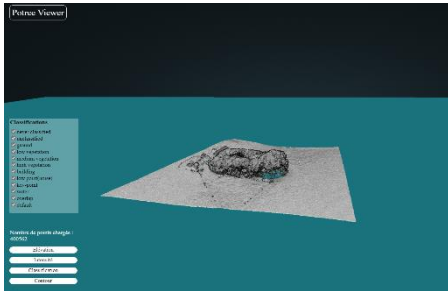


Facteur d'élévation : 11

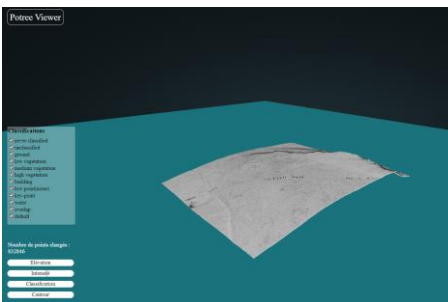
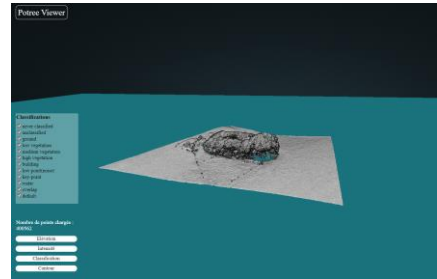
# Annexe IX

-

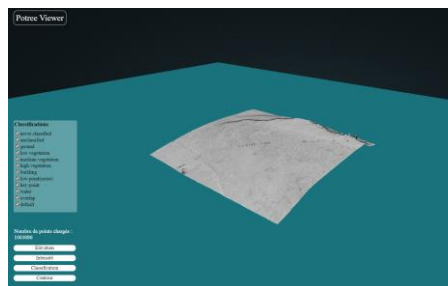
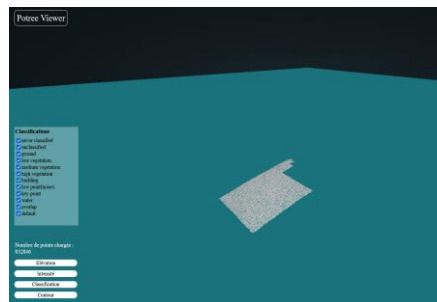
## Comparaison du chargement de nuages de point sur un serveur local et le serveur de test 3DGS



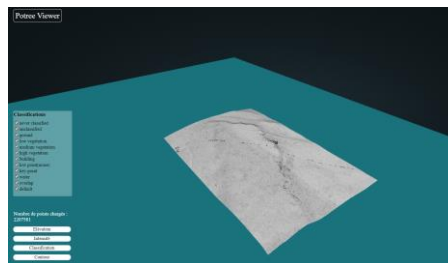
**400 562  
points**



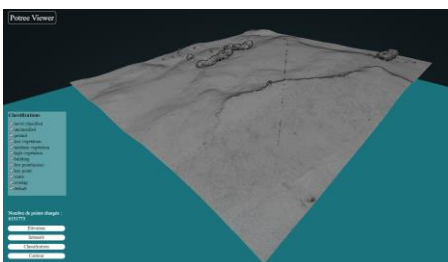
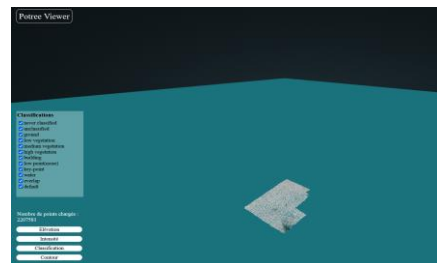
**800 562  
points**



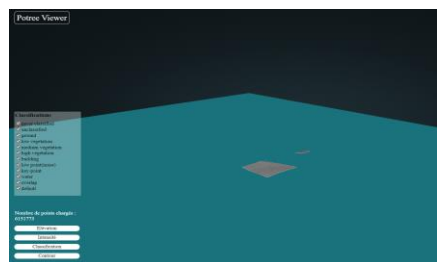
**1 000 562  
points**



**2 000 562  
points**



**6 000 562  
points**



**Serveur de test**

**Serveur local**