

Projet développement Informatique

# Couplage du moteur de jeu 3D Minetest et d'un moteur de visualisation géographique 3D



## Membres du groupe

Jules Rosier  
Joachim Comet  
Romain Coin  
Vincent Miras

## Commanditaires

François Lecordix  
Pierre-Yves Rollo  
Quentin Bouillaguet

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Contexte du projet</b>	<b>1</b>
<b>3</b>	<b>Objectifs de l'étude - Reformulation du besoin</b>	<b>2</b>
3.1	Les objectifs de l'étude . . . . .	2
3.2	Les contraintes . . . . .	2
3.3	Le recueil du besoin - Les acteurs . . . . .	3
<b>4</b>	<b>Analyse fonctionnelle - Solutions proposées</b>	<b>3</b>
<b>5</b>	<b>Etude technique : Choix des logiciels et langages - Architecture</b>	<b>5</b>
<b>6</b>	<b>Réalisation et suivi de projet</b>	<b>6</b>
6.1	Les risques . . . . .	6
6.1.1	Difficultés prévues . . . . .	6
6.1.2	Matrice des Risques . . . . .	7
6.2	Difficultés rencontrés . . . . .	7
6.3	Planning Prévisionnel . . . . .	8
<b>7</b>	<b>Conclusion et Perspectives</b>	<b>8</b>

# 1 Introduction

Notre projet de développement informatique s'inscrit dans le cadre d'une collaboration entre l'IGN*fab* et EvidenceB, en réponse à un appel d'offre de l'Éducation Nationale. L'objectif de cette initiative est d'enrichir l'outil développé par EvidenceB, Kidscode, en intégrant une fonctionnalité de liaison entre le moteur de visualisation géographique 3D iTowns et le moteur de jeu 3D Minetest.

Minetest est un jeu vidéo open-source de type bac à sable, offrant aux joueurs un monde virtuel en 3D pour explorer, construire et interagir. Gratuit et hautement personnalisable grâce à sa communauté active, il permet aux joueurs de créer et partager leurs propres contenus et serveurs multijoueurs. iTowns est quant à elle une bibliothèque open-source pour la visualisation 3D géospatiale sur le web, offrant une immersion interactive dans des données géographiques.

Notre rôle dans ce projet est de réaliser cette liaison technique entre les deux moteurs, en permettant une synchronisation en temps réel des mouvements entre Minetest et iTowns. Cette liaison vise à offrir aux utilisateurs la possibilité d'explorer des environnements virtuels dans Minetest tout en visualisant simultanément leur position sur une carte avec des couches de bâtiments en 3D sur iTowns.

La finalité du projet serait de permettre la visite à distance de zones particulières, de monuments, ou de musées grâce à cette liaison. Ainsi, une fois qu'on pénètre dans l'enceinte d'un bâtiment visitable sur Minetest, iTowns affiche un rendu 3D de l'intérieur de ce bâtiment qui devient donc visitable aisément.

## 2 Contexte du projet

Le commanditaire de notre projet, François Lecordix, est un membre de l'équipe travaillant pour l'IGN*fab*. Cette entité joue le rôle d'un incubateur de start-up, une caractéristique qui les amène régulièrement à proposer des projets à des équipes de travail et notamment à des groupes d'étudiants comme le notre. Kidscode a été développé par EvidenceB, une start-up qui a été créée dans le but de fournir un moyen d'enseignement ludique aux professeurs et aux élèves, en utilisant le jeu vidéo Minetest comme principal support. En 2019, elle s'associe à l'IGN*fab* pour répondre à un appel d'offre de l'Éducation Nationale. Cette collaboration vise à créer une association entre le service Minecraft à la carte proposé par l'IGN et Kidscode, dans le but de concevoir un outil d'apprentissage géographique novateur avec la possibilité de se situer à n'importe quel endroit dans le jeu et de retrouver sa position sur la carte topographique 2D de l'IGN correspondante. Depuis son lancement en 2021, le logiciel Kidscode est en constante évolution, alimentée par l'ajout continu de fonctionnalités. Notre projet s'inscrit dans cette dynamique d'amélioration, en contribuant à enrichir l'outil existant par le biais de fonctionnalités supplémentaires.

L'enjeu de notre projet n'est donc pas de développer une nouvelle application, mais bien d'améliorer un service déjà existant en y ajoutant une fonctionnalité spécifique. En effet, il nous a été demandé de proposer un système permettant un couplage entre la position du moteur de jeu vidéo Minetest et celle du moteur de visualisation 3D iTowns.

Nous ne connaissons pas et n'avons pas vraiment de moyen de connaître les potentielles conséquences de l'avortement de notre projet pour l'entreprise.

Dans le cadre de notre projet, nous serons amenés à utiliser un langage de programmation que nous n'avons encore jamais utilisé auparavant : le LUA. Bien que nous manquions d'une expertise préalable dans ces langages, chaque membre de notre groupe apporte des compétences et des perspectives uniques qui enrichissent notre approche collective.

## 3 Objectifs de l'étude - Reformulation du besoin

### 3.1 Les objectifs de l'étude

Notre objectif est de proposer une synchronisation entre le positionnement et l'orientation du joueur dans Minetest avec le service de visualisation 3D iTowns. Ainsi en considérant une même zone géographique sur iTowns et sur Minetest, le but est d'avoir une synchronisation de la position sur iTowns et Minetest en temps réel. Dans un second temps, si nous avions le temps, il était proposé que nous développions une deuxième fonctionnalité proposant le même service, mais au lieu d'explorer une zone extérieure, il permettrait de visiter l'intérieur d'un bâtiment modélisé en 3D. Cela pourrait ainsi permettre la visite de monument et de musée à distance.

Concernant l'architecture de notre système, elle est articulée en trois majeures parties. Une partie concernant Minetest, une autre concernant iTowns et finalement un serveur HTTP. Dans un premier temps, nous avons développé un mod pour Minetest, qui permet d'activer ou de désactiver la synchronisation avec iTowns. Ce mod communique également les informations de la position et de l'orientation du joueur au serveur via une requête, en utilisant un fichier JSON. Le serveur est un serveur HTTP Node.js. De même que pour Minetest, on récupère et on envoie les informations sur la position du joueur pour iTowns via des requêtes vers ce serveur. Ainsi le couplage entre Minetest et iTowns se fait à l'aide d'un serveur intermédiaire qui sert de moyen de communication entre iTowns et Minetest.

La problématique d'interopérabilité de notre solution ne se pose que parce que Kidscode ne fonctionne pas sous la dernière version de Minetest. Or il nous a été demandé de mettre en place notre solution en nous appuyant sur la version actuelle de Minetest. Il serait donc possible que nous fassions face à des erreurs dans le fonctionnement du système à cause de ce manque de compatibilité si l'on essayait d'utiliser notre fonctionnalité avec Kidscode.

Du fait que notre projet soit intégré à une plateforme déjà établie et largement adoptée, nous bénéficions d'une visibilité naturelle et d'un accès facilité à notre public cible. En effet, cette intégration nous dispense de la nécessité de prévoir des moyens de diffusion spécifiques pour promouvoir notre outil. Les utilisateurs de la plateforme existante seront en mesure de découvrir notre solution au sein de leur environnement familier, ce qui favorisera son adoption et son utilisation. Cependant, il convient de noter que l'IGN*fab* ne détient qu'une partie des droits de propriété de l'application. Par conséquent, l'IGN*fab* n'est pas autorisé à distribuer l'intégralité de l'application à sa guise. Cette restriction peut poser des défis supplémentaires en termes de diffusion et d'accès à notre outil pour certains utilisateurs potentiels.

### 3.2 Les contraintes

Pour notre projet, différentes contraintes nous sont imposées. Devant développer un service pour un projet déjà existant, il nous est nécessaire de nous adapter aux éléments le composant. Il nous faut donc développer un mod Minetest et un projet iTowns en se basant sur les méthodes de développement et les architectures adaptées aux systèmes. Le projet est une demande de l'Éducation Nationale. De ce fait, les outils utilisés devront être uniquement des outils open source. En ce qui concerne les contraintes liées à la performance, il ne nous est pas demandé de développer un mod et un projet rapides à l'exécution, mais ceux-ci doivent en tout cas être exécutés presque voire aussi rapidement que n'importe quel autre mod Minetest ou projet iTowns. Pour ce qui est de la contrainte de temps, des échéances nous ont été imposées par le jury du projet. En effet, il nous est demandé de rendre certains éléments du projet pour les Copil, Cotech ou encore la soutenance de fin de projet.

### 3.3 Le recueil du besoin - Les acteurs

Différents acteurs s'inscrivent dans le cadre de notre projet. Les premiers acteurs et les plus évidents sont nous, les développeurs du projet. Nous réalisons les différents éléments permettant de mener à bien le projet en fonction des contraintes et des objectifs qui nous sont donnés. Les commanditaires du projet sont aussi des acteurs majeurs. François Lecordix étant le commanditaire principal, c'est lui qui nous fournit le contexte du sujet ainsi que les contraintes et les objectifs. Mais son équipe de travail à l'IGN*fab* nous aide aussi lorsque nous avons besoin d'informations ou de conseils pour la partie programmation. Enfin, l'utilisateur du service, ou joueur, est le dernier acteur du projet. C'est lui qui effectue les actions en entrée de notre service et qui utilise la visualisation fournie en sortie de notre projet.

Afin de recueillir les besoins de nos commanditaires, nous avons convenu d'une réunion avec les commanditaires en début de projet. Cette réunion avait un but purement informatif, les commanditaires ont pu clairement nous expliquer leurs attentes et leurs besoins pour ce projet. Tout au long de l'avancée du projet, nous avons continué d'être en contact avec eux afin de les tenir au courant de l'avancée du projet et de la direction que nous prenions, afin aussi de nous assurer que ce que nous faisons restait cohérent avec leurs attentes.

Comme le décrit le diagramme des cas d'utilisation, l'utilisateur n'a pas la possibilité de réaliser beaucoup d'actions dans le cadre de l'utilisation du service que nous développons. Les seules actions qu'il peut effectuer sont de se déplacer dans l'une des deux interfaces au choix et observer ce mouvement être reproduit dans l'autre interface.



FIGURE 1 – Diagramme des cas d'utilisation

## 4 Analyse fonctionnelle - Solutions proposées

Le projet repose sur le but de synchroniser les mouvements entre le moteur de visualisation ITowns et le jeu Minetest. Pour cela, on a réalisé le diagramme d'activités 2 qui permet de voir les principales étapes nécessaires au fonctionnement du système. Ces fonctionnalités sont la récupération et la transformation de coordonnées depuis un des deux moteurs vers le second, le tout en communiquant à l'aide d'un serveur intermédiaire. Cela va permettre de modifier la vue sur le second moteur afin d'avoir un déplacement simultané.

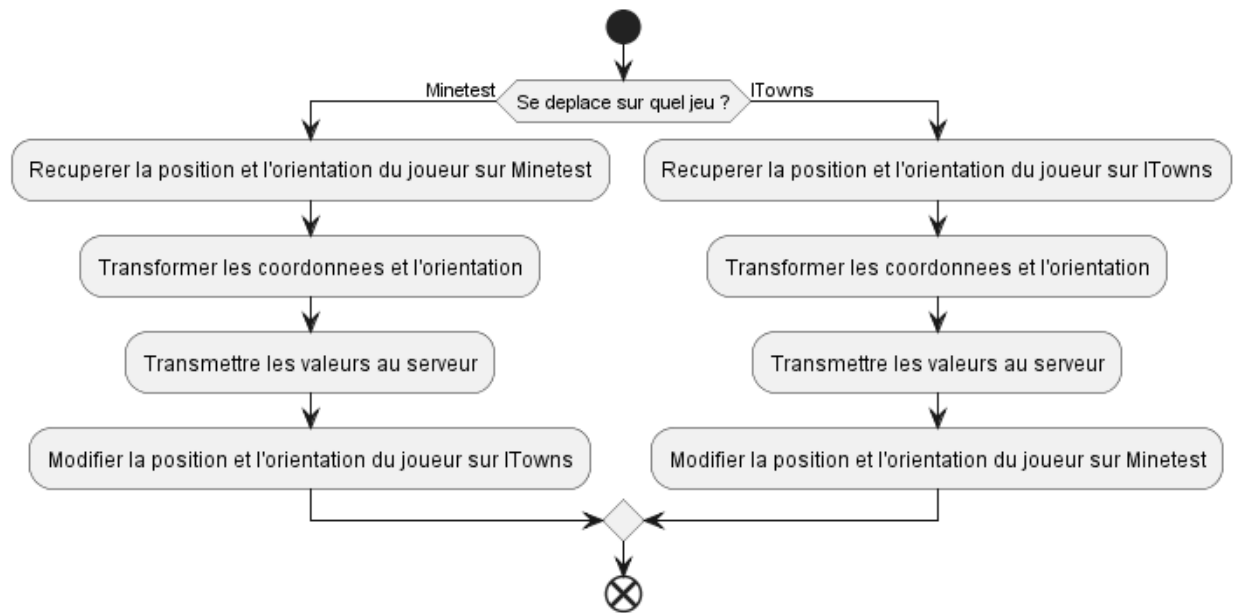


FIGURE 2 – Diagramme d'activité

Ce diagramme se complète avec le diagramme de séquence 3 qui permet de se rendre compte que le processus n'est pas dépendant de l'utilisateur mais se fait de manière continue et autonome.

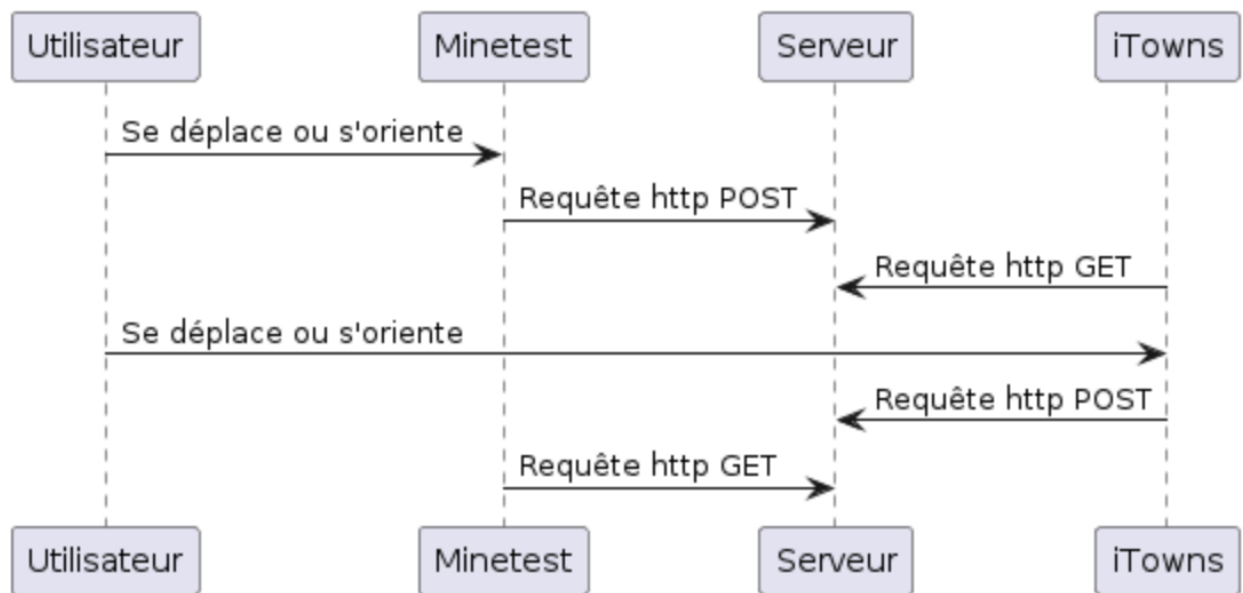


FIGURE 3 – Diagramme de séquence

Cependant, ces fonctionnalités n'utilisent pas les éléments explicites qui seront modifiés. C'est pourquoi le diagramme des classes 4 permet de mieux comprendre les éléments utilisés dans les fonctionnalités.

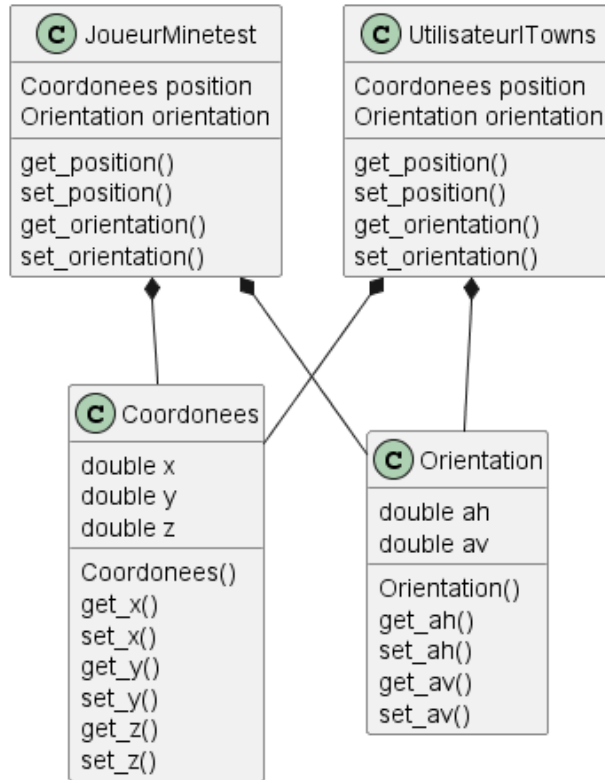


FIGURE 4 – Diagramme de classes

## 5 Etude technique : Choix des logiciels et langages - Architecture

Le choix des outils Minetest et iTowns ont été imposées par le commanditaire. Nous avons donc utiliser le logiciel Minetest et la bibliothèque iTowns existants. De plus, nous avons utilisé l'environnement Node.js afin de réaliser un serveur pour transmettre les informations. Pour la partie Minetest, nous avons fait le choix de développer un mod car cela constitue une manière simple et efficace de modifier le jeu et avoir un moyen de récupérer les coordonnées du jeu. Le mod est codé en Lua, langage du jeu. Si certaines demandes semblent irréalisables, en Lua, elles seront faites en C++ puisque le Lua est un langage qui en dérive. Le C++ est donc compatible avec le jeu. Le mod sera intégré directement sur le dossier du jeu pour être utilisé dans le jeu dès son lancement.

Pour le projet iTowns, le choix a été fait entre le langage TypeScript et JavaScript. Les deux langages sont très proches l'un de l'autre, la grosse différence étant que le TypeScript est typée alors que le JavaScript ne l'est pas du tout. Nous avons déjà codé en JavaScript dans le cadre de nos études. Ainsi le choix s'est porté sur le langage JavaScript pour éviter des problèmes liés aux types des objets qui nous sont inconnus et ainsi se concentrer sur les fonctionnalités.

Minetest est un moteur de jeu utilisant des voxels pour représenter un monde en 3 dimensions. Ce moteur a été utilisé par l'IGN pour le projet "Minecraft à la carte" qui permet de représenter la France dans un monde de voxels. Minetest étant en open-source, il est facile d'ajouter des fonctionnalités en ajoutant des mods au jeu de base.

L'outil iTowns est un modélisateur de données géographiques développé par l'IGN et disponible en téléchargement sur git. Une grande proportion de données géographiques est visualisable sur iTowns uniquement en flux : WMS, WMT, WFS et TMS. Les données peuvent être aussui bien des vecteurs que des rasters.

ITowns utilise Three.js : Les données sont intégrées dans des couches regroupées dans une vue. La vue est planaire si la projection est locale. Sinon la vue représente le globe pour une projection globale comme WGS84. Three.js est une bibliothèque JavaScript qui comprend un grand nombre de constructeurs (vecteurs 2D,3D, rotation, camera, position, vue) et méthodes qui nous permettront soit de modifier notre espace 3D dans ITowns (caméra, vue), soit d'obtenir des éléments pour les modifier (matrice de rotation). Le projet se concentrera sur une tuile WMS (en flux depuis géoplateforme) dont l'emprise correspond géographiquement à l'emprise de la tuile de "Minetest à la carte" lancée dans le jeu.

Le serveur va traiter 4 requêtes afin de gérer les deux sens de mouvement. Pour chaque sens, il y a une requête POST et une requête GET pour transmettre les données nécessaires sous un format JSON. Ces données sont sauvegardées dans une variable du serveur pour simplifier l'accès en tout temps.

## 6 Réalisation et suivi de projet

### 6.1 Les risques

#### 6.1.1 Difficultés prévues

Dans le cadre de notre projet, nous avons été amenés à étudier les risques liés à notre projet.

Nous avons tout d'abord identifié les probables difficultés que nous pensons rencontrer durant notre projet. Les différentes difficultés prévues sont les suivantes :

- **Actualisation des coordonnées en temps réel sur Minetest** : La mise à jour des coordonnées en temps réel pourrait poser des problèmes techniques et de synchronisation.
- **Minetest - Exportation des coordonnées** : L'exportation des coordonnées de Minetest vers un fichier externe pourrait nécessiter une manipulation complexe des données. Cette manipulation permettrait de synchroniser les positions en temps réel entre Minetest et iTowns.
- **Gestion de projet - Prise en main des outils de gestion de Projet** : Nous pourrions rencontrer des difficultés dans l'adaptation et l'utilisation efficace des outils de gestion de projet nécessaires à la coordination de l'équipe et au suivi des tâches.
- **Minetest et iTowns - Prise en main des langages TypeScript et Lua** : L'apprentissage et la maîtrise des langages de programmation TypeScript et Lua, utilisés dans Minetest et iTowns, pourraient représenter un défi pour certains membres de l'équipe.
- **Gestion de projet - Scission de l'équipe** : Une scission au sein de l'équipe peut entraîner un manque de communication et des retards dans le projet. Il est crucial de favoriser un environnement de travail collaboratif et de promouvoir une communication ouverte et transparente. La résolution rapide des conflits et la clarification des rôles et responsabilités peuvent aider à prévenir ce risque.
- **Mise à jour software** : Les mises à jour logicielles peuvent nécessiter des changements dans l'architecture existante, ce qui pourrait entraîner des perturbations dans le projet. Il est important de maintenir une communication étroite avec le commanditaire pour anticiper les mises à jour logicielles et élaborer des plans d'action appropriés. La documentation des processus de mise à jour et la réalisation de tests rigoureux peuvent également contribuer à atténuer ce risque.
- **Problème de droits sur le poste de travail** : Les problèmes de droits d'accès sur les postes de travail peuvent entraîner des retards dans le travail pour les personnes concernées. Pour y remédier nous avons demandé un compte administrateur à notre commanditaire.
- **Problème de Proxy sur le poste de travail** : Les problèmes de proxy sur les postes de travail peuvent entraîner des retards dans le travail pour les personnes concernées. Nous avons dû modifier la répartition des tâches car un des postes de travail était bloqué par le proxy pour travailler sur iTowns.



### 6.1.2 Matrice des Risques

Afin de prévenir les risques que nous allons rencontrer, nous avons mis en place une matrice des risques, elle sert à identifier, évaluer et gérer les risques potentiels associés à un projet.

Nature du risque	Probabilité	Conséquences	Solution	Evolution	Gravité
Manque de temps	Forte	Pas de résultats ou résultats peu concluants	Compromis dans les objectifs	↗	Forte
Perte de données	Moyen	Retard dans l'avancée du travail	Mise en place de back-up	↘	Forte
Temps de réponse qui varie pour les actions Minetest	Moyen	Chaine d'actions désordonnée	attendre que la requête soit finie	↘	Faible
Ne pas trouver de dalle correspondant à la zone étudiée	Moyen	Ne pas avoir de modèle 3D de la zone étudiée	Utiliser une dalle 2D, ou changer de dalle	=	Moyen
Difficultés d'apprentissage des langages de programmation	Faible	Retard dans l'avancée du travail	Demande d'aide extérieure	↘	Faible
Scission de l'équipe	Très faible	Manque de communication et retard dans le projet	Bien communiquer	↘	Moyen
Mise à jour software	Très faible	Obligation de changement de l'architecture	Demande d'aide au commanditaire	↘	Faible
Problème de droits sur le poste de travail	Forte	Retard sur le travail pour la personne concernée	Demande de compte administrateur	↘	Moyen
Problème de Proxy sur le poste de travail	Forte	Retard sur le travail pour la personne concernée	Réorganisation des tâches	=	Moyen
Temps de réponse des comanditaire	Forte	Retard sur le travail pour tout le groupe		↗	Forte

FIGURE 5 – Matrice des Risques

## 6.2 Difficultés rencontrés

Nous avons fait face à plusieurs problèmes pendant toute la durée du projet PDI :

- **Problème de prise en main Typescript et iTowns** : la documentation fournie pour iTowns et les fonctions déjà présentes étant en JavaScript, il était parfois difficile de comprendre les types des objets et la syntaxe à utiliser pour faire fonctionner nos programmes en TypeScript. Nous avons donc fait le choix de coder en JavaScript.
- **Problème de mise à jour des données flux à l'IGN et problème de CORS (Cross-Origin Resource Sharing)** : l'IGN est passée de géoservices à géoplateforme ce qui a rendu l'import des couches géoservices obsolètes et ralentit le projet. Les problèmes de CORS ont aussi limité le nombre de couches que l'on pouvait charger. Le groupe s'est donc limité à une simple utilisation des couches WMS et de l'élévation avec bâtiments 3D sur iTowns.
- **Problème de lecture et écriture dans un fichier** : Il était prévu initialement d'écrire nos coordonnées (position et orientation) de Minetest et iTowns dans un fichier qui servirait donc de lien entre les deux. Par soucis d'accès concourants (lecture/écriture), nous avons préféré utiliser un serveur Node.js. Ce serveur devra être lancé en même temps que iTowns et Minetest et stockera les coordonnées en flux pour empêcher des retours en arrière problématique.
- **Problème de synchronisation des positions et rotations sur iTowns** : Pour la position, les fonctions set n'étant pas défini, il a fallu calculer les  $(dx\ dy\ dz)$  pour les ajouter aux anciens  $(x\ y\ z)$  afin d'obtenir la nouvelle position. Pour la rotation, le problème était bien plus embêtant. Les angles obtenus sur Minetest sont le Yaw (angle horizontal du joueur par rapport à l'orientation de la carte) et le Pitch (angle vertical du joueur avec le sol). Il n'y a pas de Roll car le joueur reste perpendiculaire au sol. Cependant, la rotation sur iTowns s'opère sur la couche WMS et non le joueur. En utilisant les angles d'Euler, le Yaw reste donc un Yaw, mais le Pitch varie entre Pitch et Roll suivant le Yaw (suivant si l'on regarde au Nord, Sud, Est, Ouest). Il a fallu retourner aux coordonnées polaires pour calculer un vecteur dont la direction correspond à la visée dans Minetest et utiliser la fonction lookAt qui ajuste la rotation de la carte sur iTowns suivant la direction du vecteur en paramètre.
- **Problème d'import de l'emprise de la carte Minetest sur iTowns** : Nous souhaitions faire en sorte qu'iTownns récupère directement l'emprise de la carte affichée par Minetest, sans que l'utilisateur n'ait à la renseigner. Nous avons pour cela utilisé un fetch, mais le fetch n'étant pas immédiat en JavaScript, le programme s'exécutait avant que l'emprise de la carte soit obtenue depuis le serveur. La vue ne pouvait donc pas se créer correctement. Il a fallu utiliser un setTimeout.
- **Problème d'utilisation de Node.js et plus généralement cmd** : N'étant pas familier avec l'utilisation de node ou des invites de commande, l'installation des dépendances d'iTownns ou d'un serveur Node s'est révélée compliquée et a nécessité pas mal de temps.
- **Problème de synchronisation de la rotation sur Minetest** : Le problème de l'angle vertical se

repose puisque la rotation sur iTowns comporte un paramètre roll en plus qui ne vaut jamais 0. Nous avons recalculé les angles avec une matrice de rotation d'Euler obtenue en utilisant Three.js. Cette matrice a permis de retrouver la direction  $(dx\ dy\ dz)$ .

### 6.3 Planning Prévisionnel

Nous avons décidé d'attribuer des rôles précis aux membres de notre groupe afin de moins se disperser. La répartition est la suivante :

- Jules Rosier : Chef de projet et rédacteur
- Joachim Comet : Responsable de l'analyse fonctionnelle du projet
- Vincent Miras : Chef développeur Minetest
- Romain Coin : Chef développeur iTowns

Nous avons opté pour une division en deux parties distinctes au sein de notre équipe pour la partie développement du projet. Vincent et Joachim seront responsables du développement de la partie Minetest, tandis que Romain et Jules se chargeront du développement de la partie iTowns. Le chef de projet sera en communication avec le commanditaire par courrier électronique.

Par ailleurs, nous avons établi un calendrier prévisionnel (voir Figure 6 sous forme d'un diagramme de Gantt afin d'optimiser notre organisation. Ce calendrier se divise en trois parties : tout d'abord, les tâches impliquant tous les membres du groupe (voir Figure 7), suivies des tâches spécifiques aux membres travaillant sur iTowns (voir Figure 8), et enfin celles affectées aux membres travaillant sur la partie Minetest (voir Figure 10). De plus, nous avons également conçu un diagramme de répartition des tâches pour ces trois grandes parties de notre projet, ces diagrammes sont situés ci-contre. Concernant la partie serveur du projet, la moitié du groupe s'est mise à travailler dessus.

## 7 Conclusion et Perspectives

La demande de nos commanditaires était tout d'abord de créer un couplage entre le moteur de jeu 3D Minetest et le moteur de visualisation géographique 3D iTowns. Notre produit permet actuellement ce couplage : lorsque l'utilisateur se déplace sur Minetest, la vue sur iTowns se déplace simultanément. De même, lorsque l'utilisateur se déplace sur iTowns, la vue du joueur sur Minetest se déplace également, la vue iTowns étant initialisée sur la même tuile que Minetest.

Cependant, il reste des pistes d'amélioration pour ce couplage. Tout d'abord, une amélioration de la fluidité du couplage pourrait être envisagée afin de maximiser le confort de l'utilisateur. De plus, actuellement, la valeur du champ de vision n'est pas la même sur Minetest et iTowns, ce qui peut entraîner une confusion chez l'utilisateur. Résoudre ce problème serait bénéfique pour l'expérience utilisateur.

La finalité de ce projet serait de permettre la visite à distance de monuments ou de musées à l'aide de ce couplage. Le moteur de jeu 3D Minetest n'étant pas le meilleur pour la visite de l'intérieur des bâtiments, lorsqu'on entrerait dans un bâtiment sur Minetest, une nouvelle vue représentant le modèle 3D de l'intérieur du bâtiment s'ouvrirait pour pouvoir le visiter sur iTowns.

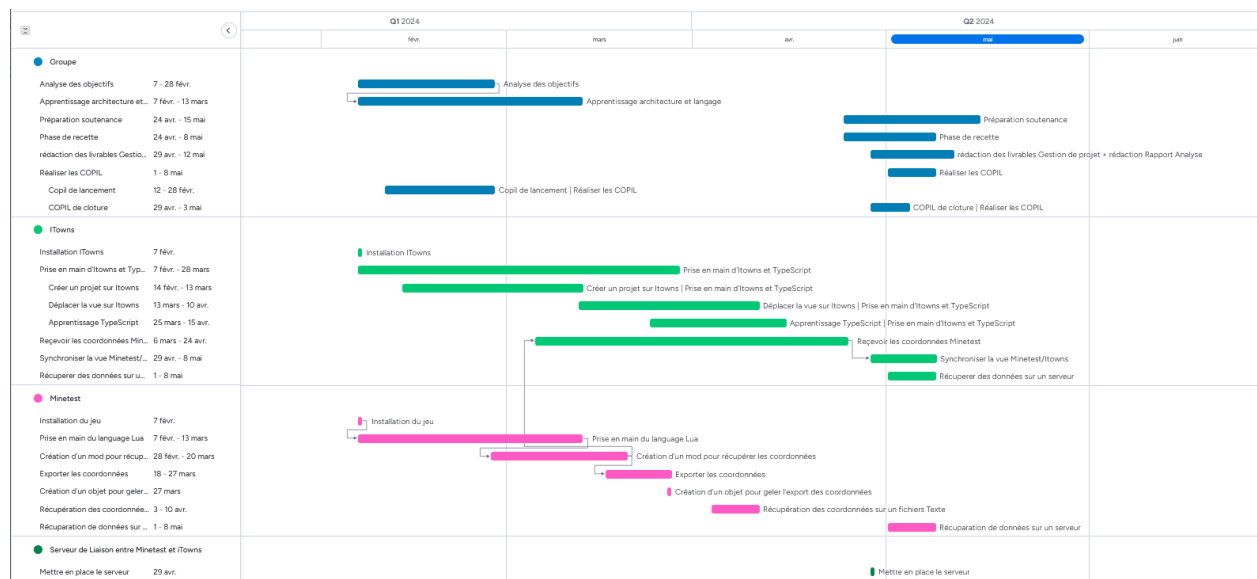


FIGURE 6 – Diagramme de Gantt

### ▼ Groupe

<input type="checkbox"/>	Tâche		Échéances	Statut	Dépendance ⓘ
<input type="checkbox"/>	Analyse des objectifs	+	7 - 28 févr.	Fait	
<input type="checkbox"/>	Apprentissage architecture et langage	+	7 févr. - 13 mars	Fait	Anal...
<input type="checkbox"/>	▼ Réaliser les COPIL 2	+	1 - 8 mai	En cours	
<input type="checkbox"/>	Sous-élément		Statut	Échéances	+
<input type="checkbox"/>	Copil de lancement	+	Fait	12 - 28 févr.	
<input type="checkbox"/>	COPIL de clôture	+	En cours	29 avr. - 3 mai	
<input type="checkbox"/>	+ Ajouter sous-élément				
<input type="checkbox"/>	Préparation soutenance	+	24 avr. - 15 mai	En cours	
<input type="checkbox"/>	Phase de recette	+	24 avr. - 8 mai	En cours	
<input type="checkbox"/>	rédaction des livrables Gestion de projet + rédacti...	+	29 avr. - 12 mai	En cours	
<input type="checkbox"/>	+ Ajouter tâche				
			7 févr. - 15 mai		

FIGURE 7 – Diagramme de Répartition pour les taches de groupes

### ▼ iTowns

<input type="checkbox"/>	Tâche		Échéances	Statut	Dépendance ⓘ
<input type="checkbox"/>	Installation iTowns	+	7 févr.	Fait	
<input type="checkbox"/>	➤ Prise en main d'Itowns et TypeScript 3	+	7 févr. - 28 mars	Fait	
<input type="checkbox"/>	Recevoir les coordonnées Minetest	+	6 mars - 24 avr.	Fait	Créa...
<input type="checkbox"/>	Synchroniser la vue Minetest/Itowns	+	29 avr. - 8 mai	Fait	Reçe...
<input type="checkbox"/>	Récupérer des données sur un serveur	+	1 - 8 mai	Fait	
<input type="checkbox"/>	+ Ajouter tâche				
			7 févr. - 8 mai		

FIGURE 8 – Diagramme de Répartition pour iTowns

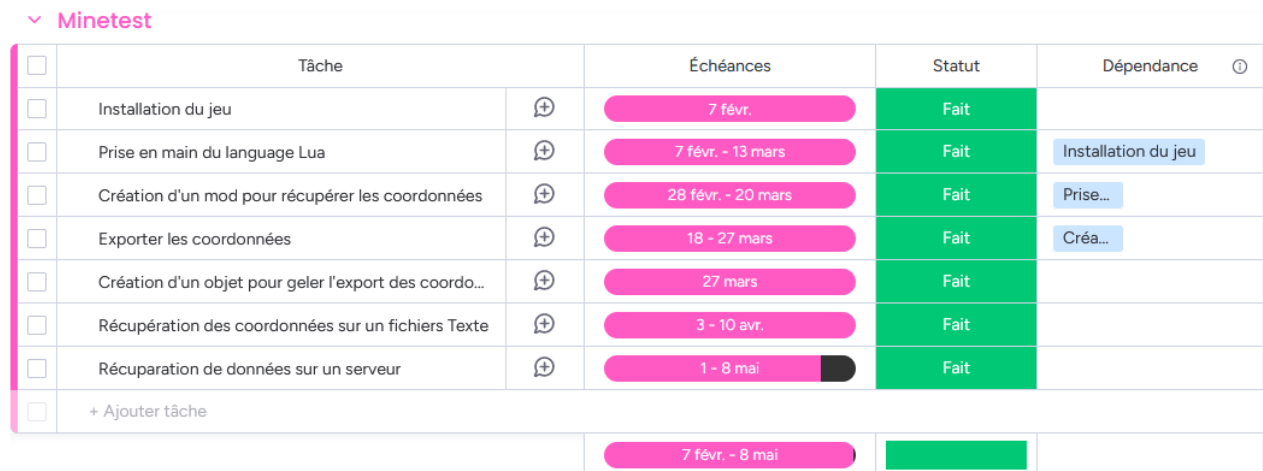


FIGURE 9 – Diagramme de Répartition pour Minetest

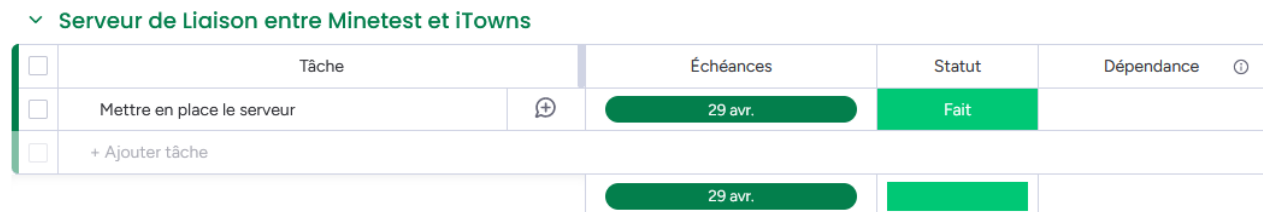


FIGURE 10 – Diagramme de Répartition pour Serveur