

*Rapport de projet de 2e année :*

# **Construction Site Assistant**




*Réalisé par*

*Oliver BELLIARD, Matthieu GOMEZ, et Saïd SAHNOUNE*

*encadré par*

*le Dr. Michel CHAPRON*

# Table des matières

<b>Table des matières.....</b>	<b>1</b>
<b>Acronymes.....</b>	<b>2</b>
<b>Objectifs.....</b>	<b>3</b>
<b>Analyse fonctionnelle.....</b>	<b>3</b>
Fonctions principales (FP).....	3
Fonctions de contrainte (FC).....	3
Fonctionnalités optionnelles.....	3
<b>Cahier des charges fonctionnel.....</b>	<b>4</b>
FP1 : Modèle virtuel interactif.....	4
Description.....	4
Spécifications.....	4
FP2 : Positionnement correct du modèle 3D dans l'espace réel.....	5
Description.....	5
Spécifications.....	5
FC1 : Ordre de construction des éléments du chantier.....	5
Description.....	5
Spécifications.....	5
FC2 : Gestion en temps réel des stocks de matériel.....	6
Description.....	6
Spécifications.....	6
FC3 : Créer une version mobile de l'assistant.....	6
Description.....	6
Spécifications.....	6
FC4 : Calcul des flux des vecteurs de ressources.....	7
Description.....	7
Spécifications.....	7
FC5 : Définition de zones de danger en direct.....	7
Description.....	7
Spécifications.....	7
<b>Diagramme d'architecture.....</b>	<b>8</b>
<b>Gantt.....</b>	<b>9</b>
<b>Management du projet.....</b>	<b>10</b>
Trello  .....	10
Notion  .....	10
GitHub  .....	12
<b>Application Meta Quest.....</b>	<b>12</b>
Unity.....	12
Scène principale.....	13



Assistant de chantier.....	16
<b>IFC à jour.....</b>	<b>17</b>
<b>Conclusion.....</b>	<b>17</b>

# Acronymes

**AR** : Augmented Reality, Réalité Augmentée

**VR** : Virtual Reality, Réalité Virtuelle

**XR** : Augmented or Virtual Reality, Réalité Augmentée ou Virtuelle

**MR** : Mixed Reality, Réalité Mixte

**3D** : [en] trois dimensions

**UI** : User Interface, Interface Utilisateur

**SDK** : Software Development Kit, Kit de développement logiciel

**API** : Application Programming Interface, Interface de Programmation d'Application

**CSA** : Construction Site Assistant

**MVC** : Model, View and Control

**IFC** : signifie Industry Foundation Classes ou Information For Construction

**BIM** : Building Information Model

# Objectifs

Créer une application exploitant la réalité augmentée appliqué à la conduite d'un chantier dans le but d'optimiser son déroulement. En affichant un modèle 3D, à l'échelle et l'emplacement prévu dans les plans, dans lequel chaque élément est interactif, l'opérateur pourra obtenir des informations sur ces derniers et personnaliser l'affichage du modèle en fonction de ses besoins.

En permettant à cette application de communiquer avec un serveur, le modèle du plan virtuel et les informations seront actualisés par des utilisateurs partie prenante du chantier en temps réel.

L'application est développée pour le casque de réalité augmentée et virtuelle Quest2 mais elle serait compatible avec les autres casques Quest, notamment le Quest3, le dernier à cette date, un appareil que nous avons pu nous procurer par la suite avant la fin du projet.

De plus, une version mobile est envisagée pour permettre de rendre l'application plus accessible une fois que la version pour casque AR/VR serait fonctionnelle.

## Analyse fonctionnelle

### Fonctions principales (FP)

1. Modèle virtuel interactif
2. Positionnement correct du modèle 3D dans l'espace réel

### Fonctions de contrainte (FC)

1. Ordre de construction des éléments du chantier
2. Gestion en temps réel des stocks de matériel
3. Créer une version mobile de l'assistant
4. Calcul des flux des vecteurs de ressources
5. Définition de zones de danger en direct

### Fonctionnalités optionnelles

1. Relier le programme à un serveur pour actualiser les données en temps réel à afficher
2. Faire en sorte que l'on puisse actualiser les données depuis l'interface AR

# Cahier des charges fonctionnel

## FP1 : Modèle virtuel interactif

### Description

Permettre à l'utilisateur d'interagir avec des plans 3D d'un chantier à dimensions réelles ou réglables. Permettre en addition de connaître lors des interactions des informations sur la composition et/ou la fabrication de l'élément sélectionné.

### Spécifications

- a. Modèle principal décomposable en éléments uniques interactif
- b. Automatiquement :
  - i. Change la couleur des éléments sélectionnés
  - ii. N'affiche plus les éléments terminés (sauf demande de l'utilisateur)
  - iii. Afficher l'ordre de construction
- c. Sur demande :
  - i. Permet de n'afficher que certains éléments sur demande
  - ii. Afficher la Composition d'un élément
  - iii. Changer l'état de construction d'un ou plusieurs éléments

## FP2 : Positionnement correct du modèle 3D dans l'espace réel

### Description

Positionner correctement dans l'espace réel perçu en réalité augmentée le modèle 3D du chantier, de façon automatique, sans intervention de l'utilisateur lors des utilisations répétées.

### Spécifications

- Précision : +/- 50 cm
- Limites : frontières du terrain de construction
- Contraintes des dimensions : idéalement 1 km<sup>2</sup>,  $\pm 100m^2$

## FC1 : Ordre de construction des éléments du chantier

### Description

A partir du fichier fourni par l'utilisateur, pouvoir établir l'ordre de construction des éléments constituant le bâtiment sans intervention manuelle de l'utilisateur. Pouvoir par la suite proposer un ordre de construction du bâtiment.

### Spécifications

- Prendre en compte les barrières physiques formées en cours de construction.
- Prendre en compte la variation des stocks quand un changement est reporté.
- Proposer un ordre de construction qui permette d'exploiter au maximum la structure du bâtiment construit pour le transport des ressources.

## FC2 : Gestion en temps réel des stocks de matériel

### Description

Permettre de gérer les variations de stocks et de maintenir l'utilisateur informé des changements grâce à l'interface utilisateur.

### Spécifications

Permettre, depuis l'interface utilisateur :

- L'affichage des quantités
- Avoir des options d'ajout de matériel
- Avoir des options de suppression de matériel

## FC3 : Créer une version mobile de l'assistant

### Description

Avoir une version mobile pour faciliter l'accès aux plans 3D aux ouvriers sur les chantiers.

### Spécifications

Permettre depuis un smartphone ou une tablette de :

- Naviguer au sein du modèle 3D interactif
- Changer le status de construction des éléments
- Changer le statut des stocks
- Signaler une zone de danger
- Voir les trajets de déplacement prévus pour les différents vecteurs



## FC4 : Calcul des flux des vecteurs de ressources

### Description

Calculer les vecteurs de déplacement des flux de ressources humaines et matérielles en tenant compte des obstacles et de l'avancement du chantier.

Calculer les parcours de livraison des ressources en prenant compte de l'état en temps réel du chantier.

### Spécifications

- Temps réel : 15 minutes après un changement du terrain ou l'apparition d'un obstacle.
- Visibilité : visible depuis tous les appareils.
- Mise à jour en temps réel

## FC5 : Définition de zones de danger en direct

### Description

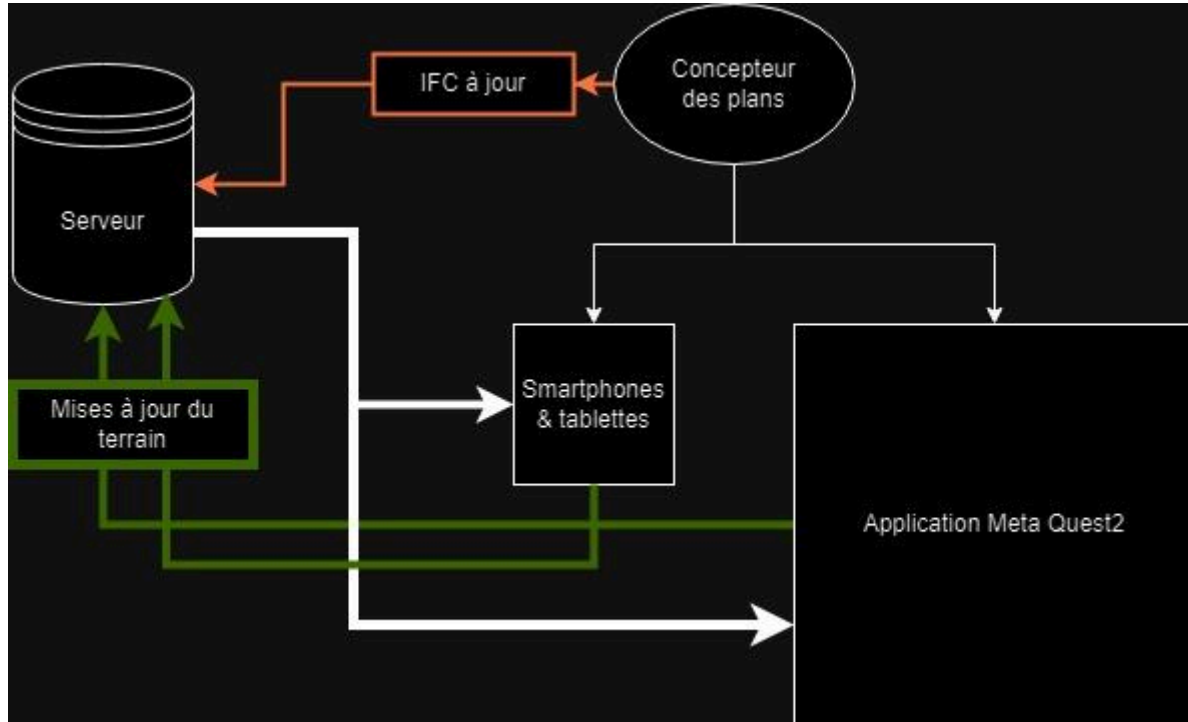
Permettre à tous les utilisateurs de signaler des obstacles et des zones de danger depuis tous les appareils supportés.

### Spécifications

Indication des obstacles : reconnaissance grâce à des outils de numérisation d'objets 3D ou un pointeur de délimitation du terrain du monde de réalité augmentée.

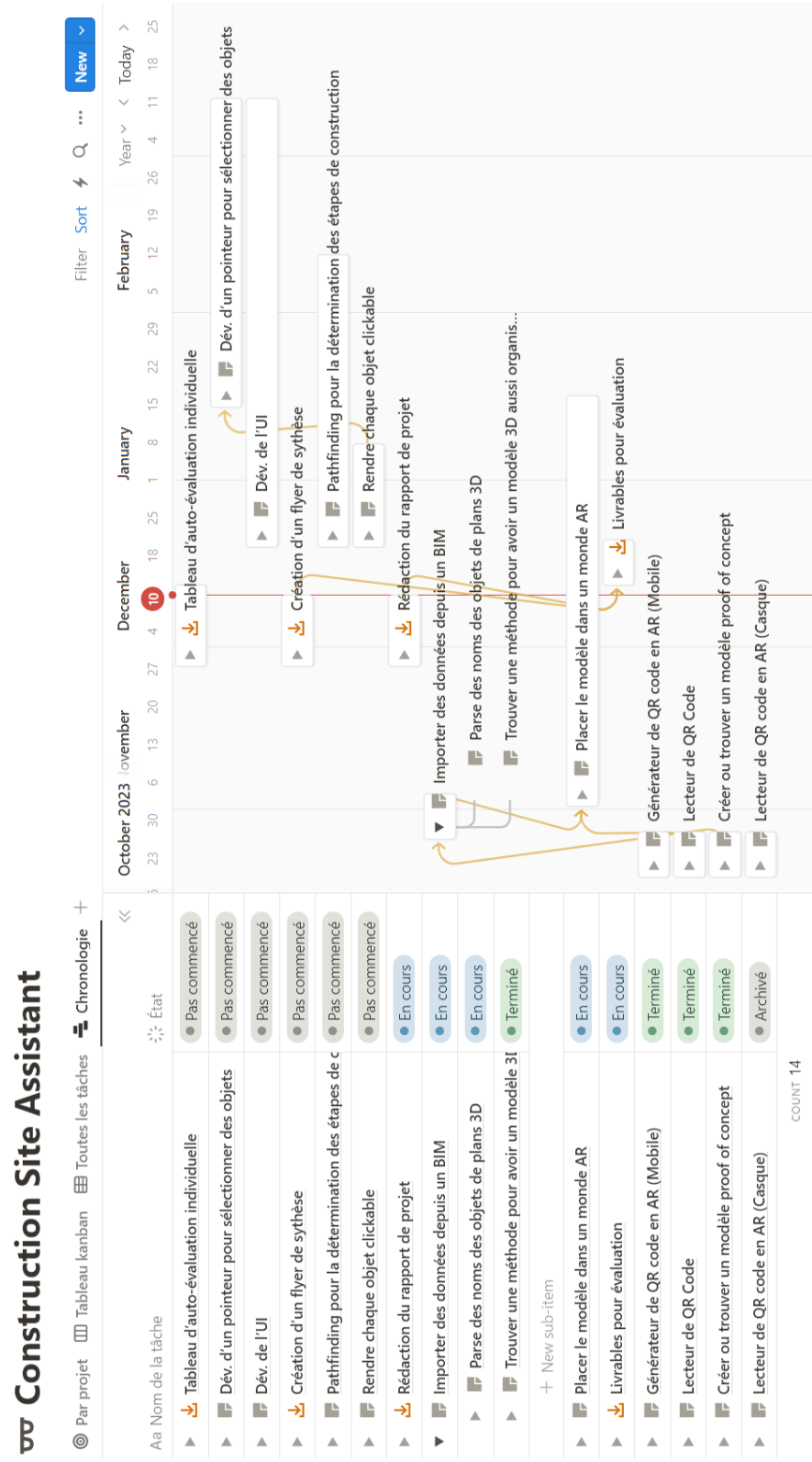
Indication des zones de danger : utilisation d'un pointeur pour marquer les surfaces du monde de réalité augmentée.

# Diagramme d'architecture



Notre diagramme d'architecture montre l'interaction entre les différents appareils du système dans leur globalité. Nous n'avons pas pu établir un diagramme d'architecture fixe dû aux différentes découvertes faites pendant les recherches de solution aux différentes contraintes. Cependant, le fonctionnement montré à travers celui-ci reste l'objectif visé décrit par l'analyse fonctionnelle.

# Gantt



# Management du projet

Pour les séances de projet nous avons suivi au maximum la planification que nous avons prévue, illustrée à travers le diagramme de Gantt. Pour gérer le projet nous avons utilisé principalement les 3 outils suivants.

## Trello

Cet outil non adapté aux projets confidentiels mais qui s'adapte totalement à notre projet nous a permis, notamment en début de projet, de pouvoir faire du brainstorming, de lister nos idées et de trier les résultats des différentes recherches que nous avons effectuées avant la planification du projet. Dessus nous avons listé les grandes lignes des étapes que nous savions que nous allions devoir suivre puis les détails et les idées que nous avons abandonnés avec les raisons détaillées du pourquoi.

Le lien vers le Trello du projet est le suivant : <https://trello.com/w/projetassistantchantier2a/>

De plus, comme il nous a servi comme dépôt de tous nos résultats de recherche, il contient également les ressources qui nous ont permis de surmonter des obstacles variés à des moments cruciaux du projet.

Ayant de l'information sensible et d'ordre confidentiel pour l'établissement, notre tableau principal est privé mais accessible sur demande avec un compte Trello. Pour cela il suffit de nous contacter.

## Notion

Cet outil gratuit nous a principalement permis de lister la liste de tâches sur une base de données spécialement conçue pour le projet grâce à ses options de personnalisation de tableaux et les outils qu'il possède qui permettent de tirer parti de données basiques comme une date, une étiquette ou même un fichier. Nous avons choisi cet outil car il nous a permis de ne pas polluer notre banc de ressources, Trello, et car la version gratuite ne permet pas le travail collaboratif. De plus, les outils de visualisation des dates comme la visualisation sous forme de diagramme de Gantt nous a permis une meilleure gestion de notre temps tout au long du projet. Veuillez trouver le lien vers la page Notion du projet est ci-dessous :

<https://oliver-benjamin.notion.site/adea81ec6a7540518e80d6fe5fd57195?v=632ea0d2a33b4f16ba4129930be465de&pvs=4>

Nous pouvons noter que lors de la dernière séance, la liste de tâches complétés que nous nous sommes fixés est celle ci-dessous. On peut voir que nous avons pu mettre en place la majorité de fonctionnalités que nous avons prévu pour l'application. Aussi, la liste n'a pas dû être changée depuis sa création si ce n'est pour mettre à jour l'état des tâches :

▼ 🍏 Project management 4

Aa Nom de la tâche	⚙ État	📅 Échéance
📄 Livrables pour évaluation	● Terminé	December 12, 2023
📄 Rédaction du rapport de projet	● Terminé	November 27, 2023 → December 10, 2023
📄 Création d'un flyer de sythèse	● Terminé	November 27, 2023 → December 10, 2023
📄 Tableau d'auto-évaluation individuelle	● Terminé	November 27, 2023 → December 12, 2023

+ New

COMPLETE 4/4

▼ ➡️ Tâches 11

Aa Nom de la tâche	⚙ État	📅 Échéance
📄 Créer ou trouver un modèle proof of concept	● Terminé	October 19, 2023 → October 24, 2023
📄 Lecteur de QR Code	● Terminé	October 19, 2023 → October 23, 2023
📄 Placer le modèle dans un monde AR	● Terminé	November 1, 2023 → January 16, 2024
▶ 📄 Rendre chaque objet clickable	● Terminé	December 19, 2023 → January 7, 2024
▶ 📄 Importer des données depuis un BIM	● Terminé	October 26, 2023 → October 30, 2023
▶ 📄 Détermination des étapes de construction	● Pas comme...	December 19, 2023 → February 11, 2024
📄 Lecteur de QR code en AR (Casque)	● Archivé	October 19, 2023 → October 23, 2023
📄 Générateur de QR code en AR (Mobile)	● Terminé	October 19, 2023 → October 23, 2023
📄 Dév. de l'UI	● Terminé	December 19, 2023 → March 12, 2024
📄 Dév. d'un pointeur pour sélectionner des objets	● Terminé	January 14, 2024 → March 12, 2024
📄 Tests d'échèle du modèle en monde AR	● En cours	December 17, 2023 → March 12, 2024

COMPLETE 9/11

Comme base de données avec suivi des versions de tout le projet nous avons choisi GitHub. Dessus nous avons stocké tous les fichiers du projet (documentation, projet Unity, modèles 3D et logiciels de conversion). Vous trouverez le lien menant au GitHub du projet ci-dessous :

<https://github.com/MatthieuGomes/ConstructionSiteAssistant>

L'essentiel du projet avec l'historique des versions est dessus. Il faut néanmoins noter que la première version du projet avec une fonctionnalité que nous pensions simple à implémenter et que nous avons mis de côté n'est pas dessus pour des soucis de cohérence avec tout ce qu'il y a dessus. En revanche, il y a la version épurée du projet avec l'avancement et la liste des changements pour chaque mise à jour (commit sur la branche principale, Main).

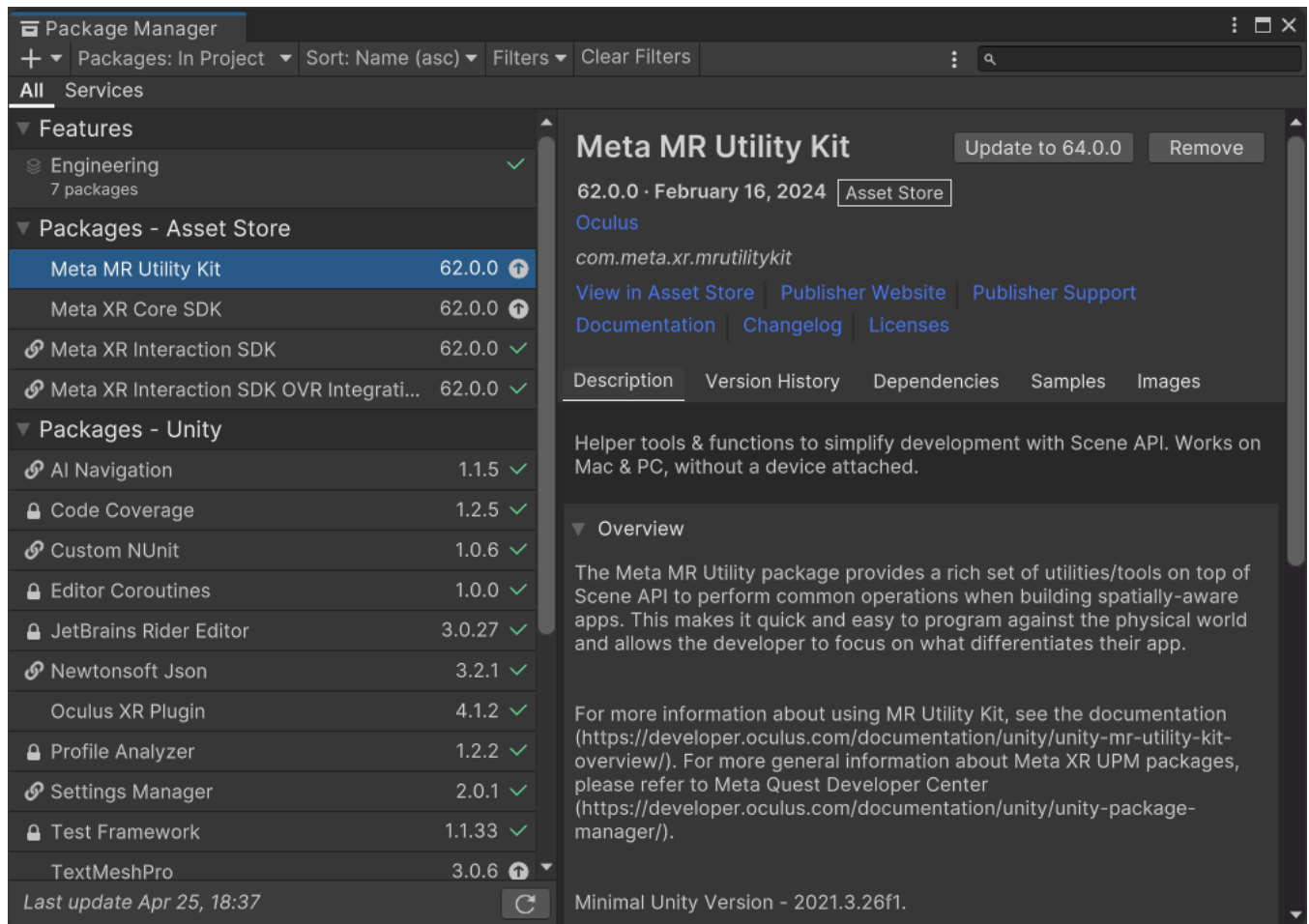
# Application Meta Quest

## Unity

Même si nous avons prévu tout un système complet pour le projet, avec le temps très limité à notre disposition, notre priorité était focalisée sur l'application car sans connaître son architecture il n'était pas possible de réaliser les autres parties du système. L'application a été réalisée sur le logiciel Unity, un moteur de jeu permettant d'exporter les applications créées avec celui-ci sur diverses plateformes telles que des consoles, les PC, des smartphones et dans notre cas, en exploitant l'exportation vers des appareils Android, des casques VR/AR (référés comme XR sur le logiciel).

La création d'applications sur Unity est relativement simple car le logiciel se charge de toute la partie technique concernant le déploiement et permet de se focaliser exclusivement sur le contenu. Lors de la création d'un projet pour des appareils Meta Quest, en plus des outils déjà existants faisant partie intégrante du moteur, il faut inclure les "package" Meta Quest. Les principaux sont le package "Meta XR Core SDK" et "Meta MR Utility Kit". Le premier se charge d'inclure dans le projet tous les fichiers nécessaires à adapter le SDK déjà existant dans Unity pour la XR aux casques Meta Quest ainsi qu'un assistant qui permet de configurer automatiquement le projet en quelques clics et un assistant permettant de construire des scènes avec des "Building Blocks", des Gameobjects préconfigurés pour fonctionner avec les appareils Meta Quest pour que la conception des scènes soit aussi proche du Plug & Play que possible.

Au jour où nous avons arrêté le développement du projet nous étions à la version 62 du SDK de Meta et il fallait seulement les packages de la capture ci-dessous pour créer une application XR fonctionnelle :



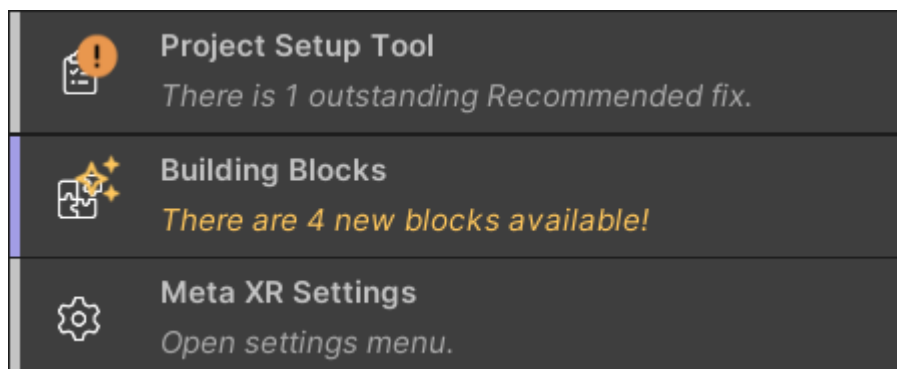
## Scène principale

La mise en place d'une scène basique pour une application XR a beaucoup évolué tout au long du projet. Quand nous avons commencé le projet nous étions à la version 54 de ce qui était connu à ce moment là comme l'API de Meta (qui a été fusionnée par la suite au Meta XR Core SDK). Ce package utilisait le SDK android présent sur Unity et l'ajout du package Meta XR Core SDK était optionnel. Mais cela demandait de configurer chaque objet d'une scène à la main, ce qui rendait très long la mise en place d'une scène simple pour qu'elle puisse fonctionner avec le casque en utilisant toutes les fonctionnalités mises à disposition.

Heureusement, pour les versions 60+ des outils qui se chargent de configurer les objets nécessaires à une scène et le projet ont été développés pour retirer ce fardeau très chronophage. La méthode pour créer une scène dans la version 62, la dernière que nous avons pu essayer, ressemble au processus qui suit. Cliquer sur le logo Meta de la partie inférieure droite de l'interface :

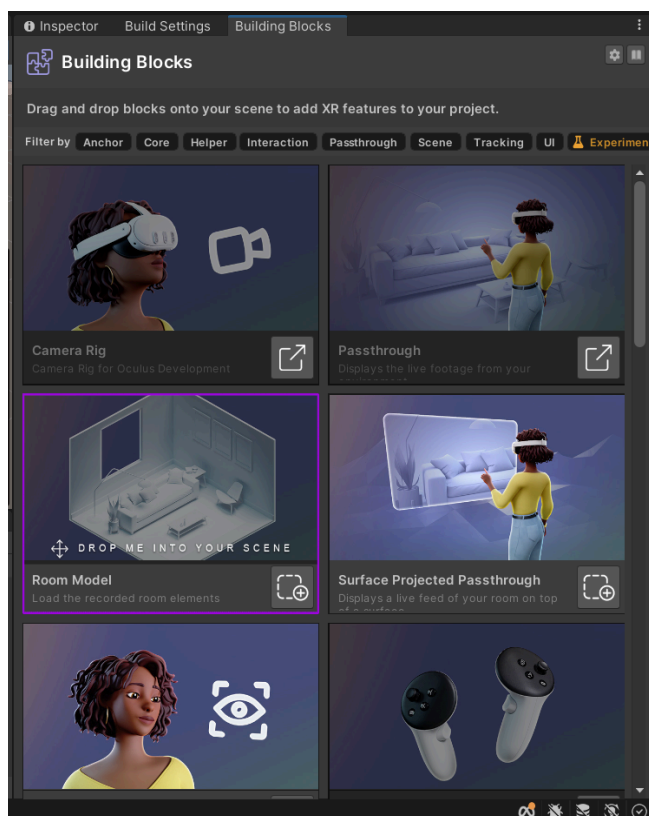


Ce qui ouvrirait le menu déroulant suivant :



On voit alors 3 catégories, la première, Project Setup Tool qui indique quand le projet est mal configuré et permet de corriger le problème automatiquement en un clic par problème (de façon générale un problème de configuration entraîne un échec au moment du build ou de l'ouverture depuis le casque), Building Blocks qui contient les parties nécessaires pour mettre en place une scène et Meta XR Settings pour paramétrer l'API plus en profondeur, chose qui n'est normalement pas nécessaire à cette version du SDK.

La fenêtre de Building Blocks est celle qui permet de remplir la scène avec les objets nécessaires pour des fonctionnalités basiques comme le Passthrough ou la visualisation d'un double virtuel 3D des manettes dans l'espace. A notre version cette fenêtre est tel que le montre la capture ci dessous :





Pour ajouter un Building Block, notamment les élémentaires comme le “Camera Rig” ou le “Passthrough”, il suffit de cliquer sur le bouton d’ajout :



Ce bouton ajoute les objets concernés à la scène au bon endroit car les éléments de l’API de Meta sont sensibles aux objets parent et fils. Pour notre scène de démonstrations, celle de la dernière version disponible sur GitHub, nous avons ajouté les Building Blocks suivants :

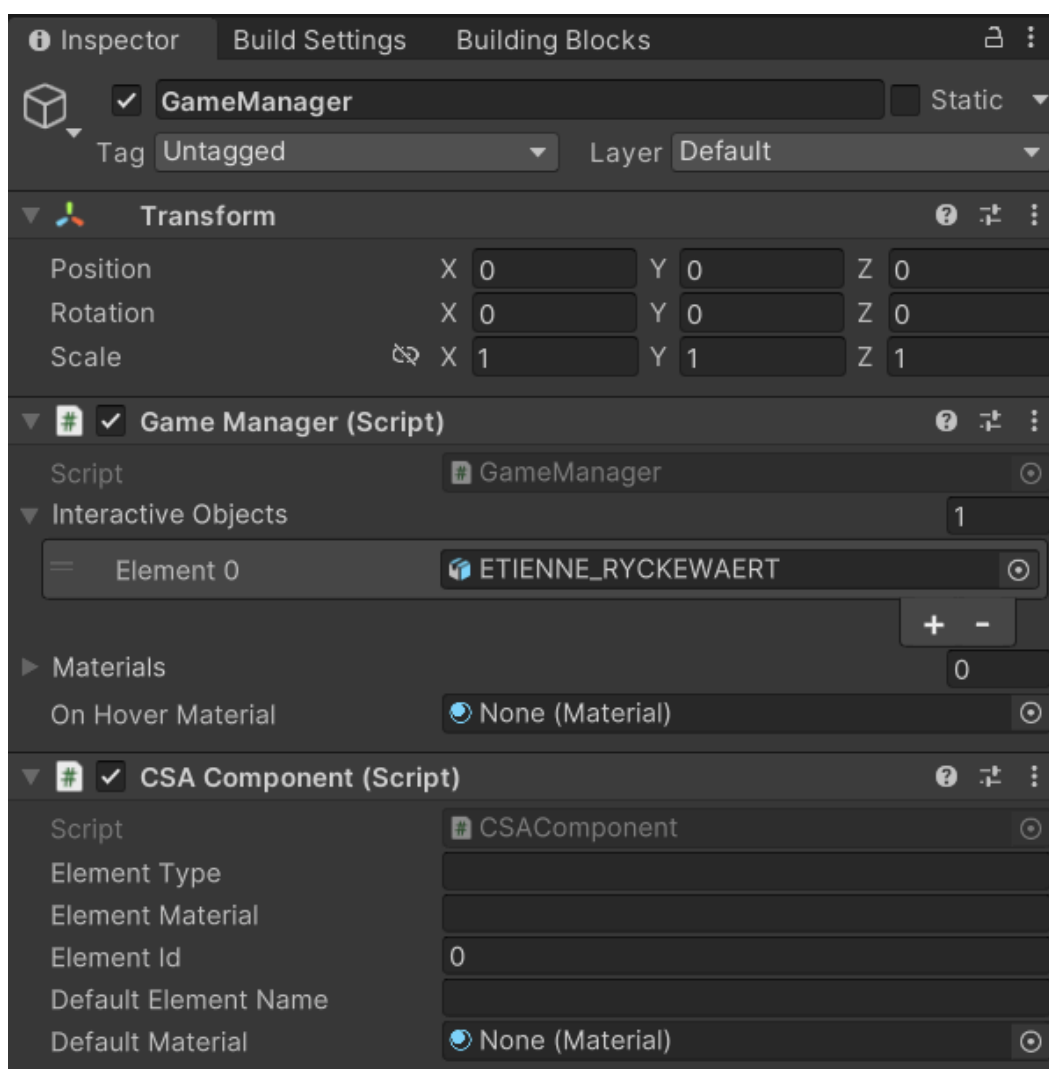


Avec ces building blocks nous obtenons une scène qui permet d'être en AR (Passthrough), d'interagir avec l'application au travers des mains ou des manettes, avec un aperçu 3D des mains et des manettes selon celui qui est la commande active.

Grâce à tous les outils qui simplifient la mise en place d'une scène, c'est tout ce qu'il faut pour faire une scène permettant d'avoir une application pour Meta Quest fonctionnelle à la version 62 de l'API de Meta.

## Assistant de chantier

L'application qui sert d'assistant de chantier est construite sur la base d'une application pour Meta Quest, expliquée juste avant. Les outils que nous avons développés pour notre application sont les objets suivants :



Nous avons développé un modèle de données (suivant la philosophie MVC) pour notre application, c'est le rôle du script "CSA Component". Il extrait les données contenues dans les noms des éléments de plans 3D avec une fonction de parse. Pour être sûrs que ce modèle de données est respecté sur l'entièreté des éléments que nous souhaitons standardiser, nous avons développé un outil reconnu comme un GameObject spécial par Unity : le

Game Manager. C'est notre objet de contrôle, toujours sous la philosophie MVC. Pour conclure par rapport à cette logique, notre objet de vue est sensé être notre UI mais à cause de contraintes de délais, notre View est l'interface de Unity qui permet en plus de déboguer et vérifier que les scripts fonctionnent.

Le Game Manager permet d'attacher le script sur tous les éléments dans le tableau de "Interactive Objects". En plus, une fonctionnalité que nous n'avons pas réussi à automatiser mais qui fonctionne est de rendre cliquable des objets avec un matériau spécial (pour pouvoir avoir un retour visuel des clics).

## IFC à jour

Les IFC<sup>1</sup> sont un format d'échange de données ouvert (non-propriétaire) qui permet d'échanger des données liées à un projet BIM, quels que soient les outils informatiques des divers acteurs. Quand nous avons un IFC à jour, pour pouvoir l'inclure dans le projet il faut le transformer en un format reconnu nativement par Unity. Pour notre cas nous avons choisi le format **.dae** qui est compatible avec la plupart des logiciels de modélisation 3D et permet de retenir la hiérarchie ainsi que les noms des éléments.

Le processus de conversion est détaillé sur le README du GitHub du projet et il fonctionne sur divers systèmes car l'outil de conversion est libre et développé de façon à être un script Bash à l'origine.

## Conclusion

Même si cela ne se reflète pas forcément sur le rapport, il y a eu un grand travail de recherche pour réussir à synthétiser les méthodes décrites avant et planifier le projet le mieux possible. Le gros du travail est surtout visible sur Git Hub et Trello étant donné qu'ils répertorient tous nos résultats aux différentes étapes de nos découvertes et surtout à chaque mise à jour de l'API de Meta. Nous tenons à mentionner que la mise à jour des versions 5X à 6X ont totalement changé le workflow de développement d'application pour un flux simplifié et plus proche de ce qui a été décrit dans ce rapport.

Comme à chaque séance nous avons pris le soin de débriefer et mettre à jour le tableau sur Trello de notre avancement, suivre le progrès et maintenir à jour le Gantt n'a pas été d'une grande difficulté. De plus, avec la quantité de recherche effectuée au début, nous avons une bonne idée de ce qui allait être possible de réaliser avec les outils à disposition.

---

<sup>1</sup> Plus d'information sur les IFC sur : <https://buildingsmartfrance-mediaconstruct.fr/comprendre-format-ifc/>