



*inria*

anr<sup>®</sup> UGA



université  
PARIS-SACLAY



Institut de Recherche  
en Informatique de Toulouse  
CNRS - INP - UT3 - UT1 - UT2J



# Rapport de Stage de PFE

*Interactions en Réalité Mixte pour la collaboration entre  
environnements multi-écrans distants*

**Nom Prénom :** SERAFIN Thomas

**Email étudiant :** thomas.serafin@grenoble-inp.org

**Tuteur IRIT :** Arthur FAGES

**Email IRIT :** arthur.fages@irit.fr

**Tuteur École :** Valérie PERRIER

**Email École :** valerie.perrier@grenoble-inp.fr

**Président du Jury :** Antoine FRENOY

**Formation :** Ensimag — 3e année

**Année universitaire :** 2024–2025

Ce stage a été financé par le PEPR eNSEMBLE

Toulouse, le November 6, 2025

# Contents

<b>1 Présentation générale du stage et de la problématique étudiée</b>	<b>3</b>
1.1 Contexte . . . . .	3
1.2 Problématiques et objectifs . . . . .	4
1.3 Approche . . . . .	5
1.3.1 La Réalité Mixte, qu'est-ce que c'est ? . . . . .	5
1.3.2 Qu'apporte la Réalité Mixte dans la résolution du problème? . . . . .	5
1.4 Outils utilisés . . . . .	6
1.5 Structure d'accueil . . . . .	7
<b>2 Synthèse de l'existant en terme de collaboration multi-écrans à distance</b>	<b>7</b>
2.1 Les outils de visioconférence et leurs limites . . . . .	7
2.2 Desk2Desk . . . . .	8
2.3 MiniMates . . . . .	9
2.4 Volumetric Hybrid Workspaces . . . . .	10
2.5 HoloDevice . . . . .	11
2.6 Using Virtual Replicas . . . . .	12
2.7 Solutions apportées et insuffisances de l'état de l'art . . . . .	13
<b>3 Solution envisagée : nouvelles interactions basées sur la métaphore du "diorama à judas"</b>	<b>14</b>
3.1 Explication du concept de "diorama à judas" . . . . .	14
3.2 Pourquoi cette idée répond aux insuffisances de l'état de l'art . . . . .	16
3.3 Cas d'usage envisagé . . . . .	16
<b>4 Représentation d'un environnement de travail distant dans le diorama</b>	<b>18</b>
4.1 La pièce: contenant les écrans et l'avatar . . . . .	18
4.2 L'avatar: une représentation 3D du collaborateur à l'intérieur du diorama . . . . .	18
4.2.1 Intégration réseau: Netcode et RPC . . . . .	19
4.2.2 Calibration de l'avatar . . . . .	20
4.2.3 Synchronisation des joints des mains . . . . .	21
<b>5 Interaction avec le diorama</b>	<b>22</b>
5.1 Le judas: voir à l'intérieur du diorama . . . . .	22
5.1.1 Les stencils de MRTK . . . . .	22
5.2 Contrôles du diorama . . . . .	23
5.2.1 Ajuster la taille du judas . . . . .	23
5.2.2 Déplacer le diorama dans l'espace de travail . . . . .	23
5.2.3 Changer de point de vue dans le diorama . . . . .	23
<b>6 Communication avec un collaborateur grâce au diorama</b>	<b>24</b>
6.1 Gestes sur un écran local . . . . .	24
6.1.1 Virtualisation et placement écrans . . . . .	24
6.2 Gestes sur les écrans du collaborateur . . . . .	25
6.2.1 Détection de la main à travers le diorama . . . . .	25
6.2.2 Affichage des mains dans la scène du collaborateur . . . . .	27
<b>7 Limites et perspectives</b>	<b>28</b>
7.1 Les limites du diorama . . . . .	28
7.2 Perspectives d'évaluation . . . . .	29
<b>8 Déroulement du stage et difficultés rencontrées</b>	<b>30</b>
8.1 Interaction avec les membres de l'équipe . . . . .	30
8.2 Divergence entre le travail prévu et le travail effectif . . . . .	30
8.3 Problèmes rencontrés d'un point de vue général . . . . .	31
8.4 Cas du positionnement des mains lors du partage d'écran: un exemple chronophage . . . . .	32

<b>9 Impacts environnemental et sociétaux du système développé</b>	<b>34</b>
9.1 Consommation énergétique . . . . .	34
9.2 Impact sur la collaboration à distance . . . . .	34
9.3 Contexte de développement . . . . .	35
<b>10 Conclusion</b>	<b>35</b>
10.1 Synthèse du travail réalisé . . . . .	35
10.2 Apports techniques et personnels . . . . .	36
<b>A Prise en main de Unity</b>	<b>37</b>
A.1 Configuration pour Android et packages . . . . .	37
A.2 Ajout du PassThrough . . . . .	37
<b>B Schéma illustrant le partage des écrans une fois virtualisés</b>	<b>37</b>
<b>C Glossaire des éléments techniques de Unity</b>	<b>39</b>
C.1 GameObject . . . . .	39
C.2 Prefab Unity . . . . .	39
C.3 Script C# . . . . .	39
C.4 Scene View et Game View . . . . .	40
C.5 Dossier Assets . . . . .	40

# 1 Présentation générale du stage et de la problématique étudiée

## 1.1 Contexte

Avec la généralisation des environnements de travail multi-écran, comme illustré sur la Figure 1 [10] (ordinateurs, tablettes, smartphones, tableaux interactifs, etc.), les pratiques collaboratives évoluent, tant dans un cadre professionnel que personnel. Toutefois, dans un contexte de télétravail croissant et de réduction des déplacements, ces collaborations doivent souvent s'effectuer à distance, ce qui pose de nouveaux défis en matière d'interaction et de perception partagée des espaces de travail.

Les outils de visioconférence traditionnels, comme Zoom ou Discord, ne permettent pas de supporter efficacement la communication gestuelle — les gestes qui accompagnent la parole ou servent à montrer des éléments sur son écran ou celui d'un collaborateur. En effet, on se limite souvent à une fenêtre de caméra pour voir les autres participants et à la souris pour indiquer des éléments, ce qui devient rapidement insuffisant, notamment lorsque plusieurs écrans sont utilisés simultanément. Ce type de limitation est particulièrement visible dans des contextes complexes, tels que les salles de gestion de crise multi-écrans, où plusieurs collaborateurs distants souhaitent interagir de manière fluide.

Par ailleurs, comme le soulignent Jiang et al. [10], les environnements numériques actuels favorisent des flux de travail fragmentés, dans lesquels les utilisateurs répartissent ou déplacent leurs activités entre plusieurs écrans. Cette fragmentation peut entraîner des ruptures dans la continuité de l'interaction et de la communication, en particulier lorsqu'il devient difficile de maintenir une coordination fluide entre les actions effectuées sur différents dispositifs. Dans le cadre d'une collaboration à distance, cette discontinuité est aggravée par l'absence de présence physique et de communication gestuelle, rendant d'autant plus nécessaire le développement de solutions qui intègrent et synchronisent mieux les interactions multi-appareils.



Figure 1: Différentes configurations d'espaces de travail multi-écran [10]

## 1.2 Problématiques et objectifs



Figure 2: Collaborateurs présents au même endroit

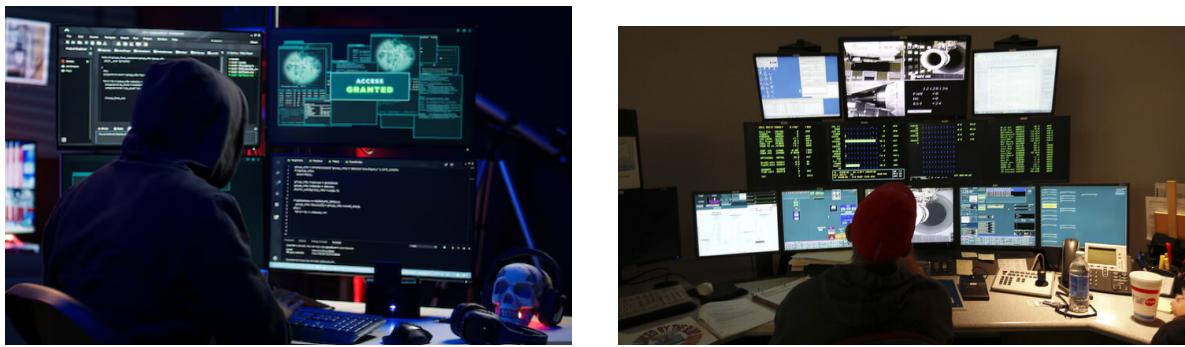


Figure 3: Deux collaborateurs distants avec chacun son environnement multi-écran

Lorsque deux collaborateurs sont présents l'un à côté de l'autre dans une même pièce contenant des écrans, tous les deux voient le même contenu. Par exemple, sur la Figure 2, l'un peut montrer des éléments sur les écrans et l'autre le verra faire ce geste. Or, lorsque les collaborateurs travaillent à distance et disposent chacun de leur propre environnement multi-écran (Figure 3), la séparation physique entre eux soulève plusieurs problèmes. En tant qu'utilisateur, je ne vois plus les écrans de mon collègue ni la manière dont ils sont agencés dans son espace de travail. Ainsi, les références déictiques ne peuvent plus être utilisées. On pourrait imaginer un utilisateur qui montre des éléments sur ses écrans avec ses mains ou produit des gestes complexes en même temps qu'il parle à son collègue distant pour produire des explications. Par exemple, cet utilisateur pourrait dire à son collègue : "regarde sur mon écran de droite et sur celui de gauche en montrant des éléments de ses écrans avec ses deux mains" ou "le phénomène de dispersion se produit de cette manière" tout en effectuant un geste complexe avec ses mains pour illustrer le phénomène dont il parle. Le collègue distant ne peut absolument pas percevoir la gestuelle de l'utilisateur qui interagit avec son environnement multi-écran. De plus, en tant qu'utilisateur, je ne peux pas non plus effectuer des gestes dans l'environnement du collègue distant comme lui montrer un élément sur ses écrans ou produire des explications sur ses écrans avec des gestes complexes. En effet, tout l'intérêt d'une collaboration réside dans sa nature bilatérale : chaque utilisateur doit pouvoir interagir avec l'environnement de travail de l'autre.

Ainsi, l'ensemble de ces questionnements pourraient se décomposer en trois problématiques :

- P1. Comment partager, entre collaborateurs, le contenu et la disposition des écrans de leur environnement de travail ?
- P2. Comment, en tant qu'utilisateur, puis-je communiquer mes gestes sur mes écrans à un collègue distant ?

- P3. Comment effectuer des gestes dans l'environnement d'un collaborateur distant?

Pour addresser ces problématiques, on cherchera à atteindre ces objectifs :

Objectif 1 : Permettre la perception de l'espace de travail du collaborateur ainsi que de l'agencement de ses écrans dans l'espace de l'utilisateur. Connaître l'agencement des écrans dans l'espace de travail du collaborateur permettra alors d'utiliser des références déictiques entre collaborateurs.

Objectif 2 : Partager la position de la tête et des mains d'un utilisateur avec son collaborateur. Ainsi, la représentation de l'agencement des écrans, couplée à celle du collaborateur distant, permettra de communiquer les gestes d'un collaborateur sur ses propres écrans.

Objectif 3 : Développer une interaction avec l'espace de travail du collaborateur distant, afin de pouvoir faire des gestes complexes sur les écrans du collaborateur distant.

En complément de la communication orale, la communication entre humains repose également sur le non-verbal, qui joue un rôle essentiel dans la compréhension mutuelle. Il est intéressant de noter que, au-delà de la transmission d'informations par des gestes sur les écrans, le partage des mouvements du collaborateur entre utilisateurs permet de rétablir la communication non verbale, laquelle disparaît lorsque les individus sont physiquement séparés.

### 1.3 Approche

Durant ce stage, j'ai utilisé la Réalité Mixte pour atteindre les objectifs. Présentons tout d'abord ce que c'est.

#### 1.3.1 La Réalité Mixte, qu'est-ce que c'est ?

La Réalité Augmentée (RA) est une technologie qui superpose des éléments virtuels (images, sons, informations) à notre environnement réel à l'aide d'un smartphone, d'une tablette ou de casques. Elle enrichit le monde réel sans le remplacer. À l'inverse, la Réalité Virtuelle (RV) plonge l'utilisateur dans un univers entièrement numérique, coupé du monde réel, généralement à l'aide d'un casque immersif. En résumé, la RA ajoute au réel, tandis que la RV crée un monde entièrement virtuel. Il est possible de se représenter l'intermédiaire entre la Réalité Virtuelle et la Réalité Augmentée comme un continuum avec une échelle de virtualité et de réalité [6]. La Réalité Mixte englobe l'ensemble de ce continuum (Figure 4).

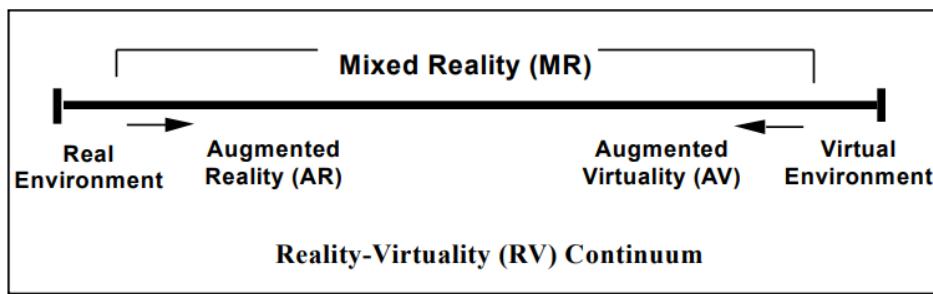


Figure 4: Continuum d'après Paul Milgram et al. [6]

#### 1.3.2 Qu'apporte la Réalité Mixte dans la résolution du problème?

Les casques de Réalité Mixte offrent un nouvel espace d'interaction qui transforme la collaboration. En effet, ils apportent une dimension 3D. Celà est intéressant à exploiter pour représenter les collaborateurs et leur environnement de travail distant.

Comme le souligne le philosophe Maurice Merleau-Ponty, « le corps est notre ‘je’ premier et notre point d’appui dans le monde », ce qui signifie que la communication humaine ne se limite pas aux

mots, mais s'appuie aussi largement sur le langage corporel et les gestes. La Réalité Mixte permet ainsi de restaurer cette dimension physique de la communication, en recréant un espace où les mouvements et postures des utilisateurs peuvent être perçus et partagés, rendant possible une interaction plus incarnée et naturelle.

Dans cet espace, il y a alors la possibilité de représenter en 3 dimensions les collaborateurs dans leur espace de travail, par des avatars par exemple, ainsi que la disposition de leurs écrans au sein de leur espace de travail. Une telle reconstruction de l'espace de travail distant d'un collaborateur chez un utilisateur permettrait de pallier le fait qu'ils ne puissent pas percevoir l'espace de travail de leur interlocuteur.

## 1.4 Outils utilisés

Pour programmer et développer le projet en Réalité Mixte. Voici la liste des principaux outils employés :

- **Unity** : Durant ce projet, l'essentiel de la conception est réalisé avec le logiciel Unity. C'est une plateforme de développement multiplateforme permettant de créer des jeux vidéo, des applications interactives en 2D et 3D et, ici en particulier, des expériences en Réalité Virtuelle et Augmentée. De plus, Unity dispose d'une grande communauté et d'une vaste documentation qui facilite la prise en main. Il existe d'autres moteurs tel que l'Unreal Engine mais ce dernier est plus destiné aux rendus graphiques élevés. Il est par exemple utilisé pour Fortnite dans le jeu vidéo ou dans le cadre de la série The Mandalorian, pour le cinéma.

Version d'Unity utilisée: 2022.3.62f1

- **C #** : Le langage de programmation C # est utilisé dans les scripts Unity. Ces derniers permettent de contrôler les entités de Unity (appelé GameObject) ainsi que leur comportement et intéraction dans l'application. Les scripts permettent de moduler à notre guise les objets de Unity. Il existe des bibliothèques et scripts fournis afin de simplifier le développement. (On peut citer les scripts permettant d'ajouter une dimension physique aux objets, par exemple)
- **MRTK** : MRTK 3 (Mixed Reality Toolkit 3) est un framework open-source de Microsoft destiné à faciliter le développement d'applications en Réalité Mixte avec Unity. Il fournit des composants prêts à l'emploi pour les interactions utilisateur, les interfaces 3D et la compatibilité multiplateforme XR.
- **Git** est aussi de la partie en ce qui concerne la sauvegarde régulière du projet et le partage des fichiers avec le tuteur et un autre stagiaire.
- **Meta Quest** : Finalement, le casque utilisé est le Meta Quest 3 de la Figure 5, permettant de faire de la Réalité Mixte avec des caméras frontales. Il existe aussi le Hololens mais non utilisé ici, le Meta Quest dispose d'une vision plus panoramique et complète.



Figure 5: Casque Meta Quest 3

## 1.5 Structure d'accueil

Le stage s'est déroulé au sein du laboratoire IRIT (Institut de Recherche en Informatique de Toulouse) qui comporte 7 départements de recherche qui regroupent 24 équipes. Le bâtiment se trouve sur le campus de Toulouse. Plus précisément, j'étais dans l'équipe ELIPSE (Etude de L'Interaction Personne Système), qui fait de la recherche dans le domaine de l'Interaction Humain-Machine (IHM). Les chercheurs de cette équipe conçoivent, développent et évaluent des modèles, outils, et techniques d'interaction pour mieux comprendre leur impact sur la perception et la cognition humaines. Cela inclut l'utilisation de la Réalité Mixte par exemple.

Ils travaillent aussi sur le développement de dispositifs ou systèmes d'assistance pour personnes en situation de handicap (ex: déficience visuelle) et adaptation des interfaces à ces usages.

Une autre thématique de recherche de l'équipe est la collaboration multi-écrans dans des environnements de Réalité Mixte, pour permettre à des utilisateurs distants ou avec différents dispositifs de travailler ensemble.

L'équipe dispose d'un environnement de travail dynamique et multidisciplinaire, avec un accès à des équipements de pointe pour la recherche. Cela inclut notamment des casques de Réalité Mixte tels que le Meta Quest 3, le Hololens 2 et des salles d'expérimentation dotées de trakers type OptiTrack. Ces dernières permettent notamment de traker des objets physiques, à l'aide de marqueurs.

## 2 Synthèse de l'existant en terme de collaboration multi-écrans à distance

Avant de passer à l'élaboration d'un prototype pour résoudre les problématiques, il est intéressant de passer en revue ce qui a déjà été fait, c'est-à-dire les travaux scientifiques dont les applications se rapprochent du sujet de ce stage. Cette partie vise à recenser les recherches et projets menés dans ce domaine et à analyser les initiatives mises en place pour répondre à la problématique qui regroupe, rappelons-les, ces trois questions :

- P1. Comment partager, entre collaborateurs, le contenu et la disposition des écrans de leur environnement de travail ?
- P2. Comment, en tant qu'utilisateur, puis-je communiquer mes gestes sur mes écrans à un collaborateur distant ?
- P3. Comment effectuer des gestes dans l'environnement d'un collaborateur distant ?

Les insuffisances de ces travaux par rapport à la problématique étudiée ici seront ainsi mises en évidence. Le but de l'outil à développer durant ce stage est précisément de combler ces lacunes.

### 2.1 Les outils de visioconférence et leurs limites

Des outils de visio-conférence comme Zoom ou Teams permettent, en partie, de répondre aux problématiques. En effet, ils proposent, tout d'abord, de pouvoir partager les contenus et le contenu seulement des écrans des collaborateurs par vidéo capture de ces derniers. Un utilisateur peut alors indiquer avec sa souris un élément présent sur un de ses écrans en fournissant des explications. Son collaborateur pourra alors suivre les explications en visualisant la souris. De plus, grâce à la webcam, il est possible de percevoir le collaborateur dans son environnement de travail avec ses expressions faciales et ses gestes s'ils entrent dans le champ de la caméra. On peut donc le voir parler, bouger les mains et indiquer des éléments avec sa souris sur ses écrans partagés.

Ainsi, les outils de visioconférence répondent partiellement à P1, puisque seul le contenu des écrans est partagé et non leur agencement dans l'espace de travail.

De plus, il n'est pas possible de percevoir l'agencement des écrans du collaborateur distant. Il n'est donc pas possible pour un utilisateur de voir son collaborateur faire des gestes sur ses écrans. P2 n'est

donc pas résolue.

De plus, dans cette configuration de visio-conférence, il est impossible d'interagir avec l'espace de travail du collaborateur distant. Il est possible à la limite d'utiliser sa souris pour indiquer un élément sur un écran partagé par un collaborateur mais celà ne résout pas le problème. En effet, on ne peut pas faire un geste complexe dans son espace distant, par exemple. P3 n'est donc pas résolue.

Une autre limite des outils de visio-conférence, est que l'espace de travail du collaborateur distant visible grâce à la webcam est limité. En effet, il est impossible de voir ce qui est hors champ. Hors, dans les objectifs nous cherchons aussi à communiquer les gestes, parfois hors champ de la caméra, pour rétablir la communication indirecte entre les collaborateurs.

## 2.2 Desk2Desk

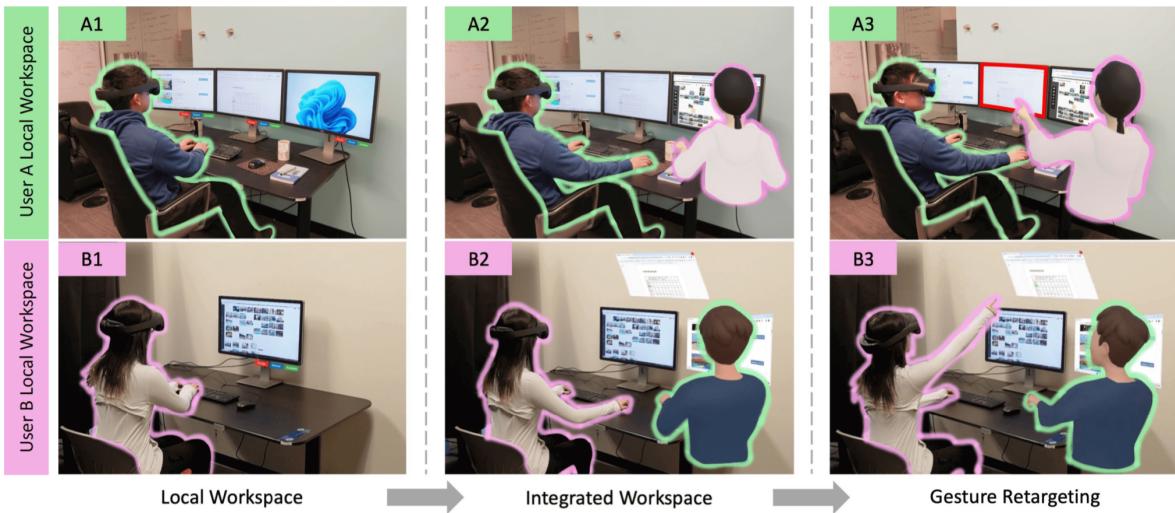


Figure 6: Desk2Desk

Dans l'article intitulé Desk2Desk [8], afin de pallier l'éloignement physique entre deux collaborateurs, les auteurs ont choisi de positionner virtuellement le collaborateur à côté de l'utilisateur équipé d'un casque de Réalité Mixte comme visible sur la Figure 6. Cette configuration permet à l'utilisateur de consulter ses propres écrans tout en percevant, à ses côtés, l'avatar du collaborateur, accompagné de ses propres écrans. Les deux collaborateurs peuvent ainsi interagir avec leurs environnements respectifs comme s'ils étaient physiquement assis l'un à côté de l'autre. Afin de garder une cohérence entre l'espace de travail local d'un collaborateur et celui d'un collaborateur distant, les espaces de travail sont virtuellement alignés. Les écrans virtuels sont alignés et orientés par rapport aux écrans réels.

Desk2Desk semble résoudre entièrement les problématiques (P1, P2 et P3). En effet, l'agencement des espaces de travail et les écrans sont partagés entre les utilisateurs, chacun peut indiquer à son collaborateur des éléments de ses propres écrans ou pointer sur ceux de l'autre. En effet, l'avatar peut aussi bien pointer ou faire des gestes sur des éléments de ses propres écrans virtuels partagés et faire des gestes aussi sur les écrans physiques du collaborateur. L'utilisation des références diéctiques est alors de nouveau possible.

De plus, le fait de représenter le collaborateur sous forme d'avatar permet de restaurer instantanément une communication gestuelle et non verbale.

Cependant, on peut se rendre compte que malgré le fait qu'il est possible de partager plusieurs écrans, l'environnement de l'utilisateur portant le casque peut vite devenir encombré par le contenu virtuel. Il y a alors un **problème d'occlusion** de l'espace environnant. Cela peut, par exemple, devenir gênant si l'utilisateur est entouré de personnes réelles dans son environnement de travail et qu'elles se retrouvent masquées par un contenu virtuel dense. Concernant un confort de travail, avoir

un contenu dense peut être visuellement fatiguant.

### 2.3 MiniMates

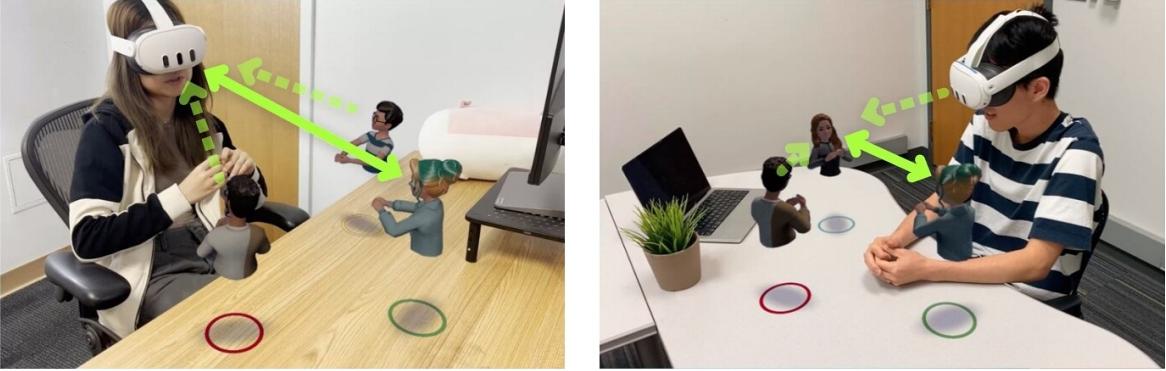


Figure 7: MiniMates

Une autre approche de la collaboration à distance a été explorée dans MiniMates [3]. L'idée consiste à intégrer des avatars miniatures dans l'environnement de Réalité Mixte, de manière à minimiser l'encombrement visuel dans le champ de vision de l'utilisateur comme visible sur la Figure 7. Ces avatars miniatures représentent les collaborateurs distants et sont disposés autour de l'utilisateur portant le casque de Réalité Mixte.

L'article MiniMates fait explicitement référence à la possibilité de partager, entre collaborateurs, un écran où chacun dispose de son pointeur qu'il peut manipuler avec sa souris, tel un tableau collaboratif. Cependant, la possibilité de partager plusieurs écrans par Minimates n'est pas clairement écrite mais citée comme scénario possible d'utilisation. Celà impliquerait, par exemple, un partage à tour de rôle des écrans de chaque collaborateur. Sous l'hypothèse qu'un MiniMates peut partager plusieurs écrans au dessus de sa tête par exemple, un problème va survenir lorsqu'un avatar va vouloir faire référence à des éléments de ses écrans. Il n'y a, en effet, pas de cohérence spatiale entre la position et l'orientation du MiniMates et la position des écrans partagés. Un Minimates, étant miniature, ne peut pas pointer directement avec sa main un élément sur l'écran partagé entre les utilisateurs. Bien que Minimates propose une représentation non occultante des collaborateurs sous forme d'avatars, l'article ne résout pas la problématique liée au partage de l'agencement des environnements multi-écrans (P1). P2 et P3 ne sont alors pas résolus non plus.

## 2.4 Volumetric Hybrid Workspaces

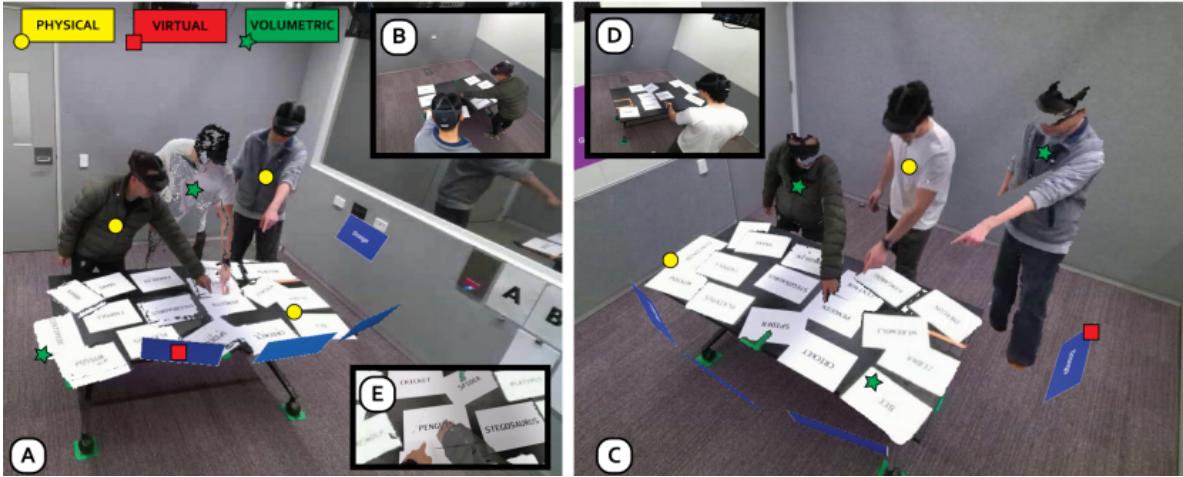


Figure 8: Fusion de deux espaces de travail distants

Andrew Irlitti et al. [2] présentent une plateforme de téléprésence volumétrique en Réalité Mixte permettant de fusionner en temps réel deux environnements physiques distants. Grâce à la capture volumétrique des espaces (personnes, objets et contexte spatial), chaque utilisateur équipé d'un casque de réalité mixte peut visualiser l'environnement distant reconstruit et interagir dans un espace de travail hybride commun, ici une table. Le système repose sur l'alignement d'un point d'ancrage partagé (comme une table ou une zone au sol), créant un espace de collaboration symétrique entre les deux sites. Trois types d'objets coexistent dans ce cadre :

- des objets **physiques** (présents localement comme les humains physiquement présents ou encore les pancartes blanches posées sur la table),
- des objets **volumétriques** (rendus à partir du site distant comme par exemple l'avatar du collaborateur ainsi que les pancartes blanches de l'espace de travail distant)
- et des objets **virtuels** (purement numériques comme les pancartes violettes qui n'existent dans aucun espace réel)

Une étude expérimentale avec des équipes partiellement distribuées a permis d'évaluer comment ces différents types d'objets influencent la communication, l'engagement, et les stratégies de collaboration.

On retrouve l'aspect de l'illusion de présence évoqué dans Desk2Desk, du à la présence de l'avatar du collaborateur chez l'utilisateur. Les résultats de ces recherches montrent que la téléprésence volumétrique permet une interaction fluide et une reconstitution efficace des dynamiques de collaboration habituellement réservées à la co-présence physique.

Cet article répond donc en grande partie à P1, P2 et P3, car l'agencement des espaces de travail entre collaborateurs est partagé et il est également possible de communiquer les gestes entre eux.

Cependant, on peut s'apercevoir que ces recherches ne traitent pas du partage d'écrans bien que les pancartes pourraient en jouer le rôle, d'où l'intérêt de nuancer le propos concernant le fait que l'article résout la problématique. De plus, il y a nécessité que les espaces des collaborateurs distants se ressemblent un minimum pour pouvoir définir un point d'ancrage pour la synchronisation des éléments virtuels. Hors, dans la collaboration à distance multi-écrans, il n'y a aucune raison que les espaces de travail soient agencés de la même manière.

De la même manière que les articles précédents, du à l'affichage des éléments virtuels (tel que l'avatar complet et les pancartes), il peut y avoir un problème d'occlusion du champ visuel chez les utilisateurs.

## 2.5 HoloDevice



Figure 9: Un utilisateur et l'avatar virtuel du collaborateur à gauche. La main du collaborateur qui apparaît dans la vision de l'utilisateur à droite

Neil Chulpongsatorn et al. [1] propose le concept d'interaction cross-device holographique” dans le contexte de collaboration à distance. L'idée est de permettre à des appareils physiques situés à des endroits différents d'interagir comme s'ils étaient dans le même espace, via une représentation holographique (dans un environnement de Réalité Mixte). Par exemple, un usager d'un téléphone portable peut voir la reconstruction virtuelle de l'écran de son collaborateur à coté de son écran physique. Cela fait d'ailleurs echo au placement des écrans dans Desk2Desk, qui sont agencés de telle manière à s'aligner avec les écrans physiques. Grâce à ce dispositif, il est possible de partager le contenu et l'agencement des écrans entre collaborateurs. Celà résout P1.

Dans cet article, il est possible de partager les gestes faits sur ses propres écrans à son collaborateur. En effet, l'utilisateur apparaît sous forme d'avatar réel dans l'espace du collaborateur et peut montrer des éléments sur ses écrans comme visible en Figure 9 à gauche. Dans cette figure, l'avatar pointe du doigt un écran qu'il tient dans sa main. De plus, un collaborateur peut pointer des éléments sur les écrans de son collaborateur distant, en particulier, grâce à une main virtuelle recréée dans l'espace virtuel du collaborateur distant. Cela est visible en Figure 9 à droite. La main blanche correspond à la main d'un collaborateur distant pointant à côté de l'écran blanc chez un utilisateur. En résumé, cet article résout P2 et P3.

Comme dans l'article précédent, il y a la possibilité de reproduire le collaborateur sous forme d'avatar réel. Cela est pertinent pour visualiser ses mains et sa tête et aussi pour transmettre la communication faciale, par exemple.

Cependant, de la même manière que dans les articles précédents, un nombre élevé d'écrans partagés, ainsi que l'espace visuel occupé par l'avatar peut poser un problème d'occlusion chez le collaborateur.

## 2.6 Using Virtual Replicas

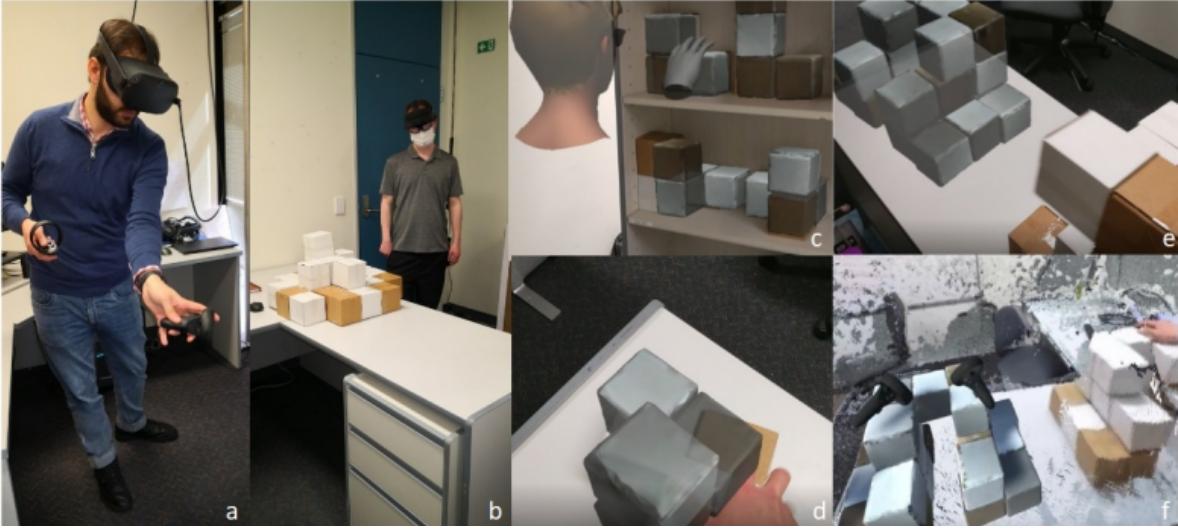


Figure 10: L'expert produit des explications en Réalité Mixte

Bien que cette recherche ne cible pas explicitement la collaboration multi-écrans, les idées abordées restent pertinentes par rapport à notre problématique. Huayuan Tian et al. [9], les auteurs explorent comment l'utilisation de répliques virtuelles peut améliorer la collaboration à distance en Réalité Mixte, en s'appuyant sur une reconstruction 3D de l'espace de travail. Ils partent du constat que, dans certains contextes, des personnes situées à distance doivent collaborer sur des tâches complexes. Par exemple, un utilisateur local peut suivre les instructions d'un expert distant pour réaliser une tâche physique. Cependant, la compréhension des intentions de l'expert peut être limitée en l'absence de mécanismes efficaces de référencement spatial et de démonstration gestuelle.

Pour répondre à cette problématique, ils proposent d'utiliser des répliques virtuelles comme support de communication spatiale. Leur approche consiste à segmenter les objets manipulables au premier plan dans l'environnement local, puis à en créer des répliques virtuelles. L'utilisateur distant peut alors manipuler ces répliques pour illustrer la tâche, montrer les gestes à effectuer, et guider efficacement l'utilisateur local. Grâce à cette méthode, l'utilisateur local est en mesure de mieux comprendre, plus rapidement et plus précisément, les instructions et intentions de l'expert distant. L'utilisation de mains virtuelles pour représenter les mains manipulant l'objet concerné permet de rajouter une dimension gestuelle dans la communication.

Si l'on fait le parallèle avec la collaboration à distance multi-écrans, les idées abordées dans cet article pourraient être extrapolées pour faire une réplique virtuelle de l'espace de travail et des écrans du collaborateur. Ainsi, les cubes utilisés dans l'article pourraient faire office d'écrans manipulables. De plus, dans cette configuration, un collaborateur pourrait être complètement immergé dans l'espace de travail de son interlocuteur. Il verrait donc ses gestes faire les mouvements.

Ce qui semble insuffisant concernant notre problématique, est que le niveau de détail de l'espace environnant reconstruit et des objets manipulés est assez faible. Hors, bien qu'on pourrait penser transposer les cubes en écran, le niveau de détail des écrans doit être élevé pour assurer une bonne lecture des informations sur ces derniers.

De plus, le problème d'occlusion est aussi présent. Le collaborateur regardant l'expert le guidant dans sa tâche, il ne voit plus son environnement réel à cause de la reconstruction virtuelle de l'espace du collaborateur.

## 2.7 Solutions apportées et insuffisances de l'état de l'art

Dans l'état de l'art, nous avons pu voir que les articles cités apportent des solutions adressant les problématiques lorsque des collaborateurs avec des espaces de travail multi-écran sont distants :

- P1. Comment partager, entre collaborateurs, le contenu et la disposition des écrans de leur environnement de travail ?
- P2. Comment, en tant qu'utilisateur, puis-je communiquer mes gestes sur mes écrans à un collaborateur distant ?
- P3. Comment effectuer des gestes dans l'environnement d'un collaborateur distant?

Tous les articles cherchent à représenter partiellement ou complètement l'espace de travail du collaborateur distant chez un utilisateur. Cela inclut des avatars pour représenter le collaborateur distant dans tous les articles traitant de Réalité Mixte ou une par une capture video pour les outils de visio conférence. Desk2Desk [8], Volumetric Hybrid Workspaces [2] et HoloDevice [1] proposent en plus de cet avatar de représenter virtuellement l'agencement des éléments (dont écrans) de l'espace du collaborateur distant. Dans ce cas, il y a cohérence spatiale entre les mouvements du collaborateur représenté et son espace de travail virtuel reconstitué. Ainsi, il est possible de communiquer des gestes complexes effectués dans l'environnement de l'utilisateur vers le collaborateur et de montrer des éléments chez le collaborateur distant, par exemple.

Cependant, l'apparence du collaborateur sous forme de contenu vidéo dans les outils de visio-conférence limite la perception de l'environnement de travail du collaborateur. De plus, toutes les solutions apportées en Réalité Mixte dans l'état de l'art, semblent présenter un problème d'**occlusion visuelle** du à l'affichage du contenu virtuel que ce soit les avatars, les écrans ou l'environnement du collaborateur distant de manière générale. Un nombre élevé d'écrans partagés peut accentuer le problème. Cela engendre inévitablement un encombrement visuel chez le collaborateur qui se voit afficher les contenus virtuels dans son propre environnement. C'est gênant surtout si ce collaborateur à besoin de consulter ses propres écrans en même temps.

### 3 Solution envisagée : nouvelles interactions basées sur la métaphore du "diorama à judas"

L'idée étudiée est alors la suivante: permettre la collaboration multi-écran à distance entre plusieurs personnes à l'aide de la Réalité Mixte en utilisant une métaphore de "diorama à judas".

#### 3.1 Explication du concept de "diorama à judas"

Un diorama est une reconstitution visuelle, à une échelle plus ou moins réaliste, d'un environnement, d'un milieu ou d'un univers. Un exemple typique est celui des scènes naturalistes présentées dans les musées, où des animaux sont disposés dans une boîte vitrée reproduisant leur habitat naturel. La Figure 11 illustre un exemple de diorama : une cuisine miniature reconstituée dans une boîte.

Dans le cadre de ce stage, l'idée est de transposer ce principe en Réalité Mixte, en représentant l'espace de travail d'un collaborateur distant. Contrairement à une cuisine contenant verres et tasses, cet environnement reconstitué contiendra plutôt des éléments propres au travail numérique, comme des écrans ainsi qu'un avatar du collaborateur. Il s'agit donc d'une miniaturisation virtuelle de son espace de travail réel.



Figure 11: Diorama Cuisine

Un judas est une petite ouverture permettant d'observer à travers une surface opaque. Un exemple courant est celui que l'on trouve sur les portes d'appartements, permettant de voir qui se trouve derrière sans avoir à ouvrir la porte. (Figure 27)

Dans le contexte de ce stage, l'idée consiste à reconstituer en Réalité Mixte, et à une échelle miniature, l'espace de travail d'un collaborateur distant, observable à travers un judas virtuel. La Figure 13 illustre ce concept à travers un diorama à judas : bien qu'il s'agisse ici d'une scène générique, on peut imaginer qu'elle représente l'environnement de travail du collaborateur, avec ses écrans et outils numériques. Il est important de noter que la boîte présentée sur la Figure 13 est totalement opaque, ne laissant voir que par l'ouverture prévue à cet effet. Dans le cadre du "diorama à judas" en Réalité Mixte, la boîte telle qu'elle apparaît marron sur la Figure 13 n'est pas visible par l'utilisateur mais uniquement le contenu visible à travers le trou l'est. Cela permet de minimiser l'occlusion.

L'objectif est de disposer ce diorama à judas, grâce à la Réalité Mixte, dans l'environnement physique de l'autre collaborateur distant. Ce dernier, équipé de son casque, pourra alors observer à l'intérieur du judas, représentant l'espace de travail de son interlocuteur, tandis que le reste de sa



Figure 12: Judas

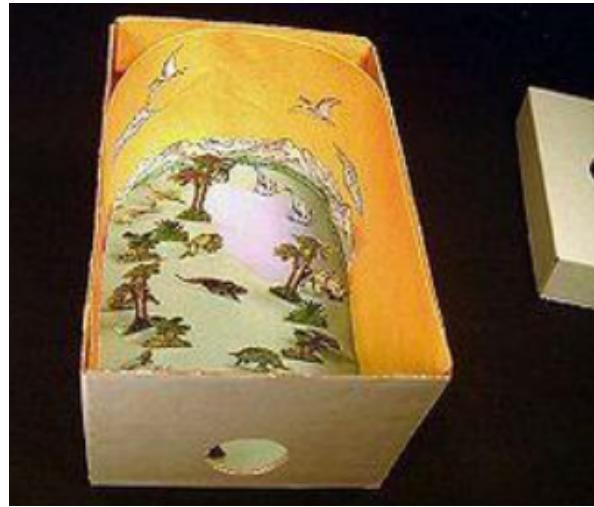


Figure 13: Diorama à Judas

vision reste ancré dans son propre environnement réel.

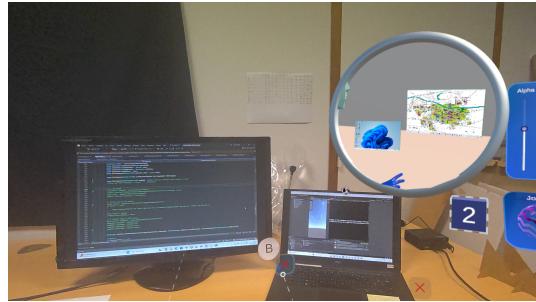


Figure 14: Utilisateur 1



Figure 15: Utilisateur 2

Sur les Figures 14 et 15, qui illustrent des "diorama à judas" en Réalité Mixte, il est possible d'avoir une vue d'ensemble de deux collaborateurs avec leurs écrans possédant chacun un diorama représentant l'espace de travail du collaborateur distant. Dans ce diorama se trouve un avatar du collaborateur distant (tête et main) ainsi que le contenu et l'agencement de ses écrans. L'espace de travail distant est visible uniquement à travers le judas.

### 3.2 Pourquoi cette idée répond aux insuffisances de l'état de l'art

Explorer une nouvelle technique d'interaction en Réalité Mixte, fondée sur la métaphore du « diorama à judas », permettrait tout d'abord de répondre, tout d'abord, à la limitation des systèmes de l'état de l'art concernant l'*occlusion visuelle* lors de la perception de l'espace de travail du collaborateur en Réalité Mixte. En effet, la métaphore du "diorama à judas" propose une visualisation partielle mais ciblée de l'espace de travail distant, tout en préservant la visibilité de l'environnement local. Il est ainsi possible d'avoir une vue en détails de la scène et des écrans virtualisés du collaborateur sans avoir à faire un compromis sur la visibilité de l'espace de travail courant. Comme sur les portes d'appartement, le judas rapproché de l'oeil permet de voir plus en détail l'espace de travail distant.

L'agencement de l'espace de travail du collaborateur est reconstitué dans le diorama de manière à respecter les positions relatives entre les écrans (Objectif 1). De plus, il y a cohérence spatiale entre la représentation du collaborateur (avatar) et ses écrans. Les références diéctiques peuvent alors être utilisées par les collaborateurs tel qu'ils les utiliseront s'ils étaient présents dans la même pièce. Cela offre la possibilité d'indiquer des écrans de manière non ambiguë dans un environnement multi-écran, comme si l'on observait le collaborateur dans son espace de travail à la troisième personne. Il est alors possible de voir, à travers le diorama le collaborateur faire des gestes sur ses écrans dans son espace de travail distant (Objectif 2).

De même, il est possible d'interagir avec le diorama en passant sa main à l'intérieur pour effectuer des gestes dans l'environnement du collaborateur distant (Objectif 3). Une main virtuelle apparaît alors dans l'environnement de ce dernier.

De plus, la position du diorama dans l'espace de travail du collaborateur ne dépend donc pas strictement de son environnement physique puisqu'il n'y a pas synchronisation avec l'espace de travail distant. L'utilisateur peut alors ajuster à sa guise la place occupée par le diorama dans son espace de travail, selon ses besoins.

### 3.3 Cas d'usage envisagé

Voici un petit scénario explicatif d'un cas d'usage :

Personnages :

- Arthémis
- Bacchus

Arthémis se trouve au pôle Sud, tandis que Bacchus est au pôle Nord. Ils sont loin ... très loin l'un de l'autre et pour des raisons environnementales ne prennent pas l'avion pour se retrouver physiquement. Ils décident, à un moment opportun, de mettre un casque de Réalité Mixte puis de se partager leur environnement de travail à l'aide du "diorama à judas". De plus, ils ont des environnements de travail comprenant beaucoup d'écrans car ce sont de grands scientifiques, glaciologue, et ils ont besoin de visualiser simultanément un grand nombre de données. Grâce au "diorama à judas", chacun dispose dans son champ de vision d'une reconstitution de l'environnement de travail de l'autre. C'est-à-dire, des écrans, ainsi que de l'avatar du collaborateur.

Pour observer l'espace de travail de Bacchus, Arthémis doit s'approcher du judas, lui permettant de regarder à l'intérieur du diorama représentant l'environnement de Bacchus — et inversement pour Bacchus.

À un moment donné, Arthémis détecte un signal ultrasonique sous la glace. Pour partager cette découverte, elle affiche, sur l'un de ses écrans, une carte de l'Antarctique indiquant la provenance du signal, et sur un autre écran, une copie de l'enregistrement audio correspondant. Depuis son propre casque, Bacchus visualise le diorama d'Arthémis. À l'intérieur de celui-ci, Arthémis lui montre les deux écrans en les pointant du doigt, tout en fournissant des explications avec des gestes. Elle lui montre, en effet, avec ses deux mains la profondeur estimée du signal. Ainsi, Bacchus peut suivre les

explications d'Arthémis tout en sachant quel écran et quel geste suivre chez Arthémis. Bacchus peut aussi consulter ses propres écrans en même temps sans être encombré par le diorama d'Arthémis.

Arthémis dispose sur un de ses écrans d'une image de la coupe verticale de la glace du lieu étudié et Bacchus veut lui montrer sur cet écran la distance qu'il pense y avoir entre leur position et la profondeur du signal. Il passe alors ses deux mains dans le diorama et pointe sur l'écran d'Arthémis. Cette dernière voit apparaître, à travers le casque de Réalité Mixte, deux mains virtuelles qui montrent chacune un endroit de son écran dans son environnement de travail.

## 4 Représentation d'un environnement de travail distant dans le diorama

L'objectif de cette partie est de comprendre comment le diorama et en particulier l'environnement de travail virtuel est structuré, et en quoi les choix de conception retenus permettent de répondre aux problématiques.

### 4.1 La pièce: contenant les écrans et l'avatar

Dans le contexte étudié, un collaborateur est défini comme une personne travaillant à un bureau équipé de plusieurs écrans. Pour représenter son environnement de travail, il a été choisi de modéliser une pièce délimitée par des murs plans. Lorsqu'un utilisateur virtualise l'un de ses écrans, pour le partager à son collaborateur, celui-ci apparaît dans le diorama correspondant à son espace de travail. Les écrans sont positionnés dans la pièce de manière cohérente avec leur disposition réelle. Par exemple, si un utilisateur virtualise deux écrans superposés, ils apparaîtront également l'un au-dessus de l'autre dans le diorama, afin de rester fidèles à leur agencement dans l'environnement de travail physique.

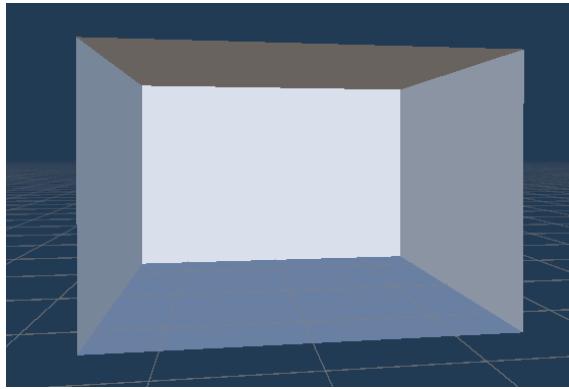


Figure 16: Modèle de pièce dans Unity



Figure 17: Ecrans dans une pièce dans Unity

### 4.2 L'avatar: une représentation 3D du collaborateur à l'intérieur du diorama

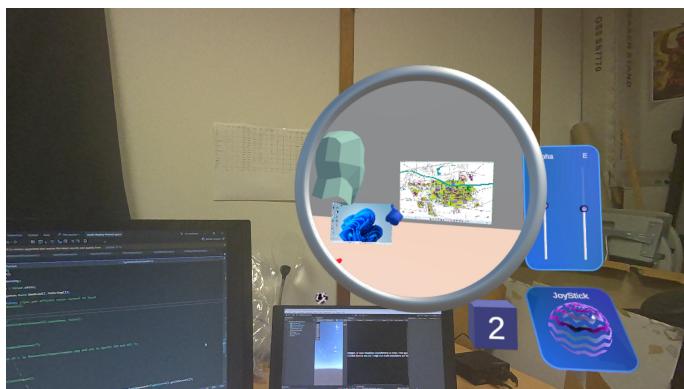


Figure 18: Avatar qui pointe un coin de son écran



Figure 19: Avatar seul

La question s'est posée de savoir comment représenter l'avatar du collaborateur dans le diorama. Au début du projet, pendant que je travaillais sur la construction du Diorama, un autre stagiaire était en charge de la mise en place de l'interface réseau ainsi que de l'intégration de l'avatar à l'intérieur du diorama. Bien que nous travaillions tous les deux sur le même projet, nos tâches respectives étaient

quelque peu distinctes. Ce collègue a ainsi choisi de représenter le collaborateur à l'aide d'un avatar minimaliste, constitué d'une tête et de deux mains. En effet, les seules informations que l'avatar a besoin de transmettre sont la direction du regard du collaborateur et ses gestes. Afficher les bras d'un collaborateur serait inutile en plus de créer de l'occlusion ce qu'on souhaite éviter. L'idée sous-jacente est que ces éléments — la tête et les mains — soient synchronisés en temps réel, via le réseau, avec les mouvements du collaborateur dans son environnement de travail réel grâce aux capteurs du casque. À travers le diorama, l'utilisateur peut percevoir le langage corporel de son collaborateur et visualiser comment ce dernier interagit avec son environnement de travail. Par exemple, en Figure 18 un collaborateur pointe le coin supérieur droit d'un de ses deux écrans dans son espace réel et le geste est reproduit par l'avatar dans le diorama qui lui est associé.

#### 4.2.1 Intégration réseau: Netcode et RPC

Cette courte section vise à présenter les points clés permettant de comprendre l'architecture réseau de l'application. Bien que je ne sois pas à l'origine de sa conception, j'ai été amené à manipuler et utiliser cette architecture par la suite dans le cadre du projet. L'objectif principal étant de permettre la collaboration entre au moins deux utilisateurs équipés de casques de réalité mixte, situés à distance, la mise en réseau des utilisateurs est donc essentielle.

L'architecture réseau du projet repose sur les bibliothèques Netcode for GameObjects fournies par Unity, qui implémentent un modèle Client/Serveur. Dans ce modèle, l'utilisateur portant le casque est considéré comme un Client et se connecte sur un Serveur. Unity propose plusieurs composants facilitant la synchronisation des objets à travers le réseau, tels que les NetworkObject ou NetworkTransform, permettant, par exemple, de synchroniser la position et la rotation d'un objet en temps réel. Cependant, ces composants peuvent montrer leurs limites dans le cas de communications plus complexes ou spécifiques. Dans ces situations, l'utilisation des Remote Procedure Calls (RPC) devient alors nécessaire et pertinente.

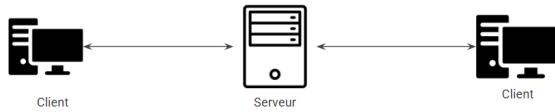


Figure 20: Architecture Clients - Server

Les RPC (Remote Procedure Calls) sont des mécanismes permettant d'exécuter une fonction à distance, c'est-à-dire sur un autre appareil connecté au réseau. Dans Unity, via Netcode for GameObjects, les RPC sont utilisés pour envoyer des instructions du client vers le serveur (ClientRPC) ou du serveur vers un ou plusieurs clients (ServerRPC). Cela permet de synchroniser des actions spécifiques, comme déclencher une animation, envoyer un message ou activer un objet, qui ne peuvent pas être gérées uniquement par la synchronisation automatique des composants comme NetworkTransform. Les fonctions RPC sont dotées d'une entête comme en Figure 21.

Ils sont particulièrement utiles lorsqu'on souhaite exécuter du code à distance, que ce soit sur un client spécifique, sur le serveur, ou de manière globale. Grâce aux RPC, il est possible de contrôler précisément où un morceau de code doit s'exécuter. Cela s'avère notamment crucial pour le spawn d'objets réseau, c'est-à-dire faire apparaître une même instance d'un game Object pour l'ensemble des utilisateurs connectés sur le réseau. Ce type d'opération ne peut être réalisé que par le serveur. Par exemple, le diorama doit être instancié côté serveur, puis propagé aux autres clients. Un client seul ne peut pas initier la création d'un objet réseau.

**[Rpc(SendTo.ClientsAndHost)]**

Figure 21: Exemple d'entête de fonction RPC

Du côté utilisateur, lorsqu'un nouveau client se connecte, le choix a été fait de faire apparaître automatiquement dans sa scène les dioramas des autres collaborateurs déjà présents sur le réseau. Inversement, son propre diorama doit également être instancié et visible chez les autres utilisateurs connectés. Une gestion efficace des dioramas sur le réseau devient donc essentielle pour assurer la cohérence de l'expérience collaborative. Dans le cadre du stage, le nombre d'utilisateurs a été limité à deux, ce qui a permis de simplifier certaines logiques de synchronisation.

#### 4.2.2 Calibration de l'avatar

L'objectif est ici de positionner un avatar dans le diorama associé à un utilisateur. Lorsque ce dernier bouge sa tête ou ses mains, les mouvements sont captés par le casque de Réalité Mixte et l'application synchronise l'avatar dans le diorama correspondant à l'espace de travail de cet utilisateur. Plusieurs approches ont été envisagées. La première consistait à faire apparaître l'avatar en tant que NetworkObject directement lors de l'instanciation du diorama. Cependant, cette méthode a soulevé des problèmes liés à la hiérarchie des NetworkObjects, notamment en ce qui concerne le parentage. En effet, si l'avatar n'est pas initialement intégré au diorama, il devient nécessaire de le rattacher manuellement à ce dernier afin qu'il fasse partie de sa structure. Cela permet de garantir que l'avatar se déplace avec le diorama comme un ensemble cohérent.

Pour pallier ce problème, il a donc été pensé d'intégrer au Prefab (préfabriqué) "diorama", le Prefab "Avatar". L'avatar n'est donc plus un objet réseau mais désormais un objet local au Diorama. Il suffit alors de déplacer l'avatar localement dans le diorama.

Le référentiel de l'avatar à l'intérieur du diorama est appelé "workspace" (Figure 22 et 23). Il s'agit du repère local servant de point d'origine pour le placement des éléments dans le diorama, en particulier le positionnement de l'avatar (et des écrans). Les gameObjects présents dans le diorama sont enfants du workspace, c'est-à-dire que si le workspace est déplacé alors tous les objets qu'il contient le sont aussi.

Ainsi, lorsque l'utilisateur se déplace dans son environnement de travail, il suffit de mettre à jour la position locale (par rapport au référentiel "workspace") de l'avatar, son orientation ainsi que la position de ses mains. Ainsi, il y a cohérence spatiale entre l'espace de travail réel d'un utilisateur et sa représentation virtuelle dans le diorama. Pour ma part, je me suis chargé de la synchronisation de la tête et fait des révisions sur la position des mains ultérieurement. L'ensemble du processus repose sur l'utilisation de requêtes RPC pour assurer la cohérence entre les instances réseau.

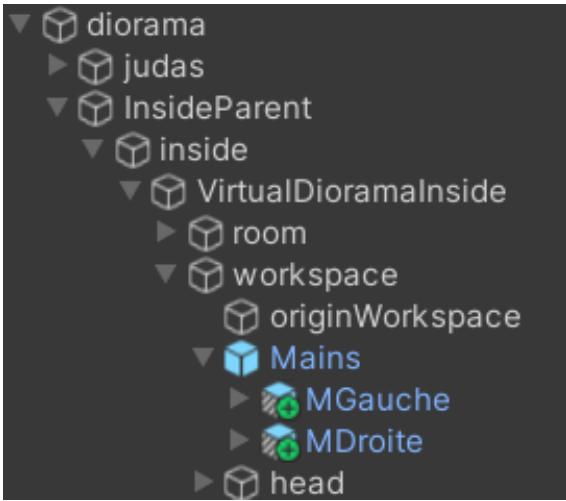


Figure 22: Place du workspace dans la hiérarchie du diorama

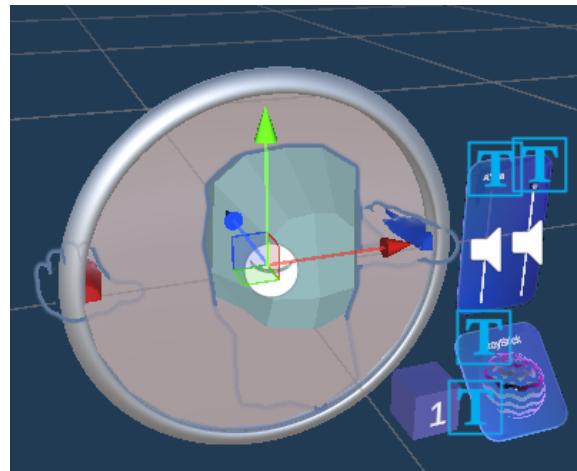


Figure 23: Référentiel du workspace dans diorama

Les gestes de l'avatar ont aussi été synchronisés côté serveur et non uniquement chez les clients afin de pouvoir debugger si nécessaire du point de vue du serveur.

#### 4.2.3 Synchronisation des joints des mains

Afin de maintenir une cohérence entre les mouvements des mains d'un collaborateur et leur représentation virtuelle dans un diorama, la position et la rotation des mains ainsi que de leurs phalanges ont dû être traitées.

Le casque Meta Quest 3 est équipé de capteurs capables de suivre avec précision les mouvements des mains et traçer leurs contours à partir de la vision du casque. Ainsi, si un collaborateur pointe un élément à l'écran avec le doigt dans son espace réel, l'objectif est que son avatar reproduise exactement ce geste — doigt tendu, main orientée dans la bonne direction dans le diorama.

Pour cela, il a été nécessaire de s'intéresser à la structure de la main de l'avatar. Comme une main humaine, celle-ci est modélisée à l'aide de joints (articulations) représentant les os de la main. Il est possible de contrôler la rotation locale de chaque phalange afin de simuler les mouvements des doigts de manière réaliste.

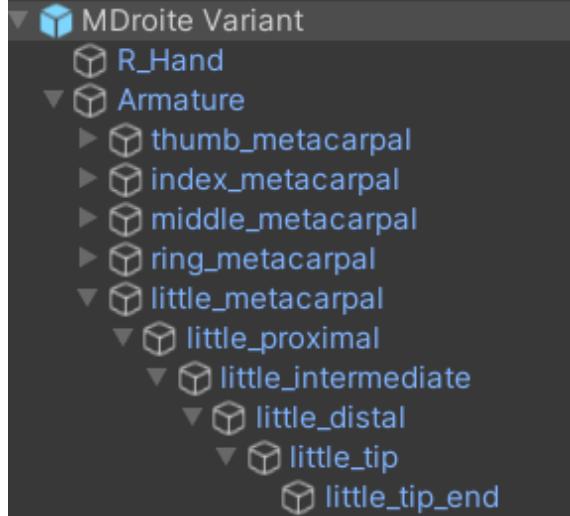


Figure 24: Structure du prefab MDroite

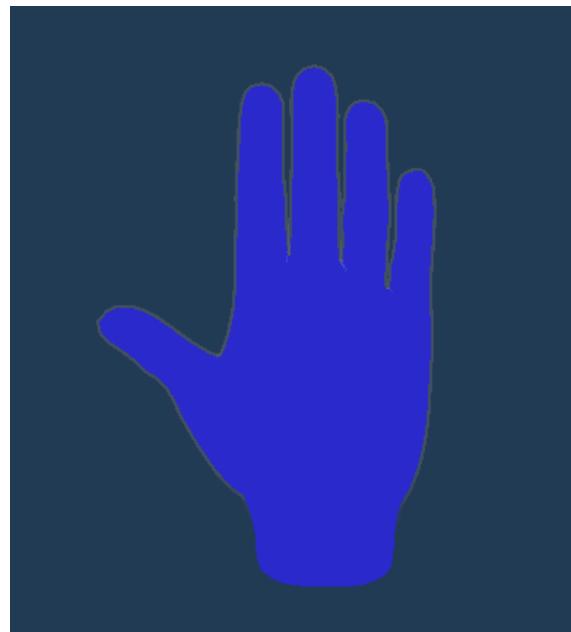


Figure 25: Prefab de la Main Droite (MDroite) de l'avatar

L'approche retenue consiste à récupérer, pour chaque collaborateur, les rotations locales de l'ensemble des joints de ses mains droite et gauche, puis à les transmettre au serveur. L'avatar correspondant dans le diorama peut alors utiliser ces données pour afficher une posture de main fidèle aux gestes du collaborateur en temps réel.

Sur un aspect plus technique, l'utilisateur local dispose d'un modèle de main basé sur MRTK (OpenXR), qui s'adapte dynamiquement à sa main réelle grâce aux capteurs du casque. Ce sont les données issues de ce modèle de main (notamment les rotations des articulations) qui sont transmises au serveur.

L'avatar distant, quant à lui, possède ses propres préfabs (préfabriqué) (Figure 25) de main — MDroite et MGauche — dont les rotations des phalanges sont mises à jour en temps réel à partir des données reçues depuis le serveur. Pour récupérer ou changer la rotation des phalanges, il suffit de changer les propriétés de rotations locales des gameObjects constituant les doigts, visibles en Figure 24. Par exemple, le petit doigt comprend l'os métacarpien, la phalange proximale, la phalange intermédiaire, la phalange distale, ainsi que l'extrémité du doigt (tip).

## 5 Interaction avec le diorama

### 5.1 Le judas: voir à l'intérieur du diorama

Le principe du judas est de permettre la visualisation de l'intérieur du diorama uniquement lorsqu'on y porte le regard. Il vient résoudre le problème d'occlusion des éléments virtuels dans la scène de l'utilisateur. Il est donc nécessaire de masquer les éléments que l'on souhaite dissimuler, afin de ne laisser visible qu'une ouverture restreinte. Une solution envisageable pour parvenir à cet effet dans Unity consiste à utiliser des buffers ou des masques.

#### 5.1.1 Les stencils de MRTK

Le MRTK (Mixed Reality Toolkit) fournit des packages permettant de créer des portails visuels, appelés Stencils. Un objet utilisant un shader ou un matériau avec la propriété Stencil peut ainsi être masqué, à condition qu'un portail (doté du matériau StencilPortal) se trouve entre la caméra et cet objet. Par exemple, sur la Figure 26, le portail carré masque une partie de l'objet orange ayant les propriétés stencil.

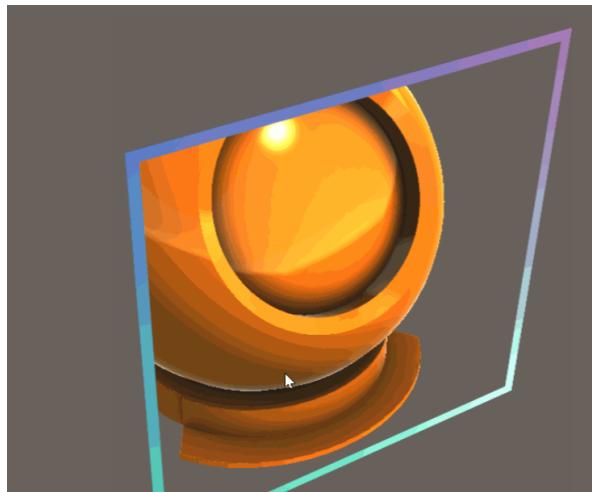


Figure 26: StencilPortal

Par défaut, les portails proposés par MRTK sont de forme carrée. Or, le choix a été fait d'utiliser une forme circulaire pour représenter le judas. La forme ronde du judas, s'inspire du judas d'appartement. Il a donc été nécessaire d'appliquer le matériau StencilPortal à un cylindre aplati, afin de donner l'illusion d'une ouverture ronde. De surcroît, j'ai modélisé sous Blender un bord circulaire à des fins esthétiques. Ainsi, en Figure 27, l'ouverture ne permet de voir l'espace de travail qu'à travers le trou, le reste de l'intérieur du Diorama est masqué.

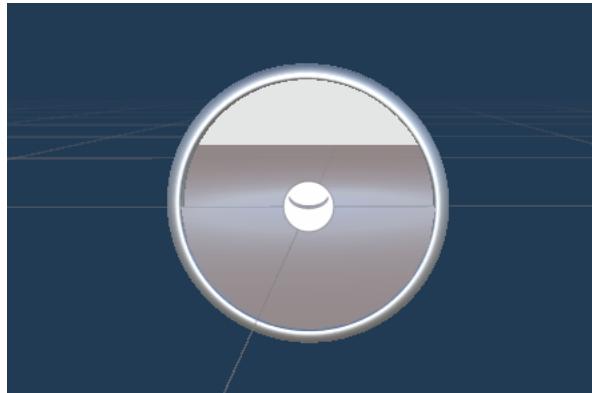


Figure 27: Judas avec diorama vide

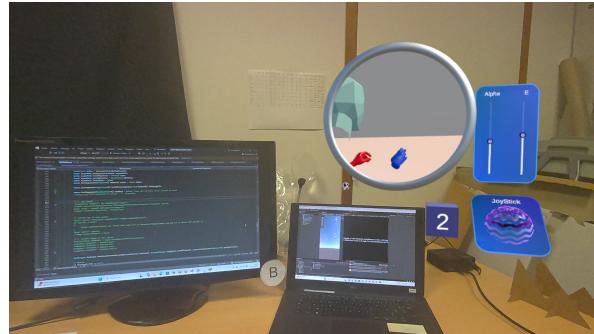


Figure 28: Les contrôles du diorama, sliders, poignée et JoyStick

## 5.2 Contrôles du diorama

Le diorama doit être modulable afin de s'adapter aux différentes problématiques rencontrées lors de la collaboration. Dans cette optique, plusieurs interfaces de contrôle ont été explorées pour le rendre à la fois **non obstruant** et **amovible**.

- **Réglage du diamètre du judas** : permet de définir à quel point le diorama masque l'environnement réel, pour éviter toute gêne visuelle.
- **Positionnement dans l'espace de travail** : l'utilisateur peut déplacer le diorama afin de l'intégrer au mieux à son environnement physique.
- **Modification du point de vue** : il est possible d'ajuster l'angle ou la perspective depuis laquelle la scène du collaborateur est visualisée.

### 5.2.1 Ajuster la taille du judas

Selon le niveau de visibilité souhaité par l'utilisateur sur l'espace de travail du collaborateur, il est judicieux de pouvoir régler le diamètre du judas. Une méthode, pour un utilisateur équipé d'un casque de réalité mixte, consiste à contrôler des sliders (Figure 28) avec la main droite, par exemple. Lorsque le slider est au maximum, le diamètre du judas est également au maximum, et inversement. Ainsi, l'utilisateur peut, selon ses besoins, élargir sa vue en se rapprochant du judas ou en augmentant son diamètre. Lorsqu'il n'a plus besoin d'une grande visibilité dans le diorama, il peut réduire considérablement le diamètre.

### 5.2.2 Déplacer le diorama dans l'espace de travail

L'utilisateur n'a pas nécessairement besoin de consulter le diorama en permanence. La possibilité de déplacer le diorama, de le rapprocher ou de l'éloigner, s'avère donc pertinente. Pour cela, un cube (cube violet de la Figure 28) a été conçu comme poignée permettant de déplacer l'ensemble du diorama ainsi que ses contrôles. Il peut aussi positionner le diorama par rapport à ses écrans comme il le souhaite, pour ne pas les masquer par exemple.

### 5.2.3 Changer de point de vue dans le diorama

Un joystick offre une possibilité de navigation dans l'espace virtualisé du collaborateur distant. Il permet à l'utilisateur de changer de point de vue à l'intérieur du Diorama. Par exemple, il peut être utilisé pour rapprocher les écrans du collaborateur distant afin d'en obtenir un aperçu plus précis, ou au contraire pour les éloigner et bénéficier d'une vue d'ensemble de son espace de travail.

J'ai fait le choix d'une "boule" (Figure 28) de contrôle qui offre la possibilité de bouger l'intérieur du diorama dans toutes les directions possibles.

## 6 Communication avec un collaborateur grâce au diorama

Cette section s'intéresse aux apports du diorama en terme de communication gestuelle et de l'intégration des écrans au sein du diorama.

### 6.1 Gestes sur un écran local

Afin d'assurer une bonne transmission des informations entre collaborateurs, ces derniers peuvent choisir de partager leurs écrans à l'intérieur de leur diorama correspondant. Le collaborateur peut donc suivre les explications de son interlocuteur tout en suivant les écrans pointés par l'avatar dans le diorama.

#### 6.1.1 Virtualisation et placement écrans

Voici un scénario qui illustre le partage des écrans, en reprenant l'histoire d'Arthémis et Bacchus, les glaciologues :

Arthémis dispose de trois écrans dans son espace de travail physique et souhaite les partager avec Bacchus via le diorama. Pour ce faire, Arthémis utilise son casque pour désigner du doigt les extrémités de son premier écran. Un écran virtuel est alors généré dans le diorama correspondant à l'espace de travail d'Arthémis, visible par les autres utilisateurs. Arthémis répète le processus pour les deux autres écrans. Ainsi, en regardant à travers le diorama d'Arthémis, Bacchus peut voir l'avatar d'Arthémis ainsi que les écrans que celle-ci vient de partager. Ces écrans sont placés dans le diorama en respectant la disposition qu'ils ont dans l'espace d'Arthémis.

Ainsi, le processus se décompose en deux étapes.

- La première consiste en la **virtualisation de l'écran** : cela inclut la récupération de sa taille, de son contenu et de sa position dans l'espace de travail réel de l'utilisateur. Le stagiaire avec qui j'ai collaboré en début de projet a développé une interface utilisateur, permettant de générer un écran virtuel dans le diorama du collaborateur pour représenter un écran physique de l'utilisateur. (Figure 29) En positionnant trois points dans l'espace à l'aide de son doigt, l'utilisateur peut définir un rectangle, qui sera interprété comme un écran virtuel. Le but est de partager le contenu de cet écran grâce au diorama.



Figure 29: Virtualisation de l'écran

- La seconde étape concerne **l'affichage de cet écran sous forme d'objet 3D** : les paramètres de l'écran virtualisé (position dans la scène de l'utilisateur, taille, orientation) sont ensuite envoyés sur le réseau afin d'être reconstruits dans l'architecture locale des dioramas présents dans la scène des collaborateurs. De la même manière, un prefab Game Object "cube" est instancié pour générer l'écran. L'écran est disposé de manière à respecter la cohérence spatiale entre l'environnement physique du collaborateur qui le génère et la position de l'écran dans l'espace virtuel reconstruit dans le diorama. En effet, il est placé dans le "workspace" partageant le même repère que l'avatar, comme développé en Section 4.2.2. Cela permet de préserver la validité des

références déictiques lors de la communication entre utilisateurs. Sur la Figure 30 il est possible de voir l'utilisateur qui vient de virtualiser son écran pointer le coin en haut à droite. La Figure 31 correspond à la scène du collaborateur qui voit l'utilisateur qui vient de virtualiser son écran pointer le coin de cet écran.



Figure 30: Utilisateur pointant le coin de son écran virtualisé



Figure 31: L'avatar reproduit le geste dans le diorama correspondant chez le collaborateur

## 6.2 Gestes sur les écrans du collaborateur

Cette interaction a pour but de permettre à un utilisateur de passer sa main à travers le diorama pour effectuer un geste sur l'écran du collaborateur distant. Cela permet, par ailleurs, de rétablir la co-présence dans l'échange.

Sur la Figure 32, l'utilisateur passe son bras droit à travers le diorama. Le contenu virtuel masque une partie du bras; seul le bout de la main, en gris clair, est visible à l'intérieur du diorama. Cette main droite, apparaît en bleu chez l'autre utilisateur au niveau de l'endroit pointé (Figure 33).



Figure 32: Un utilisation passe la main dans le diorama pour montrer un élément sur l'écran virtuel du collaborateur

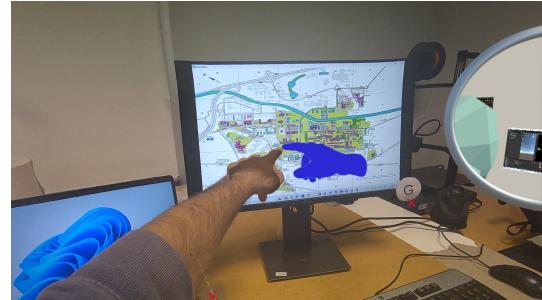


Figure 33: Cette main apparaît dans la scène de son collaborateur

### 6.2.1 Détection de la main à travers le diorama

Dans un premier temps, il a fallu se poser la question de ce que signifiait concrètement « mettre la main à l'intérieur du diorama ». Faut-il que la main passe uniquement par le judas ? Cependant, dans le cas d'un petit judas, il est possible que la main ne puisse pas le traverser ou, du moins, ne soit pas détectée.

L'approche retenue a donc plutôt été basée sur un critère de visibilité : tant que l'utilisateur voit sa propre main à travers le judas dans le diorama, celle-ci reste visible dans la scène de l'autre. Dès qu'elle n'est plus visible par le judas, elle disparaît de l'environnement du collaborateur.

L'approche conceptuelle retenue repose sur ce qu'on pourrait appeler un "cône de champ de vision".

La Figure 34 illustre le fonctionnement des clipping planes utilisés par le module caméra de Unity pour le rendu 3D. Dans Unity, la caméra définit le point de vue à partir duquel la scène est observée : elle projette l'espace tridimensionnel sur un plan image, qui correspond à ce qui sera affiché à l'écran — que ce soit sur un moniteur classique ou dans un casque Meta Quest. Seuls les objets situés entre le near clipping plane et le far clipping plane sont pris en compte par la pipeline graphique et projetés sur ce plan image, formant ainsi la portion visible de la scène pour l'utilisateur.

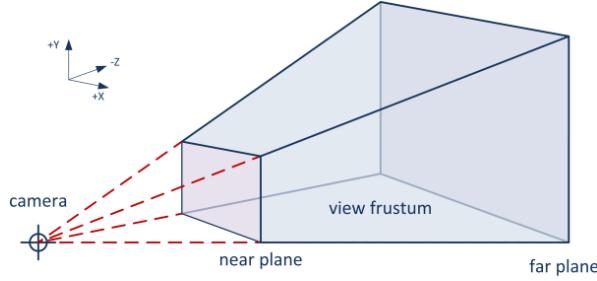


Figure 34: Caméra et clipping planes

En revenant au diorama, la main de l'utilisateur ne doit être affichée chez le collaborateur distant que si elle se trouve spatialement derrière le judas et que, du point de vue de la caméra, elle est visuellement contenue dans le cercle formé par le judas. En d'autres termes, elle doit être dans le cône de vision dont le sommet est la position de la caméra, et dont la base est déterminée par le diamètre du judas. La Figure 35 illustre cette configuration. La zone achurée en bleu correspond à l'espace dans lequel si la main entre alors elle doit être transmise à l'autre utilisateur. Le disque gris représente l'ouverture du portail.

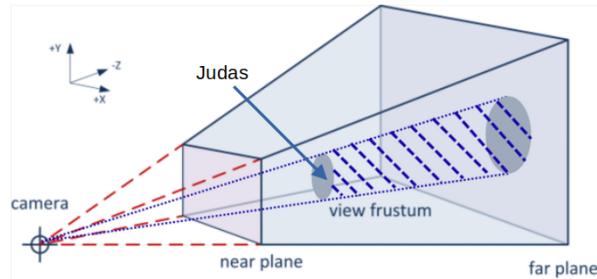


Figure 35: Caméra et clipping planes

Ainsi, pour qu'une main soit visible dans la scène de l'autre collaborateur, deux conditions doivent être remplies :

- Elle doit se situer **dans le cône de vision**,
- Elle doit être positionnée **derrière le judas**.

Déterminer de manière analytique l'espace visible à travers le judas, en fonction du point de vue de la caméra, s'est révélé trop complexe. Une méthode envisagée consistait à effectuer des calculs d'angles par rapport à l'axe de la caméra, mais leur mise en œuvre s'est avérée difficile.

Une autre approche plus pragmatique a alors été envisagée : utiliser les Raycasts ou Spherecasts de Unity. Le principe est simple : on trace un rayon entre la main et la position de la caméra, et on vérifie si ce rayon intersecte le judas. Si c'est le cas, la main est considérée comme visible dans l'espace défini par le diorama ; sinon, elle ne l'est pas. Sur la Figure 36 et 37 il est possible d'avoir deux vues de la situation dans laquelle la main est à l'intérieur du Diorama. Le rayon rouge de la Figure 36 correspond au Raycast.

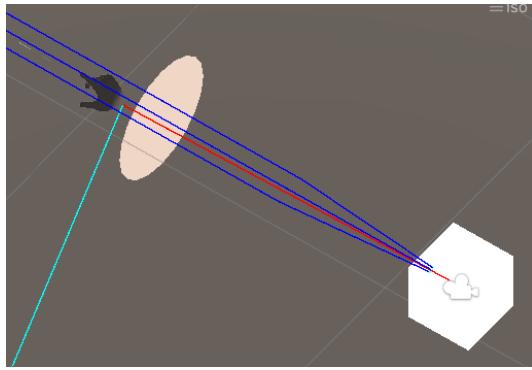


Figure 36: Vue dans éditeur Unity: Raycast à travers le judas

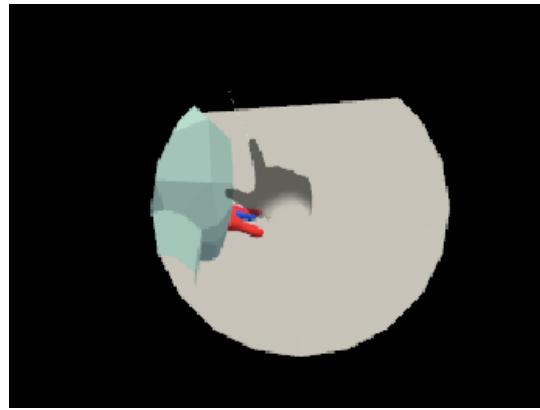


Figure 37: Vue dans le simulateur : main à l'intérieur du Diorama

Dans la Figure 38 et 39 on peut constater que lorsque le rayon ne traverse pas le judas alors la main n'est pas visible à travers le judas depuis la camera. Elle n'est plus dans le "cône" de champs de vision de l'utilisateur à travers le judas.

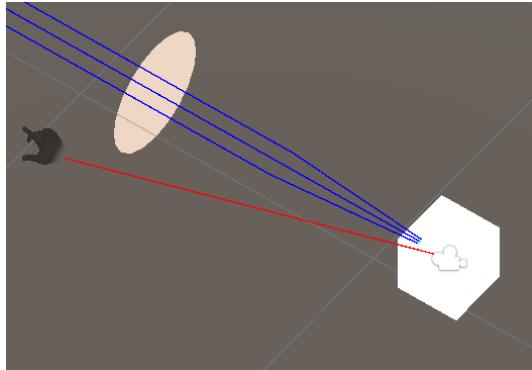


Figure 38: Vue dans éditeur Unity: Raycast hors du judas

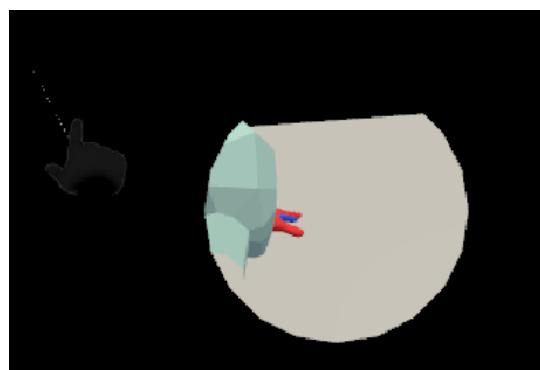


Figure 39: Vue dans le simulateur : main à l'extérieur du Diorama

Ainsi, de cette manière, il est possible de déterminer à chaque instant si la main est oui ou non dans le champ de vision de l'utilisateur à travers le judas.

Cependant, pour que les Raycasts ou Spherecasts détectent une intersection, l'objet interrogé doit disposer d'un Collider. Dans notre cas, le judas est modélisé comme un cylindre aplati, de manière à ressembler à un portail circulaire. Un Mesh Collider a donc été utilisé pour épouser au mieux la forme du judas. À noter qu'initialement, les Raycasts classiques ne fonctionnaient pas correctement avec le Mesh Collider du judas. Après avoir consulté un doctorant, il m'a été conseillé d'utiliser un Spherecast, c'est-à-dire un rayon avec un diamètre (semblable à un laser élargi). Cette méthode s'est révélée plus fiable et compatible avec notre configuration.

Grâce à cette approche, la détection de la présence de la main dans le Diorama est désormais fonctionnelle. Il reste ensuite à gérer l'affichage de cette main dans la scène du collaborateur distant.

### 6.2.2 Affichage des mains dans la scène du collaborateur

Il faut maintenant afficher dans l'espace de travail du collaborateur la main de manière à rester cohérent avec la position relative de la main dans le diorama par rapport au workspace. C'est à dire, par exemple, que si un utilisateur applique au workspace une rotation avec le Joytick du diorama alors lorsqu'il va passer sa main dans le diorama, une main virtuelle doit apparaître dans la scène du collabora-

teur avec aussi une certaine rotation. La main doit être reproduite chez le collaborateur telle qu'elle est vue depuis la tête de l'avatar du collaborateur dans le diorama. Par exemple, sur les Figures 32 et 33 la main bleue apparaît bien à droite de la main du collaborateur tout en pointant aussi sur l'écran.

Pour récupérer les positions et rotations relatives de la main par rapport à l'avatar et aux écrans, j'ai décidé d'utiliser le repère du workspace. Tous les gameObjects présents dans le diorama sont enfants du workspace. Puisque l'origine du "workspace" dans le diorama correspond à l'origine du monde dans la scène du collaborateur qui va voir apparaître la main, alors il suffit de transmettre à ce collaborateur la position et rotation locale de la main par rapport au workspace. Afin de manipuler un objet moins complexe que la main, j'ai décidé d'utiliser une sphère positionnée au niveau du poignet comme visible en Figure 40. Sur cette Figure, la sphère est visible mais lors d'un cas d'usage, elle est rendue invisible par l'utilisateur.



Figure 40: Sphère au niveau du poignet, le bras est partiellement caché par les éléments virtuels

Le principe est le suivant: lorsque l'utilisateur passe sa main dans le cône de champs de vision et tel qu'il voit sa main à travers le judas, alors cette sphère est instanciée. Sa position correspond, à tout instant, à la position du poignet de la main et sa rotation locale, à la rotation du poignet. La sphère est rendue enfant du "workspace" donc la rotation locale et position locale de la sphère et donc de la main par rapport au workspace est connue à tout instant. Il suffit ensuite de transmettre ces données au collaborateur distant pour que le système affiche une main virtuelle dans sa scène. Ainsi, une main (prefab) est instanciée dans la scène du collaborateur avec comme position et rotation les données transmises.

## 7 Limites et perspectives

### 7.1 Les limites du diorama

Dans les parties précédentes, les fonctionnalités offertes par « le Diorama à Judas » ont été détaillées. Cependant, par rapport aux problématiques, il présente tout de même certaines limites que j'ai pu identifier lors de la phase de développement.

Tout d'abord, l'avatar tel qu'il est conçu ici ne permet pas de transmettre les expressions faciales du collaborateur, contrairement à ce que certains articles de l'état de l'art proposent. Or, comme mentionné précédemment, une communication et une collaboration efficaces entre humains reposent également sur le non-verbal. Cela peut donc constituer une limitation dans les échanges. Il serait intéressant, à ce titre, de remplacer la tête de l'avatar par une tête animée reflétant les expressions faciales, ou par une tête d'avatar réaliste.

De plus, actuellement, l'outil de collaboration est limité à deux utilisateurs, ce qui peut poser problème, car une collaboration implique souvent plus de participants. Plusieurs solutions peuvent être envisagées : on pourrait, par exemple, disposer plusieurs dioramas dans notre champ visuel via le casque, mais cela risquerait de l'occulter. Une autre approche consisterait à modifier le contenu à l'intérieur du diorama, en affichant successivement l'espace de travail du collaborateur que l'on souhaite consulter. Avec cette approche, on a besoin de 1 seul diorama où l'on peut, par exemple, changer de

collaborateur en choisissant son numéro.

La représentation de l'espace du collaborateur à travers le Diorama reste assez sommaire. En effet, elle se limite à un avatar minimaliste et à des écrans représentés par de simples rectangles. Il serait intéressant, par exemple, de réaliser une reconstruction 3D de l'environnement d'un collaborateur à l'aide de caméras et de l'afficher dans le Diorama. Cette approche reprendrait l'idée développée par Huayuan Tian et al. [9]. Les seuls éléments de son environnement de travail qu'un collaborateur peut partager sont les écrans. Il pourrait être intéressant d'ouvrir la possibilité de pouvoir partager d'autres éléments comme le font certains articles de l'état de l'art. On pourrait penser à partager des feuilles sur lesquelles du contenu serait écrit et qu'un collaborateur souhaiterait partager. Cela pourrait passer par une virtualisation de la feuille et de son contenu puis afficher virtuellement cette dernière dans l'espace de travail distant grâce à des cameras 3D et nuages de points. Cette approche peut avoir des inconvénients car faire une reconstitution en nuage de points de la feuille peut manquer de précision à cause des caméras 3D. Si un objet se trouve entre la camera 3D et la feuille alors une partie de la feuille ne sera pas correctement reconstruite dans l'espace distant. Cela peut aussi manquer de précision concernant le niveau de détail de la feuille.

Enfin, il n'y a pas de tracking entre le moniteur réel et l'écran virtuel. Si le moniteur est déplacé dans l'espace de travail d'un collaborateur (il le bouge) alors l'écran virtuel associé construit dans le diorama n'est pas déplacé.

## 7.2 Perspectives d'évaluation

Bien que je n'aie pas eu le temps de mener des expériences et de concevoir des protocoles expérimentaux complets pour tester le système présenté avec des utilisateurs, je propose ci-dessous des pistes de réflexions pour évaluer son utilisation dans des conditions proches de l'usage réel.

L'objectif de ces tests serait de vérifier si la collaboration est effectivement plus efficace avec le diorama qu'avec des outils tels que Zoom. Le prototype a été développé sur l'hypothèse qu'il présente des avantages par rapport à ces outils, mais ceux-ci restent à confirmer expérimentalement.

Ces tests permettraient, par exemple, de déterminer des caractéristiques optimales pour le diorama, tels que la taille idéale du judas, le meilleur point de vue dans le diorama. Impliquer des utilisateurs dans ces évaluations permettrait de recueillir des informations précieuses pour améliorer le dispositif.

Un protocole de test pourrait être le suivant:

Deux utilisateurs seront placés :

- soit dans deux pièces séparées,
- soit dans une même pièce, mais de manière à ne pas pouvoir se voir directement.

Cette configuration vise à simuler une situation typique de visioconférence où les collaborateurs sont physiquement éloignés. Chaque participant disposera d'un ou plusieurs écrans physiques. Il disposeront chacun d'un casque de Réalité Mixte Meta Quest 3.

Les utilisateurs seront ensuite invités à évaluer les aspects suivants :

**Vue de l'espace de travail distant via le judas** Les participants devront manipuler la taille du judas et déplacer l'intérieur du Diorama. Le but est de vérifier :

- la visibilité de l'espace de travail du collaborateur,
- la facilité de navigation et d'observation via le judas.

### Virtualisation des écrans

Il leur sera demandé de virtualiser un ou plusieurs de leurs écrans physiques. Cette opération se fait en marquant trois coins de chaque écran afin d'en définir le contour.

L'objectif est d'évaluer si le dispositif de virtualisation est :

- facile à utiliser,
- précis dans la reproduction des écrans physiques dans l'espace virtuel.

Les écrans virtualisés devront apparaître dans l'espace de travail du collaborateur distant.

**Interaction gestuelle avec le Diorama** Les utilisateurs devront passer leur main dans le Diorama pour effectuer un geste sur l'écran virtualisé du collaborateur. Ce test vise à évaluer :

- la précision du geste,
- la compréhension mutuelle (le collaborateur doit percevoir correctement le geste et l'endroit où il a été effectué) en voyant une main pointer l'élément concerné dans sa scène.

Bien que je n'ai pas pu faire tester à des utilisateurs l'application en suivant un protocole de test, j'ai pu l'utiliser avec un autre utilisateur (*doctorant*) dans une situation qui se rapproche d'un contexte d'usage. On était, en effet, assis chacun à un espace multi-écran avec le casque de Réalité Mixte.

## 8 Déroulement du stage et difficultés rencontrées

### 8.1 Interaction avec les membres de l'équipe

À mon arrivée en début de stage, un autre stagiaire travaillait déjà sur le projet depuis environ un mois. Nous avons collaboré durant un mois et demi, avant qu'il termine son stage et que je poursuive seul le développement. Le projet était entièrement nouveau : aucune base ni historique n'existaient avant notre arrivée, ce qui impliquait de tout construire à partir de zéro. Les interactions avec les membres de l'équipe, et en particulier avec mon tuteur, se sont inscrites dans une démarche plutôt exploratoire. Au départ, j'étais guidé sur les fonctionnalités à implémenter afin de bien cerner le sujet, tout en disposant d'une certaine liberté technique dans mes choix d'implémentation. Au fil de l'avancement du prototype, les échanges avec mon tuteur portaient sur les prochaines étapes du développement, et je devais proposer des solutions techniques adaptées aux nouvelles fonctionnalités envisagées pour le diorama. J'ai également échangé avec plusieurs doctorants du laboratoire, notamment lorsque je rencontrais des difficultés techniques ou que j'avais besoin d'idées pour implémenter certaines fonctionnalités. Ils ont aussi pu m'aider à tester le prototype et à recueillir des retours pertinents sur son fonctionnement.

### 8.2 Divergence entre le travail prévu et le travail effectif

Le développement d'un tel système en Réalité Mixte m'a confronté à de nombreuses difficultés et imprévus, que j'ai dû surmonter avec persévérance et rigueur. Ces challenges ont eu pour conséquence d'impacter le planning prévisionnel et m'ont appris à m'adapter et à rebondir face aux obstacles.

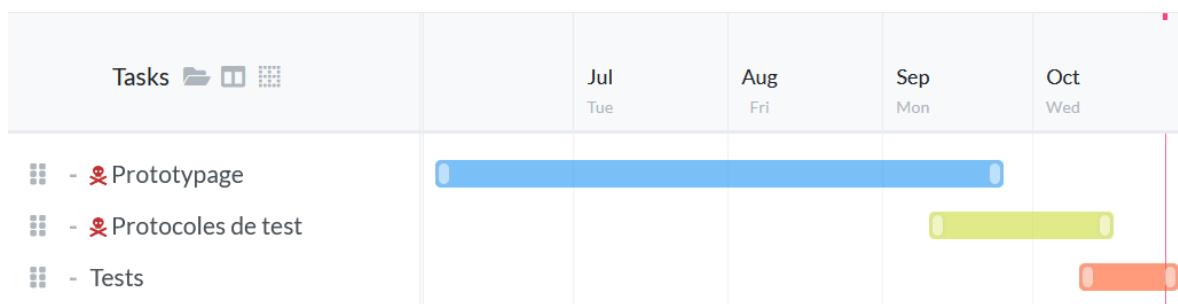


Figure 41: Gantt Prévisionnel

Des problèmes techniques sont survenus durant le stage ce qui a impacté le déroulé prévu. En Figure 41, est visible le Gantt prévisionnel qui incluait en fin de stage une mise en oeuvre des protocoles de test et leur execution avec des utilisateurs. Ces problèmes m'ont fait prendre du retard et je n'ai donc pas pu mener à bien la rédaction d'un protocole complet et le faire tester.

Sur les Figures 42 et 43 est visible le déroulé effectif du stage. Sur 43 la grande barre bleue (4 semaines) et jaune donne une idée de la durée qu'a pris la résolution du problème du positionnement

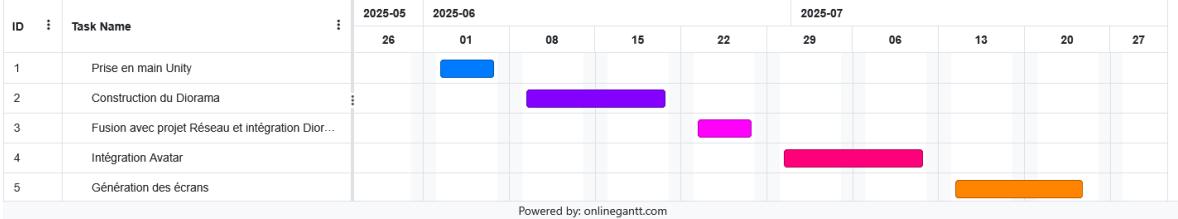


Figure 42: Gantt effectif Juin - Fin Juillet

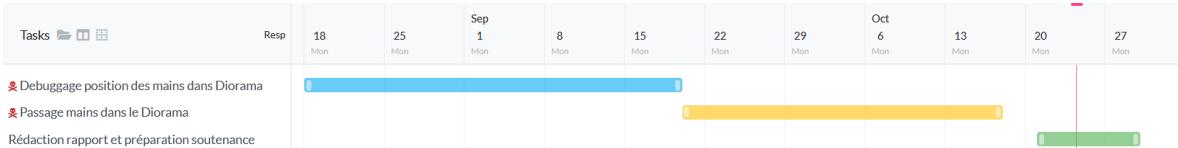


Figure 43: Gantt effectif Mi-Août Fin-Octobre

des mains, détaillé par la suite. Dans les lignes qui vont suivre, seront explicitées les différentes tâches qui étaient prévues ainsi que les problèmes rencontrés pendant leur exécution:

Dans un premier temps, il a été nécessaire que je prenne en main Unity et de le configurer pour un usage en Réalité Mixte.

Une fois l'environnement maîtrisé, j'ai du implémenter un « diorama à judas » en Réalité Mixte: une représentation miniature de l'espace de travail d'un collaborateur, modélisé comme une pièce fermée dont les écrans sont visibles à travers un trou ou un portail.

Par la suite, il a fallu intégrer ce diorama au travail réalisé en parallèle par un autre stagiaire, qui s'était concentré sur la mise en place d'une architecture réseau. Cette intégration visait à permettre une collaboration en temps réel entre au moins deux utilisateurs connectés. Le stagiaire avec qui je travaillais avait fini son stage donc je me suis retrouvé seul sur le projet.

J'ai ensuite développé des interfaces d'interaction pour permettre aux utilisateurs de manipuler le diorama dans leur propre espace de travail, ainsi que de gérer dynamiquement le contenu affiché à l'intérieur (par exemple, les écrans virtuels, le point de vue). C'est à ce moment qu'est intervenu le problème du positionnement des mains lorsque j'ai commencé à tester l'affichage des écrans au sein du diorama.

Une fois ce problème résolu, le placement des mains était alors cohérent avec celui des écrans. J'ai alors pu développer la possibilité d'interagir avec l'espace du collaborateur distant, c'est-à-dire pouvoir passer la main dans le diorama.

Le stage s'est ensuite terminé par un moment dédié à la rédaction du rapport et la préparation de la soutenance orale.

### 8.3 Problèmes rencontrés d'un point de vue général

Le développement de l'application a été plus long que prévu, principalement en raison des nombreux bugs rencontrés au fil du processus. Étant seul à travailler sur cette partie, leur résolution a parfois été chronophage, malgré la possibilité ponctuelle de demander de l'aide.

Tester une application en Réalité Mixte représente également un défi particulier : bien que le simulateur intégré à Unity permette de faire une première évaluation sur ordinateur, il ne garantit pas un comportement identique une fois déployée sur les casques. Cela a nécessité des tests fréquents sur les appareils réels (casques Meta Quest), impliquant d'autres personnes pour simuler une utilisation

collaborative. Cette contrainte a parfois ralenti l'avancement, mais elle était essentielle pour valider le bon fonctionnement de l'application dans des conditions réalistes.

#### 8.4 Cas du positionnement des mains lors du partage d'écran: un exemple chronophage

Un problème a été mis en évidence lors des tests de virtualisation et de partage d'écrans. En effet, alors que je pensais que tout fonctionnait correctement, j'ai constaté un décalage entre la position des mains et celle de l'écran dans le diorama. Par exemple, si je créais un écran virtuel dans mon espace de travail et pointais le coin supérieur droit, mon collaborateur ne voyait pas la main de mon avatar pointer ce coin, mais légèrement à côté. Sur le Diagramme de Grantt en Figure 43, ce problème est représenté par la barre bleue clair qui s'étale sur près de 4 semaines.

La première étape a donc été d'identifier l'origine de ce décalage : provenait-il de la position de l'écran ou de celle de la main ? Pour cela, j'ai vérifié les coordonnées de l'écran dans le diorama en affichant des sphères sur ses coins, accompagnées de leurs coordonnées. Cette vérification a confirmé que l'écran était correctement positionné. Le décalage provenait donc de la main.

J'ai ensuite tenté de suivre le mouvement du tip (extrémité) de la main virtuelle dans le diorama. Pour cela, j'ai ajouté un gameObject cube au niveau du tip, que j'ai déplacé au sein de la hiérarchie de la main afin de comprendre le comportement des différents composants. En observant les GameObjects côté client et serveur pendant l'exécution dans l'éditeur Unity, j'ai remarqué qu'en pointant le coin de l'écran virtuel en tant qu'utilisateur, le centre de la main correspondante dans le Diorama était aligné avec ce coin. Par exemple, sur la figure 44, en fonctionnement correcte le bout de l'index devrait pointer sur la sphère (coin de l'écran virtuel).

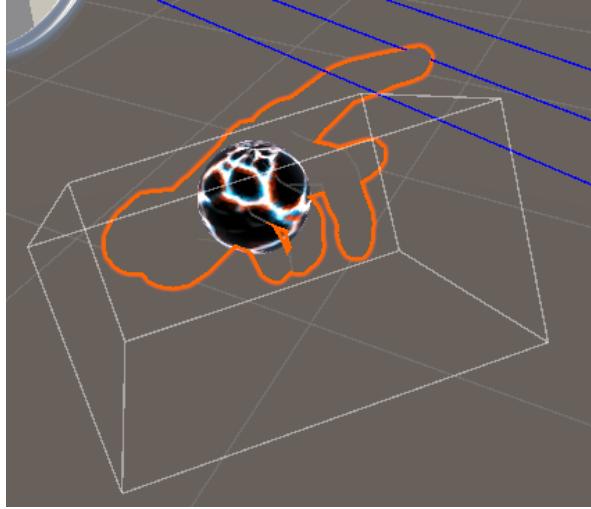


Figure 44: Décalage entre tip et le coin de l'écran (sphère noire)

En analysant plus en profondeur la structure de la main, j'ai compris que le repère d'origine n'était pas situé au niveau du tip, mais au centre de la main. Le problème provenait donc d'un offset entre les coordonnées envoyées au serveur et la position réelle de la main dans le diorama.

J'ai ainsi mis en évidence un décalage entre la main OpenXR (via MRTK) et la main de l'avatar côté client. Pour mieux comprendre l'origine de cet écart, j'ai affiché les joints (articulations) des mains MRTK et ceux de l'avatar, afin d'identifier le point de divergence. Sur la Figure 45 j'ai affiché les joints de la main OpenXr (trackée par le casque) et la main de l'Avatar (player prefab sur le réseau). Le décalage est flagrant.

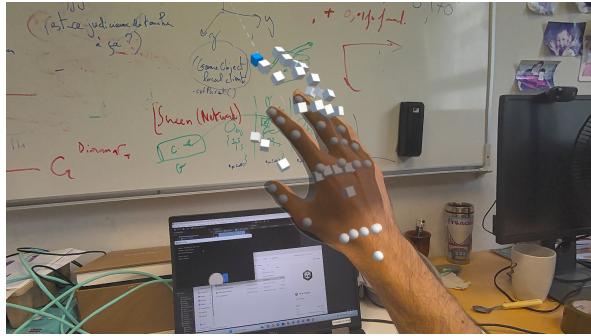


Figure 45: Main OpenXR (sphères) Main Avatar (cubes)

Après plusieurs tests et tentatives, j'ai compris que la coordonnée envoyée au serveur correspondait au centre de la main, et non à l'armature (ou structure osseuse) complète de la main, ce qui explique le décalage observé.

Il suffisait alors de faire en sorte que la main MRTK et la main de l'avatar (player prefab) étaient superposées lorsqu'on se connecte en tant que client avec le casque. Sur la Figure 46 on peut observer le résultat obtenu lorsque les mains sont correctement synchronisées.

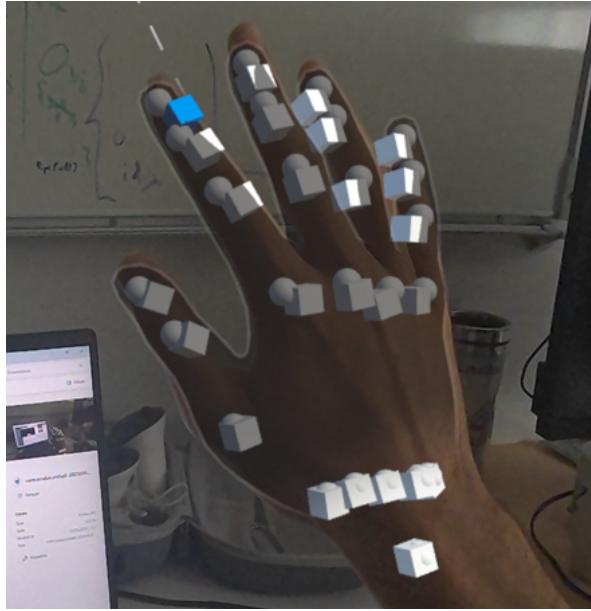


Figure 46: Mains correctement synchronisées

Ainsi, l'Armature de la main dans le Diorama recevait les coordonnées de l'Armature de la main MRTK. La main dans le diorama était donc correctement synchronisée, dans son espace local, avec celle de la main utilisateur dans son espace de travail.

En résumé, j'ai synchronisé le poignet de la main de l'avatar avec celui de la main MRTK suivie par le casque.

Jusqu'à présent, la main de l'avatar avait une taille fixe (issue d'un prefab). J'ai donc modifié le code ainsi que certains composants Unity afin d'adapter dynamiquement, à la connection du client, la taille des joints de l'avatar à ceux de la main MRTK suivie en temps réel. Cela permet à la main de l'avatar de mieux correspondre à la morphologie de la main réelle de l'utilisateur.

Ce processus de débogage a nécessité la mise en place de plusieurs stratégies. La première a été

d'ajouter de la visibilité en affichant des objets supplémentaires dans la scène, ou encore en inscrivant certaines coordonnées dans la console. Cependant, pour des raisons techniques, il m'était impossible de connecter les casques lorsque le serveur est lancé depuis l'éditeur Unity . Il a donc fallu trouver des alternatives pour afficher les logs ou les joints directement dans le casque. Ce problème de connection du casque à l'éditeur Unity en mode server a, par la suite, été résolu en désactivant des parefeux Windows.

## 9 Impacts environnemental et sociétaux du système développé

Lors du développement logiciel, de nombreux outils technologiques et numériques sont utilisés. Qu'il s'agisse de leur production ou de leur consommation électrique, ils ont un impact sur l'environnement.

Au-delà de cet impact environnemental, ce système est destiné à être utilisé par des êtres humains dans le cadre de la collaboration à distance multi-écrans, ce qui peut influencer les conditions de travail quotidiennes des utilisateurs.

Il est donc pertinent d'examiner à la fois les impacts environnementaux du système développé et ses répercussions sur le plan social.

### 9.1 Consommation énergétique

Le système développé repose principalement sur l'utilisation d'un ordinateur portable, de casques de réalité mixte (type Meta Quest), d'un serveur local, ainsi que d'un ventilateur ponctuellement utilisé en cas de forte chaleur. L'ensemble de ces équipements, ainsi que l'utilisation d'Internet et du moteur Unity, engendre une certaine consommation énergétique.

Contrairement aux outils de collaboration traditionnels comme Zoom ou Microsoft Teams, qui reposent sur un échange continu de flux audio-vidéo, l'application développée ici utilise des casques de Réalité Mixte. Ceux-ci ne transmettent pas directement de vidéo, mais permettent de visualiser un espace de travail distant sous forme d'objets virtuels, dans lequel les écrans du collaborateur peuvent être partagés. L'usage des casques induit donc une consommation énergétique supplémentaire, notamment lors de la phase d'affichage et de traitement 3D en temps réel.

### 9.2 Impact sur la collaboration à distance

Pour évaluer plus finement l'impact environnemental de cette solution, il serait pertinent de la mettre en perspective avec celui d'un déplacement physique. Par exemple, on pourrait comparer la consommation énergétique d'une session de collaboration immersive à distance — entre deux collaborateurs situés à des centaines de kilomètres, chacun dans une salle équipée — avec celle d'un déplacement en présentiel (par train ou avion). Dans ce cas, la solution en Réalité Mixte pourrait présenter un intérêt environnemental en réduisant les émissions liées au transport, notamment si elle est utilisée régulièrement pour éviter des trajets longues distances.

Du point de vue des utilisateurs, certains aspects du système en Réalité Mixte peuvent présenter des limites. Par exemple, le fait de devoir bouger constamment les bras pour interagir avec les éléments à l'écran peut devenir fatigant sur de longues périodes, surtout en comparaison avec l'utilisation d'une souris, qui est beaucoup moins exigeante sur le plan physique.

Par ailleurs, le port prolongé d'un casque de réalité mixte peut engendrer une fatigue visuelle et physique, ce qui impose une vigilance quant à la durée d'utilisation. Le poids du casque constitue également un facteur à ne pas négliger lors d'un usage prolongé. À ce titre, on peut le comparer aux lunettes de réalité mixte actuellement en développement chez Meta : bien que moins performantes en matière de détection de l'environnement, elles se distinguent par leur légèreté, offrant un meilleur confort à l'usage.

### 9.3 Contexte de développement

J'ai rejoint l'IRIT, en tant que stagiaire, qui met en place des initiatives qui adressent l'impact environnemental du laboratoire. En particulier, l'IRIT a une mission dédiée à la transition écologique du laboratoire [7].

Cette mission a pour objectifs de susciter et d'accompagner les changements de pratiques en faveur d'une transition écologique, ainsi que de favoriser les débats autour de l'évolution des métiers et des recherches dans ce contexte.

Parmi les objectifs principaux, il s'agit notamment de sensibiliser le personnel à l'impact environnemental des outils informatiques, en particulier en ce qui concerne leur fabrication et leur utilisation. Par exemple, les dispositifs de Réalité Mixte peuvent consommer une quantité significative d'électricité. Il est donc important de penser à les éteindre ou les mettre en veille lorsqu'ils ne sont pas utilisés. J'ai pu faire le constat, dans des salles de classes, espaces administratifs ou même parfois à l'IRIT que certains postes restent allumés ou en veille alors que les utilisateurs ne sont pas présents. Cela induit alors une consommation énergétique inutile.

Un autre objectif important de la mission est le calcul du bilan carbone du laboratoire, afin de disposer de données chiffrées sur lesquelles s'appuyer. L'IRIT vise notamment à réduire son empreinte carbone de 35% à 55% d'ici 2030, par rapport à l'année 2019. Parmi les mesures envisagées, on peut citer la préférence donnée au train plutôt qu'à l'avion pour les déplacements professionnels. Des ateliers participatifs sont organisés pour délibérer collectivement sur les scénarios et les stratégies à adopter pour atteindre ces objectifs.

## 10 Conclusion

### 10.1 Synthèse du travail réalisé

Dans un premier temps, ce rapport a mis en évidence l'usage croissant des dispositifs multi-écran pour la collaboration à distance, notamment accentué par la pratique du télétravail. La distance physique entre les collaborateurs soulève 3 problématiques:

- P1. Comment partager, entre collaborateurs, le contenu et la disposition des écrans de leur environnement de travail ?
- P2. Comment, en tant qu'utilisateur, puis-je communiquer mes gestes sur mes écrans à un collaborateur distant ?
- P3. Comment effectuer des gestes dans l'environnement d'un collaborateur distant ?

Ensuite, l'état de l'art a permis d'explorer plusieurs travaux menés, qui apportent des pistes de réponse à ces problématiques. L'utilisation de la Réalité Mixte dans ces articles ajoute une dimension spatiale/3D, elle en effet des possibilités pour enrichir les échanges à distance. Néanmoins, des limites persistent comme l'occlusion du champ visuel par les contenus virtuels lors de l'affichage de l'espace de travail du collaborateur distant chez un utilisateur.

Dans ce contexte, le concept du "diorama à judas" a été introduit et positionné comme une solution originale pour la collaboration à distance en environnement multi-écran. Les interactions basées sur la métaphore du "diorama à judas" présentées dans ce rapport permettent une représentation spatiale cohérente et non-occultante de l'espace de travail du collaborateur distant, grâce à une reconstitution virtuelle de ses écrans et à la visualisation de son avatar.

La virtualisation et le partage des écrans ont ensuite été abordés comme manière de partager à la fois le contenu des écrans et l'agencement de ces derniers dans l'environnement de travail distant. De plus, la possibilité d'interagir avec l'intérieur du diorama, par exemple en pointant des éléments sur les écrans du collaborateur, ouvre la voie à une communication bilatérale.

Plusieurs pistes de protocole expérimental ont été proposées afin d'évaluer concrètement la pertinence du diorama dans un contexte d'usage réel. Ces protocoles visent à tester la facilité de virtualisation, la qualité de la visualisation à travers le judas, ainsi que la précision des interactions gestuelles.

Les limites du diorama témoignent qu'il peut être amélioré. Des pistes comme l'amélioration de l'apparence de l'avatar et l'augmentation du nombre de collaborateurs seraient à explorer ultérieurement.

Enfin, une réflexion sur la divergence entre le travail prévu et le travail effectif pendant ce stage a été abordée afin de mettre en évidence que toute aventure ne se passe jamais sans rebondissements !

## 10.2 Apports techniques et personnels

Grâce à ce stage, j'ai pu acquérir de solides compétences techniques, notamment en explorant des outils informatiques et technologiques que je ne maîtrisais pas auparavant. J'ai particulièrement approfondi mes connaissances sur l'environnement de développement Unity, le langage C#, ainsi que sur l'utilisation des casques de Réalité Mixte Meta Quest 3. La prise en main de ces outils m'a permis de mieux comprendre les enjeux techniques liés au développement d'interfaces immersives, tout en m'exerçant à la conception d'expériences interactives en Réalité Mixte.

Par ailleurs, j'ai significativement développé mon autonomie, en particulier lors de la phase de développement, où j'ai été confronté à de nombreux problèmes techniques. La résolution de ces bugs a souvent nécessité de longues phases de recherche, d'expérimentation et de test. Cette expérience m'a permis de renforcer ma capacité d'analyse, ma patience et ma résilience, des qualités essentielles dans tout travail de développement logiciel.

Au-delà des compétences techniques, ce stage m'a également offert une expérience humaine et professionnelle enrichissante. J'ai eu l'opportunité de travailler avec une équipe de recherche en participant aux séminaires hebdomadaires du laboratoire. Ces moments d'échange m'ont permis de découvrir des sujets variés, d'élargir ma culture scientifique, et de mieux comprendre la manière dont les chercheurs communiquent, collaborent et structurent leurs projets.

Ce stage m'a donné un aperçu concret du travail en laboratoire de recherche, avec ses exigences de rigueur, d'organisation, et de documentation. Il m'a aussi appris à mieux communiquer mes avancées à l'oral lors de réunions, par exemple. Il est parfois nécessaire de repartir de zéro pour mieux comprendre ou corriger un comportement. Git facilite ces retours en arrière si besoin. Il reste toutefois indispensable de tester l'application directement sur les casques, même si tout semble fonctionner en simulation. Comprendre un problème issu du travail d'un tiers peut être long et décourageant, surtout lorsqu'il persiste malgré les efforts. Dans ces situations, il est important de rester patient et de s'approprier progressivement le code existant.

Enfin, communiquer régulièrement avec les tuteurs permet de prendre du recul, de partager les avancées et de demander de l'aide si nécessaire.

## A Prise en main de Unity

Cette courte partie concerne la configuration de Unity afin de pouvoir travailler avec la Réalité Mixte. La mise en place de cet environnement de développement a nécessité plusieurs jours, en s'appuyant sur des tutoriels disponibles en ligne, notamment ceux des wikis de Microsoft et de Unity. Le processus d'installation sera accompagné de références vers ces tutoriels, qui offrent des explications plus complètes et structurées étape par étape.

### A.1 Configuration pour Android et packages

Afin que Unity reconnaisse le Casque Meta Quest 3, il est nécessaire de configurer les "Build settings" afin de basculer sur une compilation sur Android.

Ensuite, il est nécessaire d'installer quelques packages supplémentaires afin de pouvoir travailler en Réalité Mixte sur Unity. En particulier, nous avons besoin du "Unity OpenXR Plugin" et enfin les "Meta XR SDKs" pour l'utilisation du "passthrough", notamment, dont nous reviendrons dessus plus loin. Une explication plus détaillée du processus d'installation est disponible sur le site de Meta [5].

Concernant le casque Quest 3, pour qu'il soit reconnu par l'ordinateur et puisse communiquer avec Unity via un câble USB (par exemple pour y transférer un programme), il est nécessaire d'installer le Meta Developer Hub. Une fois installé, chaque fois qu'une connexion filaire sera établie entre le PC et le casque, une demande d'autorisation de connexion devra être validée directement dans le casque.

Il est alors possible de travailler avec Unity et d'exporter / build le programme sur le casque.

### A.2 Ajout du PassThrough

Le passThrough permet de visualiser l'environnement réel de l'utilisateur à travers le casque, grâce aux caméras situées à l'avant et sur les côtés de celui-ci. L'intégration du passThrough dans Unity s'est faite en ajoutant à la caméra fournie par MRTK les scripts OVRManager et OVR Passthrough Layer. Ces scripts ont ensuite été configurés pour garantir le bon fonctionnement de la fonctionnalité. Des tutoriels détaillés étaient disponibles sur le site de Meta [4].

Dans le tutoriel proposé par Meta, l'exemple repose sur l'utilisation de la OVRCameraRig. Cependant, j'ai finalement opté pour MRTK, qui dispose de son propre module de caméra, utilisé à la place.

## B Schéma illustrant le partage des écrans une fois virtualisés

Les données indiquées sur les flèches représentent les informations transmises. Dans cette situation, le Client 1 crée un écran en local. Il envoie au serveur l'identifiant de son Diorama — ici nommé "D1" à titre d'exemple — indiquant que le Diorama D1 existant sur le réseau représente son espace de travail. Il transmet également au serveur la position, la rotation et la taille de l'écran dans son propre référentiel. Le serveur relaie ensuite ces informations à tous les autres Clients, en leur demandant de retrouver le Diorama D1 dans leur scène et d'y ajouter l'écran. Les actions comme l'instanciation de l'écran sont effectuées localement par chaque Client.

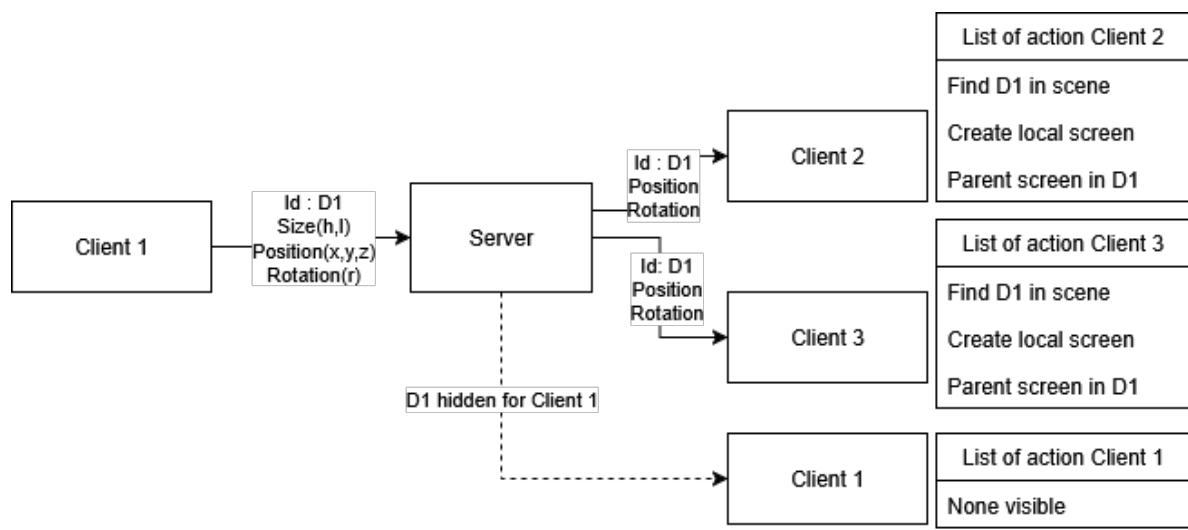


Figure 47: Génération Ecrans

## C Glossaire des éléments techniques de Unity

### C.1 GameObject

Un **GameObject** est l'unité de base dans Unity. Il agit comme un *conteneur* auquel on peut ajouter des composants pour définir son apparence, ses interactions ou son comportement. Par exemple, un GameObject peut représenter un cube visible, un joueur contrôlable, une lumière, une caméra, etc. Seul, un GameObject ne fait rien : ce sont ses *composants* qui lui donnent vie.

### C.2 Prefab Unity

Un **Prefab** (ou préfabriqué) est un GameObject (ou un ensemble hiérarchique de GameObjects) sauvegardé en tant que ressource réutilisable. Il peut être *instancié dynamiquement* dans la scène à tout moment, ce qui permet de dupliquer facilement des objets complexes avec leurs composants et scripts déjà configurés.

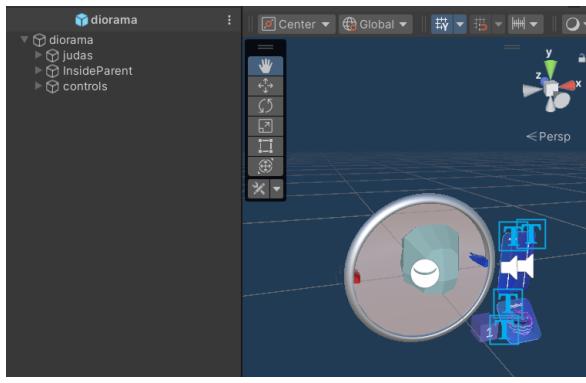


Figure 48: Prefab du Diorama

### C.3 Script C#

Les **scripts C#** permettent de définir les comportements des GameObjects dans Unity. Ils jouent un rôle essentiel : sans script, un objet peut être visible ou physique, mais il ne pourra pas réagir, interagir ou évoluer dans le temps. Un script peut, par exemple, gérer le mouvement d'un personnage, détecter une collision ou modifier l'apparence d'un objet.

Le C# utilisé dans Unity est un langage orienté objet, proche de Java ou C++. Il est particulièrement adapté à la programmation de comportements interactifs, car Unity propose un ensemble de classes prêtes à l'emploi (comme `MonoBehaviour`) pour gérer les objets, événements et interactions du moteur.

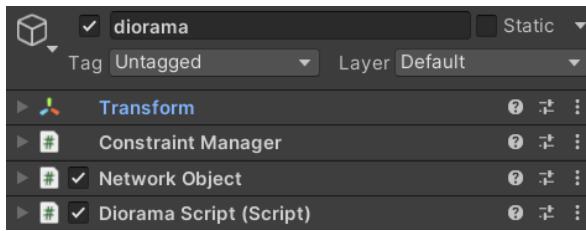


Figure 49: Scripts rattachés à la racine du prefab Diorama

## C.4 Scene View et Game View

La **Scene View** est la vue de l'éditeur dans laquelle on construit, place et ajuste les objets de l'application. Elle sert principalement à l'organisation spatiale des éléments dans la scène, en mode édition.

La **Game View** est la vue du joueur : elle montre ce que verra un joueur pendant l'exécution du jeu. Elle dépend entièrement de la *caméra principale* active dans la scène.

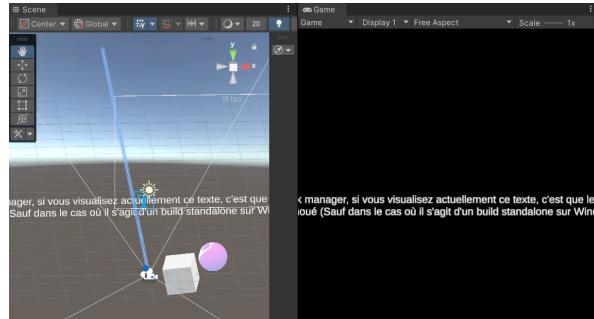


Figure 50: Scene View et Game View

## C.5 Dossier Assets

Le **dossier Assets** contient l'ensemble des ressources du projet Unity : modèles 3D, Prefabs, scripts, matériaux, textures, sons, bibliothèques, etc. C'est la base de données du projet, où sont stockés tous les éléments que Unity peut utiliser dans les scènes.

## References

- [1] Neil Chulongsatorn et al. *HoloDevice: Holographic Cross-Device Interactions for Remote Collaboration*. 2024. URL: <https://arxiv.org/pdf/2405.19377.pdf>.
- [2] Andrew Irlitti et al. *Volumetric Hybrid Workspaces: Interactions with Objects in Remote and Co-located Telepresence*. 2024. URL: <https://dl.acm.org/doi/10.1145/3613904.3642814>.
- [3] Akihiro Kiuchi et al. *MiniMates*. 2024. URL: <https://dl.acm.org/doi/10.1145/3706598.3714328>.
- [4] Meta. *Get Started with Passthrough*. 2024. URL: <https://developers.meta.com/horizon/documentation/unity/unity-passthrough-gs>.
- [5] Meta. *Set up Unity for XR development*. 2024. URL: <https://developers.meta.com/horizon/documentation/unity/unity-project-setup>.
- [6] Paul Milgram et al. *Augmented reality: A class of displays on the reality-virtuality continuum*. 1994. URL: [https://www.researchgate.net/publication/228537162\\_Augmented\\_reality\\_A\\_class\\_of\\_displays\\_on\\_the\\_reality-virtuality\\_continuum](https://www.researchgate.net/publication/228537162_Augmented_reality_A_class_of_displays_on_the_reality-virtuality_continuum).
- [7] *Mission transition écologique de l'IRIT*. 2024. URL: <https://www.irit.fr/missions/transition-ecologique/la-mission/>.
- [8] Ludwig Sidenmark et al. *Desk2Desk: Optimization-based Mixed Reality Workspace Integration for Remote Side-by-side Collaboration*. 2024. URL: <https://dl.acm.org/doi/10.1145/3654777.3676339>.
- [9] Huayuan Tian et al. *Using Virtual Replicas to Improve Mixed Reality Remote Collaboration*. 2023. URL: <https://ieeexplore.ieee.org/document/10049700>.
- [10] Ye Yuan et al. *Understanding Multi-Device Usage Patterns: Physical Device Configurations and Fragmented Workflows*. 2024. URL: <https://dl.acm.org/doi/10.1145/3491102.3517702>.