

Filière Systèmes industriels

Orientation Infotronics

Travail de bachelor Diplôme 2021

Alexandre Figueiredo

Augmented reality for remote assistance

■ Professeur
Dr. Francesco Carrino

■ Expert
Alain Gruet

■ Date de la remise du rapport
20.08.2021, 12 :00

Ce rapport est l'original remis par l'étudiant.
Il n'a pas été corrigé et peut donc contenir des inexactitudes ou des erreurs.

HES-SO Valais

**Données du travail de diplôme
Daten der Diplomarbeit**

FO 1.2.02.07.CB
haj/11/01/2016

SYND	ETE	TEVI
X	X	X

Filière / Studiengang SYND	Année académique / Studienjahr 2020/21	No TD / Nr. DA IT/2021/49
Mandant / Auftraggeber <input type="checkbox"/> HES—SO Valais <input checked="" type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire Partnerinstitution	Etudiant / Student Alexandre Ferreira Figueiredo Professeur / Dozent Francesco Carrino	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input type="checkbox"/> Etablissement partenaire Partnerinstitution
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja ¹ <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) Alain Gruet, Y-Parc Rue Galilée 15, 1400 Yverdon	
Titre / Titel Augmented reality for Remote assistance		
Description / Beschreibung Les outils de mise en service virtuelle sont utilisés pour le déploiement de nombreuses technologies, systèmes, machines et processus d'automatisation. L'utilisation de ces outils limite le nombre de personnes requises sur place et permet donc de réduire considérablement les coûts de mise en service. En outre, ces mêmes outils peuvent être utilisés pour mettre en place un environnement de formation réaliste : – Pour former les opérateurs avant que le système réel ne soit mis en service – Pour former les nouveaux opérateurs aux séquences de démarrage, de transition et d'arrêt. La Réalité Augmentée (RA) peut être un précieux allié pour aider à l'installation et à la maintenance des projets d'automatisation. Objectifs / Ziele – Analyser et comparer les capacités et les limites des dispositifs de RA (Head Mounted Displays et dispositifs mobiles) et des services existants (TeamViewer Pilot, Vuforia chalk, WorkLink, etc.). – Concevoir, développer et tester un proof-of-concept pour le cas d'utilisation proposé par le partenaire industriel du projet (ALPS Automation). – [Objectif secondaire/optionnel] Effectuer des tests d'utilisabilité avec les utilisateurs finaux. Tâches Analyse – Bref état de l'art sur l'utilisation de la réalité augmentée dans l'industrie 4.0, les technologies existantes (HoloLens, RealWear, etc.) et les services (TeamViewer Pilot, Vuforia chalk, WorkLink, etc.) – Prise en main de Microsoft HoloLens 2 (d'autres technologies peuvent être envisagées) – Comparaison entre Unity, Unreal Engine, Microsoft Mesh – [Facultatif] développement de mini-démonstrateurs avec les trois technologies. Conception : – Définition de l'interface utilisateur (modalités d'interaction) – Définition de l'architecture du système Développement et tests – Développement d'un prototype pour le cas d'utilisation d'ALPS – Test (test d'utilisabilité) Keywords: Reality Mixte, Interaction Homme-Machine, Computer vision		
Signature ou visa / Unterschrift oder Visum Responsable de l'orientation / filière Leiter der Vertiefungsrichtung / Studiengang: ... ¹ Etudiant / Student : ...		Délais / Termine Attribution du thème / Ausgabe des Auftrags: 10.05.2021 Présentation intermédiaire / Zwischenpräsentation 07 – 08.06.2021 Remise du rapport / Abgabe des Schlussberichts: 20.08.2021, 12:00 Exposition / Ausstellung der Diplomarbeiten: 25 – 27.08.2021 (si autorisé / falls genehmigt) Défense orale / Mündliche Verfechtung: 30.08 – 09.09.2021

¹ Par sa signature, l'étudiant-e s'engage à respecter strictement la directive DI.1.2.02.07 liée au travail de diplôme.
Durch seine Unterschrift verpflichtet sich der/die Student/in, sich an die Richtlinie DI.1.2.02.07 der Diplomarbeit zu halten.

Rapport reçu le / Schlussbericht erhalten am Visa du secrétariat / Visum des Sekretariats

Informations à propos de ce rapport

Information de contact

Auteur: Alexandre Figueiredo
Etudiant en Bachelor
HES-SO//Valais Wallis
Switzerland
Email: alexandre.ferreirafigueiredo@students.hevs.ch

Déclaration d'honneur

Je soussigné, Alexandre Ferreira Figueiredo, déclare que le travail présenté est le résultat d'un travail personnel. Je certifie que je n'ai pas eu recours au plagiat ou à d'autres formes de fraude. Toutes les sources d'informations utilisées et les citations de l'auteur ont été clairement mentionnées.

Lieu et date : Grône, le 28 août 2021

Signature :



Validation

Accepté par la HES-SO//Valais Wallis (Suisse, Sion) sur proposition de :

Dr. Francesco Carrino, Superviseur du projet

Alain Gruet, ALPS Automation SA, Expert

Lieu et date : _____

Dr. Francesco Carrino

Pierre Pompili

Superviseur

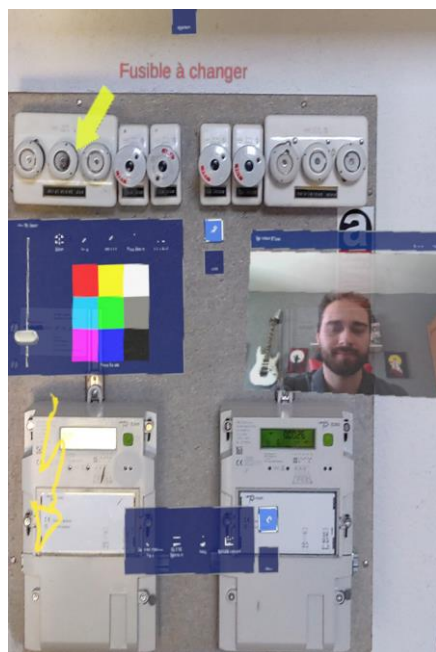
Chef de section, HEI-VS//Systèmes industriels

Remerciements

En premier lieu, je tiens sincèrement à exprimer ma gratitude envers les personnes ayant été présentes à mes côtés durant la réalisation de ce travail de Bachelor. Les conseils de ces personnes m'ont fortement aidés et encouragés, tant sur le plan professionnel que moral.

Parmi ces personnes, je remercie, tout d'abord, M. Francesco Carrino qui a accepté de me superviser et de m'apporter ses précieux conseils tout au long du projet. Il s'est également montré très disponible en étant présent à tous les meetings hebdomadaires organisés.

Je tiens également à remercier M. Alain Gruet, qui a accepté d'être l'expert de ce travail de diplôme. Il s'est montré très intéressé et dévoué au projet tout au long de sa réalisation en apportant son avis, ses idées et en organisant certains événements avec l'entreprise ALPS Automation SA tels que les tests avec les utilisateurs finaux.



Augmented reality for Remote Assistance

Diplômant/e Alexandre Figueiredo

Objectif du projet

Le but de ce projet est d'analyser les capacités des technologies de réalité augmentée (RA) existantes afin de développer un prototype permettant de réaliser des assistances à distance à l'aide de la RA pour le cas de ALPS Automation SA.

Méthodes | Expériences | Résultats

L'industrie 4.0 se voit offrir des nouvelles technologies informatiques aux entreprises. Parmi ces dernières se trouve la réalité augmentée. ALPS Automation SA, ayant bien compris les enjeux de cet ère, souhaite s'équiper de la RA afin de pouvoir effectuer des assistances à distance de haute qualité pour des mises en service ou des maintenances de systèmes.

Trois applications prototypes (PC, HoloLens 2 et smartphone) ont été réalisées en s'équipant de la norme WebRTC pour la communication audio/vidéo et d'un protocole TCP/IP pour la communication de messages entre les appareils. Ces messages (JSON) transmis permettent aux « spécialistes » de réaliser divers types d'annotations virtuelles en 3D dans l'environnement de la « personne assistée ». Equipée d'un HoloLens 2, la personne assistée a également accès à un bon nombre de fonctionnalités : création d'annotations, manipulation d'objets virtuels, instances de modèles 3D, enregistrements vidéo, etc. Parmi les annotations possible se trouvent : dessins libres, modèles 3D, textes. Il est également possible d'effacer ces annotations. Ces dernières disposent également d'un bon nombre d'options configurables par les utilisateurs.

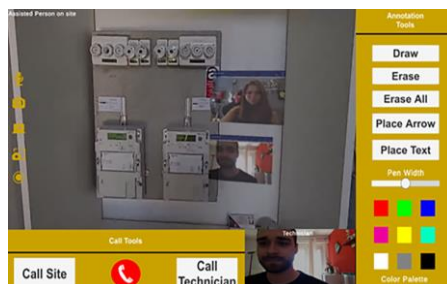
Travail de diplôme
| édition 2021 |

Filière
Systèmes Industriels

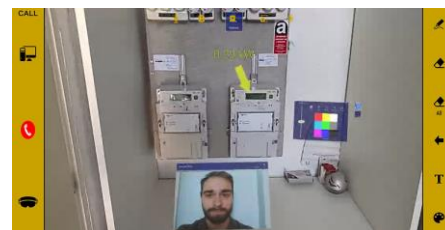
Domaine d'application
Infotronique

Professeur responsable
Dr. Carrino Francesco
francesco.carrino@hevs.ch

Partenaire
ALPS Automation SA



Rendu depuis l'application
PC



Rendu depuis l'application
Smartphone

Table des matières

Table des illustrations	14
Table des tableaux.....	18
Préambule.....	19
1. Introduction	19
1.1 Contexte et problème.....	19
1.2 Cahier des charges et objectifs	19
1.2.1 Objectif 1.....	19
1.2.2 Objectif 2.....	19
1.2.3 Objectifs secondaires	20
1.3 Structure du rapport.....	20
2. Outils de développement.....	21
2.1 Software.....	21
2.2 Hardware	21
3. Analyse.....	22
3.1 Etat de l'art de la réalité augmentée dans l'industrie	22
3.1.1 Définition de l'industrie 4.0	22
3.1.2 La réalité augmentée dans l'industrie	23
3.2 Etude des différentes technologies et services de développement.....	27
3.2.1 Prise en main de Unity	27
3.2.2 Prise en main de Unreal Engine	29
3.2.3 Prise en main de Microsoft Mesh	31
3.2.4 Comparaison entre Unity, Unreal Engine, Microsoft Mesh	35
3.2.5 Présentation et prise en main de Mixed reality toolkit (MRTK)	37
3.2.6 Dispositifs de déploiement pour la réalité augmentée	40
3.3 Analyse du cas d'utilisation ALPS Automation SA	41
3.3.1 Personnes en jeu et dispositifs utilisés	41
3.3.2 Diagramme physique	43
3.3.3 Description des besoins d'ALPS Automation SA	44
3.3.4 Comparaison entre les différents casques de réalité augmentée pour ALPS Automation SA.....	49
3.3.5 Comparaison entre les différents services existants pour ALPS Automation SA	52
3.3.6 Conclusion de l'analyse du cas d'ALPS Automation SA.....	54
4. Conception Générale.....	55
4.1 Conception de l'architecture générale.....	55
4.1.1 Layer 1 : Liens entre les utilisateurs et les appareils.....	55
4.1.2 Layer 2 – Diagramme de composants général (Spécialiste)	56
4.1.3 Layer 2 – Diagramme de composants général (Site)	57

4.1.4 Conclusion de la conception de l'architecture générale.....	58
4.2 Communication audio et vidéo	59
4.2.1 Principe de fonctionnement.....	60
4.2.2 Conclusion de la conception de la communication audio et vidéo	62
4.3 Annotations à distances.....	63
4.3.1 UDP VS TCP/IP.....	63
4.3.2 Utilisation de TCP/IP	65
4.3.3 Conclusion de la conception des annotations à distances	67
5. Implémentation générale.....	68
5.1 Projets Unity	68
5.1.1 Structure des projets.....	68
5.1.2 Fenêtre de Hiérarchie Unity	69
5.1.3 Fenêtre Scène Unity	69
5.1.4 Fenêtre « Game » Unity	70
5.1.5 Fenêtre « Inspector ».....	70
5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC	71
5.2.1 Fonctionnement exclusif de la caméra locale et du microphone local.....	71
5.2.2 Attribution de IDs spécifiques pour chaque paire de connexion	71
5.2.3 Adresse http du serveur Node.js.....	72
5.2.4 Désactivation de la caméra locale avant de créer un appel	73
5.3 Installation des paquets MRTK.....	73
6. Prototype PC	74
6.1 Conception	74
6.1.1 Définition de l'architecture du système	74
6.1.2 Définition de l'interface utilisateur	77
6.1.3 Conclusion de la conception PC	78
6.2 Implémentation	79
6.2.1 Paquets et librairies externes	79
6.2.2 Hiérarchie Unity.....	80
6.2.3 Implémentation de MRTK-WebRTC	81
6.2.4 Implémentation des scripts.....	81
6.2.5 Conclusion de l'implémentation PC	87
6.3 Déploiement de l'application	88
6.4 Validation.....	89
7. Prototype HoloLens 2	90
7.1 Conception	90
7.1.1 Définition de l'architecture du système	91
7.1.2 Définition de l'interface utilisateur	94

7.1.3 Conclusion de la conception HoloLens 2	98
7.2 Implémentation	99
7.2.1 Paquets et librairies externes	99
7.2.2 Hiérarchie Unity	100
7.2.3 Configuration des profils MixedRealityToolkit du prototype HoloLens 2	101
7.2.4 Implémentation de MRTK-WebRTC	102
7.2.5 Implémentation des scripts	102
7.2.6 Modèles 3D Préfabriqués sur Blender	113
7.2.7 Outils d'annotations Tracking	114
7.2.8 Activation et désactivation des outils d'annotations en local	115
7.2.9 Problème ProximityLight (résolu)	115
7.2.10 Conclusion de l'implémentation HoloLens 2	116
7.3 Déploiement de l'application	116
7.4 Validation	119
8. Prototype Smartphone/Tablette : Spécialiste	120
8.1 Conception	120
8.1.1 Définition de l'architecture du système	120
8.1.2 Définition de l'interface utilisateur	122
8.1.3 Conclusion de la conception smartphone spécialiste	123
8.2 Implémentation	124
8.2.1 Paquets et librairies externes	124
8.2.2 Hiérarchie Unity	125
8.2.3 Implémentation de MRTK-WebRTC	126
8.2.4 Implémentation des scripts	126
8.2.5 Conclusion de l'implémentation smartphone spécialiste	131
8.3 Déploiement de l'application	132
8.4 Validation	133
9. Prototype Smartphone/Tablette : Personne assistée	134
9.1 Conception	134
9.1.1 Définition de l'architecture du système	134
9.1.2 Conclusion de la conception smartphone personne assistée	136
10. Scénario d'ALPS Automation SA et tests d'usabilité	137
10.1 Contexte	137
10.1.1 Acteurs impliquées	137
10.2 Tests des différents prototypes	138
10.2.1 Interfaces entre les participants	138
10.2.2 Déroulement du scénario de test	138

10.3	Questionnaire de feedback pour les participants.....	139
10.3.1	SUS	139
10.4	Résultats obtenus pour le prototype de l'HoloLens 2	141
10.4.1	Points positifs appréciés	141
10.4.2	Points négatifs relevés	141
10.4.3	Quelques remarques relevées	141
10.4.4	Quelques suggestions apportées.....	141
10.4.5	Problèmes rencontrés	141
10.5	Résultats obtenus pour le prototype du PC	142
10.5.1	Points positifs appréciés	142
10.5.2	Points négatifs relevés	142
10.5.3	Quelques remarques relevées	142
10.5.4	Quelques suggestions apportées.....	142
10.5.5	Problèmes rencontrés	143
10.6	Résultats obtenus pour le prototype du smartphone	143
10.6.1	Points positifs appréciés	143
10.6.2	Points négatifs relevés.....	143
10.6.3	Quelques remarques relevées	143
10.6.4	Quelques suggestions apportées.....	143
10.6.5	Problèmes rencontrés	144
10.6.6	Conclusion	144
11.	Conclusion Générale	145
11.1	Résumé du projet.....	145
11.2	Comparaison avec les objectifs initiaux.....	146
11.3	Retour sur la planification du projet	147
11.4	Problèmes rencontrés	148
11.4.1	Erreur de connexion HoloLens 2	148
11.4.2	Mise en œuvre des appels multi-participants	149
11.4.3	Pare-feu bloquant la connexion Android	149
11.4.4	Transfert des coordonnées statiques 2D (PC, smartphone) aux coordonnées dynamiques 3D (HL2).	150
11.4.5	Faible précision des annotations smartphone.....	150
11.4.6	Microphone smartphone.....	150
11.4.6	Problème d'utilisation de JSON avec Newtonsoft.json	150
11.5	Perspectives.....	151
11.5.1	Configuration d'un serveur sécurisé pour la norme WebRTC.....	151
11.5.2	Utilisation d'un serveur Web pour les connexions multi-participants	151

11.5.3 Précision des annotations Smartphone.....	152
11.5.4 Autre méthode pour les annotations à distance.....	152
11.5.5 Création d'outils individuels propre à chaque spécialiste connectés	152
11.5.6 Adresses IP Dynamiques	152
11.5.7 Quelques suggestions supplémentaires	153
11.6 Conclusion personnelle	153
12. Bibliographie.....	154
12.1 Références	154
12.2 Liens supplémentaires utiles	156
13. Acronymes	157
15. Annexes.....	158
A. Code de l'Application Unity réalisé dans la prise en main	158
B. Tableau comparant Unity et Unreal Engine ^[13]	160
C. Plateformes et appareils pris en charge par MRTK-Unity	162
D. Tests Black-box PC	163
E. Tests Black-box Smartphone.....	165
F. Tests Black-box HL2	167
G. Configuration du « Player » pour déployer sur PC	171
G.1 Other Settings.....	171
G.2 XR Settings	172
H. Configuration du « Player » pour déployer sur smartphone.....	172
H.1 Other Settings.....	172
H.2 XR Settings	173
I. Modules de déploiements installés dans Unity	174
J. Scénario de test des prototypes	175
K. Formulaire d'autorisation pour l'utilisation de données de feedback 180	
L. Résultats des tests d'usabilité obtenus et calculés.....	183
M. Architecture du software PC Spécialiste	184
N. Architecture du software HoloLens 2 de la personne assistée	185
O. Layer 4 : Remote Assist Specialist Smartphone Software	186
P. Layer 4 : Remote Assist specialist smartphone or tablette software.....	187
Q. Planning initial	188
R. Planning Réel	189

Table des illustrations

Figure 1 - Evolution de l'industrie de 1765 à nos jours [1].....	22
Figure 2 - Résultat de l'enquête sur les facteurs limitants de la RA dans l'industrie Source : Relflections on the Limited Pervasiveness of Augmented Reality in Industrial Sectors, Alberto Martinetti, Henrique Costa Marques, Sarbjeet Singh et Leo van Dongen, Applied Science – MDPI.....	26
Figure 3 - Motivations pour le manque d'omniprésence de la RA dans les industries Source : Relflections on the Limited Pervasiveness of Augmented Reality in Industrial Sectors, Alberto Martinetti, Henrique Costa Marques, Sarbjeet Singh et Leo van Dongen, Applied Science - MDPI	27
Figure 4 - Prise en main de Unity : App Android	28
Figure 5 - Prise en main de Unreal Engine 4 : App android.....	29
Figure 6 - Level Blueprint de l'application réalisée pendant le workshop	30
Figure 7 - Création d'un espace de collaboration	32
Figure 8 - Espace des members.....	32
Figure 9 - Personnalisation d'avatar	33
Figure 10 - Importation de modèles 3D	33
Figure 11 - Palette d'annotations de Microsoft Mesh	34
Figure 12 - Exemple d'espace Mesh	34
Figure 13 - Application réalisée pour la prise en main de MRTK.....	38
Figure 14 - Communication RTC PC-PC	39
Figure 15 - Paradigme d'interaction avec la réalité augmentée Source : CARRINO, Francesco & CAON Maurizio. Virtual, Augmented & Mixed Reality. MSC BA HES-SO, 2020-2021. Chapitre Mixed Reality Interaction, p.73.	40
Figure 16 - Casque de réalité augmentée VS Smartphone et tablette	42
Figure 17 - Diagramme Physique : Exemple de setup et de mise en situation	43
Figure 18 - Description des besoins de ALPS Automation SA	44
Figure 19 - Spécifications nécessaires de l'Hardware	45
Figure 20 - Spécifications générales nécessaires du software	46
Figure 21 - Diagramme de cas d'utilisation UML du spécialiste au bureau et la personne assistée sur site.....	47
Figure 22 - RealWear HMT-1	49
Figure 23 - Vuzix M400.....	49
Figure 24 - Hololens 2	49
Figure 25 - Magic Leap One	49
Figure 26 - Comparaison des casques de réalité augmentée.....	49
Figure 27 - Comparaison générale de quatre services réputés.....	52
Figure 28 - Comparaison des fonctionnalités des quatre services	53
Figure 29 - Layer 1 : Liens entre les appareils et les utilisateurs.....	55
Figure 30 - Layer 2 : Diagramme de composants général du prototype du spécialiste	56
Figure 31 - Layer 2 : Diagramme de composants général du prototype site	57
Figure 32 - Information concernant la prise en charge de HoloLens 2 []	59
Figure 33 - Connexions WebRTC entre les différents pairs de ce projet	60
Figure 34 - Node.js server listening	61
Figure 35 - Comparaison entre TCP et UDP Sources : SAWAKINOME, TCP VS UDP, https://fr.sawakinome.com/articles/technology/tcp-vs-udp.html LIFESIZE, TCP vs. UDP : What's the difference : https://www.lifesize.com/en/blog/tcp-vs-udp/	64
Figure 36 - Etablissement d'une connexion entre un client TCP et un Serveur TCP Source : GABIOUD, Dominique. Systèmes d'informations / TCP/IP, 2019-2020.	

Connexion TCP p.37 TCPIPGUIDE.COM, TCP Connection establishment Sequence Number Synchronization and parameter Exchange. http://www.tcpiptide.com/free/t_TCPConnectionEstablishmentSequenceNumberSynchroniz-2.htm (consulté le 5 août 2021)	65
Figure 37 - Format d'un message JSON.....	66
Figure 38 - Diagramme de séquence : De la création d'un message à l'exécution d'une action à distance.....	67
Figure 39 - Structure du projet Unity	68
Figure 40 - Hiérarchie par défaut lors de la création d'un projet.	69
Figure 41 - Exemple de scène Unity	69
Figure 42 - Exemple de fenêtre de jeu Unity	70
Figure 43 - Fenêtre Inspector	70
Figure 44 - Composants Peer Connection pour se connecter aux appareils pairs (HoloLens 2 à gauche et Smartphone à droite)	71
Figure 45 - Schéma des ID locaux et distants pour les « Signalers » du prototype PC.....	71
Figure 46 - Schéma des ID locaux et distants pour les « Signalers » du prototype HL2	72
Figure 47 - Schéma des ID locaux et distants pour les « Signalers » du prototype smartphone	72
Figure 48 - Diagramme de séquence : Contrôle de la caméra source pour un appel WebRTC	73
Figure 49 - Layer 3 : Desktop PC	74
Figure 50 - Layer 4 : Architecture du Software PC Spécialiste	75
Figure 51 - Interface utilisateur PC du spécialiste	77
Figure 52 - Paquets présents dans le prototype PC	79
Figure 53 - Paquet utilisé pour travailler avec les objets Json : Newtonsoft.Json	79
Figure 54 - Ajout des références sur Visual Studio	79
Figure 55 - Emplacement du plugin.....	80
Figure 56 - Ajout du plugin Newtonsoft.json dans les assets du projet Unity	80
Figure 57 - PC : Hiérarchie des objets Unity	80
Figure 58 - Enfants de HoloLens 2 et Smartphone	81
Figure 59 - Répartition des scripts dans le projet Unity PC.....	81
Figure 60 - Diagramme de transition : DetectMouseOverVideo.cs.....	82
Figure 61 - Méthode CheckInput() du script Typer.cs (PC)	83
Figure 62 - Relations entre les scripts (classes) du dossier TCPClient (PC)	83
Figure 63 - Connection du client TCP (PC) à son serveur TCP	84
Figure 64 - Méthode SendMessage() pour envoyer les messages JSON au serveur	84
Figure 65 - Principe de création de messages JSON lorsqu'un bouton de l'UI spécifique est pressé.....	85
Figure 66 - Algorithme dans la méthode Update de ManageRemoteCommands.cs	86
Figure 67 - Build Settings PC	88
Figure 68 - Exécutable généré par la compilation	88
Figure 69 - Layer 3 : Interfaces HoloLens 2 VS RealWear	90
Figure 70 - Layer 4 : Architecture du software HoloLens 2 de la personne assistée	91
Figure 71 - Layer 4 : Interfaces fournies nativement par l'HoloLens 2	93
Figure 72 - UI Natif : Accès au menu principal.....	94
Figure 73 - UI Natif : Menu Principal.....	94

Figure 74 - Exemple de l'utilisation de OneDrive et de la superposition d'éléments	95
Figure 75 - UI Prototype : Menu de la main	95
Figure 76 - UI Prototype : Menu d'annotation	96
Figure 77 - Outils d'annotations	97
Figure 78 - UI Prototype : Menu d'appel	97
Figure 79 - Paquets présents dans le prototype Smartphone	99
Figure 80 - HoloLens 2 : Hiérarchie des objets Unity	100
Figure 81 - Clone du profil HL2 par défaut	101
Figure 82 - Clone du profil Spatial Awareness ainsi que ces propriétés	101
Figure 83 - Enfants de PC et Smartphone	102
Figure 84 - Répartition des scripts dans le projet Unity PC	102
Figure 85 - Diagrammes de classe du dossier LocalHL2Functions	103
Figure 86 - Informer le système en cas d'utilisation des interfaces IMixedRealityInput	103
Figure 87 - Algorithme pour créer des lignes de dessin. Méthode Update() dans LineDrawer.cs	104
Figure 88 - Algorithme pour effacer une annotation. ElementsEraser.cs	105
Figure 89 - Algorithme pour effacer toutes les annotations instanciées. Methode EraseAll() de EraseAllElements.cs	106
Figure 90 - Algorithme pour instancier des flèches locales. Méthode Update() de PlaceArrows.cs	106
Figure 91 - Prefab de texte à instancier	107
Figure 92 - Changement de couleur d'un matériau. MaterialManager.cs	107
Figure 93 - Relation entre les classes des scripts situés dans RemoteCommands	108
Figure 94 - Principes du script TCPServer.cs	109
Figure 95 - Algorithme de la méthode DispatchRemoteCommands() de EventDispatcherJson.cs	110
Figure 96 - Algorithme lorsqu'un bouton distant a été pressé. EventDispatcherJson.cs	111
Figure 97 - Algorithme lorsqu'un bouton distant a été relâché. EventDispatcherJson.cs	111
Figure 98 - Algorithme de la méthode Update de EventDispatcherJson.cs	112
Figure 99 - Modèles 3D créés sur Blender	113
Figure 100 - Composants permettant de faire du tracking d'objet	114
Figure 101 - Articulations de la main Source : ResearchGate.net https://www.researchgate.net/figure/Joints-of-the-right-hand-dorsal-view-Note-that-the-terms-trapeziometacarpal-TM-joint_fig1_257045252	114
Figure 102 - Algorithme d'activation et désactivations d'outils d'annotations	115
Figure 103 - Build Settings HoloLens 2	116
Figure 104 - Fonctionnalités à activer pour un bon fonctionnement de l'application	117
Figure 105 - XR Settings HoloLens 2	117
Figure 106 - Génération du fichier .sln	117
Figure 107 - Configuration du déploiement	118
Figure 108 - Configuration de l'adresse IP de l'HoloLens 2 sur Visual Studio	118
Figure 109 - Application déployée sur l'HoloLens 2 avec certaines menus et annotations présents	118
Figure 110 - Layer 3 : Smartphone/Tablette Spécialiste	120
Figure 111 - Layer 4 : Remote Assist Specialist Smartphone Software	121

Figure 112 - Interface utilisateur de l'application spécialiste déployé sur smartphone.....	122
Figure 113 - Paquets importés dans Unity pour le prototype Smartphone (Spécialiste).....	124
Figure 114 - Smartphone : Hiérarchie des objets Unity.....	125
Figure 115 - MixedRealityToolkit Component.....	125
Figure 116 - Enfants de HoloLens 2 et PC	126
Figure 117 - Répartition des scripts dans le projet Unity Smartphone	126
Figure 118 - Tyler.cs Smartphone : OpenKeyboard().....	127
Figure 119 - Tyler.cs Smartphone : Update()	127
Figure 120 - Smartphone : Diagramme de classes du dossier Remote Scripts..	128
Figure 121 - ManageRemoteCommands.cs Smartphone : Update()	129
Figure 122 - ManageRemoteCommands Smartphone : in Update(). How to move in Z axis the remote tool.....	130
Figure 123 - Build Settings Android.....	132
Figure 124 - Application déployée sur Android avec certaines fenêtres et annotations présentes	132
Figure 125 - Layer 3 : Smartphone/Tablette Assisted person.....	134
Figure 126 - Layer 4 : Remote Assist specialist smartphone or tablette software	135
Figure 127 - Smartphone Native Interfaces	136
Figure 128 - Armoire électrique du scénario	137
Figure 129 - Réponses SUS	139
Figure 130 - Echelle de satisfaction	140
Figure 131 - Score final SUS des trois prototypes.....	144
Figure 132 - Erreur de connexion de l'HoloLens 2 au serveur Node.js	148
Figure 133 - Cause de l'erreur de la connexion HL2 et du serveur Node.js source : MICROSOFT DOCS, Initialisation de votre projet et déploiement de votre première application https://docs.microsoft.com/fr-fr/windows/mixed-reality/develop/unity/tutorials/mr-learning-base-02?tabs=openxr (consulté le vendredi 13 août 2021).....	148
Figure 134 - Création d'une nouvelle règle dans le pare-feu pour autoriser la connexion TCP provenant du smartphone.	149
Figure 135 - Lignes de code permettant de résoudre le problème de fonctionnement de JSON. (Source : https://stackoverflow.com/questions/16359628/json-net-under-unity3d-for-ios)	150
Figure 136 - Plateformes et appareils pris en charge par MRTK-Unity Source : https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05	162

Table des tableaux

Tableau 1 - Matériel à disposition depuis le bureau	41
Tableau 2 - Matériel à disposition depuis le site.....	41
Tableau 3 - Liste des commandes créés par le client TCP.....	87
Tableau 4 - Interfaces entre les participants du scénario	138
Tableau 5 - Tableau 3 - Test Call0_PC.....	163
Tableau 6 - Test Call1_PC	163
Tableau 7 - Test Ann1_PC	163
Tableau 8 - Test Ann2_PC	163
Tableau 9 - Test Ann3_PC	164
Tableau 10 - Test Ann4_PC	164
Tableau 11 - Test Ann5_PC	164
Tableau 12 - Test Ann6_PC	164
Tableau 13 - Tableau 3 - Test Call0_Smartphone	165
Tableau 14 - Test Call1_PC	165
Tableau 15 - Test Ann1_Smartphone	165
Tableau 16 - Test Ann2_Smartphone	166
Tableau 17 - Test Ann3_Smartphone	166
Tableau 18 - Test Ann5_Smartphone	166
Tableau 19 - Test Ann6_PC	167
Tableau 20 - Test Call0_HL2.....	167
Tableau 21 - Test Call1_HL2.....	167
Tableau 22 - Test Ann1_HL2	167
Tableau 23 - Test Ann2_HL2	168
Tableau 24 - Test Ann3_HL2	168
Tableau 25 - Test Ann4_HL2	168
Tableau 26 - Test Ann5_HL2	169
Tableau 27 - Test Ann6_HL2	169
Tableau 28 - Test UI1_HL2	169
Tableau 29 - Test UI2_HL2	169
Tableau 30 - Test UI3_HL2	170
Tableau 31 - Test UI4_HL2	170
Tableau 32 - Test UI5_HL2	170

Préambule

Afin d'avoir une meilleure lecture et lisibilité de ce rapport, certaines figures et certains tableaux ont été mis en annexe.

1. Introduction

1.1 Contexte et problème

A l'aide de la réalité augmentée (RA), l'entreprise ALPS Automation SA souhaite déployer de nombreuses technologies, systèmes, machines et processus d'automatisation à distance. La réalité augmentée, qui est la superposition de la réalité et d'éléments virtuels (images, sons, vidéos, etc.), permet ainsi de limiter le nombre de personnes requises sur le site d'opérations lors des différentes mises en services et maintenances. Cette technologie permet donc de réduire considérablement les coûts de ces différentes opérations. De plus, elle permet de :

- Former les opérateurs avant que le système réel ne soit mis en service.
- Former les nouveaux opérateurs aux séquences de démarrage, de transition et d'arrêt.

De ce fait, dans le cadre de la Haute Ecole d'Ingénierie du Valais, ce travail de diplôme est réalisé dans le but de concevoir et développer, pour ALPS Automation SA, un concept s'alliant de la réalité augmentée afin de satisfaire leurs besoins pour effectuer de l'assistance à distance.

1.2 Cahier des charges et objectifs

1.2.1 Objectif 1

Le premier objectif de ce travail de diplôme est d'analyser et comparer les capacités et les limites des dispositifs de réalité augmentée actuels et des services existants dans le but de trouver un concept répondant aux besoins de ALPS Automation SA.

1.2.2 Objectif 2

Le second objectif est de concevoir, développer et tester un prototype pour le cas d'utilisation proposé par ALPS Automation SA et répondant à son cahier des charges.

L'analyse du cas d'utilisation de ALPS Automation SA permettra de connaître leurs critères et leurs exigences. De ce fait, elle se trouvera dans le chapitre « 3.3 Analyse du cas d'utilisation ALPS Automation SA ».

1.2.3 Objectifs secondaires

Si la durée du travail de bachelor le permet et que les objectifs principaux sont atteints, certains objectifs secondaires pourraient être réalisés. Ces derniers consisteraient à :

- Effectuer des tests d'utilisations avec les utilisateurs finaux du prototype.
- Développer des mini-démonstrateurs avec Unity, Unreal Engine, Microsoft Mesh.

1.3 Structure du rapport

Ce rapport peut être commodément divisé en six parties.

Cette première partie présente le projet et son contexte ainsi que les différents objectifs. Elle présente également les outils nécessaires au bon développement de ce travail de diplôme.

La deuxième partie détaillera l'analyse de la réalité augmentée dans l'industrie ainsi que les technologies existantes qui lui sont liées et analysera également le cahier des charges de ALPS Automation SA.

Une troisième partie traitera des points conception et d'implémentation générales des divers prototypes réalisés dans ce projet.

Dans la quatrième partie, la réalisation (conception, implémentation, déploiement et validation) de chaque prototype sera individuellement présentée. Dans les chapitres de cette partie, il se peut que certaines informations soient répétées de la sorte à disposer de certaines informations courtes sans avoir à feuilleter le rapport pour les retrouver. En revanche, d'autres informations communes plus longues seront référencées à d'autres chapitres pour alléger le rapport.

Une cinquième partie vient compléter ce rapport en présentant le déroulement des différents tests utilisateurs réalisés avec ALPS Automation SA avec les discussions les accompagnant.

Enfin, dans la dernière partie, une conclusion relèvera les points importants, traitera de l'état des objectifs, soulèvera quelques problèmes rencontrés et soulèvera également des éléments de perspective de projet. De plus, les annexes ainsi que les éléments bibliographiques se trouveront à la fin du rapport.

2. Outils de développement

Ce chapitre présente les différents outils de développement ayant été nécessaires au bon déroulement du projet. Les technologies testées et prise en mains seront également listées. Le chapitre de l'analyse expliquera les choix faits par rapport à ces technologies et décrira également dans quelle mesure elles ont été utilisées.

2.1 Software

- Unity 2019.4.26f1
- Visual Studio 2019 (VS2019)
- Unreal Engine 4 (UE4)
- Microsoft Mesh (preview)
- MRTK SDK for Unity
- MRTK-WebRTC for Unity
- AR Foundation SDK for Unity
- Microsoft .NET Framework 4.7.1 SDK
- Microsoft .NET Runtime – 5.0.6 (x64)
- Microsoft Visual C++
- Node.js
- Windows Software Development Kit 10.0.19041.685
- Windows 10

2.2 Hardware

- 2 PC
- HoloLens 2 (HL2)
- Smartphone Android
- RealWear HMT-1

3. Analyse

Ce chapitre présente les différentes analyses réalisées durant ce travail de diplôme. Une première partie présentera les résultats des recherches de la réalité augmentée dans l'industrie ainsi que les différentes technologies et services déjà existants. Une deuxième partie détaillera les résultats de l'analyse du cas de ALPS Automation SA découlant des diverses recherches et discussions avec eux. Enfin, une conclusion relèvera les points importants de ce chapitre.

3.1 Etat de l'art de la réalité augmentée dans l'industrie

3.1.1 Définition de l'industrie 4.0

« Animée par l'émergence de nouvelles technologies, l'industrie 4.0 désigne une nouvelle génération d'usines connectées, robotisées et intelligentes. Avec la révolution numérique, les frontières entre le monde physique et digital s'amenuisent pour donner vie à une usine 4.0 interconnectée dans laquelle les collaborateurs, les machines et les produits interagissent. L'industrie 4.0 est un défi et une véritable opportunité.^[1]»

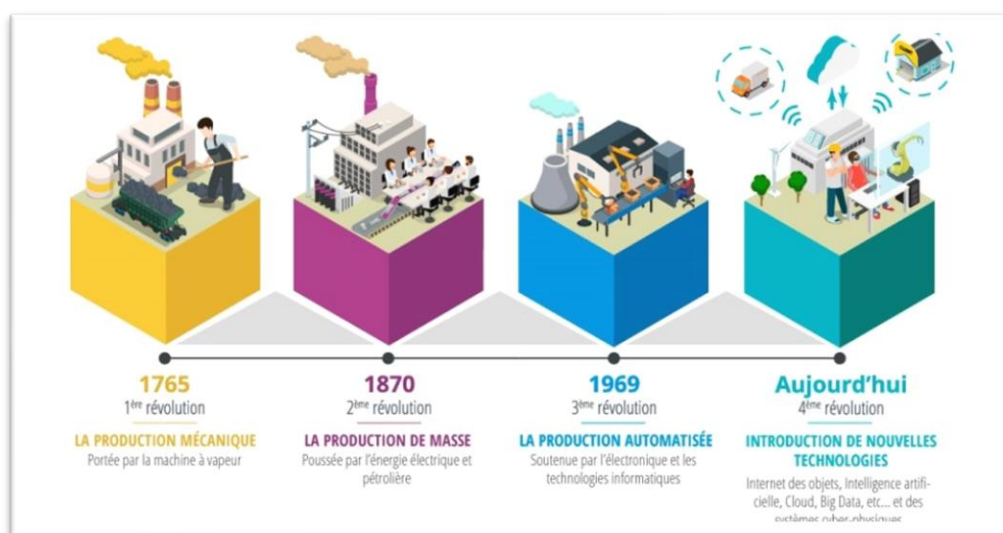


Figure 1 - Evolution de l'industrie de 1765 à nos jours [1]

Cette définition de l'industrie 4.0 situe bien les enjeux de la technologie dans l'industrie et permet de comprendre la raison d'être de ce projet ainsi que la raison pour laquelle il s'intéresse à la réalité augmentée dans le secteur industriel et plus particulièrement pour le cas de l'entreprise ALPS Automation SA.

¹ VISIATIV SOLUTIONS, *Industrie 4.0 : définition et mise en œuvre vers l'usine connectée*
<https://www.visiativ-solutions.fr/industrie-4-0> (Consulté le 17 mai 2021)

3.1.2 La réalité augmentée dans l'industrie

Bien que la RA ait vu le jour en 1968 grâce à Ivan Sutherland^[2], celle-ci n'est que, depuis peu, un souffle nouveau pour le secteur industriel. En effet, grâce aux améliorations que cette technologie a reçu au fil des années, elle est maintenant un grand atout pour les entreprises et offre de nombreux avantages et bénéfices.

Quelques chiffres

Pour situer la réalité augmentée dans le monde d'aujourd'hui, financesonline.com^[3] a récolté les informations suivantes :

- Le taille du marché de la RA est actuellement de 12 milliards de dollars. (Statista, 2021)
- D'ici 2023, le nombre d'utilisateurs de RA est estimé à 2.4 milliards. (AR Insider, 2021)
- 70% des professionnels disent que la RA a le potentiel de les aider à affiner leurs compétences personnelles et professionnelles. (ISACA, 2016)
- 65% des entreprises de RA développent cette technologie pour des cas industriels (Forbes, 2020)

Les avantages majeurs de la réalité augmentée dans l'industrie

Devenue l'un des piliers de la digitalisation, la RA apporte de nombreux bénéfices. Après plusieurs recherches, les éléments principaux suivants ont pu être listés :

- **Mettre en synergie le terrain et le numérique^[4] :**

Analyse rapide : La réalité augmentée va aider l'opérateur à détecter qu'un modèle initial conçu par ordinateur est différent du modèle réel fabriqué car les modèles virtuels et modèles physiques peuvent être superposés.

Rapports précis : La RA permet de générer des rapports précis interfacés avec les différents systèmes d'informations de l'entreprise.

Fluidité de l'information : Ces rapports font remonter l'information du terrain vers les bureaux d'études, du contrôle qualité, etc.

Exploitation des données optimisées : La réalité augmentée permet d'exploiter les données numériques sur le terrain et les données du terrain (capture du réel visualisé) dans les systèmes numériques.

² WIKIPEDIA (EN), *Ivan Sutherland*

https://en.wikipedia.org/wiki/Ivan_Sutherland (Consulté le 17 mai 2021)

³ FINANCES ONLINE, *Technology statistics, Augmented Reality*

<https://financesonline.com/technology-statistics> (Consulté le 17 mai 2021)

⁴ VISIATIV SOLUTIONS, *6 bénéfices de la réalité augmentée dans l'industrie*

<https://www.visiativ-solutions.ch/realite-augmentee-dans-l-industrie> (Consulté le 17 mai 2021)

- **Améliorer l'apprentissage^[5]** : La RA permet de créer des simulations très difficiles à mettre en place permettant aux opérateurs de se former et d'entreprendre les bonnes actions le jour où la situation en question leur fera face.
- **Gagner en productivité^[5]** : En permettant d'optimiser les temps de manipulation et en permettant de visualiser des informations numériques superposées au monde physique en temps réel, la RA permet un gain en productivité.
- **Renforcer la sécurité^[5]** : Certains sites d'interventions pouvant être dangereux pour un opérateur, les informations visuelles ou sonores fournies par la réalité augmentée permettent aux opérateurs de rester vigilants et limitent ainsi leurs risques d'accidents. La RA permet d'identifier les éléments dangereux à portée de l'opérateur.
- **Améliorer la qualité, réduire le temps d'arrêt de production et minimiser les erreurs^[5]** : La réalité augmentée permet de contrôler rapidement la conformité d'un montage. Des problèmes pourraient être détectés lors du contrôle et pourraient être ainsi immédiatement corrigés.
- **Accéder aux données en temps réel^[6]** : Grâce à la réalité augmentée, les opérateurs peuvent avoir sur leur champ visuel de la documentation ainsi que des données importantes en temps réel leur permettant de continuer leur travail sans devoir interrompre le flux de production en allant chercher ces informations ailleurs.
- **Faire du marketing industriel^[6]** : La réalité augmentée permet de faire une visite d'entreprise à distance, de faire des meetings également à distance et de montrer des produits sous forme virtuelle comme si les clients étaient à côté de ce dernier.

Il existe bien d'autres bénéfices de la réalité augmentée dans l'industrie n'ayant pas été listés ci-dessus, comme le déploiement d'une chaîne de production virtuelle dans une pièce vide, avant même qu'elle n'existe. ALPS Automation SA a bien saisi ces différents concepts, ce qui leur permettra de prendre une longueur d'avance sur la concurrence en faisant des mises en services et maintenances à distance afin de diminuer les coûts et améliorer la qualité globale des diverses productions.

⁵ VISIATIV SOLUTIONS, *6 bénéfices de la réalité augmentée dans l'industrie*
<https://www.visiativ-solutions.ch/realite-augmentee-dans-l-industrie> (Consulté le 17 mai 2021)

⁶ AUDROS, *La réalité augmentée industrielle au service de l'industrie 4.0*
<https://www.audros.fr/realite-augmentee-industrie> (Consulté le 17 mai 2021)

Assistance à distance grâce à la réalité augmentée

Le site FinancesOnline.com dit ceci : « Les agents des centres d'appels reçoivent jusqu'à 50 appels par jour (HubSpot, 2019). Traditionnellement, lorsque les clients appellent l'assistance technique, ils font de leur mieux pour décrire le problème, tandis que l'agent propose des étapes que le client doit suivre. Cependant, le plus souvent, cela ne résout pas les problèmes et ne fait que créer de la frustration.^[7] »

En superposant le monde physique à des éléments virtuels, l'AR et ses bénéfices énumérés précédemment, changeront les Call Center afin de résoudre le problème cité plus haut et créeront ainsi de la satisfaction au lieu de la frustration.

Facteurs limitants de la réalité augmentée dans l'industrie

En 2019, une étude sur les facteurs limitants de la RA dans l'industrie a été publiée dans le journal « Applied Science » de MDPI. Dans cette dernière, ses auteurs ont intégré un système de réalité augmentée dans le domaine de l'industrie et ont interviewé les entreprises et employés afin de connaître leur point de vue sur la réalité augmentée au travail.

Leur enquête a été basée sur une approche semi-quantitative dans trois secteurs industriels (industrie automobile, industrie ferroviaire et industrie manufacturière). Dans ce rapport, seuls les résultats de l'industrie manufacturière seront présentés car c'est celle qui se rapproche le plus du partenaire de projet ALPS Automation SA. L'application de réalité augmentée, réalisée dans le secteur « Manufacturing », consistait à assister et guider numériquement les utilisateurs dans les opérations d'assemblage et de désassemblage d'un composant d'une machine d'emballage à grande vitesse. Elle a été testée sur 24 employés dans ce secteur. Le dispositif de déploiement utilisé était un Microsoft Hololens, qui est également un très grand candidat pour la réalisation de ce travail de Bachelor.

Leur recherche est réalisée à l'aide des quatre étapes ci-dessous :

1. Session de Brainstorming : Les trois groupes réfléchissent sur quels sont les facteurs limitant la réalité augmentée dans les industries.
2. Collecte des facteurs : Les interviewers collectent tous les facteurs identifiés dans les trois groupes.
3. Regroupement des facteurs : Les interviewers regroupent toutes les entrées en 11 principaux facteurs.
4. Classement des facteurs : Les facteurs ont été proposés aux trois groupes dans le but de les classer sur une échelle de Likert 0-5 où la valeur 0 est la valeur la moins critique.

⁷ FINANCESONLINE, 20 Current Augmented Reality Trends & Prediction for 2021/2022 and Beyond, 11. AR Remote Assistance

<https://financesonline.com/augmented-reality-trends> (Consulté le 17 mai 2021)

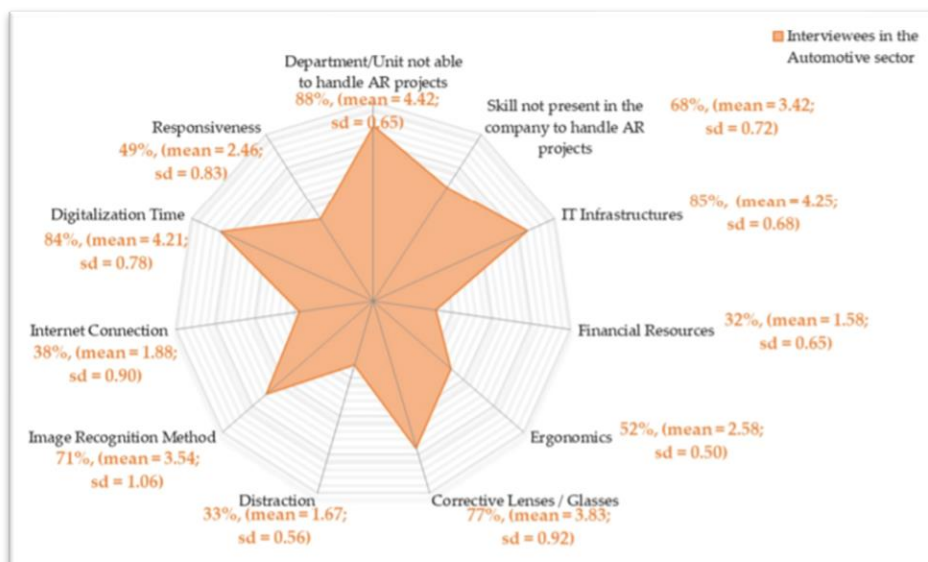


Figure 2 - Résultat de l'enquête sur les facteurs limitants de la RA dans l'industrie
Source : Reflections on the Limited Pervasiveness of Augmented Reality in Industrial Sectors, Alberto Martinetti, Henrique Costa Marques, Sarbjeet Singh et Leo van Dongen, Applied Science – MDPI

La figure ci-dessus montre les 11 facteurs limitants ressortis de l'enquête avec leur pourcentage, la valeur moyenne et l'écart-type (sd). Ces derniers sont :

- Le département/unité n'est pas prêt pour gérer les projets de RA
- Compétences non présentes dans l'entreprise pour gérer les projets de RA
- Infrastructures informatiques
- Ressources financières
- Ergonomie
- Lentilles correctrices/lunettes
- Distraction
- Méthode de reconnaissance d'image
- Connexion Internet
- Temps de numérisation
- Réactivité

Enfin, dans le but de mieux comprendre ces différentes limitations, les 11 premiers facteurs ont été regroupés en trois catégories finales étant considérées comme limitations globales à l'omniprésence de la réalité augmentée dans l'industrie. Comme le montre la figure ci-dessous, il s'agit de limites organisationnelles (40%), de limites liées à l'interaction humaine (24%) et de limites technologiques et techniques (36%). Ces trois catégories empêchent la réalité augmentée de s'épanouir dans l'industrie mais diminueront progressivement avec le temps.

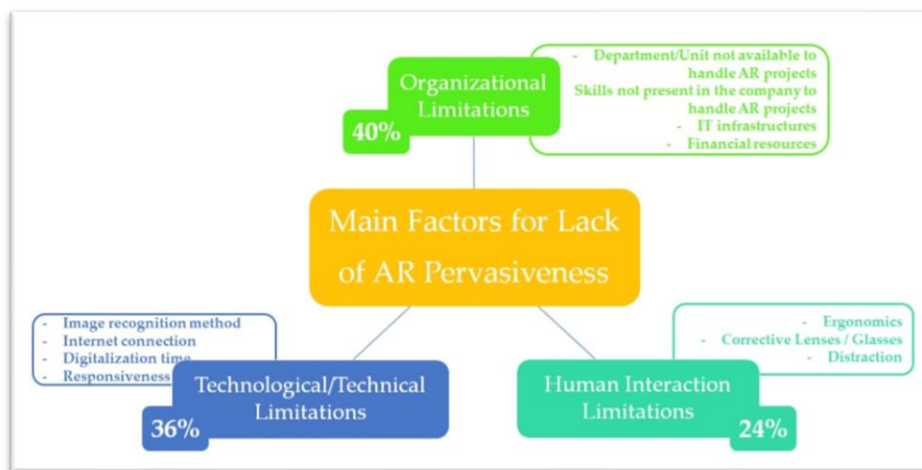


Figure 3 - Motivations pour le manque d'omniprésence de la RA dans les industries
Source : *Reflections on the Limited Pervasiveness of Augmented Reality in Industrial Sectors*, Alberto Martinetti, Henrique Costa Marques, Sarbjeet Singh et Leo van Dongen, *Applied Science - MDPI*

3.2 Etude des différentes technologies et services de développement

N'ayant aucune expérience dans le domaine de la réalité augmentée et avec les technologies qui lui sont liées, une étude de ces dernières a été nécessaire afin de prendre en main ce projet. Il existe des « moteurs » comme Unity ou Unreal Engine (UE) permettant de réaliser des applications de RA. Ensuite, il existe également divers services comme TeamViewer Assist AR, Vuforia Chalk ou Microsoft Mesh permettant déjà d'effectuer des assistances à distance. Cependant, il se peut qu'ils ne correspondent pas aux exigences de ALPS Automation SA. Ces derniers peuvent être déployés sur des smartphones, sur des tablettes ou encore sur des casques de réalité augmentée, lesquels seront également le sujet d'une analyse. Cette section traitera de la prise en main de ces technologies et montrera le résultat des recherches et des comparaisons faites sur ces dernières.

3.2.1 Prise en main de Unity

Ce moteur de jeu, développé par Unity Technologies, est très pratique pour le développement d'applications de RA. Ces derniers disent : « Unity fournit de puissants outils pour créer des expériences riches et attrayantes en réalité augmentée, qui interagissent intelligemment avec le monde réel.^[8]»

Unity a l'avantage d'être gratuit pour un usage personnel ou pour les étudiants, ce qui est très pratique à la réalisation de ce projet. Cependant, il coûterait 1833,33€ par mois et pour 10 postes aux entreprises ayant des revenus supérieurs à 200'000 \$.^[9]

⁸ UNITY, *Réalité augmentée*
<https://unity.com/fr/unity/features/ar> (Consulté le 18 mai 2021)

⁹ UNITY, *Unity Store*
<https://store.unity.com/fr/compare-plans> (Consulté le 18 mai 2021)

En intégrant AR Foundation (Framework conçu pour le développement en réalité augmentée et pour le déploiement sur divers systèmes d'exploitation (OS), appareils mobiles et dispositifs de RA) à Unity, une première application simple de réalité augmentée a été réalisée sur Android dans le but d'apprendre et connaître cette plateforme.

Cette application consistait à utiliser la caméra de l'appareil pour filmer le monde réel et détecter les divers plans (horizontaux et verticaux) sur le champ de vision du dispositif. Une fois ces derniers détectés, il était possible de cliquer sur une zone de l'écran (possédant un de ces plans) pour placer différents cubes 3D dans l'espace. Ensuite, en utilisant deux doigts, il était possible déplacer un des cubes, de modifier la taille des cubes ou encore de leur faire subir des rotations. Ci-dessous se trouve une photo de l'application.

En annexe se trouve le code ayant permis la réalisation de cette application. Celle-ci, nommée « Placing and Manipulating Objects in AR^[10] » a été réalisée en suivant un tutoriel de Unity Learn. Le lien vers ce dernier peut être trouvé dans les références pour reproduire cette première prise en main.



Figure 4 - Prise en main de Unity : App Android

¹⁰ Unity Learn, *Placing and Manipulating Objects in AR*
<https://learn.unity.com/tutorial/placing-and-manipulating-objects-in-ar#> (Consulté le 4 juin 2021)

Unity a l'avantage d'avoir beaucoup de collaborations lui fournissant plusieurs Software Development Kit (SDK) (p. ex. : Manomotion SDK) très utiles à la réalisation d'une application de RA.

3.2.2 Prise en main de Unreal Engine

Unreal Engine (UE), développé par Epic Games, est également un moteur de jeu très populaire permettant de réaliser des applications en réalité augmentée.

Il dispose de licences gratuites pour les créateurs et personnes souhaitant publier leurs applications. Toutefois, ils disposent d'une version personnalisable, dont le prix est à définir dépendant des termes de négociation, et d'une versions entreprise pour le prix de 1500\$ par poste et par année^[11].

A l'instar de Unity, Unreal Engine dispose de Google ARCore et ARKit (inclus dans AR Foundation pour Unity). Pour prendre en main ce moteur de jeu avec la réalité augmentée, un petit workshop de « Circuit Stream » a été suivi afin de réaliser une application ayant la capacité de tracker des images en réalité augmentée. Il est possible de retrouver le lien du workshop dans les sources^[12].

Lorsque la caméra du smartphone détecte un logo HES-SO comme le montre la figure ci-dessous, un robot bleu apparaît alors dans l'espace réel et effectue « la danse du robot ».

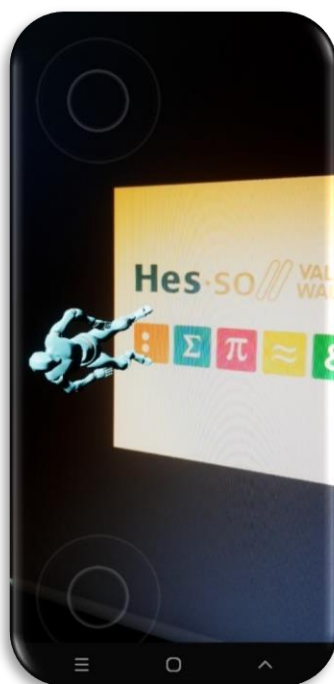


Figure 5 - Prise en main de Unreal Engine 4 : App android

¹¹ Unreal Engine, *Unreal Products*
<https://www.unrealengine.com/en-US/custom-license> (Consulté le 21 mai 2021)

¹² Circuit Stream, *Introduction to AR Development with Unreal Engine 4 — Build an Image Tracker App under 90 Minutes!*
<https://www.youtube.com/watch?v=yQKJkGenTnA> (Consulté le 20 mai 2021)

Ce workshop a permis de découvrir le système de « blueprints » de Unreal Engine. Les bluesprints, représentés par des nœuds et ayant plusieurs fonctionnalités, permettent de réaliser des scripts de jeu sans écrire de code. Ci-dessous se trouve le script de blueprints réalisé pendant ce workshop.

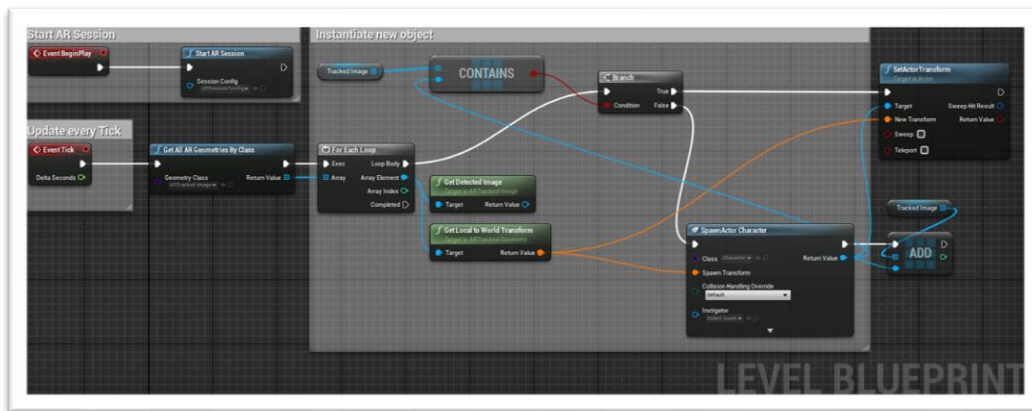


Figure 6 - Level Blueprint de l'application réalisée pendant le workshop

Au lancement de l'application, elle débute au nœud « Event BeginPlay ». Ce nœud est semblable à la méthode « start() » dans les scripts Unity. C'est à ce moment-là qu'une session de réalité augmentée est commencée. Celle-ci permet à l'application de demander et de lui donner l'accès de la caméra du smartphone.

Ensuite, un nœud Event Tick (semblable à la méthode « update() » dans Unity) est appelé à chaque frame permettant ainsi de rafraîchir l'application. Ici, la première chose à faire est de détecter toutes les images « trackable » dans le monde physique à l'aide du nœud « Get All AR Geometries By Class » et de toutes les manipuler grâce au nœud « For Each Loop ».

Le nœud « Get Local to World Transform » permet ensuite de prendre la position de l'image dans le monde physique et de la retransmettre dans l'application afin de faire apparaître un personnage à cette même position. Dès lors qu'il est apparu, il est ajouté dans un tableau nommé « Tracked Image », à l'aide du nœud « SpawnActor Character ». En contrôlant le contenu de ce tableau, il est possible d'ajouter une condition, grâce aux nœuds « Contains » et « Branch », servant à n'instancier qu'un personnage à la fois. En effet, si un personnage est déjà existant, on lui réassigne sa position grâce à « Set Actor Transform », sinon on le crée pour la première fois.

3.2.3 Prise en main de Microsoft Mesh

Microsoft Mesh est une plateforme développée par Microsoft permettant à des personnes de communiquer et collaborer dans un espace de réalité mixte commun à travers différents appareils à n'importe quel emplacement. Elle permet d'obtenir des informations numériques pertinentes à l'aide de la réalité augmentée et mixte et d'accélérer la prise de décisions et de résolutions de problèmes.

A l'aide de « l'holoportation » il est possible de se projeter, aussi réel que possible, dans l'espace des interlocuteurs. Une deuxième façon de se projeter, moins réaliste, est faite à l'aide d'avatars virtuels personnalisables.

La plateforme peut être utilisée à des fins d'apprentissages, d'assistance à distance ou de cocréations de projets.

Etant actuellement en phase de « preview », Microsoft Mesh est gratuite pour le moment. Cependant, elle n'est disponible, actuellement, que sur HoloLens et les fonctionnalités ne sont probablement pas toutes implémentées.

Selon Microsoft, cette plateforme sera payante lorsqu'elle sera officiellement sortie. Cependant, ils ne donnent actuellement aucun détail concernant ses futurs prix.

Ayant un HoloLens 2 à disposition, l'application a pu être testée dans son état de « preview ».

Voici les fonctionnalités actuelles de Microsoft Mesh :

- Rejoindre des sessions collaboration à distance
- Personnaliser un avatar expressif
- Importer du contenu 3D depuis OneDrive
- Sauvegarder des espaces pour la collaboration synchrone et asynchrone
- Dessiner et annoter du contenu

Ci-dessous se trouve un aperçu de l'état actuel des fonctionnalités de Microsoft Mesh.

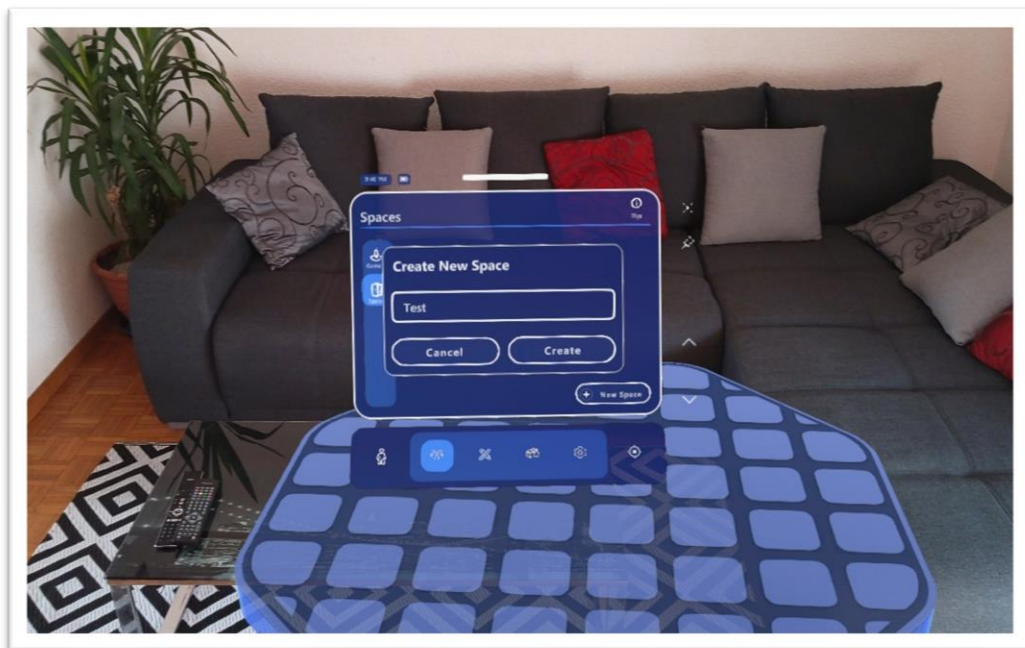


Figure 7 - Création d'un espace de collaboration

Comme le montre la figure ci-dessus, un espace de collaboration peut être créé en lui donnant un certain nom. Dans cet espace il est alors possible d'inviter des personnes pour collaborer.



Figure 8 - Espace des membres

Il est possible d'ajouter des membres ou de les retirer afin de modérer l'espace.

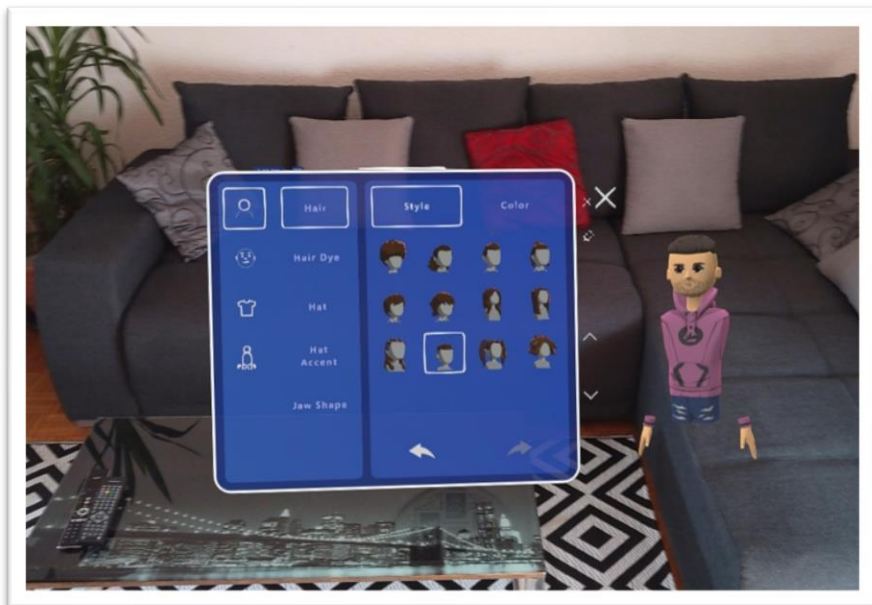


Figure 9 - Personnalisation d'avatar

Il semblerait que pour la phase de « Preview » l'holoportation n'est toujours pas possible. Cependant, il est possible de personnaliser son avatar afin de le matérialiser dans l'espace des interlocuteurs et d'avoir une sensation de « présence » lors des collaborations.

La personnalisation permet de modifier le style et la couleur du visage, des habits, d'éventuels chapeaux et d'autres détails.



Figure 10 - Importation de modèles 3D

La figure ci-dessus montre l'importation d'un objet 3D. La voiture rouge de l'image fait partie de la bibliothèque Mesh, mais il est tout à fait possible d'importer ses propres modèles 3D à l'aide de OneDrive.

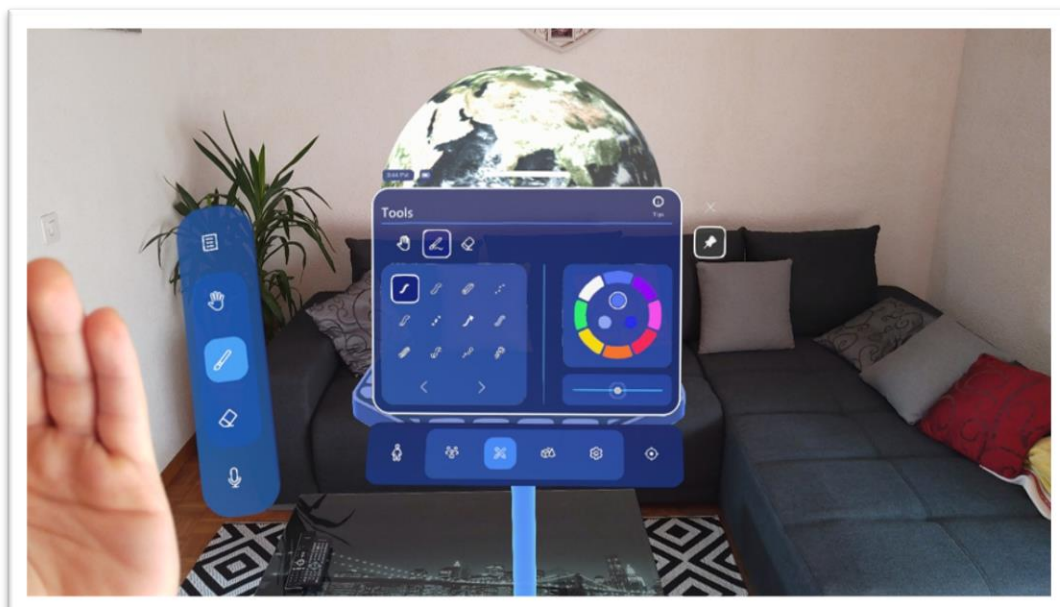


Figure 11 - Palette d'annotations de Microsoft Mesh

Comme le montre la figure ci-dessus, la plateforme dispose de plusieurs outils pour créer des annotations (Traits pleins, traitillés, gomme, changement de taille, etc.). L'outil « main » permet de manipuler les objets à sa guise.



Figure 12 - Exemple d'espace Mesh

Ci-dessus se trouve un exemple d'espace qui peut être sauvegardé afin que des collaborateurs puissent regarder son contenu en tout temps même lorsque la personne qui l'a modifié est absente. L'exemple présente une Lune et une Terre matérialisées dans la réalité avec des annotations les indiquant.

3.2.4 Comparaison entre Unity, Unreal Engine, Microsoft Mesh

Unity vs Unreal Engine

Comparer les technologies entre elles n'a rien d'anodin lorsqu'il faut faire un choix d'outil de développement. Etant tous deux des moteurs de jeu, le débat entre Unity et Unreal Engine existe, maintenant, depuis quelques années. Ils sont tous deux très puissants et permettent de réaliser de grandes applications. Toutefois, les deux possèdent leurs propres avantages et inconvénients.

Dans les annexes se trouve un tableau comparatif de ces deux moteurs, inspiré d'un travail ayant déjà effectué cette comparaison en 2017^[13]. Néanmoins, les informations ont été vérifiées par rapport aux spécifications de 2021 et de les mettre à jour si besoin.

Ce tableau compare beaucoup de fonctionnalités étant, pour la plupart, plus utiles à la création de jeux vidéo. Cependant, dans le cadre de ce projet et la réalité augmentée, il permet de retirer les informations suivantes :

1. Bien que Unity ait globalement plus de plateformes cibles que Unreal Engine, les deux se valent dans le domaine de la réalité augmentée. Les deux moteurs peuvent déployer des applications sur les mêmes dispositifs supportant la réalité augmentée.
2. Le langage de programmation de Unity est le C#, alors que pour Unreal Engine, c'est le C++ qui est utilisé. De plus, Unreal Engine 4 (UE4) permet de réaliser des applications sans coder à l'aide du système de « Blueprints ». Un développeur pourrait porter son choix sur le moteur utilisant le langage dans lequel il est le plus à l'aise.
3. Unity supporte plus de formats que Unreal Engine, ce qui peut être important dans la réalité augmentée, notamment dans les formats pour modèles 3D.

Outre ce tableau, d'autres éléments peuvent être pris en compte pour faire un choix de moteur de développement.

1. Bien que cette notion soit subjective à chaque développeur, Unity tend à être plus intuitif pour le développement d'applications pour les débutants que Unreal Engine.
2. Les deux moteurs possèdent une forte communauté et tutoriels pour le développement de jeux vidéo. Néanmoins, en comparant les sites officiels de Unity et Unreal Engine, Unity est plus présent que Unreal Engine pour l'apprentissage de la réalité augmentée. Le même phénomène a été remarqué sur la plateforme « YouTube » en recherchant des tutoriels de prises en main. Cet élément peut être crucial pour faire un choix entre deux plateformes.

¹³ SMÍD, Antonín, Comparison of Unity and Unreal Engine, Czech Technical University in Prague, 2017. Chapter 6, Results and Comparison, p. 41 - 60

La comparaison de Microsoft Mesh fait-elle sens ?

Dans la donnée de ce travail de bachelor, il était demandé de comparer les trois technologies de ce chapitre. Toutefois, à l'aide des informations retirées lors de la prise en main de Microsoft Mesh, il a été constaté qu'une comparaison entre celle-ci et Unity ou Unreal Engine 4 ne fait guère de sens. En effet, Alors que Unity et UE4 sont des moteurs de jeu permettant de réaliser diverses applications, Microsoft Mesh est une plateforme de communication et de collaboration mélangeant la réalité et des éléments virtuels dans le but d'avoir de meilleures interactions entre les interlocuteurs. De ce fait, une comparaison supplémentaire de Mesh avec les deux autres technologies ne sera pas faite.

Conclusion

Enfin, après ces diverses prises en main et comparaisons, l'éditeur Unity a été choisi pour réaliser le prototype pour diverses raisons qui seront énoncées ci-dessous.

1. Comme expliqué plus haut, Mesh ne permet pas de réaliser des applications prototypes. Toutefois, elle peut être un excellent outil de communication, et l'utilisation du « software development kit (SDK) » de ce dernier peut être très intéressante. Cependant, étant en phase de « preview », le SDK n'est pas encore disponible. Il faudra encore attendre quelques temps afin que celui-ci le soit pour les développeurs. Pour information, un SDK serait un ensemble d'outils d'aide à la programmation pour développer différents logiciels.
2. Etant toujours à l'affût de nouvelles technologies et n'ayant pas d'expérience en C# (langage de programmation des scripts Unity), ce serait l'occasion idéale d'expérimenter et d'apprendre ce langage tout en concevant ce prototype pour ALPS Automation SA à l'aide de Unity alors que UE4 préconise le langage de programmation C++.
3. Bien que Unreal Engine possède également une large communauté, il semblerait qu'après différentes recherches, la communauté Unity est un peu plus impliquée dans la réalité augmentée, ce qui peut être intéressant dans les différents forums lorsqu'il y a besoin d'aide ou de « back-up ».
4. Enfin, subjectivement parlant, l'interface de Unity était plus agréable pour quelqu'un qui n'avait jamais travaillé ces avec technologies.

3.2.5 Présentation et prise en main de Mixed reality toolkit (MRTK)

Comme expliqué précédemment, le choix d'éditeur de développement s'est porté sur Unity. Dans ce dernier il est possible d'importer « Mixed Reality Toolkit », développé par Microsoft, qui fournit des composants et des fonctionnalités servant au développement d'applications de réalité mixte.

Application de prise en main déployée sur HoloLens 2

Tout d'abord, cette application a été déployée sur le casque de réalité mixte HoloLens 2, qui sera décrit plus tard dans le chapitre traitant des différentes technologies. Cette dernière, ayant pour but de découvrir MRTK et ses fonctionnalités dans Unity, a été réalisée en suivant un guide de prise en main fait par Microsoft. Afin de pouvoir la reproduire, le lien vers ce tutoriel^[14] sera écrit dans les sources.

Elle a pour objectif de :

- Apprendre à configurer Unity pour l'utilisation de MRTK.
- Apprendre à développer des applications et les déployer sur un appareil.
- Apprendre à utiliser certaines fonctionnalités de MRTK
- Créer une expérience de réalité mixte complète.

Le guide est séparé en différentes catégories séquentielles :

1. Introduction.
2. Initialisation d'un projet et déploiement d'une première application.
3. Configuration des profils MRTK.
4. Positionnement d'objets dans une scène.
5. Création de contenu dynamique en utilisant des « solvers ».
6. Création d'interfaces utilisateurs.
7. Interaction avec des objets 3D.
8. Utilisation du suivi du regard.
9. Utilisation des commandes vocales.

¹⁴ MICROSOFT, *1. Introduction to the MRTK tutorials*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/tutorials/mr-learning-base-01> (Consulté le 7 juin 2021)

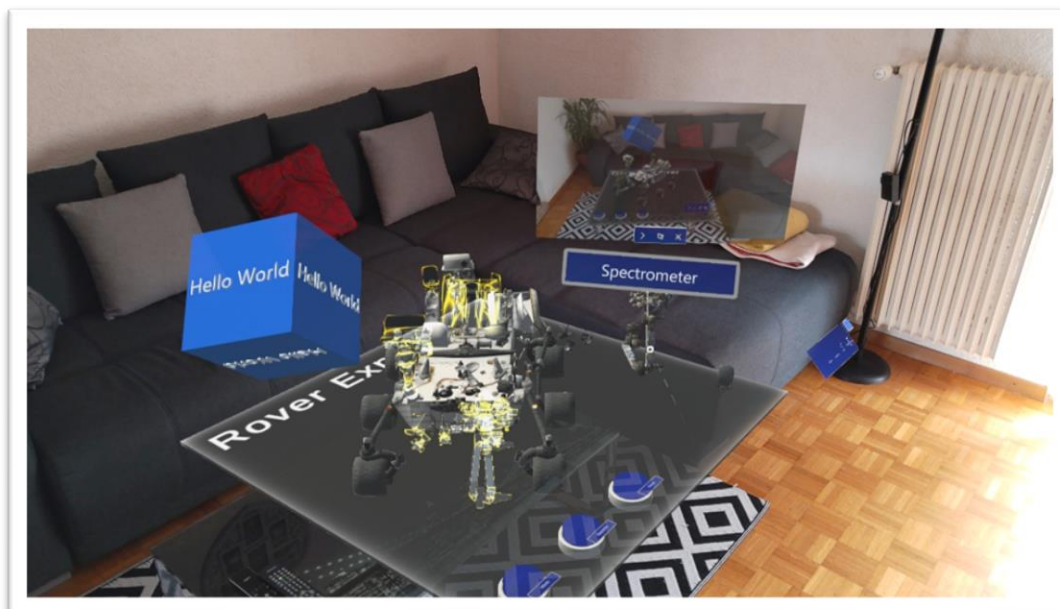


Figure 13 - Application réalisée pour la prise en main de MRTK

La figure ci-dessus est le résultat final de l'application déployée sur l'HoloLens 2. Le cube bleu « Hello World » a été le premier élément à être déployé sur le casque de réalité mixte. L'utilisateur peut le manipuler dans l'espace en le déplaçant, en le faisant subir des rotations ou encore en lui changeant sa taille.

La suite du guide était centrée sur le rover explorateur de Mars appartenant à la NASA se trouvant dans la figure ci-dessus. L'utilisateur dispose de plusieurs boutons et éléments d'interface lui permettant de manipuler le rover de la même manière que le cube « Hello World » ainsi que d'effectuer certaines opérations. Il peut prendre des pièces détachées, comme le spectromètre sur la figure, et les attacher au rover afin de compléter sa construction. Les zones en surbrillance jaune sont les emplacements des différentes pièces à installer. Les trois boutons sur le plateau permettent de désactiver les zones en surbrillance, d'afficher le rover en vue éclatée et de réinitialiser l'application. A l'aide du petit menu bleu situé sur la droite de la figure, l'utilisateur peut, en appuyant sur les divers boutons, utiliser les fonctionnalités suivantes :

- Activer un indicateur dirigeant l'utilisateur jusqu'à la plateforme du Rover lorsque celui-ci n'est plus dans son champ de vision.
- Prendre uniquement le Rover afin de le placer à une position souhaitée.
- Activer les bordures de la plateforme entière afin de la manipuler dans l'espace.
- Désactiver les bordures de la plateforme.

Prise en main de MRTK-WebRTC

Pour réaliser la communication vidéo et vocale, MRTK propose aux développeurs une collection de bibliothèques intégrables à Unity. De la sorte, le spécialiste et l'opérateur sur le site pourront communiquer afin que la mise en service ou la maintenance d'un système se passe correctement.

Une fois de plus, Microsoft met à disposition une prise en main de cette technologie, qui sera également mise dans les sources^[15] afin de la reproduire si nécessaire.

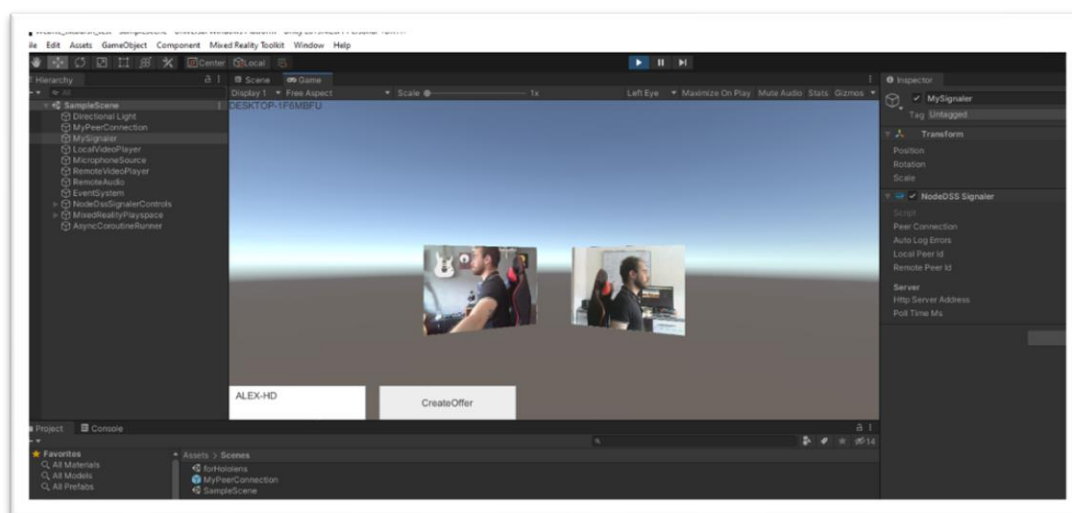


Figure 14 - Communication RTC PC-PC

Pour prendre en main cette technologie, la communication a été réalisée entre un ordinateur fixe et un ordinateur portable connecté par Wi-Fi sur le même réseau. La figure ci-dessus montre l'image des caméras propre à chaque PC.

De la même manière, cette fonctionnalité pourrait être déployée sur HoloLens 2 par exemple.

¹⁵ MICROSOFT.GITHUB.IO, Hello, Unity World!
<https://microsoft.github.io/MixedReality-WebRTC/manual/unity/helloworld-unity.html>
(Consulté le 7 juin 2021)

3.2.6 Dispositifs de déploiement pour la réalité augmentée

A ce jour, il existe de nombreux dispositifs et manières d'utiliser la réalité augmentée. Ci-dessous se trouve un tableau récapitulant ces derniers :



Figure 15 - Paradigme d'interaction avec la réalité augmentée

Source : CARRINO, Francesco & CAON Maurizio. Virtual, Augmented & Mixed Reality. MSC BA HES-SO, 2020-2021. Chapitre Mixed Reality Interaction, p.73.

Chacun ayant leur points forts et points faibles, il faudra faire un bon choix pour répondre aux exigences de ALPS Automation SA.

Par exemple, un ouvrier qui travaille sur une machine a besoin d'avoir ses mains libres afin d'être efficace tout en ayant à disposition l'aide de la réalité augmentée. Le choix pourrait donc se tourner vers un casque de réalité augmentée. A l'inverse, une personne visualisant un certain nombre de fauteuils dans son salon, pour décider lequel lui conviendrait le mieux, n'a pas forcément besoin de ses mains libres et pourrait donc le visualiser à l'aide d'une tablette.

Dans le chapitre « 3.3 Analyse du cas d'utilisation ALPS Automation SA », un choix sur les technologies à utiliser sera fait et une description détaillée sera faite afin de comprendre et argumenter le choix de ces dernières.

3.3 Analyse du cas d'utilisation ALPS Automation SA

Avoir une bonne analyse solide et claire permet de faire les bons choix dans un projet et de partir sur une bonne route. C'est à cet objectif que se consacre ce chapitre.

En effet, grâce aux recherches effectuées dans les chapitres précédents, un lien sera fait entre ces dernières et les besoins de ALPS Automation SA pour avoir le meilleur choix et le plus optimisé à leur fournir.

3.3.1 Personnes en jeu et dispositifs utilisés

Tout d'abord, les personnes en jeu ainsi que leurs possibilités d'interactions ont été déterminées. Ces dernières sont les suivantes :

1. B (Bureau) : Spécialiste qui fournit l'assistance depuis son bureau.
2. S (Site) : Personnel qui bénéficie de l'assistance depuis le site.

Ces derniers peuvent avoir à disposition les dispositifs suivants :

Matériel B	Obligatoire	Nice to have
Laptop	X	
Smartphone	X	
Tablette	X	
Pointpad (hands-overlay)		X

Tableau 1 - Matériel à disposition depuis le bureau

Matériel S	Obligatoire	Nice to have
Smartphone	X	
Lunettes caméra + écran	X	
Tablette	X	
Laptop	X	

Tableau 2 - Matériel à disposition depuis le site

L'application doit donc être cross-platform afin d'être utilisable, par exemple, depuis un laptop au bureau et d'un casque de réalité augmentée sur le site.

Grâce à ces informations sur les besoins de ALPS Automation SA, plusieurs dispositifs du paradigme d'interaction montré dans les pages précédentes peuvent ne plus être considérés. En effet, les projecteurs et dispositifs installés dans l'environnement ne sont pas pratiques car la personne sur le site a besoin d'un dispositif efficace, ergonomique et mobile.

Le meilleur choix reste donc un casque de réalité augmentée qui permet d'avoir ses mains libres, tout en bénéficiant d'une très haute qualité d'interaction avec la réalité augmentée. Toutefois, bien que les smartphones et tablettes utilisent les mains de la personne sur le site, ils peuvent être intéressants en cas de maintenance non planifiées étant très souvent à portée de main.





































COMPARAISON ENTRE CASQUES AR & SMARTPHONES/TABLETTES							
CASQUE DE RÉALITÉ AUGMENTÉE				SMARTPHONE TABLETTE			
	GOOD	MODERATE	BAD		GOOD	MODERATE	BAD
PRIX	 < 1200 CHF	 2000 < CHF < 4000	 > 4000 CHF	PRIX	 < 1200 CHF	 2000 < CHF < 4000	 > 4000 CHF
MANIABILITÉ ET ERGONOMIE	 2 Mains libres	 1 Main libre	 2 Mains prises	MANIABILITÉ ET ERGONOMIE	 Mains libres	 1 Main libre	 2 Mains prises <i>Tablette</i>
ACCESSIBILITÉ ET FACILITÉ D'ADAPTATION (Pour les entreprises)	 Immersion totale (VR)	 Immersion non totale	 Taille de l'écran	ACCESSIBILITÉ ET FACILITÉ D'ADAPTATION (Pour les entreprises)	 Immersion totale (VR)	 Immersion non totale	 Taille de l'écran
CHAMP DE VISION	 Plusieurs caméras et sensors	 Une Caméra	 Pas de caméra et sensors	CHAMP DE VISION	 Plusieurs caméras et sensors	 Une Caméra	 Pas de caméra et sensors
AR TRACKING	 Gestes, voix, eye track, touchpad	 Gestes, voix, eye track, touchpad	 Gestes, voix, eye track, touchpad	AR TRACKING	 Gestes, voix, eye tracking, screen	 Gestes, voix, eye tracking, screen	 Gestes, voix, eye tracking, screen
QUALITÉ DE L'INTERACTION HUMAINE	 Gestes, voix, eye track, touchpad	 Gestes, voix, eye track, touchpad	 Gestes, voix, eye track, touchpad	INTERACTION HUMAINE	 Gestes, voix, eye tracking, screen	 Gestes, voix, eye tracking, screen	 Gestes, voix, eye tracking, screen

Figure 16 - Casque de réalité augmentée VS Smartphone et tablette

La figure ci-dessus permet de voir rapidement les points forts et faibles des deux types de technologies choisies pour ce projet.

En termes de prix, les smartphones sont plus avantageux étant donné que même les plus performants restent moins chers que les casques de RA les plus courants. En ce qui concerne la maniabilité et l'ergonomie des appareils, les casques se défendent mieux grâce à leur capacité à être « hands-free », alors que les smartphones et tablettes ont besoin d'une main, voire deux mains, pour être utilisés correctement. C'est un « must have » pour un employé en maintenance ou mise en service qui a besoin de ses mains pour travailler. Cependant, cette technologie est plus compliquée à être acceptée dans les entreprises comme décrit dans le chapitre sur les facteurs limitant la réalité augmentée dans les industries. Un autre point fort des casques est leur champ de vision plus grand que la limite d'un écran smartphone. Toutefois, cette dernière notion est assez générale et est propre à chaque casque de réalité augmentée. En effet, il sera décrit plus bas le champ de vision de certains casques de RA, et il sera possible de constater que le champ de vision du casque RealWear est bien plus petit que celui du Magic Leap One. Les casques de RA disposent de plus de caméra et sondes, permettant un meilleur « tracking » de la réalité augmentée. Enfin, lorsque la technologie est maîtrisée, le casque de RA est beaucoup plus performant et permet une meilleure et plus grande interaction avec l'homme.

Il peut donc être relevé que le casque de réalité augmentée est une meilleure solution à long terme après avoir rentabilisé son coût d'achat et dès lors que les employés se sont habitués à cette technologie. Néanmoins, le smartphone et la tablette sont une très bonne solution de secours dans le cas où un casque ne serait pas disponible lors d'une intervention non planifiée.

3.3.2 Diagramme physique

Ci-dessous se trouve un exemple de setup qui pourrait s'avérer concret dans la réalité. Les dispositifs et les moyens de connexion peuvent varier, mais le raisonnement reste le même.

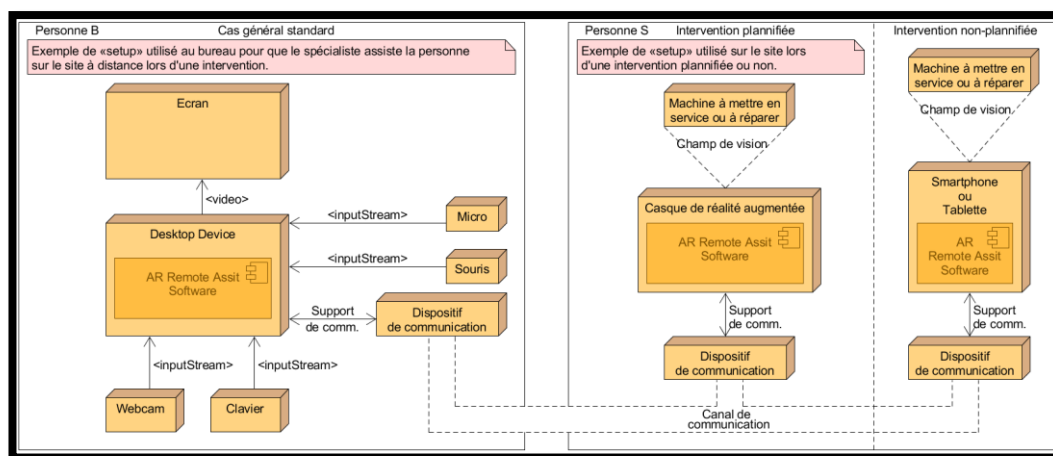


Figure 17 - Diagramme Physique : Exemple de setup et de mise en situation

La personne assistée sur le site filme la scène réelle à l'aide de son casque de RA (ou autre dispositif) et envoie les informations par internet au spécialiste se trouvant dans son bureau. Depuis son ordinateur (ou autre dispositif), le spécialiste guide la personne sur le site à l'aide d'annotations et des moyens de communications grâce à ses entrées et sorties (clavier, souris, micros, webcam, etc.).

Grâce au software intégré dans le dispositif de la personne assistée et du spécialiste, des interactions sont possibles entre eux afin de résoudre le problème dans les meilleures conditions. Ce software est le prototype étant développé dans le cadre de ce projet et devra pouvoir être déployé dans le casque de réalité augmentée pour le cas standard et dans un smartphone ou dans une tablette dans une intervention non-planifiée.

La différence de l'application dans les différents systèmes embarqués, est la manière d'interagir avec elle. En effet, sur PC, elle sera guidée à l'aide d'un clavier et d'une souris. Avec un casque de RA, elle sera utilisée à l'aide des « mains dans l'air » et, avec un smartphone, elle sera utilisée avec un écran tactile.

Le mode « dégradé », représenté par un smartphone ou une tablette dans le schéma précédent, est là en cas d'intervention non-planifiée. En effet, si une panne se produit et qu'elle doit être urgemment réparée, il faut réagir sans plus attendre. Un casque de réalité augmentée n'est pas toujours disponible à l'étranger, et envoyer un opérateur sur place demande trop de temps. C'est pourquoi ce mode « dégradé » est prévu. Une personne sur place pourra prendre en main la situation à l'aide d'un smartphone ou d'une tablette en étant guidé par un spécialiste grâce à l'application. Un mode « dégradé » peut aussi être réalisé pour les spécialistes.

Ce scénario est très généralisé et simple. L'analyse plus détaillée qui suivra dans les sections suivantes, permettra de choisir les dispositifs et services appropriés.

3.3.3 Description des besoins d'ALPS Automation SA

Ci-dessous se trouve un tableau avec la description des besoins d'ALPS Automation SA. Ce dernier contient les fonctionnalités demandées, s'il est obligatoire ou « nice to have » et si la fonctionnalité peut être réglable par l'utilisateur ou non.

Fonctionnalité	Obligatoire Dans tous les cas.	Configurable Réglage utilisateur	Nice to have
Interactions en direct	X		
Voix (B=>S + S=>B)	X		
Vidéo (S=>B)	X		
Vidéo (B=>S)			X
Annotations (B=>S)	X		
Annotations (S=>B)	X		
Fonctionnel sur réseau 4G	X		
Vidéo exploitable par faible luminosité			X
Mode "éco" (faible débit données)	X		
Mains libres sur site	X		
Annotations en différentes couleurs	X	X	
Libres	X		
Textes	X		
Formes (d'une librairie)			X
Superposition de fichier en transparence/sur une zone de l'écran	X		
Partage d'écran en transparence/sur une zone de l'écran	X		
Sauvegarde d'instantanées	X		
Zoom sur image/vidéo	X		
Large compatibilité software/hardware	X		
Mise en œuvre facile	X		
Utilisation intuitive	X		
Sessions multi-participants	X		
Coût compétitif (software, hardware, licences)	X		
Fonctionnel dans un environnement bruyant	X		
Matériel certifié ATEX			X
Enregistrement vidéo et annotations		X	
Traduction instantanée			X
Main virtuelle (B=>S)			X
Support technique	X		
Accessibilité annuaire contacts			X

Figure 18 - Description des besoins de ALPS Automation SA

Focus sur les fonctionnalités hardware

Les critères précédents étant à disposition, il est possible d'investiguer les différents casques de réalité augmentée afin de déterminer, lequel répondrait au mieux aux besoins de ALPS Automation SA.

Concernant les spécifications hardware, les nécessités suivantes ont été retirées :

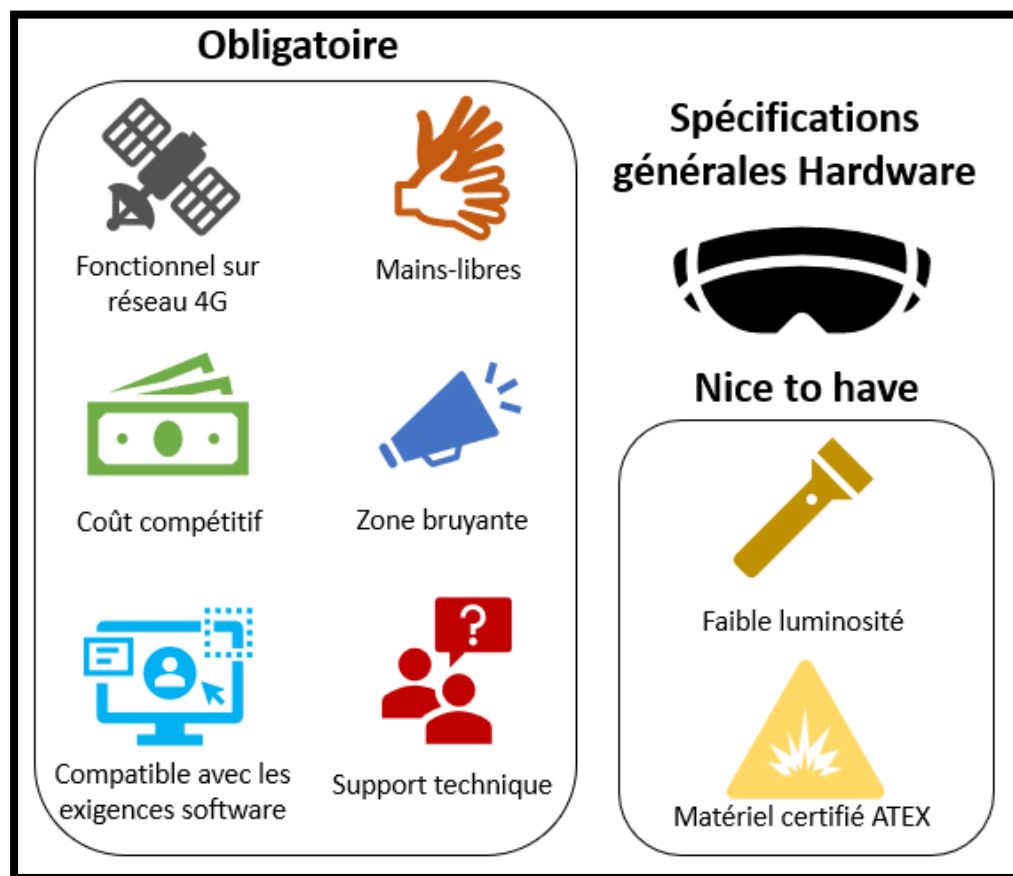


Figure 19 - Spécifications nécessaires de l'Hardware

Le dispositif choisi devra pouvoir se connecter au réseau 4G (Modem LTE ou WiFi Hotspot), il devra également être « main-libres » pour permettre à l'utilisateur de travailler dans de bonnes conditions. Le micro du dispositif devra être assez bon pour avoir des commandes vocales claires et précises dans les zones bruyantes. L'appareil devra également être apte à supporter toutes les fonctionnalités software. De plus, il devra avoir un coût compétitif (sur le long terme) et bénéficier d'un bon support technique en cas de nécessité.

Dans l'idéal, mais pas obligatoirement, la caméra du dispositif devrait être assez bonne pour pouvoir fournir des images qualitatives dans les zones à faible luminosité. De plus, c'est un plus s'il est certifié ATEX (ATmospheres EXplosives).

Focus sur les fonctionnalités softwares

En analysant les critères de ALPS Automation SA, il est possible d'investiguer les différents services existants, lesquels répondraient au mieux aux besoins de l'entreprise.

Concernant les spécifications software, les nécessités générales suivantes ont été retirées :

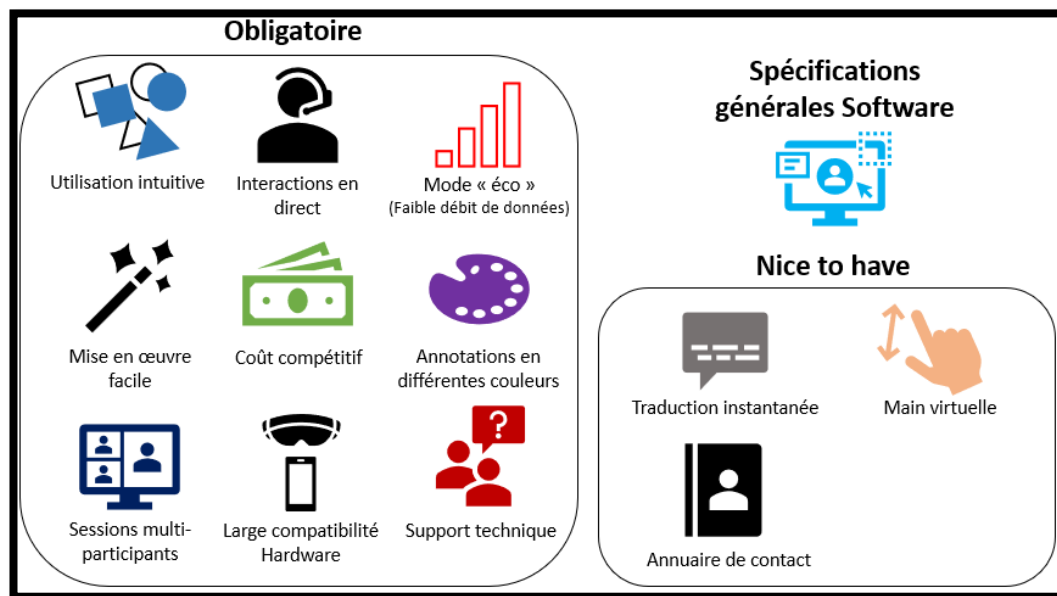


Figure 20 - Spécifications générales nécessaires du software

Ce dernier doit être facile à mettre en œuvre et doit fournir une utilisation intuitive pour que les utilisateurs puissent bien le manier. Les interactions doivent se faire en direct entre deux ou plusieurs participants. Dans le cas où la connexion à Internet n'est pas bonne et stable, un mode « éco » doit être disponible afin de limiter le débit de données en, par exemple, coupant la diffusion vidéo et ne garder qu'une communication vocale. Les utilisateurs doivent pouvoir faire différentes sortes d'annotations en différentes couleurs. Le software ainsi que les licences doivent être compatibles avec plusieurs dispositifs afin d'être le plus flexible possible et ce, à un coût compétitif. Enfin, un support technique devrait être à disposition pour résoudre des pannes en cas de nécessité.

Le software devrait également implémenter des fonctionnalités idéales mais non-obligatoires. Dans ces dernières se trouve un système de traduction instantanée, qui aurait la possibilité de sous-titrer et/ou traduire ce qui est dit en anglais ou en français. Il serait également idéal de pouvoir implémenter une main virtuelle du côté bureau pour venir en aide à la personne sur site. Enfin, un annuaire de contact serait intéressant également pour effectuer des appels rapidement.

Les fonctionnalités générales étant définies, il est possible de définir précisément les actions réalisables par le spécialiste et par la personne assistée. Pour ce faire, un diagramme de cas UML a été réalisé.

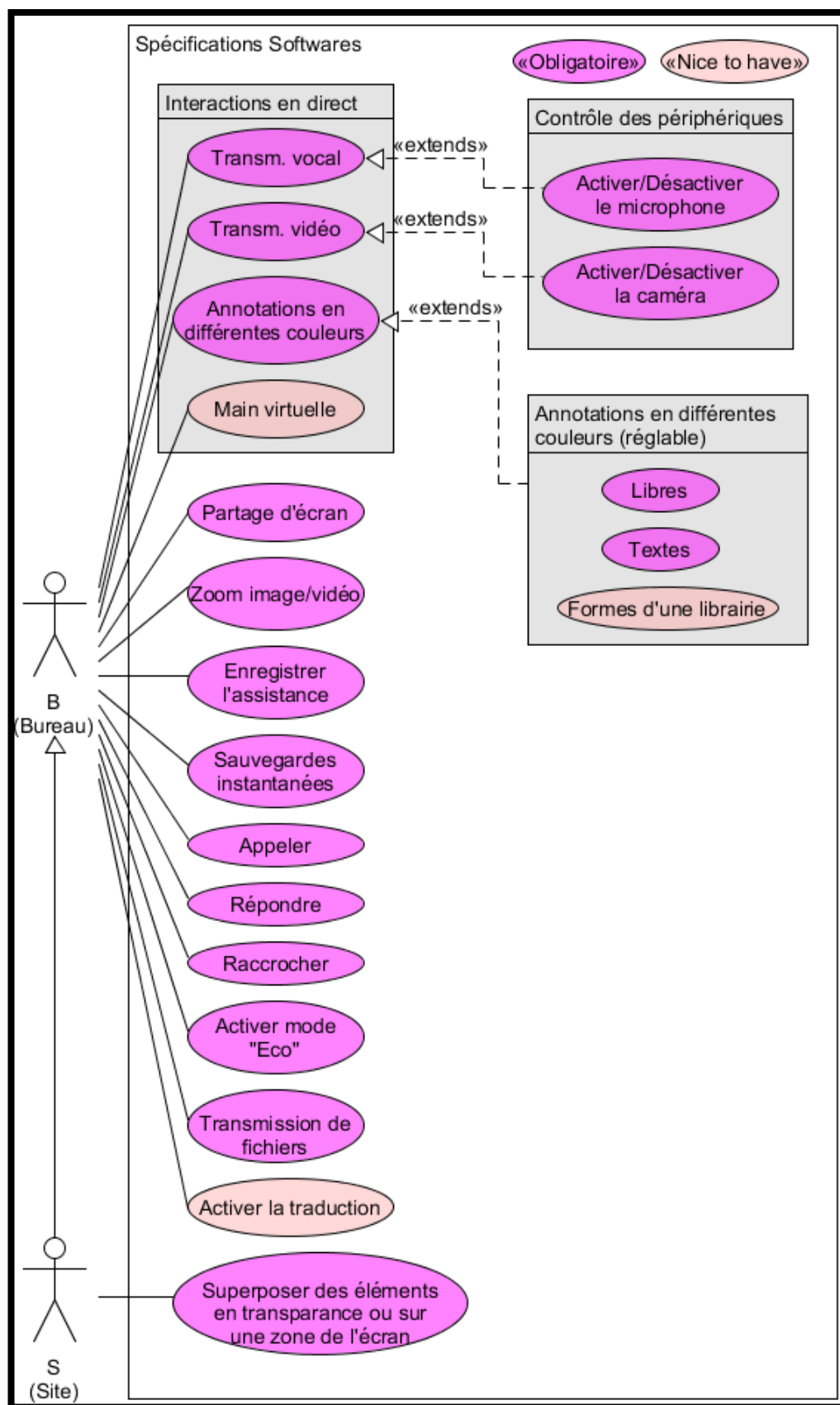


Figure 21 - Diagramme de cas d'utilisation UML du spécialiste au bureau et la personne assistée sur site

Sur la figure précédente se trouvent les différentes fonctionnalités de l'application utilisable par le spécialiste et la personne assistée. En magenta se trouvent les fonctionnalités obligatoires pour ALPS Automation SA et, en rose saumon, se trouvent les fonctions dites « Nice to have ».

Le spécialiste et la personne sur le site doivent être capables de s'appeler, de se répondre, de raccrocher, de se parler directement et de se transmettre diverses annotations (libres, textes, formes optionnelles) de couleurs réglables. Ils doivent être également capables de contrôler leur caméra et leur microphone. Ils doivent disposer d'une fonction leur permettant de partager leur écran, de transmettre des fichiers et doivent également zoomer sur l'image ou sur la vidéo. Ils doivent également avoir la possibilité d'enregistrer l'assistance ainsi que de faire des « shots » instantanés avec les annotations visibles. En outre, ils doivent pouvoir mettre la communication en mode « éco » en cas de faible débit de données.

Optionnellement, le spécialiste devrait pouvoir filmer et transmettre en direct ce qu'il filme à la personne sur le site. Une main virtuelle et l'option de traduction peuvent être, comme déjà énoncé plus tôt, une option intéressante.

La personne sur le site doit, obligatoirement, pouvoir filmer ce qu'il voit et le transmettre au spécialiste. Il doit également pouvoir superposer des éléments virtuels (fichiers, écrans, etc.) dans la réalité qu'il voit.

3.3.4 Comparaison entre les différents casques de réalité augmentée pour ALPS Automation SA

Les recherches menées permettent de lister quatre casques de RA courant et pouvant être des candidats potentiels pour ce projet. Ces quatre dispositifs sont les suivants :



Figure 24 - Hololens 2



Figure 25 - Magic Leap One



Figure 22 - RealWear HMT-1



Figure 23 - Vuzix M400

Pour ce faire, la figure ci-dessous a été réalisée pour permettre de comparer facilement les spécifications de chaque appareil.

Comparaison des casques de réalité augmentée				
Ces informations ont été retrouvées sur les sites officiels des appareils				
(*) Les prix sont obtenus après conversion des dollars américains vers des CHF (12.05.2021)				
COMPAGNIE	HOLOLENS 2	MAGIC LEAP ONE	REALWEAR HMT-1	VUZIX M400
	Microsoft	Magic Leap	RealWear	Vuzix
PRIX	3799 CHF	2075 CHF (*)	2260 CHF (*)	2244 CHF (*)
PLATEFORME	Holographique Windows	Lumin OS	Android + WearHF hands-free interface	Android (OS and apps OTA upgradeable)
CONNECTIVITÉ	WiFi, Bluetooth 5.0 USB-C, USB Ethernet for 5G/LTE	WiFi Bluetooth 4.2 USB-C PD 2.0	WiFi, GPS, USB-C Bluetooth 4.1 LTE 4G Modem	WiFi, GPS, USB-C Bluetooth 5.0 LTE 4G Modem
DURÉE DE VIE DE LA BATTERIE	2-3 heures	3.5 heures	9-10 heures	2-12 heures en dépend de la batterie externe
CHAMP DE VISION	43° x 29°	50 ° (wide)	20°	16.8° (Diagonal)
AR TRACKING	6DoF, Cartographie spatiale, Caputre de réalité mixte	6DoF	9-DoF (3-axis accelerometer, magnetometer, and gyroscope)	3 DoF head tracking, 3 axes gyro, 3 axes accel, 3 axes mag/integrated compass
INTERACTION HUMAINE	Gestes, voix, eye tracking	6 DoF contrôleur, gestes, voix, eye tracking	Voix (Détection précise), regard	Voix, touchpad
RÉSOLUTION DE LA CAMÉRA	8-MP stills, 1080p30 video	2 MP color sensor, 1080p30 video	16 MP 4-axis optical image stabilization with LED flashlight 1080p30 video	12.8 MP stills 4k30 video

Figure 26 - Comparaison des casques de réalité augmentée

Les quatre technologies ont leurs propres points forts et points faibles. La comparaison entre ces dernières sera faite ci-dessous afin de faire un choix judicieux pour la suite de ce projet.

Hololens 2

Points forts :

- Appartient à Microsoft

Explication : Ce casque à l'avantage d'avoir été conçu par Microsoft, qui est une entreprise qui peut fournir du support sur le long terme en cas de besoin. De plus, cela permet à l'Hololens 2 de s'adapter aux différentes technologies appartenant à Microsoft.

- Plateforme Windows
- Connectivité : LTE 5G Modem via USB-C
- Champ de vision (43°x29°)
- AR Tracking : Cartographie spatiale
- Interaction humaine

Points faibles :

- Prix (3799CHF)
- Durée de vie de la batterie (~2,5 heures)

Magic Leap One

Points forts :

- Prix (~2075 CHF)
- Champ de vision (50° Wide)

Points faibles :

- Durée de vie de la batterie (~3,5 heures)
- Interaction humaine : 6-DoF Contrôleur

Explication : Ce contrôleur pourrait, en général, être un bon outil pour l'utilisateur. Mais pour le cas d'utilisation de ALPS Automation SA, dans lequel il est nécessaire d'avoir les mains libres, c'est encombrant. Sans ce contrôleur, les capacités du Magic Leap One ne seraient pas utilisées à son plein potentiel.

RealWear HMT-1

Points forts :

- Prix (~2260 CHF)
- Plateforme Android
- Connectivité : LTE 4G Modem
- Durée de vie de la batterie (~9,5 heures)
- Résolution caméra : 16 MP + Flashlight

Points faibles :

- Champ de vision (20°)
- Interaction humaine : Pas d'interaction avec les gestes

Vuzix M400

Points forts :

- Prix (~2244 CHF)
- Plateforme Android
- Connectivité : LTE 4G Modem
- Durée de vie de la batterie : Dépend de la batterie choisie (12h)
- Résolution de caméra : 4K

Points faibles :

- Champ de vision (16,8° diagonal)
- Interaction humaine : Seulement voix + touchpad
- Durée de vie de la batterie : Dépend de la batterie choisie (2h)

Conclusion

Le tableau comparatif a pu ressortir des éléments importants servant à faire un choix de casque de réalité augmentée. Le choix s'est porté sur le casque HoloLens 2 pour les raisons suivantes :

Bien qu'il soit le plus cher, c'est un investissement rentable à long terme. De plus, après avoir discuté avec ALPS Automation SA, si l'appareil se démarque et réponds aux exigences de l'entreprise, le coût d'achat est moins important. En effet, c'est plutôt le coût mensuel du service (d'une application) qui devient important pour ALPS Automation SA. En réalisant une application qui répond à leurs besoins et en la déployant sur l'HoloLens, les coûts mensuels pourront être diminués, voir supprimés.

Ensuite, la valeur de la durée de vie de la batterie, qui était un autre point négatif de l'HoloLens 2, correspondait à une utilisation maximale des capacités de ce casque. L'utilisateur a la possibilité d'augmenter la durée de batterie en diminuant, par exemple, la luminosité des hologrammes, le volume ou en changeant d'autres paramètres. De plus, le casque se recharge complètement en moins d'une heure, ce qui peut être fait pendant la pause de midi par exemple.

Ensuite, ses capacités de tracking de la réalité augmentée sont simplement supérieures à ses adversaires avec une cartographie spatiale et une interaction humaine très performantes. Son champ de vision n'est pas autant bien que Magic Leap One, mais largement suffisante pour une intervention de maintenance. Les propriétés de Magic Leap One ne sont pas assez bonnes pour faire pencher la balance de son côté malgré son faible prix et son champ de vision plus grand.

De plus, HoloLens 2 a des fonctionnalités de bases très utiles. Par exemple, OneDrive est intégré par défaut et peut être facilement accédé pour avoir accès aux documents importants. Aussi, l'assistance vocale permet de prendre des photos instantanées ou d'enregistrer une session vidéo.

Enfin, la HES-SO met un casque HoloLens 2 à disposition de ce travail de diplôme, ce qui colle parfaitement bien aux besoins de ce projet. Un autre candidat très intéressant était également le RealWear HMT-1, ce dernier sera tout de même fourni par ALPS Automation SA afin de tester ses différentes capacités et limites.

3.3.5 Comparaison entre les différents services existants pour ALPS Automation SA

Une analyse des limites et capacités des différents services existants, permettant de faire de l'assistance à distance à l'aide de la réalité augmentée, a été faite dans le but de comprendre pourquoi ces derniers ne répondent pas aux besoins de ALPS Automation SA. Elle permet également de retirer le meilleur de chaque service pour les intégrer dans le prototype à développer ainsi que de le compléter par les fonctionnalités manquantes.

Comparaison des services de réalité augmentée				
	TEAMVIEWER PILOT	VUFORIA CHALK	WORKLINK	DYNAMICS 365 REMOTE ASSIST
INTERACTIONS TEMPS RÉEL	✓	✓	✓	✓
PRIX (PAR MOIS)	Lite (Small Teams) 37,90 CHF /user	Premium Plan (5 users package) 349 CHF	Informations non-disponibles	64 CHF /user
APPAREIL D'UTILISATION	iOS, Android, RealWear, Vuzix, Hololens 2, MAC, Windows	iOS, Android, RealWear, MAC, Windows	iOS, Android, RealWear, Vuzix, Hololens 2, MAC, Windows	iOS, Android, Hololens 2, Windows
SDK POUR DÉVELOPPEURS	Dans l'offre "Enterprise"	✗ Vuforia Engine	✗	✗

Figure 27 - Comparaison générale de quatre services réputés

La figure ci-dessus présente une comparaison générale de quatre services d'assistance à distance :

1. TeamViewer Pilot (Assist AR)
2. Vuforia Chalk
3. Worklink by Scope AR
4. Dynamics 365 Remote Assist (Microsoft)

Ces quatre services permettent aux spécialistes et aux personnes assistées d'interagir ensemble en temps-réel.

Concernant les prix, certaines informations n'ont pas été retrouvées étant donné qu'elles n'étaient disponibles qu'après une analyse spécifique des besoins d'entreprises et d'un devis. Des courriels ont été envoyés pour essayer d'obtenir les informations voulues, mais les destinataires n'ont jamais répondu. Toutefois, sur le site officiel de TeamViewer il est possible de retrouver le prix de l'offre « Lite ». Grâce aux informations de ALPS Automation SA, qui ont déjà testé Vuforia Chalk, le prix de la licence de ce dernier a pu être retrouvé pour 5 utilisateurs. En outre, Sur le site officiel de Microsoft, il est possible de retrouver le prix de Dynamics 365 Remote Assist. Seul le prix de Worklink n'a pu être, malheureusement, retrouvé.

Concernant les appareils de déploiement, les quatre services s'en sortent bien, avec un avantage de Worklink et TeamViewer Pilot qui s'adaptent plus facilement. Toutefois, Dynamics 365 Remote Assist synergise parfaitement avec les fonctionnalités de l'HoloLens 2.

Du côté développeur, TeamViewer Pilot propose un SDK (payant) dans la version « Entreprise » afin de réaliser des applications sur mesure. Vuforia propose un SDK nommé « Vuforia Engine » qui n'est, cependant, pas directement lié à Vuforia Chalk.

Comparaison des services de réalité augmentée											
	TRANSM. VOCALE ET VIDÉO EN DIRECT	ANNOTATIONS EN DIFFÉRENTES COULEURS	ANNOTATIONS LIBRES + FLÈCHES	ANNOTATIONS TEXTES	ANNOTATIONS AVEC DES FORMES D'UNE LIBRAIRIE	"SEE WHAT I SEE"	MODE ECO	ENREGISTRER L'ASSISTANCE	PAUSE DE L'IMAGE POUR SAUVEGARDES INSTANTANÉES	SUPERPOSITION DE DOCUMENTS EN TRANSPARENCE	SESSIONS MULTIPARTICIPANTS
TEAMVIEWER PILOT	✓	✓	✓	○	○	✓	○	✓	✓	○	○
VUFORIA CHALK	✓	✓	✓	○	○	✓	✓	✗ Exportation d'images de la session	✓	○	✓
WORKLINK BY SCOPE AR	✓	✓	✓	✓	○	✓	○	✓	✓	○	○
DYNAMICS 365 REMOTE ASSIST	✓	✓	✓	✗ Texte dans le tchat	○	✓	✓	✓	✓	✓	✓

Figure 28 - Comparaison des fonctionnalités des quatre services

La figure ci-dessus montre les principales fonctionnalités demandées par ALPS Automation SA ainsi que les capacités et limites des quatre services présentés plus tôt. Cette figure montre que l'application de Microsoft est celle qui correspondrait le mieux aux exigences du partenaire industriel. De plus, Dynamics 365 Remote Assist, faisant partie de Microsoft, dispose d'un très bon support fiable et à long terme. En outre, comme énoncé dans les sections précédentes, le dispositif choisi pour la réalisation de ce projet est l'HoloLens 2, qui est l'appareil de prédilection de ce service. Il serait fort intéressant pour ALPS Automation SA d'avoir la possibilité d'essayer la version d'essai de ce service. De plus, le coût total de ce dernier pour 5 utilisateurs reviendrait à 320 CHF par mois, ce qui reste moins cher que l'offre de Vuforia Chalk proposé à l'entreprise. Bien qu'il soit également adapté pour iOS, un de ses points faibles serait sa mauvaise adaptation avec MacOS.

3.3.6 Conclusion de l'analyse du cas d'ALPS Automation SA

Cette section a permis de mettre en lumière les différents besoins de ALPS Automation SA ainsi que de voir ce qu'il manquait dans les technologies et services existants pour satisfaire l'entreprise.

Appareils de déploiement

Selon les différents points énoncés plus tôt, il a été décidé de réaliser ce proof-of-concept pour PC, smartphone ainsi que HoloLens 2. ALPS Automation SA possède un casque RealWear-HMT1, sur lequel pourrait être développé et déployé une application d'assistance à distance. Mais faute de temps, cette dernière ne pourra être réalisée.

Applications et Services

Il était compliqué de regrouper les différentes exigences de ALPS Automation SA dans un seul service existant.

Ci-dessous se trouvent les points manquant dans les services existants :

- Le manque d'annotations sous formes de textes et de formes d'une librairie
- Le manque du mode économique
- Le manque de superpositions de documents/partage d'écran
- La limite d'appels à deux personnes

Néanmoins, l'application Remote Assist de Microsoft serait celle qui correspondrait au plus à ALPS Automation SA.

Ce proof-of-concept réunira au mieux, selon le temps à disposition, les différents besoins manquant dans les applications existantes.

4. Conception Générale

4.1 Conception de l'architecture générale

Ce chapitre décrit l'architecture générale du système à l'aide de différents diagrammes de composants UML. En mettant une bonne structure en place et en séparant bien les différents systèmes, il sera plus aisé de réaliser les différents prototypes. La modularité de ces derniers sera également mise en avant.

4.1.1 Layer 1 : Liens entre les utilisateurs et les appareils

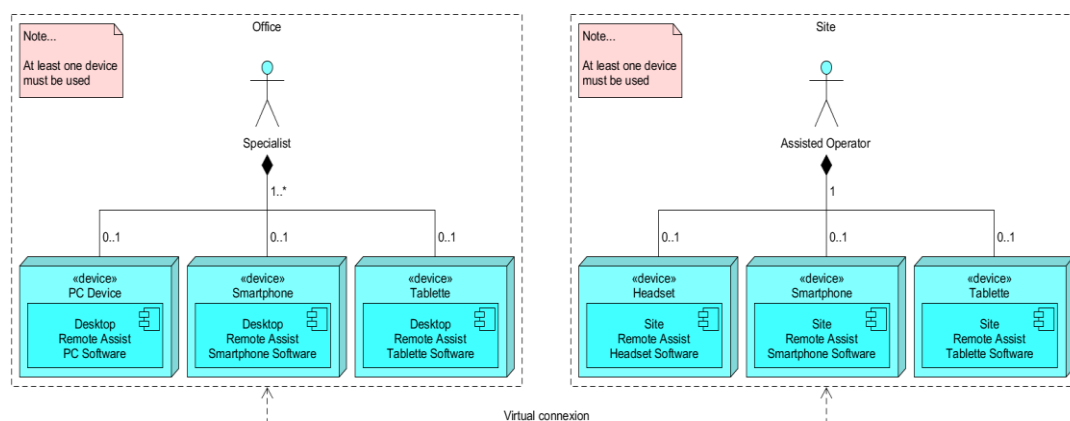


Figure 29 - Layer 1 : Liens entre les appareils et les utilisateurs

Cette couche, proche du diagramme physique présenté à la Figure 17, présente les liens entre les utilisateurs (spécialistes et personnes assistées) et les différents appareils de déploiement. Pour utiliser correctement ce proof-of-concept, les utilisateurs doivent choisir au moins un appareil contenant l'application réalisée dans ce projet. Ces derniers seront connectés ensemble à l'aide d'une technologie de communication (« virtual connexion ») permettant de transmettre et recevoir un flux vidéo/audio ainsi que d'envoyer et recevoir certaines données entre le « Site » et le « Desktop ». Cette connexion peut être, par exemple, une connexion TCP/IP ou http selon les données à envoyer. Elle sera détaillée dans les sections suivantes.

La figure précédente montre également que chaque appareil dispose de l'application réalisée. Il est à noter que, pour diverses raisons de fonctionnalités, les applications Smartphones et Tablettes sont différentes entre le côté site et le côté bureau. De plus, bien qu'ici les noms soient différents, les applications smartphones et tablettes sont les mêmes dues à leur similitude de prises en main et de fonctionnalités.

4.1.2 Layer 2 – Diagramme de composants général (Spécialiste)

La figure ci-dessous présente le diagramme de composants général des appareils utilisables par un spécialiste. Ces derniers peuvent être : Un PC, un smartphone ou une tablette.

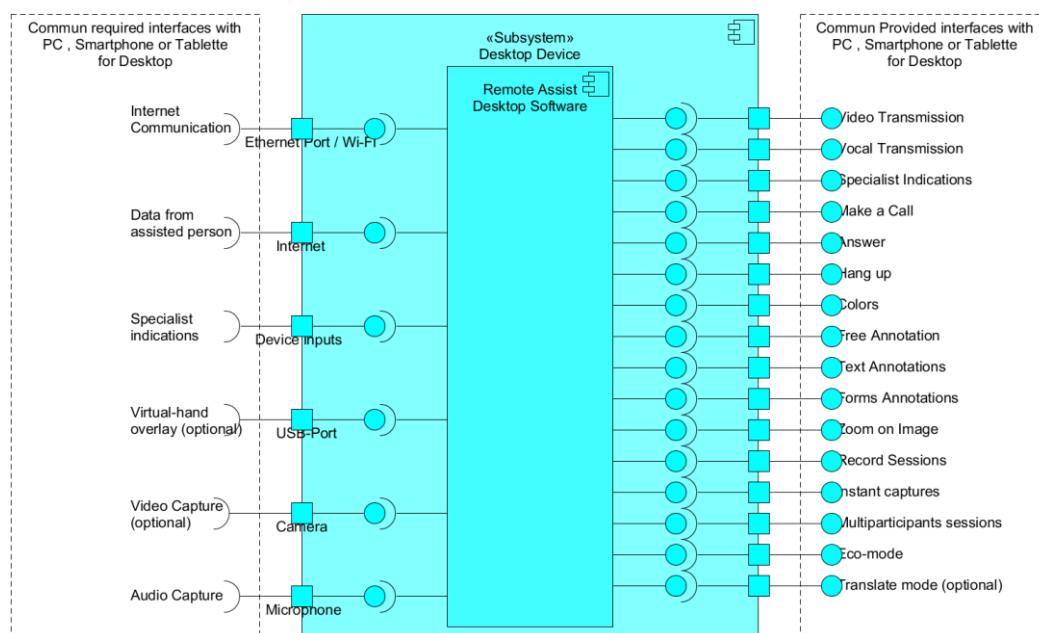


Figure 30 - Layer 2 : Diagramme de composants général du prototype du spécialiste

Pour fonctionner correctement, les prototypes *requièrent* certaines interfaces afin d'en *fournir* d'autres. À l'aide des différents ports, si différents appareils dotés de mêmes ports le permettent, ils peuvent être utilisés par le spécialiste.

Interfaces requises

À l'aide d'un port Ethernet ou d'une connexion par Wi-Fi, l'appareil se connecte à Internet afin que l'application prototype se connecte à son pair se trouvant sur le site. Cette connexion permet de recevoir des données de la personne assistée. Ces données peuvent être, par exemple, être un flux vidéo ou audio.

À l'aide des différents périphériques d'entrées, le spécialiste peut interagir avec l'application afin d'effectuer certaines actions permettant de créer des indications à envoyer à la personne sur le site.

Le microphone de l'appareil permet d'enregistrer la voix du spécialiste qui sera transmise à la personne sur le site. Optionnellement, à l'aide de la caméra de l'appareil, il est également possible d'obtenir une vidéo en direct afin d'envoyer également au site.

Enfin, pour ce qui est du PC, il se pourrait qu'un appareil permettant de simuler une main virtuellement se connecte à ce dernier afin de la retransmettre au site et d'indiquer certaines informations à la personne assistée.

Interfaces fournies

Le prototype du spécialiste peut fournir plusieurs interfaces. Le spécialiste peut transmettre sa vidéo, sa voix ainsi que ces différentes annotations (libres, textes, formes) en différentes couleurs via la connexion internet. L'application fournit également les différentes actions d'appels de bases, telles qu'appeler un pair, répondre ou encore raccrocher.

Le prototype devrait également pouvoir fournir des fonctions de zoom, d'enregistrements, de photos, de d'appels multi-participants ainsi qu'un mode économique. Optionnellement, une fonction de translation peut être fournie.

4.1.3 Layer 2 – Diagramme de composants général (Site)

La figure ci-dessous présente le diagramme de composants général des appareils utilisables par la personne assistée sur le site. Ces derniers peuvent être : un casque de réalité augmentée, un smartphone ou une tablette.

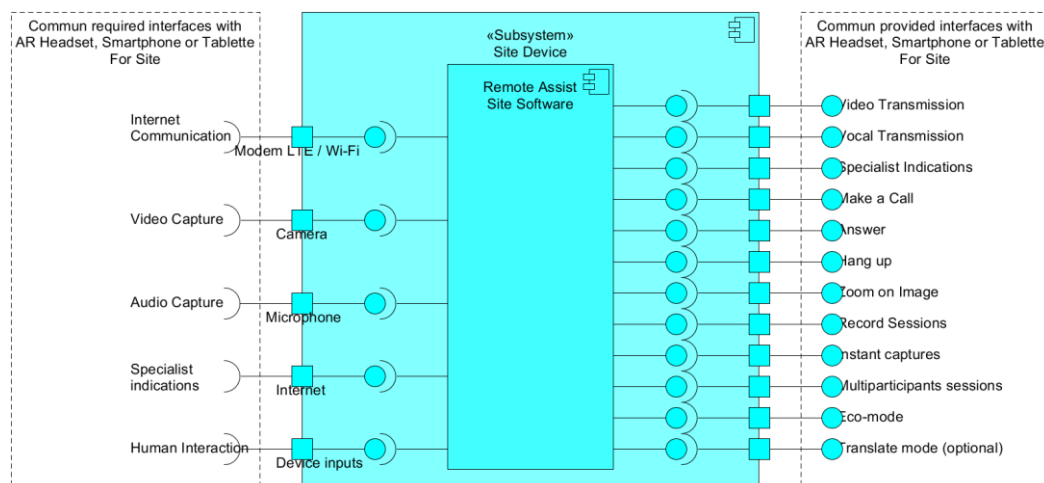


Figure 31 - Layer 2 : Diagramme de composants général du prototype site

Comme pour le prototype spécialiste, ici, les prototypes *requièrent* certaines interfaces afin d'en *fournir* d'autres. À l'aide des différents ports, si différents appareils dotés de mêmes ports le permettent, ils peuvent être utilisés par la personne assistée.

Interfaces requises

Ici, l'appareil étant portable, ce dernier doit disposer d'un récepteur Wi-Fi ou la possibilité de se connecter à un model LTE portable afin de se connecter à Internet pour se connecter à son pair qui se trouve au bureau. Cette connexion permet, au prototype, de recevoir un flux vidéo/audio ou d'autre données du spécialiste.

Une caméra ainsi qu'un microphone servent ici de port pour la capture vidéo et audio de la personne assistée. Ces captures seront retransmises au spécialiste. À l'aide des différents périphériques d'entrées, la personne assistée est capable d'interagir avec le prototype. Ces interactions varient selon l'appareil utilisé. Par exemple, un écran tactile servira de port pour un smartphone alors que la reconnaissance gestuelle sera utilisée pour un HoloLens 2. Ces différents détails seront traités, dans un niveau de profondeur supplémentaire, à la couche 3.

Interfaces fournies

Comme pour le modèle général du prototype spécialiste, le prototype de la personne assistée est capable de transmettre sa capture vidéo et audio par Internet. Cependant, les différentes annotations (libres, textes, formes) en différentes couleurs transmises par Internet, peuvent ne pas être fournies selon le casque de réalité augmentée. Par exemple, l'HoloLens 2 peut facilement créer des annotations à l'aide de sa reconnaissance gestuelle, alors qu'il s'avère plus compliqué pour un RealWear ne disposant que de sa reconnaissance vocale. C'est pourquoi ces dernières fonctions ne sont pas présentes sur la figure de cette couche et seront traitées séparément dans la couche suivante.

Hormis cela, les différentes fonctionnalités d'appels de bases peuvent être fournies, ainsi que les fonctions de zoom, d'enregistrement, de prises de photos, de sessions multi-participants, du mode économique et, optionnellement, le mode de traduction.

4.1.4 Conclusion de la conception de l'architecture générale

Ce chapitre a montré les différentes interfaces générales requises et fournies pour les prototypes spécialistes et sites. Il a permis de voir, à un niveau général, la modularité des prototypes. En effet, pour qu'un nouvel appareil puisse utiliser le prototype réalisé dans ce projet, il devrait être capable d'acquérir les différentes interfaces requises à l'aide de ces propres ports ainsi que d'avoir la capacité d'offrir les interfaces fournies.

Par un simple exemple, un appareil ne pouvant pas se connecter à Internet ne pourrait pas transmettre ou recevoir les données utilisateurs, ce qui causerait un mauvais fonctionnement du prototype.

Ces diagrammes de composants, étant assez généraux, permettent de mettre en évidence, qu'à ce niveau-là, le système d'exploitation d'un appareil importe peu. Par exemple, un smartphone Android ou iOS pourrait être utilisé.

Le reste des couches sera traité dans les chapitres « Conception » propre à chaque prototype, afin d'avoir une suite logique lors de la description des travaux réalisés pour ces derniers. Des détails supplémentaires au niveau des diagrammes de composants seront également apportés.

4.2 Communication audio et vidéo

Sans communication vocale et vidéo, ce proof-of-concept n'aurait pas lieu d'être. En effet, pour qu'un spécialiste soit capable d'assister un ouvrier à distance, il doit au moins pouvoir se faire entendre par celui-ci et recevoir de ce dernier un flux vidéo du site d'intervention.

Des recherches ont été menées afin de trouver la meilleure solution pour ce proof-of-concept. Un des points qui rendait cette tâche compliquée était le fait que la solution devait supporter la plateforme « Windows Universal Platform (UWP) » avec une architecture ARM afin de l'intégrer dans l'HoloLens 2. Par exemple, Agora.IO propose un SDK pour les développeurs, permettant de réaliser des communications audio et vidéo. Malheureusement, selon la documentation d'Agora, ce SDK est compatible avec HoloLens 1 et non HoloLens 2.

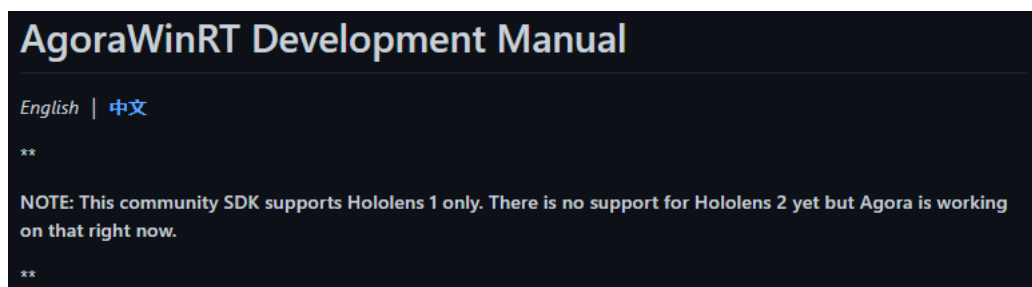


Figure 32 - Information concernant la prise en charge de HoloLens 2 [16]

Ensuite, une autre piste qui avait été trouvée était « MixedReality-SpectatorView » créer par Microsoft. Toutefois, d'après sa documentation^[17], cet outil est plutôt utilisé dans le but de permettre à des personnes tierces de voir ce qu'un HoloLens voit sur un autre appareil, sans réelle communication audio et vidéo à proprement dit.

Une autre piste trouvée est la norme WebRTC. Avec cette dernière il est possible de créer un tel type de communication en temps réel entre pairs. Ce protocole de communication prend en charge la vidéo, la voix ou encore d'autres données génériques entre pairs. Après des recherches, Microsoft offre une solution nommée « MRTK-WebRTC » permettant d'intégrer cette norme dans Unity et dans l'HoloLens 2. Les recherches montrent que, actuellement, cette solution semble être la seule pour le développement de communication sur la plateforme UWP. Par ces faits, cette solution sera choisie.

Dans un futur proche, il est possible que le projet WinRTC^[18], développée par Microsoft pour utiliser la norme WebRTC, pourrait s'intégrer à l'HoloLens 2. Il serait donc bien de garder un œil sur cette solution. Dans les références se trouve un lien vers cette technologie.

¹⁶ GITHUB.COM, *AgoraIO-Community/AgoraUWP*

<https://github.com/AgoraIO-Community/AgoraUWP> (Consulté le 5 août 2021)

¹⁷ MICROSOFT, *Spectator View pour HoloLens et HoloLens 2*

<https://docs.microsoft.com/fr-ch/windows/mixed-reality/develop/platform-capabilities-and-apis/spectator-view> (consulté le 5 août 2021)

¹⁸ GITHUB.COM, *Microsoft/winrtc*

<https://github.com/microsoft/winrtc> (consulté le 5 août 2021)

Ainsi, dans le cadre de ce projet, les différentes connexions pourraient être envisagées selon le modèle présenté ci-après.

4.2.1 Principe de fonctionnement

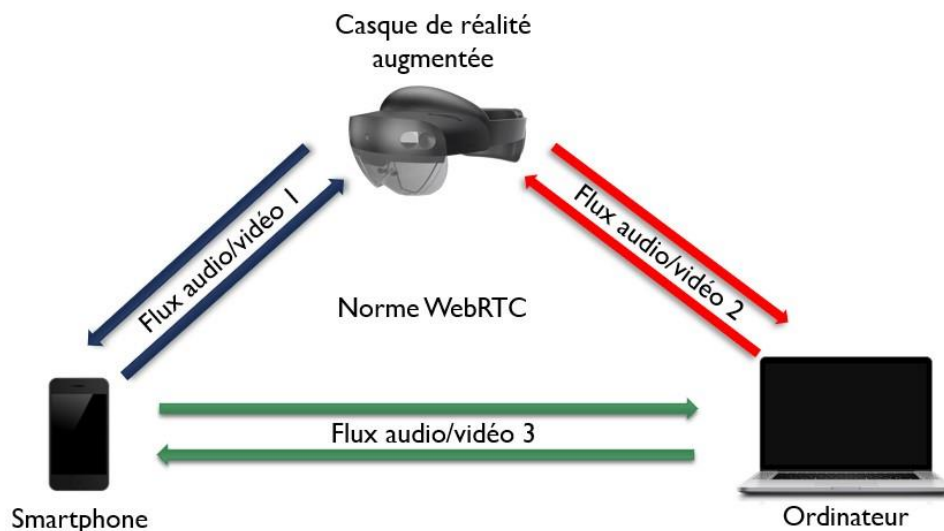


Figure 33 - Connexions WebRTC entre les différents pairs de ce projet

MRTK-WebRTC, déjà mentionné dans sa prise en main au chapitre « Prise en main de MRTK-WebRTC », permet d'intégrer ce protocole dans le prototype. Les deux principaux composants fournis par cette collection sont : « Peer Connection » et « NodeDSS Signaler ».

Peer Connection est un composant de haut niveau pour les fonctionnalités WebRTC d'Unity. C'est le point d'entrée de l'API pour établir une connexion avec un pair distant.

Pour qu'une connexion puisse être établie entre pairs, la norme WebRTC spécifie cette dernière à l'aide de Session Description Protocol (SDP). Selon Wikipedia, SDP est un protocole de communication de description de paramètres d'une session de diffusion en flux^[19]. Il permet de décrire des sessions de communication comme pour l'annonce d'une session, l'invitation à une session ou encore la négociation de paramètres.

Le composant « NodeDss Signaler » permet de spécifier ces paramètres de communication, d'envoyer et de recevoir des messages SDP ainsi que de lire leur contenu pour effectuer certaines actions relatives à leur type. Les types des messages SDP peuvent être :

- Offer : Message d'offre utilisé pour initier une nouvelle session.
- Answer : Message de réponse utilisé pour accepter une offre de session.

Attention, selon la documentation de cette collection, il faut savoir que cette solution fournie par MRTK-WebRTC se prête bien à la réalisation de ce prototype,

¹⁹ WIKIPEDIA, *Session Description Protocol*
https://fr.wikipedia.org/wiki/Session_Description_Protocol (Consulté le 4 août 2021)

mais ne convient pas pour une solution de qualité lors d'une production officielle. En effet, ce composant n'offre pas de sécurité car les communications se font en textes clairs sur http sans aucune authentification. Une personne malveillante pourrait alors compromettre le signal sans trop de difficultés.

Le composant « NodeDss Signaler » utilise un serveur Node.js dans lequel les pairs connectés vérifient constamment si un message est disponible. L'implémentation de ce serveur est disponible sur le lien Github « bengreenier/node-dss^[20] » mis en références. Dans ce projet, le PC du spécialiste sera l'hôte de ce serveur où lui ainsi que ces pairs viendront s'y connecter.

Pour ce faire, Node.js doit être téléchargé et installé sur l'ordinateur depuis nodejs.org^[21]. Ensuite, le répertoire Github précédemment énoncé doit être cloné également sur l'ordinateur à l'aide de la commande suivante :

```
git clone https://github.com/bengreenier/node-dss.git
```

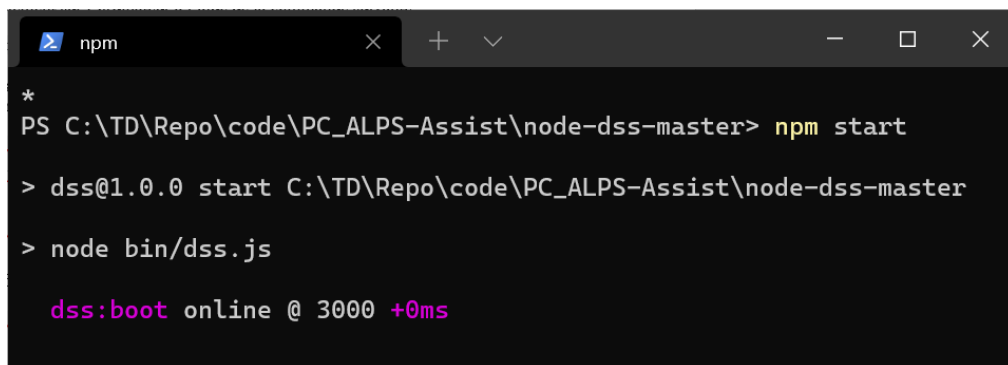
Ensuite, pour le configurer correctement, les commandes suivantes doivent être entrées dans une invite de commande :

```
cd node-dss
:env:DEBUG='dss*'
set DEBUG=dss*
npm install
```

Enfin pour lancer le serveur, la commande suivante doit être lancée :

```
npm start
```

Le serveur est opérationnel si l'invite de commande ressemble à ceci :



```
PS C:\TD\Repo\code\PC_ALPS-Assist\node-dss-master> npm start
> dss@1.0.0 start C:\TD\Repo\code\PC_ALPS-Assist\node-dss-master
> node bin/dss.js
dss:boot online @ 3000 +0ms
```

Figure 34 - Node.js server listening

À ce stade, le serveur est en attente de connexions provenant des différents pairs.

²⁰ GITHUB.COM, *beengrenier/node-dss*
<https://github.com/bengreenier/node-dss> (Consulté le 4 août 2021)

²¹ NODEJS, Download for Windows (x64)
<https://nodejs.org/en/>

Le composant « Peer Connection » étant 1-à-1, il est nécessaire d'en créer un par paire d'appareils selon la figure précédente. Cette méthode fonctionne pour des groupes de trois ou quatre personnes mais, au-delà, des perturbations au niveau de la fluidité de la communication se font ressentir. Pour ce faire, il faut bien faire attention à séparer correctement les ID de chaque appareil. Un appareil doit disposer de n ID pour n Peer Connections. Si cette condition n'est pas respectée, étant donné que chaque pair récupère les messages SDP sur un même serveur, ces derniers peuvent être récupérés par un mauvais destinataire et créerait des erreurs de connexions. Dans le chapitre d'implémentation propre à chaque appareil, cette méthode sera plus détaillée.

Pour contourner cette limite de 1-à-1 provenant du composant « Peer Connection », il faut également désactiver le composant « Webcam Source » qui donne à l'application l'accès à la caméra avant le début d'une négociation (SDP de type « offer ») et le réactiver à la fin de la négociation (SDP type « answer »). Sans cette méthode, l'application ne peut donner de rendu vidéo à tous ses clients et risque également d'engendrer des erreurs.

4.2.2 Conclusion de la conception de la communication audio et vidéo

Pour conclure cette section, il est constatable que la solution de MRTK-WebRTC se prête bien pour ce prototype mais n'est pas fiable pour une production officielle.

En effet, cette solution est plutôt faite pour des communications d'un-à-un et a dû être « contournée » afin d'être fonctionnelle pour des communications de 1-à-plusieurs, ce qui cause certains problèmes de fiabilité comme expliqué auparavant. De plus, le signal étant transmis en texte clair sur http, le signal peut facilement être compromis.

Dans les chapitres sur les recommandations et perspectives, des pistes seront présentées pour remplacer cette solution dans le cas d'une mise en production officielle.

4.3 Annotations à distances

La conception de la communication vidéo et audio étant terminée, il faut, maintenant, résoudre le problème suivant : « Comment un spécialiste peut-il transmettre, depuis son bureau, ses annotations à une personne nécessitant une assistance sur un site d'intervention ? »

Les appareils de ce proof-of-concept disposant d'internet, il suffirait naturellement de trouver un protocole de communication adapté à la situation. L'idée serait d'envoyer, depuis l'appareil du spécialiste, un certain nombre de données importantes permettant d'effectuer des actions à distances sur un appareil pair.

Aujourd'hui, plusieurs protocoles existent. Parmi ceux-là peuvent être retrouvés :

- TCP/IP
- UDP
- REST
- RPC

Parmi ces protocoles, UDP et TCP/IP ont été considérés comme les plus potentiels pour les raisons qui suivent :

Le protocole « Remote Procedure Call (RPC) » de haut-niveau est plutôt utile pour exécuter du code à distance en faisant appel à des procédures ou des fonctions en spécifiant un certain nombre de paramètres sur une machine externe. Or, ici, les paramètres peuvent être variables et complexes ce qui peut demander un plus grand travail que les autres protocoles. Le protocole REST quant à lui, est plutôt utilisé pour créer des services web à l'aide d'URL (Uniform Resource Locator) et d'URI (Uniform Resource Identifier), qui n'est pas le cas de ce proof-of-concept.

4.3.1 UDP VS TCP/IP

En effectuant des recherches, il a été constaté que chacun de ces protocoles possède ses propres forces et faiblesses. Ci-après se trouve un tableau mettant en évidence les différences majeures entre ces deux protocoles de communication. Dans les références, il est possible de retrouver les sources desquelles proviennent les informations à ce sujet.

TCP VS UDP		
	TCP	UDP
Connexion	Requiert qu'une connexion soit établie.	Ne requiert pas de connexion établie.
Garantie de livraison des données	Garantie absolue que les données transférées restent intactes et arrive dans le même ordre qu'envoyé.	Il n'y a aucune garantie que les messages ou les paquets envoyés sont livrés à la destination.
Retransmission de données	Retransmission de paquets perdus possible.	Pas de transmission de paquets perdus.
Methode de transfert	Les messages sont lus sous forme de flux d'octets.	Paquets UDP avec des limites définies, envoyés individuellement
Ordonnance des paquets	Les paquets sont organisés dans l'ordre spécifiés.	Pas d'ordre inhérent car les paquets sont indépendants.
Vitesse	TCP est plus lent que UDP	UDP est plus rapide que TCP car il n'y a pas de récupération en cas d'erreur.

Figure 35 - Comparaison entre TCP et UDP

Sources : SAWAKINOME, TCP VS UDP, <https://fr.sawakinome.com/articles/technology/tcp-vs-udp.html>
LIFESIZE, TCP vs. UDP : What's the difference : <https://www.lifesize.com/en/blog/tcp-vs-udp/>

Bien que TCP soit plus lent que UDP, il possède des avantages primordiaux pour ce proof-of-concept. En effet, ici, la sécurité des paquets envoyés est un facteur clé. Si des données sont perdues en cours d'envoi, il se peut que les fonctionnalités à distance ne soient pas correctement effectuées. La garantie que les paquets soient correctement transmis au destinataire et dans le bon ordre, fait pencher la balance du côté de TCP. De plus, il est facilement réalisable de transformer des données en flux d'octets à l'aide de la sérialisation. De plus, contrairement à UDP, il est garanti que les messages sont envoyés uniquement si une connexion avec un serveur est établie. Par ces faits, le protocole TCP sera utilisé dans le cadre de ce projet.

4.3.2 Utilisation de TCP/IP

Cette section détaillera la méthode utilisée pour transmettre des informations depuis les appareils du spécialiste (PC et smartphone) en destination de l'appareil de la personne assistée (HoloLens 2).

Connexion entre les Clients TCP et Serveurs TCP

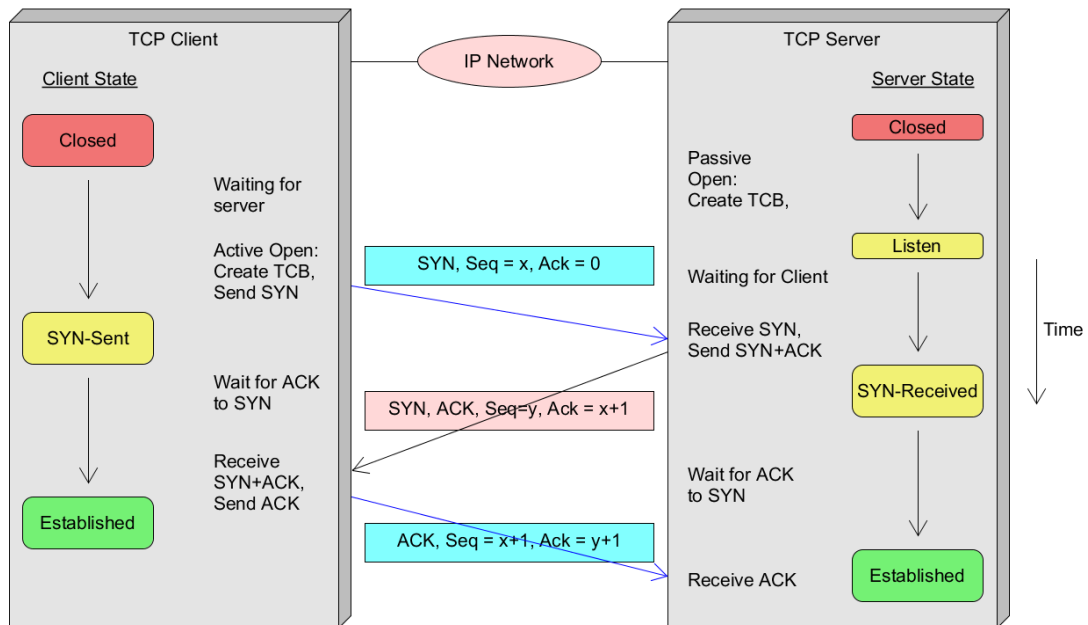


Figure 36 - Etablissement d'une connexion entre un client TCP et un Serveur TCP

Source : GABIOUD, Dominique. *Systèmes d'informations / TCP/IP*, 2019-2020. Connexion TCP p.37
TCPIPGUIDE.COM, *TCP Connection establishment Sequence Number Synchronization and parameter Exchange*. http://www.tcpiptide.com/free/t_TCPConnectionEstablishmentSequenceNumberSynchroniz-2.htm (consulté le 5 août 2021)

Les divers clients TCP (spécialistes) de ce proof-of-concept se connecteront au serveur TCP (Personne assistée, HoloLens 2) selon le schéma ci-dessus. Tout d'abord, les clients et serveurs sont fermés. Au lancement de l'application, le serveur crée un Transmission Control Block (TCB) contenant les divers éléments servant à la connexion, tels que les buffers de réceptions et d'envois. Dès lors, il se met en écoute et attend qu'un client tente de se connecter à lui. Le serveur accepte alors la connexion du Client et la connexion est établie. Ayant possiblement plusieurs clients TCP, une liste de ces derniers doit être gérée dans le script du TCP Server.

Envoi de données dans le flux entre les clients et le serveur

Une fois la connexion établie, les clients et le serveur peuvent s'envoyer des messages dans le buffer créé au préalable et les lire. Au départ, il avait été pensé qu'envoyer uniquement des chiffres entiers (*Integers*) suffirait à répondre aux besoins de ce prototype. Cependant, la conception de cette méthode a vite montré des limites car, hormis ces chiffres servant de commandes, aucune donnée ne pouvait être transmise. Pour résoudre ce problème, il a été décidé qu'envoyer des messages JavaScript Objet Notation (JSON) sérialisés (sous forme de chaîne de caractères) serait la solution car ils permettent d'envoyer plusieurs données en plus des fameuses commandes représentées par les chiffres entiers.

Messages JSON dans ce proof-of-concept

Selon json.org, « JSON est un format léger d'échange de données. Il est facile à lire et à écrire pour les humains. Il est facile à analyser et à générer pour les machines. Il est un format texte totalement indépendant du langage.^[22] »

Les données contenues dans les messages JSON échangés seront les suivantes :

- Les numéros de commandes à distances représentés par des chiffres entiers (int).
- Des coordonnées dans les axes X, Y et Z servant à la position des annotations à distances représentés par des chiffres à virgules flottantes (float) stockés dans un tableau de ce type.
- Les annotations textuelles tapées au clavier représenté par des chaînes de caractères (string).
- La largeur de la ligne de dessin libre représenté par un chiffre à virgule flottante (float).

Toujours selon json.org, ci-dessous se trouve une illustration représentant la formation d'un message JSON standard.

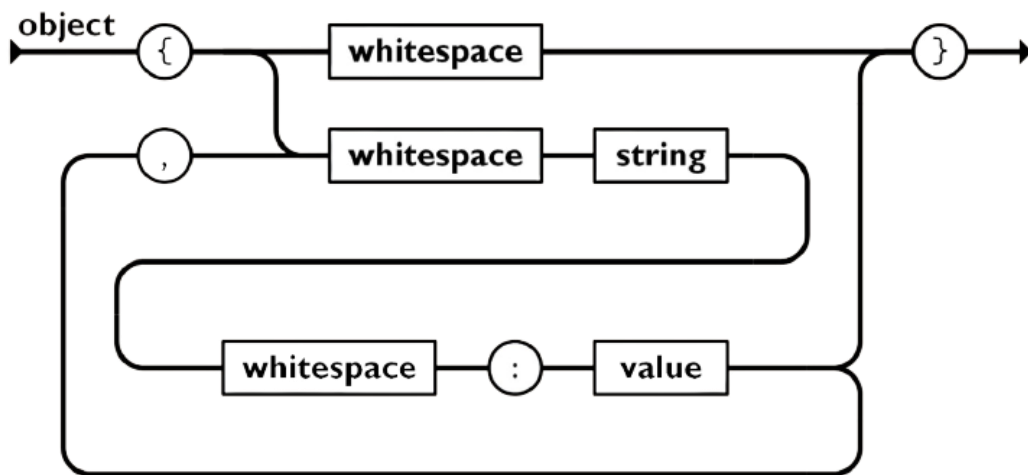


Figure 37 - Format d'un message JSON

Un des grands avantages de cette méthode est qu'il est facilement envisageable de rajouter des données si des nouvelles fonctionnalités venait à être ajoutées.

²² JSON.ORG, *Introducin JSON*

<https://www.json.org/json-en.html> (consulté le 5 août 2021)

De la création du message JSON à l'exécution d'une action à distance

Cette partie décrit comment, de façon générale, un message est créé par un spécialiste, transmis par la connexion TCP et reçu par la personne nécessitant une assistance afin de recevoir des annotations ou informations de la part du spécialiste.

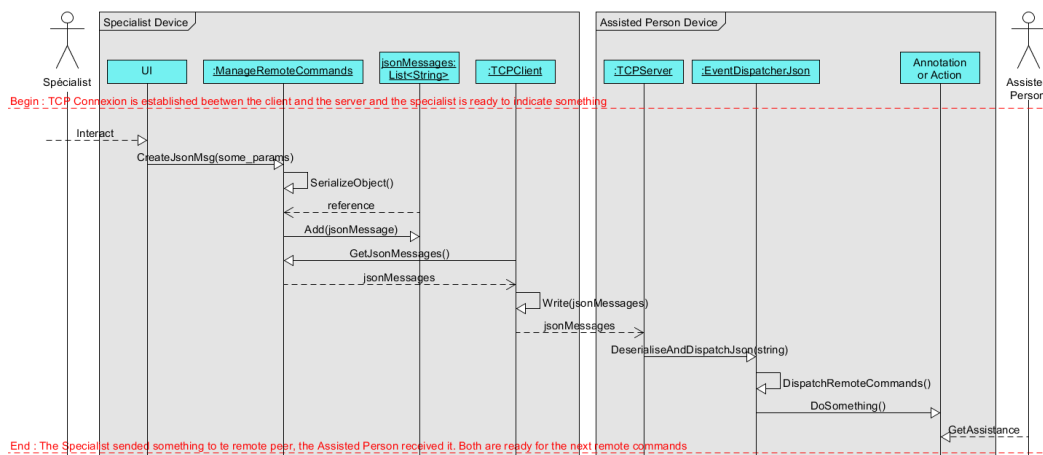


Figure 38 - Diagramme de séquence : De la création d'un message à l'exécution d'une action à distance

La figure ci-dessus présente un diagramme de séquence général montrant le cheminement entre une interaction, venant d'un spécialiste, jusqu'à la réception de ce dernier par la personne assistée.

Le spécialiste peut interagir avec l'interface utilisateur de l'application à l'aide de son clavier et de sa souris pour un PC et de l'écran tactile pour un smartphone. En fonction de l'action qu'il a effectué, un message JSON est construit selon le modèle présenté dans le chapitre précédent. Celui-ci est ensuite sérialisé, signifiant qu'il est transformé en chaîne de caractère avant d'être converti en tableau de bytes, et envoyé dans le flux de données entre le client TCP et le serveur TCP. A la réception du message par le serveur, celui-ci reconvertit les bytes reçus en chaînes de caractères et récupère les données en fonction de leurs types avant la sérialisation. Dès lors, le message peut être distribué en fonction de ses valeurs. L'idée est d'avoir une donnée représentant un numéro de commande à distance et de la distribuer dans un « Switch case » qui peut ainsi effectuer des actions définies. Cette partie sera détaillée dans la partie implémentation du prototype. Enfin, de cette manière, la personne assistée peut voir les différentes actions effectuées à distance.

4.3.3 Conclusion de la conception des annotations à distances

Pour conclure, les annotations du spécialiste, chez la personne assistée, se feront à l'aide d'outils pilotables à distance grâce aux données envoyées dans un flux créé par une connexion TCP/IP entre les clients et les serveurs. TCP/IP semblait être plus approprié que UDP pour ne pas perdre certaines données importantes durant la transmission. Les données seront transmises sous le format JSON et seront distribuées dans l'appareil distant en fonction des actions à effectuer.

5. Implémentation générale

Avant de passer à l'implémentation individuelle de chaque prototype (PC, HL2, Smartphone), certains éléments, communs à ces derniers, seront expliqués dans ce chapitre. Ceci permet de ne pas répéter certains points et d'alléger ce rapport.

5.1 Projets Unity

Tout d'abord, ces trois prototypes ont été réalisés sur la version d'Unity 2019.4.26f1. Les application PC et smartphone ont été réalisés avec un environnement 2D alors que celle pour l'HoloLens 2 a été réalisée dans un environnement 3D étant donné que la personne assistée a besoin de se déplacer dans ces trois axes X, Y et Z.

5.1.1 Structure des projets

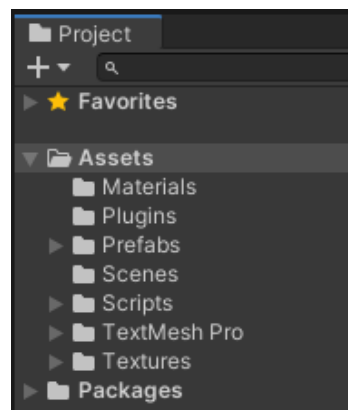


Figure 39 - Structure du projet Unity

Chaque projet Unity possède un dossier « Assets ». Il est le principal dossier dans le projet Unity et contient les différents assets nécessaires à la réalisation de celui-ci. Un asset est une ressource utilisable dans l'application. Il peut être un fichier provenant de l'extérieur de Unity, tel qu'un fichier audio, une image ou un modèle 3D.

Comme le montre la figure ci-dessus, plusieurs dossiers ont été créés afin de séparer les ressources en fonctions de leurs types. Parmi ces dossiers communs aux trois projets existants se trouvent des dossiers pour : les matériaux des objets, les plugins offrant des fonctionnalités ne provenant pas de Unity (p. ex. : Les fonctions de sérialisation JSON), des objets dits « préfabriqués », les scènes, les différents scripts, les ressources de textes et les différentes textures. Il est possible de créer des sous-dossiers afin de faire des séparations de ressources plus précises.

Le projet contient également un dossier « Package » qui regroupe les diverses librairies intégrées au projet Unity tel que MRTK-WebRTC.

5.1.2 Fenêtre de Hiérarchie Unity

Les projets propres à chaque prototype disposent d'une fenêtre « Hiérarchie ». Cette dernière montre tous les objets instanciés dans une scène. Il est possible d'en ajouter ou d'en supprimer durant le développement d'une application. Il est également possible de créer des sous-groupes d'objets groupés dans un objet supérieur. L'objet supérieur est appelé « Parent » et les sous-objets sont appelés les « Enfants ».

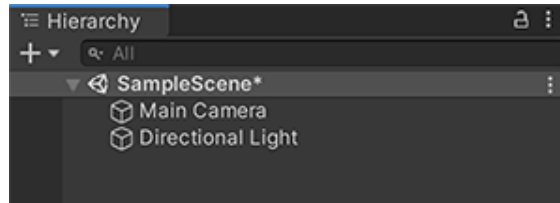


Figure 40 - Hiérarchie par défaut lors de la création d'un projet.

Par défaut, les projets possèdent une lumière directionnelle qui peut être considérée comme un soleil. Celle-ci éclaire alors les objets instanciés dans la scène. Les projets possèdent aussi une caméra qui est un dispositif à travers lequel le joueur voit le monde développé.

Dans les chapitres d'implémentations propres à chaque prototype, leur hiérarchie sera détaillée.

5.1.3 Fenêtre Scène Unity

Chaque projet dispose d'une scène avec laquelle les développeurs travaillent dans Unity. C'est dans cette dernière que les différents Assets seront instanciés.

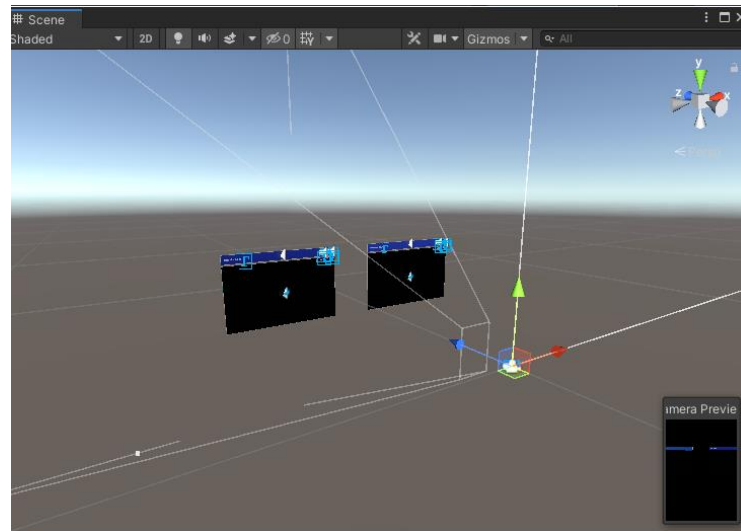


Figure 41 - Exemple de scène Unity

Les éléments présents dans cette scène sont les objets situés dans la fenêtre hiérarchie. L'utilisateur peut effectuer des modifications sur ces objets directement depuis la scène.

5.1.4 Fenêtre « Game » Unity

Les projets disposent également d'une fenêtre « Game » qui une fenêtre dans laquelle la vue de l'application est rendue à partir de la position d'une ou plusieurs caméras. Ceci permet également de voir à quoi ressemblera le jeu ou l'application final.

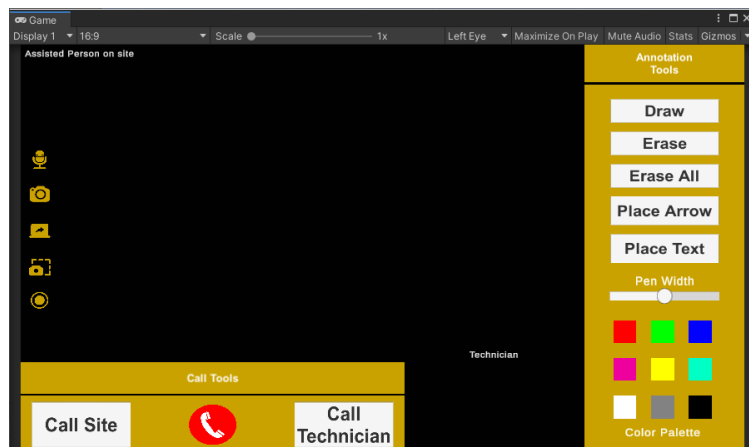


Figure 42 - Exemple de fenêtre de jeu Unity

En effet, depuis cette fenêtre, il est possible de lancer l'application et de la tester comme si elle était finie. Ceci permet également de déboguer l'application.

5.1.5 Fenêtre « Inspector »

La fenêtre « Inspector » permet d'éditer certaines propriétés des différents objets instanciés. C'est également à travers cette fenêtre que des scripts peuvent être ajoutés à ces derniers. Ci-dessous se trouve un exemple avec les différentes propriétés de l'objet « Directional Light ».

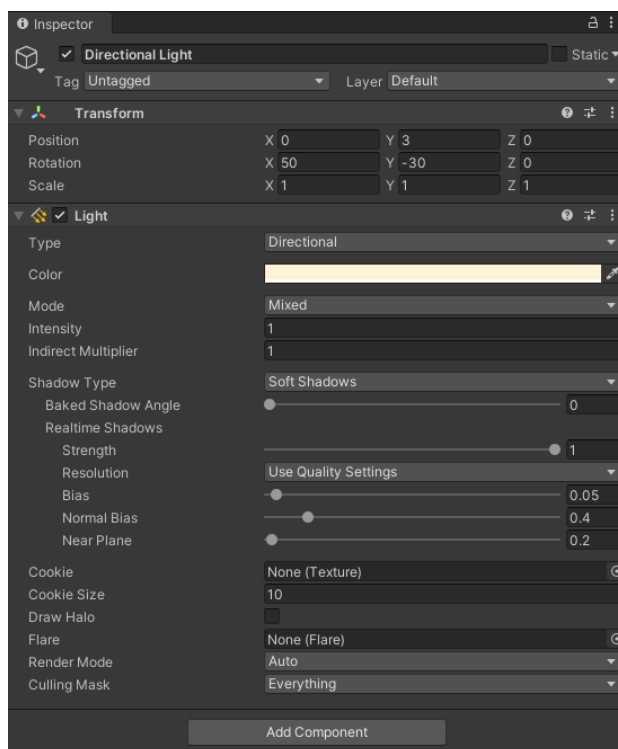


Figure 43 - Fenêtre Inspector

5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC

Pour pouvoir effectuer des appels multi-participants avec la librairie MRTK-WebRTC, il faut mettre en lumière certains points et modifier le tutoriel en référence¹⁵.

5.2.1 Fonctionnement exclusif de la caméra locale et du microphone local

Tout d'abord, il faut veiller à ce qu'uniquement un seul composant « Local Video Player » et « Microphone Source » ne soit instancié dans la scène. En effet, si deux paires de composants sont créés, l'application tentera d'accéder deux fois à la caméra et au microphone, ce qui causera une erreur et fera « crash » l'application. De ce fait, chaque « Peer Connection » aura accès au même microphone et caméra comme suit :

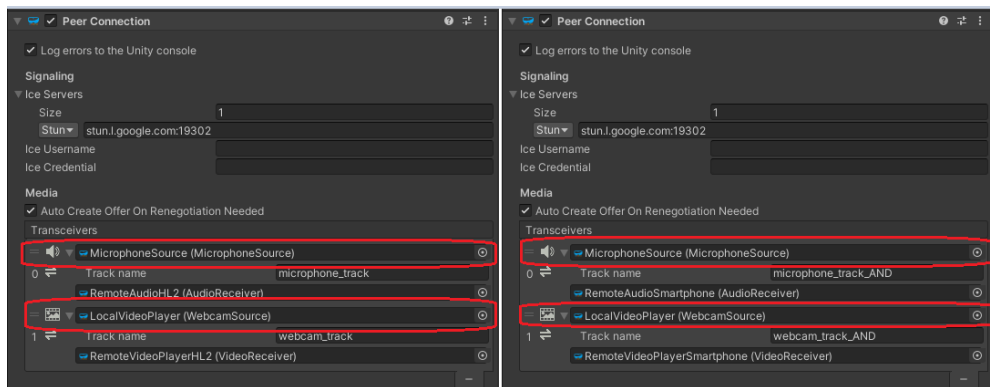


Figure 44 - Composants Peer Connection pour se connecter aux appareils pairs (HoloLens 2 à gauche et Smartphone à droite)

Bien entendu, pour les rendus vidéo et audio distants, il faut attribuer spécifiquement aux trancheurs les composants appropriés comme le montre la figure ci-dessus.

5.2.2 Attribution de IDs spécifiques pour chaque paire de connexion

Dans les composants « NodeDssSignaler » propres à chaque appareil, les ID locaux et distants doivent respecter les modèles ci-dessous.

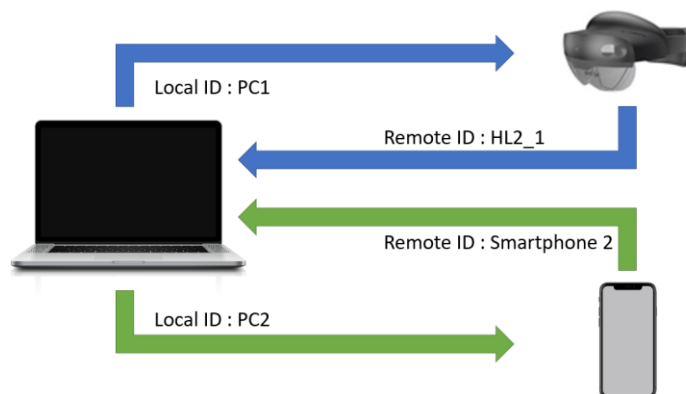


Figure 45 - Schéma des ID locaux et distants pour les « Signalers » du prototype PC

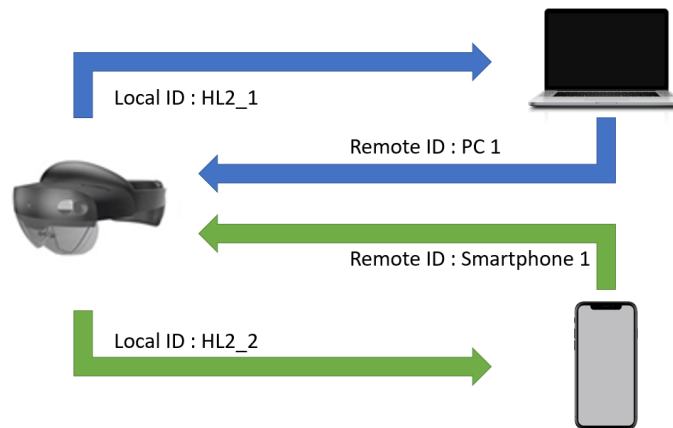


Figure 46 - Schéma des ID locaux et distants pour les « Signalers » du prototype HL2

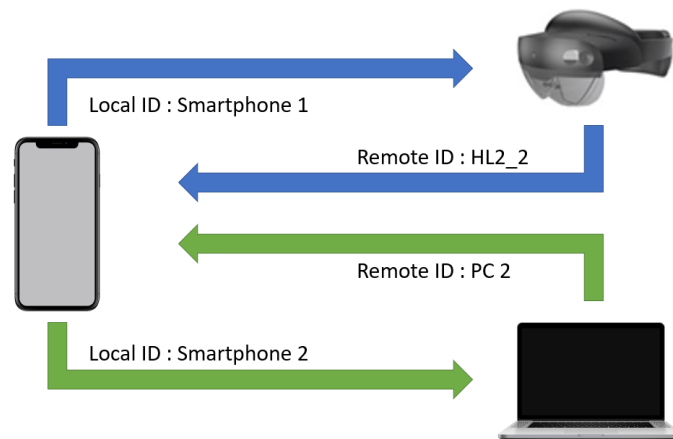


Figure 47 - Schéma des ID locaux et distants pour les « Signalers » du prototype smartphone

Si cela n'est pas respecté, les messages dans le serveur seront mal distribués et causeront donc un mauvais comportement lors des communications.

5.2.3 Adresse http du serveur Node.js

Lorsque des communications se font entre plusieurs appareils, il faut veiller à ce que le champ de l'adresse http dans les composants « Signaler » contienne l'adresse IP de l'appareil servant d'hôte au serveur node.js. Par défaut, cette valeur vaut « 127.0.0.1 » (localhost) et ne permet pas aux appareils externes de se connecter au serveur local.

5.2.4 Désactivation de la caméra locale avant de créer un appel

Il a été remarqué que pour que les flux vidéo se transmettent correctement dans un appel multi-participants, il fallait désactiver la caméra source de chaque appareil avant la création d'une offre d'appel. Pour ce faire, il fallait modifier le code de la librairie MRTK-WebRTC selon le diagramme de séquence ci-dessous.

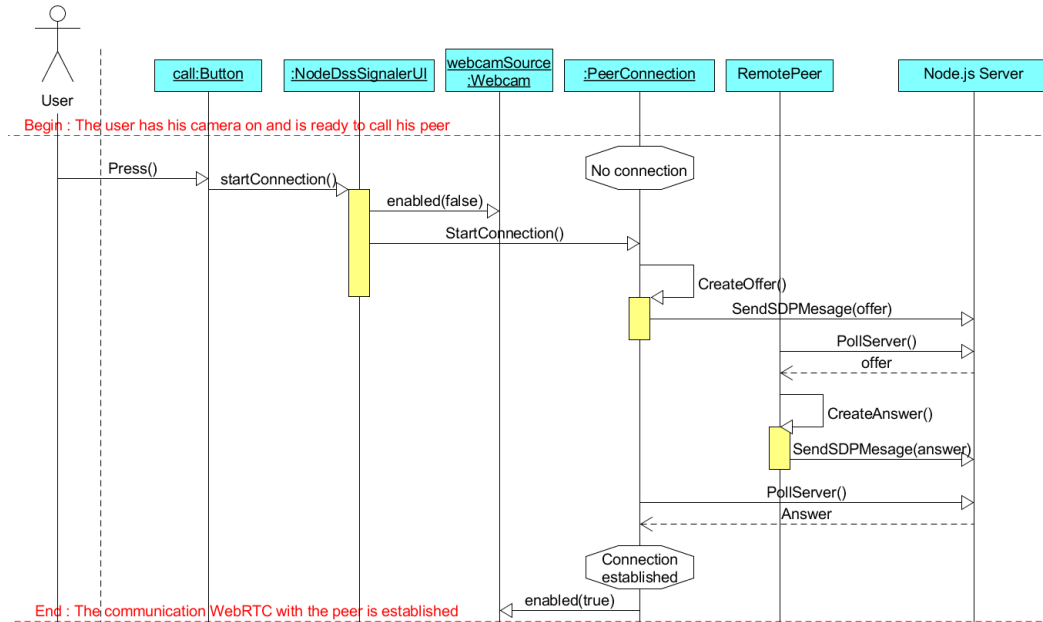


Figure 48 - Diagramme de séquence : Contrôle de la caméra source pour un appel WebRTC

Lorsque l'utilisateur appuie sur un bouton d'appel de l'interface utilisateur (UI), ce dernier appelle une méthode « StartConnection() » servant d'interface entre l'utilisateur et le composant « PeerConnection ». Cette méthode initialise certaines variables de connexion et appelle à son tour une autre méthode « StartConnection() » du composant « PeerConnection ». Cependant, pour résoudre le problème mentionné dans ce chapitre, il faut désactiver la caméra source avant cet appel comme le montre le diagramme de séquence. Après les différentes transactions de messages SDP (offres et réponses) avec le serveur, la connexion est établie et la caméra peut être à nouveau réactivée.

5.3 Installation des paquets MRTK

Afin d'installer les différents paquets de la collection « MixedRealityToolkit », Microsoft a mis au point un logiciel simplifiant cette tâche. Celui-ci se nomme « Mixed Reality Feature Tool^[23] » et se trouve en référence avec les liens permettant de travailler avec cet outil correctement.

²³ MICROSOFT DOCS, *Welcome to the Mixed Reality Feature Tool*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/welcome-to-mr-feature-tool> (consulté le 13 août 2021)

6. Prototype PC

Ce chapitre traite du développement du prototype se déployant sur un PC pour un spécialiste. Il sera séparé en quatre parties : Conception, Implémentation, Déploiement et Validation.

6.1 Conception

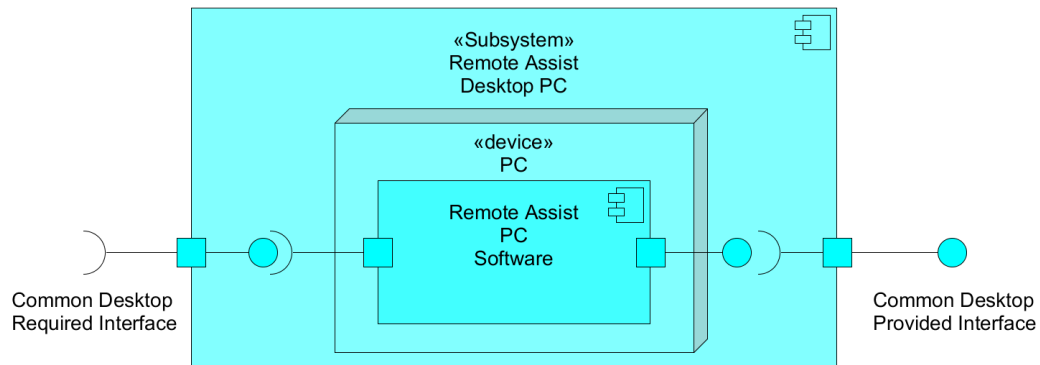


Figure 49 - Layer 3 : Desktop PC

Cette section traitera spécialement de l'application prototype « Remote Assist PC Software » utilisée par le spécialiste sur un PC. Cette dernière requiert et offre les différentes interfaces communes discutées au chapitre « 4.1.2 Layer 2 – Diagramme de composants général (Spécialiste) ».

6.1.1 Définition de l'architecture du système

Cette partie décrit l'architecture de l'application prototype spécialiste. Les outils et bibliothèques de développement servant à la réalisation de ce projet seront présentés avec certains de leurs composants permettant de fournir les différentes interfaces.

Layer 4 – Remote Assist PC Software

Tout d'abord, la figure suivante présente le sous-système représentant l'application PC. Cette dernière a été réalisée à l'aide de Unity 2D 2019.4.26.f1 qui dispose encore d'un support à long terme.

Ce sous-système requiert et fournit les différentes interfaces discutées plus tôt à travers ces différents ports. Ici, spécifiquement, les ports servant à la capture d'interactions utilisateurs sont un clavier et une souris branchés, par exemple, en USB. Ensuite, la capture audio et vidéo ainsi que la connexion Internet se font comme expliqué plus tôt.

Les composants internes seront maintenant expliqués afin de comprendre comment les interfaces fournies sont créées.

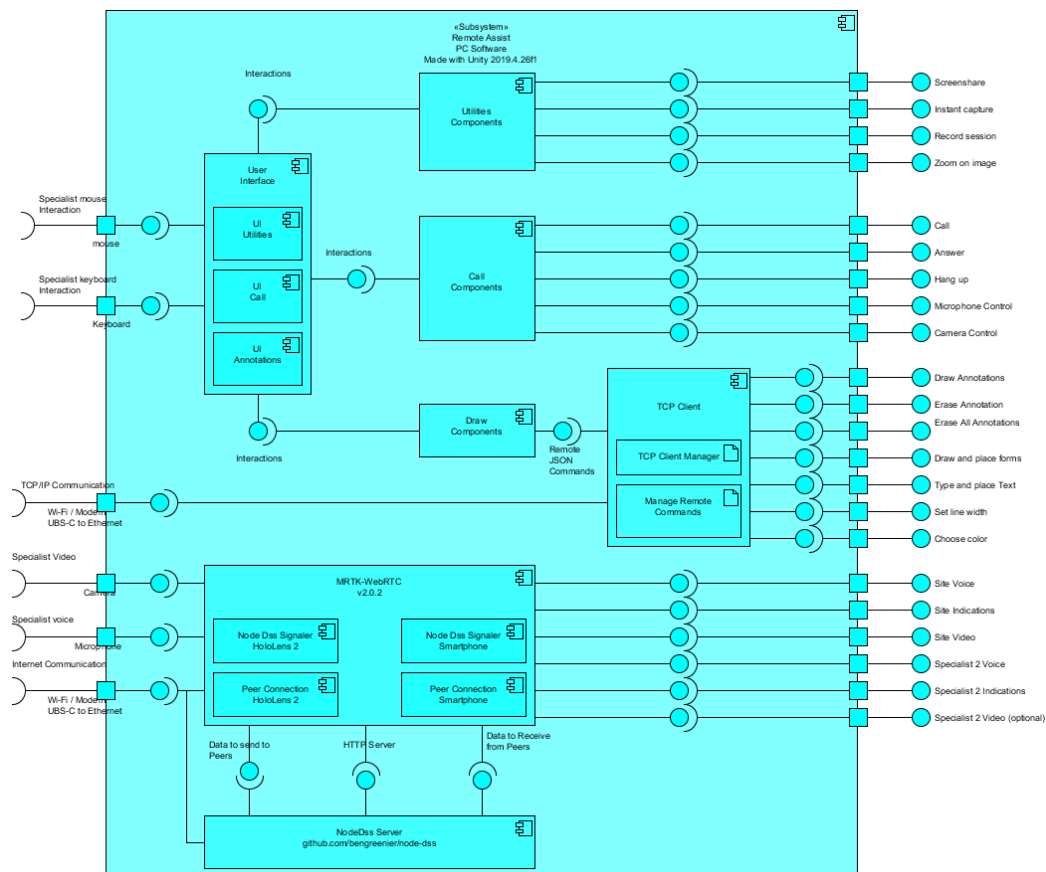


Figure 50 - Layer 4 : Architecture du Software PC Spécialiste

UI et fonctions locales

Parmi les différents composants internes de la figure ci-dessus se trouve l'interface utilisateur (UI). C'est à travers cette interface que l'utilisateur peut interagir, par exemple à l'aide de boutons, avec l'application au moyen de son clavier et de sa souris. Cette interface peut être séparée en trois composants. Une UI pour gérer les actions d'appels, une UI pour gérer les différentes fonctions servant aux annotations à distance et une UI pour certaines fonctionnalités utiles.

Une fois les interactions effectuées, les composants, servant aux appels, aux annotations et aux fonctions utilitaires, transmettent les informations à l'application permettant de réaliser les différentes fonctions nécessaires.

« Utilities Components » fournit diverses interfaces utiles au bon déroulement d'une session d'assistance à distance. Ces diverses fonctions sont : le partage d'écran du PC, la possibilité de prendre des photos instantanées, la possibilité d'enregistrer une séance ainsi que la possibilité de faire des Zooms sur l'image vidéo.

« Call Components » fournit les fonctions d'appels, de réponses, de raccroche et offre le contrôle du microphone et de la caméra. Ces fonctionnalités sont liées à la librairie MRTK-WebRTC, qui sera présentée plus tard.

Envoi d'annotations et de commandes à distances

Pour réaliser les annotations à distance ainsi que modifier certaines options de dessin, les données sont transmises en JSON dans un flux TCP/IP reliant le prototype du spécialiste et le prototype de la personne assistée. Les prototypes spécialistes jouent le rôle de clients, alors que l'appareil sur le site joue le rôle de serveur. Cette technologie sera présentée en détail dans la phase de l'implémentation. De cette manière, le spécialiste contrôle des outils de dessin directement dans l'environnement de la personne sur le site. Ces derniers peuvent servir à réaliser des annotations libres, des annotations textuelles et des annotations à l'aide de différentes formes. En outre, le spécialiste peut également, à distance, effacer les annotations, choisir la couleur de ces dernières ainsi que d'ajuster la taille de la ligne de dessin.

Communication audio et vidéo

La prochaine partie importante du prototype est la connexion lui permettant de communiquer avec ses pairs, autrement dit, avec la personne assistée et éventuellement avec un deuxième spécialiste. Pour ce faire, MRTK-WebRTC v2.0.2 a été utilisé.

Cette librairie fournit des outils de communication entre pairs. A l'aide d'un composant « Peer Connection » servant à se connecter à un pair, ainsi que d'un composant « Signaler » permettant de transmettre les données de connexions, une connexion peut être établie avec un serveur « Node-Dss » fonctionnant sur le PC lui-même.

Malheureusement, cette librairie a été réalisée pour des connexions « one-to-one ». Toutefois, des modifications ont été faites dans le code fourni par la librairie afin de contourner cette limite empêchant de réaliser des sessions multi-participants. De cette manière, le spécialiste peut envoyer et recevoir le flux vidéo et audio à deux participants. Ces modifications seront présentées dans la partie traitant de l'implémentation.

L'idée est d'avoir un composant « Peer Connection » et « Signaler » pour chaque pair muni d'un ID différent. Dans le cadre de ce projet, cette méthode est fonctionnelle. Mais dans le cas d'une production officielle, une alternative devrait être utilisée car de la latence est ressentie dans le flux vidéo lorsqu'un troisième utilisateur rejoint la session. De plus, la communication n'est pas sécurisée. En effet, les communications se transmettent en texte clair sur http avec aucune authentification. Il serait alors facile pour une personne malveillante de contourner cette connexion et compromettre le signal. Des pistes ont été trouvées et seront présentées dans le chapitre sur les recommandations.

6.1.2 Définition de l'interface utilisateur

L'interface utilisateur est la passerelle entre les fonctionnalités de l'application et l'utilisateur. Cette dernière devrait être simple et intuitive afin qu'un maximum d'utilisateurs se sentent à l'aise avec celle-ci.

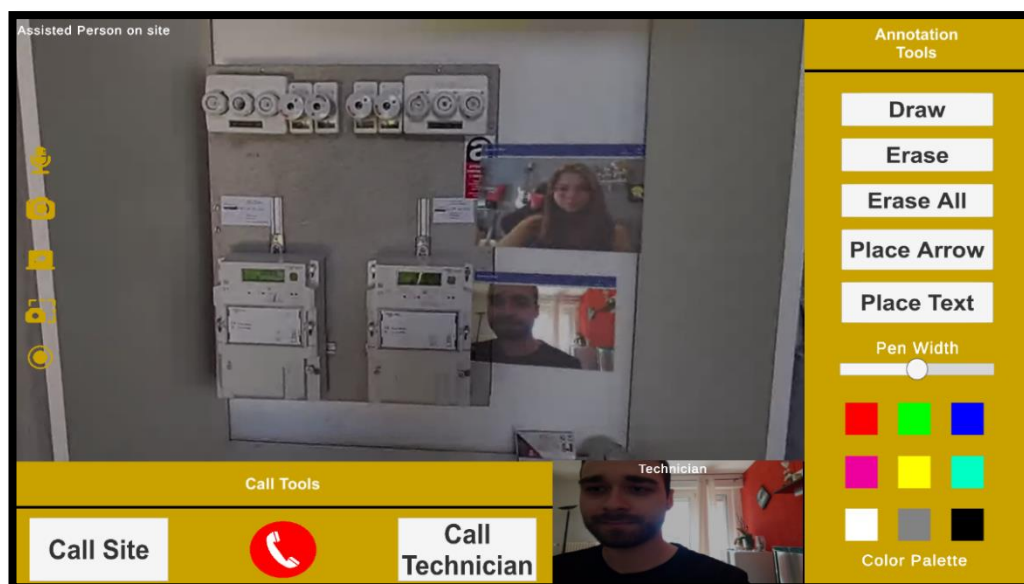


Figure 51 - Interface utilisateur PC du spécialiste

Ci-dessus se trouve l'interface utilisateur du spécialiste utilisant un PC. Les cinq parties suivantes servent de base pour cette dernière : la partie annotation, la partie appel, la partie « utilitaires », la vidéo de la personne sur le site ainsi que la vidéo d'une personne tierce.

L'image principale est un plan recevant un flux vidéo provenant de l'appareil de la personne assistée. Celle-ci est, évidemment, mise en évidence dans l'UI au format 16:9 afin d'avoir une bonne qualité et une facilité de lecture. C'est dans cette fenêtre que le spécialiste interagira avec des éléments virtuels dans l'environnement de la personne assistée.

En bas à droite de l'application se trouve un deuxième lecteur vidéo. Celui-ci est de plus petite taille que celui de la personne assistée, mais reste au format 16:9. Il reçoit le flux vidéo venant d'une personne tierce, dont par exemple, un technicien.

Sur la droite de la figure se trouvent les différents boutons permettant d'interagir avec les annotations. Le bouton « Draw » permettra d'effectuer des annotations libres, le bouton « Erase » permettra d'effacer un élément, le bouton « Erase All » permettra d'effacer tous les éléments, le bouton « Place Arrow » permettra de placer une flèche et le bouton « Place Text » permettra d'écrire un texte au clavier afin d'en faire une annotation. Toutes ces annotations seront alors envoyées et contrôlables dans l'environnement de la personne sur le site. En outre, un « Slider » permettra d'ajuster la taille de la ligne de dessin. Enfin, une palette de couleur sera à disposition pour changer la couleur des différentes annotations. Les outils distants seraient contrôlés à l'aide des mouvements de la souris et leur niveau de profondeur serait contrôlable via la molette de cette dernière.

En bas à gauche se trouvent les boutons servant à l'appel. Grâce à ces derniers, le spécialiste peut appeler la personne sur le site ou un technicien en cas de besoin. Le bouton rouge central permettrait au spécialiste de raccrocher et de mettre un terme à la communication.

A gauche de la fenêtre se trouvent quelques éléments utilitaires, tels que les capacités à couper le microphone et la caméra ainsi que de les réactiver. Un bouton permettant d'activer le partage d'écran, un bouton permettant de faire une capture d'image ainsi qu'un bouton permettant d'enregistrer une communication seront également disponibles.

6.1.3 Conclusion de la conception PC

Avec les différents éléments de conception mis en place, il est maintenant possible de commencer l'implémentation. La structure du projet ainsi que la définition de l'interface utilisateur servent de guide pour implémenter correctement les différentes fonctionnalités.

La modularité a été également mise en avant. Lorsqu'un appareil dispose des différents ports nécessaires à l'implémentation des interfaces, il peut être utilisé pour ce prototype. Le système d'exploitation de l'appareil de bureau importe peu pour cette application car ils n'ont aucun effet sur les composants implémentés. Il doit toutefois être compatible avec les bibliothèques utilisées.

Finalement, un smartphone ou une tablette ne disposant pas de tous les ports nécessaires à la réalisation de ce projet, une autre application doit être conçue pour ces plateformes afin que les différentes fonctionnalités puissent être utilisées à l'aide de l'écran tactile et non pas d'un clavier et d'une souris.

6.2 Implémentation

Cette section traitera de l'implémentation du prototype PC. Elle présentera les paquets utilisés, la hiérarchie des composants du projets de façon générale et détaillera certains principes implémentés.

6.2.1 Paquets et bibliothèques externes



Figure 52 - Paquets présents dans le prototype PC

Ce prototype PC est le plus léger en termes de bibliothèques. En effet, seul le paquet MixedReality-WebRTC 2.0.2, pour la communication audio et vidéo en temps réel, a été nécessaire d'importer dans ce projet Unity. Les autres paquets sont, par défaut, intégrés à la création de ce dernier. Toutefois, il faut veiller à importer les différentes ressources de TextMeshPro pour pouvoir travailler avec des composants de textes performants.

Afin de travailler avec des objets JSON, il a fallu installer dans Visual Studio la bibliothèque Newtonsoft.Json via « Manage NuGet Packages ».

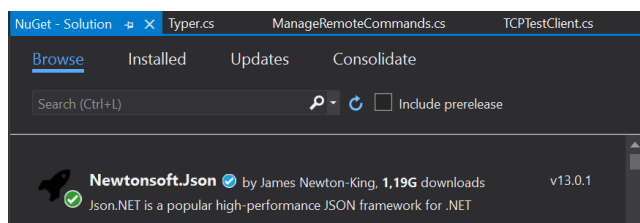


Figure 53 - Paquet utilisé pour travailler avec les objets Json : Newtonsoft.Json

De plus, les références ont dû être ajoutées sur Visual studio et sur Unity.

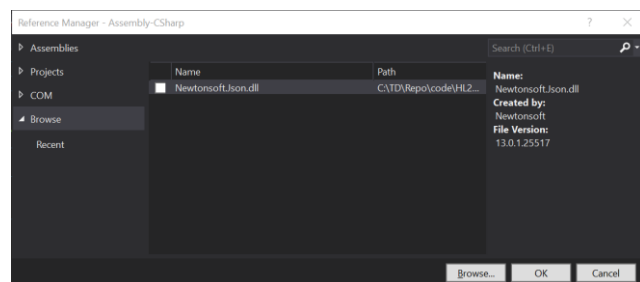


Figure 54 - Ajout des références sur Visual Studio

Dans Unity, il faut veiller à utiliser la version netstandard2.0 du plugin et à l'ajouter au dossier « assets » du projet.

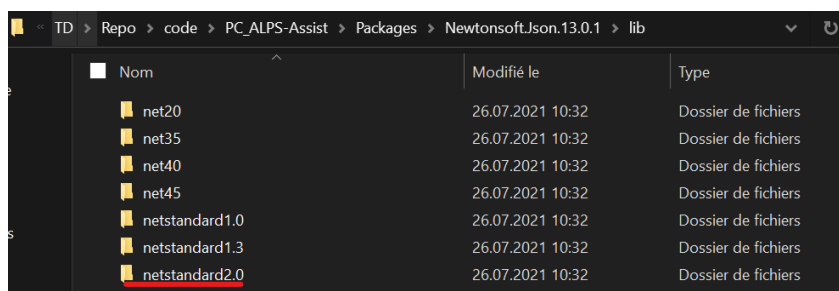


Figure 55 - Emplacement du plugin

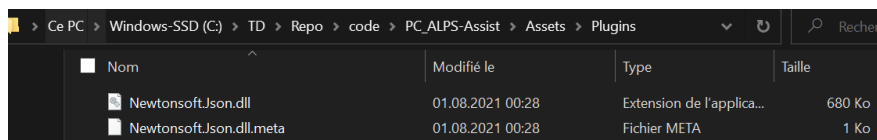


Figure 56 - Ajout du plugin Newtonsoft.json dans les assets du projet Unity

6.2.2 Hiérarchie Unity

La hiérarchie ci-dessous présente les principaux objets instanciés dans la scène. Ces derniers peuvent disposer d'objets « enfants », mais pour des raisons de lisibilité, seuls les objets de « haut-niveau » sont présentés ici.

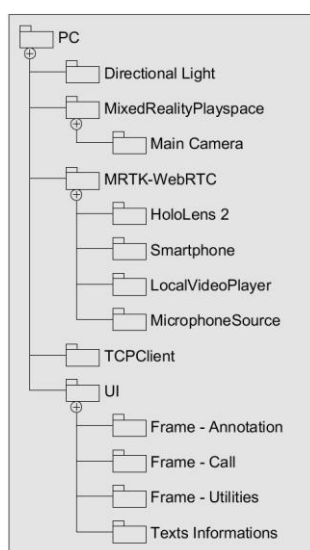


Figure 57 – PC : Hiérarchie des objets Unity

Tout d'abord, dans cette hiérarchie se trouvent les objets « Directional Light » et « Camera ». Ces éléments sont déjà expliqués dans le chapitre « 5. Implémentation générale ».

Un premier « Parent » important est « MRTK-WebRTC ». C'est dans celui-ci que se trouveront les objets liés à la communication vidéo et audio. Dans ce dernier, les objets seront, une fois de plus, séparés de manière logique. Ensuite, un composant « TCPClient » disposera des scripts liés à la connexion TCP pour transmettre les commandes à distances. Enfin, un dernier « Parent » est consacré à la réalisation de l'interface utilisateur. Ce dernier est séparé en quatre parties : UI d'annotations, UI d'appels, UI de fonctions utiles et les différents textes à afficher sur l'UI.

6.2.3 Implémentation de MRTK-WebRTC

Dans les « parents » HoloLens 2 et Smartphone se trouveront les objets propres à chaque appareil. Ces derniers disposeront des objets suivants : PeerConnection, Signaler, NodeDssSignalerControls, RemoteVideoPlayer et RemoteAudio player. De cette façon il est possible de créer des appels multi-participants.



Figure 58 - Enfants de HoloLens 2 et Smartphone

Enfin, « LocalVideoPlayer » et « Microphone Source » font également partie du parent « MRTK-WebRTC ». Ce sont ces composants qui donneront accès à la caméra et au microphone du PC. En référence se trouve un lien vers le tutoriel¹⁵ utilisé pour implémenter ces composants. Ainsi, ceci ne sera pas décrit ici pour alléger ce document. Les informations de mise en œuvre d'appels multi-participants se trouvent dans le chapitre « 5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC ».

6.2.4 Implémentation des scripts

Ce chapitre décrira les points importants des différents scripts utilisés dans ce prototype. Des dossiers séparent ces derniers selon le schéma ci-dessous.

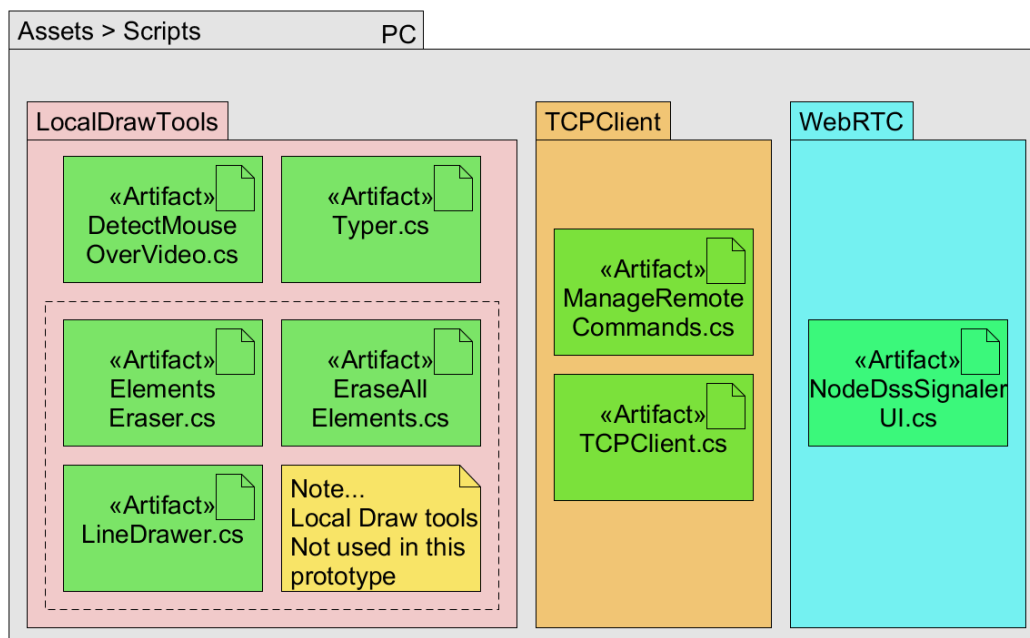


Figure 59 - Répartition des scripts dans le projet Unity PC

LocalDrawTools Scripts

Dans le dossier LocalDrawTool se trouvent les scripts relatifs aux fonctions d'interactions avec l'UI locale. Il est à noter qu'au départ, avant de contrôler les outils d'annotations à distance, certaines fonctionnalités d'annotations en local ont été réalisées. Ces dernières permettaient de dessiner en 2D sur le lecteur vidéo mais n'étaient pas visibles par la personne assistée. Ces fonctions n'étant plus utilisées, elles ne seront pas décrites ici. Cependant, le prototype HoloLens 2 possède des scripts semblables pour dessiner en local et, de ce fait, ils seront décrits dans le chapitre du prototype HL2 (cf. 7. Prototype HoloLens 2).

Cela étant dit, il reste deux scripts utilisés.

DetectMouseOverVideo.cs :

Ce script empêche d'envoyer des commandes à distance non désirées lorsque l'utilisateur n'est pas sur un lecteur vidéo. Ceci permet à ce dernier d'utiliser les éléments de l'UI sans que des actions indésirables ne se produisent à distance.

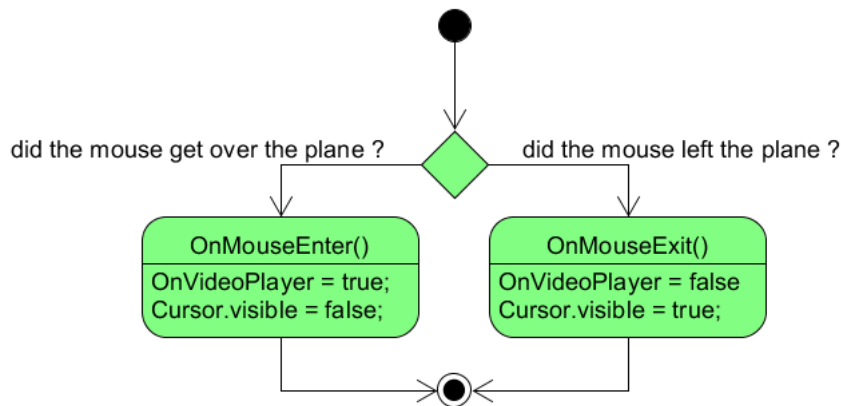


Figure 60 - Diagramme de transition : DetectMouseOverVideo.cs

Deux méthodes, OnMouseEnter() et OnMouseExit(), fournies par Unity permettent de détecter si la souris passe par-dessus l'objet qui a ce script attaché à lui. Ici, ces objets sont les lecteurs vidéo. Ainsi, si la souris passe sur un plan vidéo, les fonctionnalités à distance sont activées et, à l'inverse désactivées. Le curseur est également désactivé lorsqu'il passe sur le lecteur vidéo. De cette manière, seul l'outil contrôlable à distance est perçu par l'utilisateur PC et empêche une éventuelle confusion dans le cas où le curseur de la souris et l'outil distant ne sont pas à la même position.

Typewriter.cs :

Ce script a pour but de construire une chaîne de caractères, écrite par l'utilisateur PC, et de l'insérer dans le message JSON qui sera transmis à l'appareil pair via TCP. Dans la méthode Update() de ce script, une méthode CheckInput() est appelée à chaque frame afin de récupérer les caractères tapés au clavier par l'utilisateur.

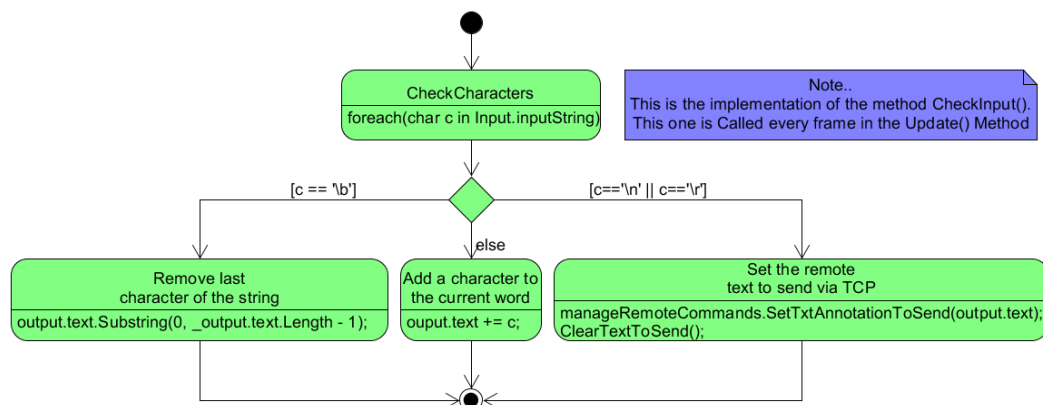


Figure 61 - Méthode CheckInput() du script Typer.cs (PC)

Tout d'abord, une boucle récupère chaque caractère tapé par l'utilisateur durant la frame où la méthode a été appelée. Ensuite, en fonction du caractère récupéré, certaines actions sont réalisées. La touche « Backspace » permet d'effacer la dernière lettre de la chaîne de caractères. Ensuite, la touche « Enter » permet de valider la chaîne de caractères et de l'insérer dans le message JSON de la classe dû à cet effet. La chaîne de caractères est, ensuite, vidée afin d'en construire une nouvelle. Enfin, pour toute autre touche, la valeur de celle-ci est ajoutée à la chaîne de caractères en cours de construction.

TCPClient Scripts

Dans ce dossier se trouvent deux scripts. Ces derniers servent à la création de la liaison TCP avec le serveur distant et à la création des messages JSON qui seront envoyés via cette connexion.

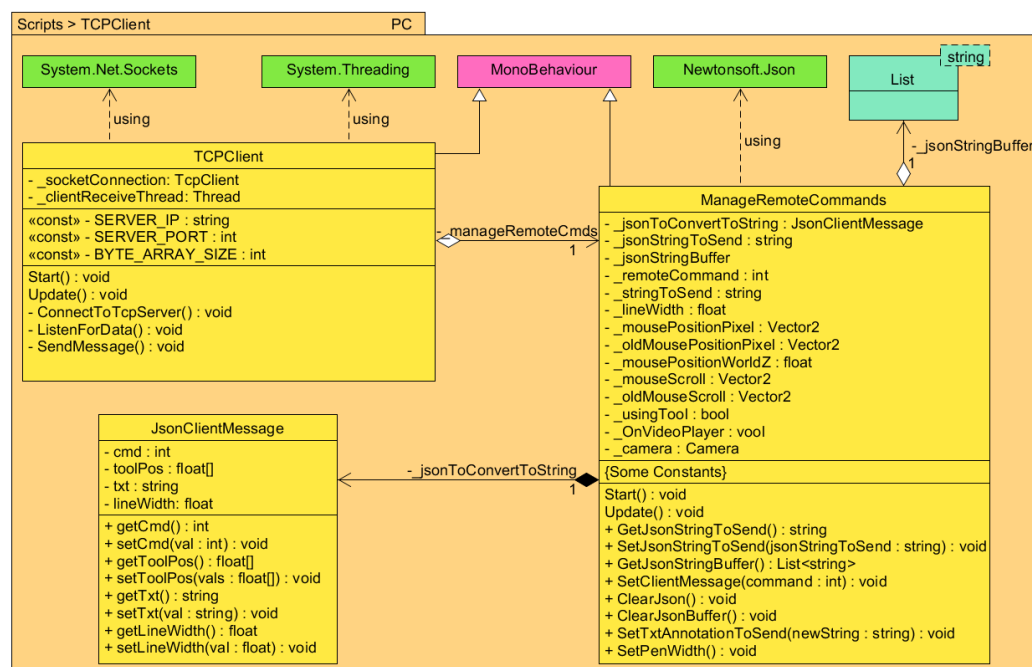


Figure 62 - Relations entre les scripts (classes) du dossier TCPClient (PC)

TCPClient.cs

Tout d'abord, ce script permet à l'application de se connecter en tant que client au serveur TCP. Il a besoin de « System.Net.Sockets » et « System.Threading » pour être implémenté.

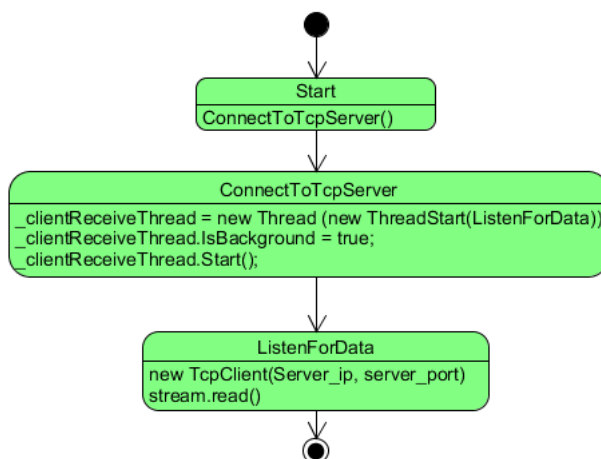


Figure 63 - Connection du client TCP (PC) à son serveur TCP

Pour réaliser la connexion TCP au début de l'application, une méthode « ConnectToTcpServer() » est appelée dans « Start() ». Dès lors, un Thread est exécuté en arrière-plan afin d'écouter les potentiels messages provenant du serveur. Pour ce faire, une nouvelle connexion « Socket » doit être faite en spécifiant l'adresse IP et le port du serveur à l'aide de la classe TcpClient. Il est ensuite possible de lire le flux de données de cette connexion.

Ensuite, dans la méthode Update() qui est appelée à chaque frame, la méthode SendMessage() est appelée. C'est cette dernière qui permet d'envoyer les messages JSON du client en direction du serveur TCP.

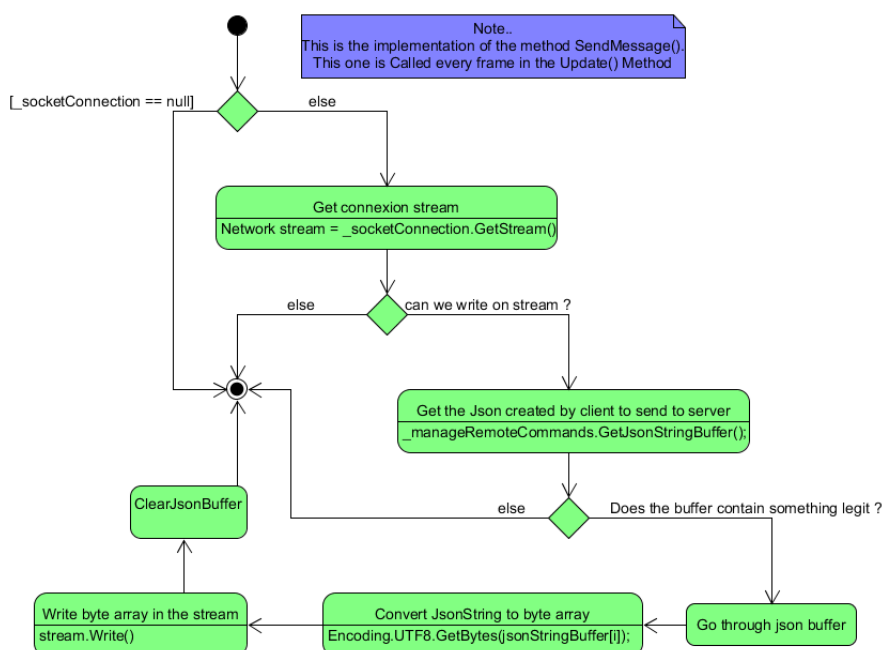


Figure 64 - Méthode SendMessage() pour envoyer les messages JSON au serveur

Dans cette méthode, il est contrôlé que la connexion socket soit bien établie. Si tel est le cas, le canal servant de flux de communication est récupéré dans le but d'y écrire des messages. Si le flux est libre, le script récupère les commandes du buffer contenant les messages JSON depuis la classe « ManageRemoteCommands ». Il est alors vérifié que le buffer contienne bel et bien un message légitime à envoyer en vérifiant que son contenu ne soit pas « null ». Si celui-ci est valide, une « boucle for » parcourt le buffer et converti son contenu en tableau de bytes qui peut être alors envoyé dans le flux de communication. À la fin de l'opération, le buffer est vidé afin de pouvoir recommencer l'opération proprement.

ManageRemoteCommands.cs :

Ce script permet de construire les messages JSON. Pour ce faire, le paquet « Newtonsoft.json » présenté plus tôt donne accès aux méthodes JSON telles que la sérialisation et la désérialisation. De nombreux messages JSON étant créés dans un petit laps de temps, ces derniers sont sérialisés et stockés dans un buffer.

Une classe nommée « JsonClientMessage » est également présente dans ce script. Celle-ci permet de définir les attributs des messages JSON. De la sorte il est possible de supprimer ou d'ajouter de nouveaux attributs de données. Alors, la classe « ManageRemoteCommands » possède un attribut de cette classe.

Des messages peuvent être créés dans plusieurs cas. Lorsque l'utilisateur clique sur un bouton de l'UI, lorsqu'il souhaite envoyer une annotation textuelle, lorsqu'il désire interagir avec un outil d'annotation distant ou changer certaines options.

En cas d'interaction avec un bouton de l'UI, le principe de création de message JSON est le suivant :

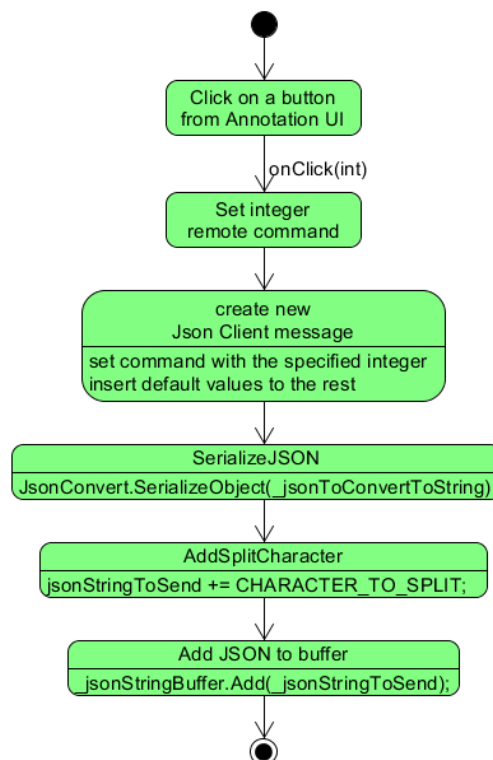


Figure 65 - Principe de création de messages JSON lorsqu'un bouton de l'UI spécifique est pressé

Lorsqu'un bouton de l'UI est pressé, il crée un événement avec un nombre entier défini au préalable. Dès lors, un message JSON est créé avec le chiffre en question comme attribut. Le reste des valeurs n'étant pas intéressantes dans ce cas de figure, elles prennent des valeurs par défaut. L'objet JSON est ensuite sérialisé et converti en chaîne de caractères, à laquelle un caractère servant à séparer les différents messages créés est ajouté à la fin. Elle est ensuite ajoutée à la liste de JSON existante.

Le reste des messages est créé dans la méthode « Update() » qui met à jour les attributs JSON si nécessaire. Par exemple, l'attribut « Txt » est mis à jour lorsque l'utilisateur souhaite envoyer une annotation textuelle. Celle-ci est expliquée dans l'algorithme de Typer.cs situé dans ce même chapitre. Ci-dessous, la méthode « Update() » de ce script est expliquée.

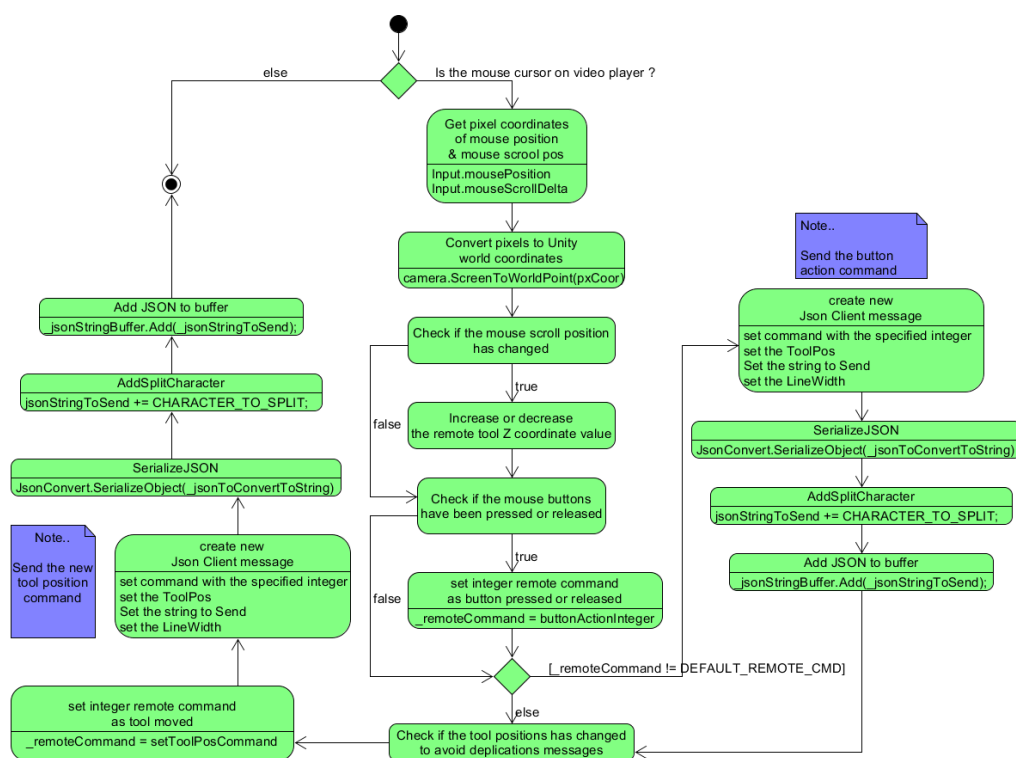


Figure 66 - Algorithme dans la méthode Update de ManageRemoteCommands.cs

Tout d'abord, comme expliqué auparavant, une condition contrôle que la souris se trouve sur le plan vidéo de l'application. Si c'est le cas, la position en pixels de celle-ci est récupérée et convertie en unités du monde Unity. Ensuite, la valeur de l'axe en Z de la souris est mise à jour au moyen de la molette de cette dernière. Dans un deuxième temps, la position de la souris ayant été traitée, un contrôle est effectué au niveau des interactions des boutons de cette dernière. En fonction de l'état du bouton, un message JSON sera créé avec un chiffre entier spécifique pour servir de commande. Le reste des attributs sont également mis à jour avec les valeurs actuelles. Le JSON est ensuite transformé, comme expliqué au préalable, et ajouté dans un buffer. Dans un troisième temps, si la position de l'outil a changé par rapport à la frame précédente, un nouveau JSON sera créé avec une commande spécifique servant uniquement à déplacer l'outil distant. Comme avant, le JSON sera transformé et ajouté à la liste des JSON actuels.

WebRTC Scripts

Ce script provenant de la bibliothèque MRTK-WebRTC et étant facilement compréhensible, il ne sera pas expliqué ici. Il a été modifié de sorte à pouvoir désactiver la caméra source afin de pouvoir utiliser cette dernière dans une communication multi-participants. Cette explication est faite dans le chapitre dédié à aux appels multi-participants. (cf. 5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC)

Liste des commandes créées depuis le PC

Ci-dessous se trouve la liste des commandes créées depuis le PC.

Provenance	Option	Nombre entier
Bouton UI	Activer annotation libre	1
Bouton UI	Activer la gomme	2
Bouton UI	Tout effacer	3
Bouton UI	Activer l'annotation flèche	4
Bouton UI	Activer l'annotation texte	5
Script - Update()	Début d'une annotation	6
Script - Update()	Fin d'une annotation	7
Script - Update()	Position de l'outil à modifier	8
Bouton UI	Changer la couleur de l'outils distant : Rouge, vert, bleu, cyan, magenta, jaune, blanc, gris, noir	15 à 23

Tableau 3 - Liste des commandes créées par le client TCP

Cette technique permet d'ajouter simplement des commandes pour acquérir des fonctionnalités supplémentaires.

6.2.5 Conclusion de l'implémentation PC

Dans ce prototype, les fonctionnalités d'appels et d'annotations à distance ont été implémentées. La fonctionnalité de partage d'écran et de contrôle de périphériques n'ont pu être mise en œuvre dû au manque de temps et aux priorités du scénario de ALPS Automation SA.

Des fonctionnalités de dessin en local (similaire à l'application « Paint » de Windows) ont été implémentées. Toutefois, elles n'ont finalement pas été utilisées mais les scripts restent à disposition dans le projet Unity si besoin.

Les principaux algorithmes ont été présentés. Il a également été montré comment la création de commandes à distance est réalisée, mettant en lumière la simplicité d'ajout ou de suppression de nouvelles fonctionnalités.

6.3 Déploiement de l'application

Ce chapitre présente les différentes configurations Unity afin de déployer ce prototype sur un ordinateur.

Tout d'abord, la plateforme de déploiement doit être sélectionnée dans les « Builds Settings ». Ici, « PC, Mac & Linux Standalone » doit être choisi. Il est alors possible de choisir le système d'exploitation de l'ordinateur ainsi que son architecture. De plus, il faut sélectionner les scènes Unity à compiler.

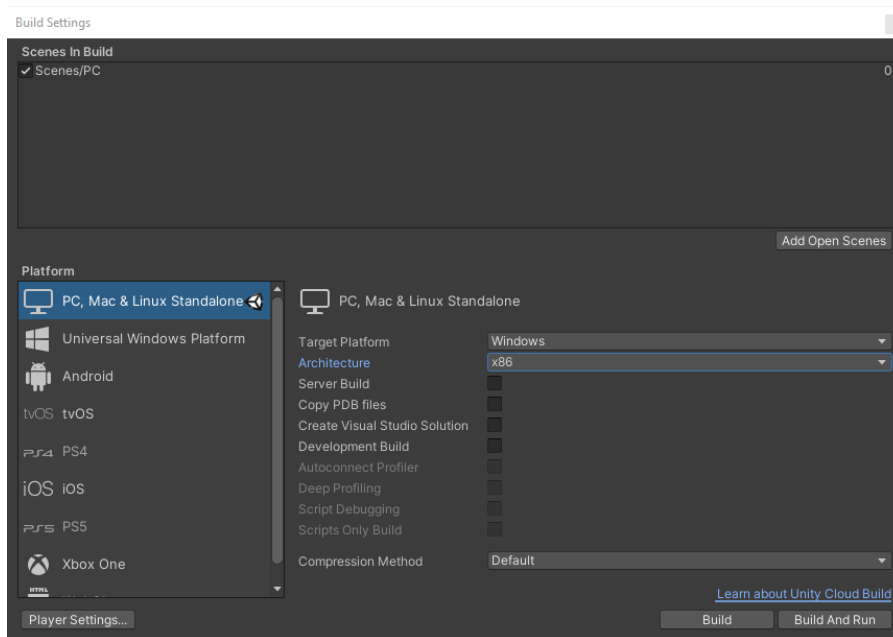


Figure 67 - Build Settings PC

Ces modules de déploiement doivent être ajoutés à la version d'Unity installée. Ceci se fait à l'aide de la plateforme « Unity Hub ». (Unity Hub > Installs > Version Unity > Add Modules)

Dans les annexes se trouve une capture d'écran montrant les différentes configurations du « player » et des modules Unity Hub afin de déployer correctement le prototype.

Cela étant fait, l'utilisateur peut « Build » son application. A la fin de la compilation, un fichier « .exe » est généré et permet de lancer l'application. Il est à noter que certaines fois, cet exécutable doit être lancé en mode administrateur.

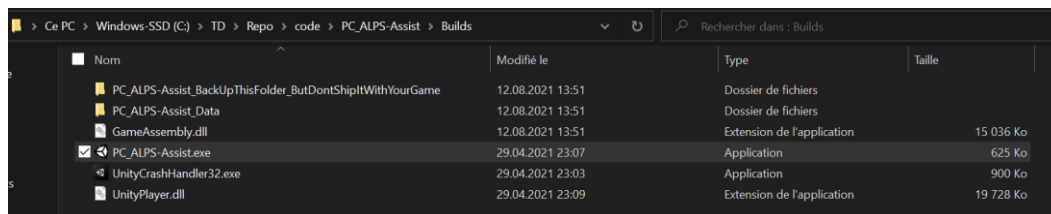


Figure 68 - Exécutable généré par la compilation

Dans la définition de l'interface utilisateur de ce prototype se trouve une image illustrant l'application déployée sur le PC.

6.4 Validation

Ici seront présentés les résultats des différents tests « black-box » effectués pour le prototype PC. Ces derniers seront présentés sous forme de tableaux avec les résultats obtenus. Afin de ne pas surcharger la lecture de ce rapport, ces tableaux sont placés en annexes.

Les tests liés à la communication audio et vidéo possèdent un ID « Callx_PC » et les tests liés aux annotations ont un ID « Annx_PC ».

Les résultats ont montré que le prototype PC est capable de créer une communication audio et vidéo avec les prototypes smartphone et HoloLens 2. De plus, il est également capable de recevoir les appels provenant de ces derniers. L'image vidéo et le son sont correctement transmis et reçus.

Ensuite, les différentes fonctionnalités d'annotations montrent également un bon fonctionnement. Toutefois, il est compliqué de réaliser des annotations libres correctement car l'image n'est pas stable du fait que la caméra de la personne assistée est constamment en mouvement. Un système permettant de figer l'image à un instant donné pourrait résoudre ce problème. De plus, en fonction du décor et des déplacements de la personne assistée, les outils d'annotations peuvent sortir du champ de vision du spécialiste et être irrécupérables. Un bouton permettant de réinitialiser la position des outils résoudrait temporairement ce problème. Enfin, via le flux TCP, les messages JSON parviennent au serveur TCP et, de la sorte, le spécialiste peut faire toutes sortes d'annotations ainsi que d'en modifier les options.

Un chapitre consacré aux tests utilisateurs est présenté plus tard dans le rapport. (cf. 10. Scénario d'ALPS Automation SA et tests d'usabilité)

7. Prototype HoloLens 2

Ce chapitre traite du développement du prototype se déployant sur un HoloLens 2 utilisé par une personne nécessitant une assistance à distance. Il sera séparé en quatre parties : Conception, Implémentation, Déploiement et Validation.

7.1 Conception

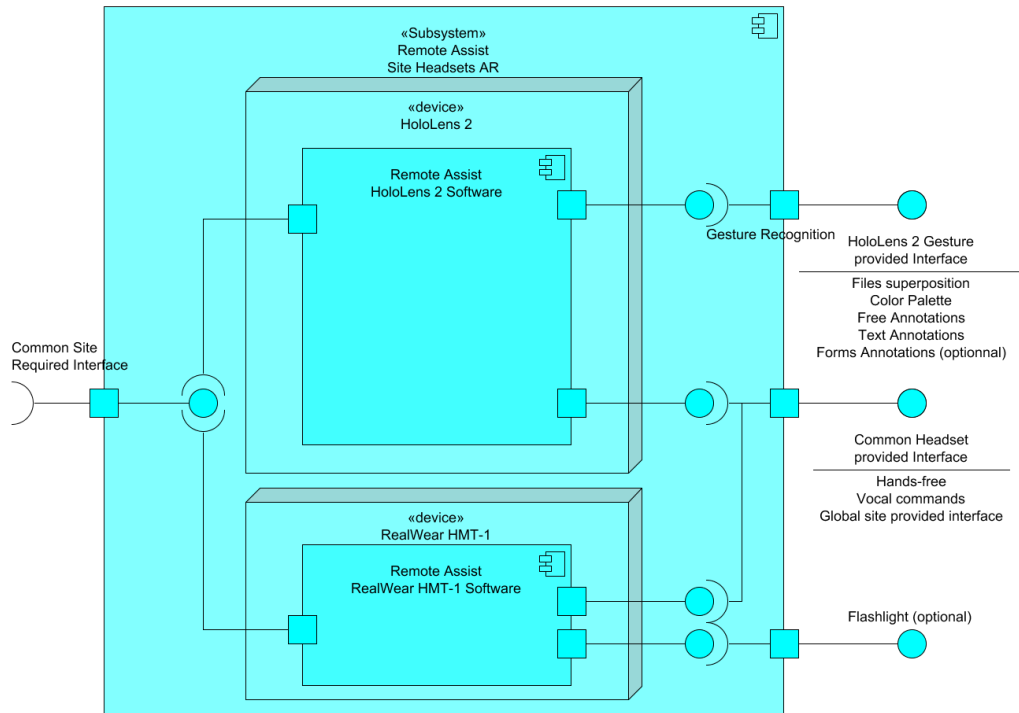


Figure 69 - Layer 3 : Interfaces HoloLens 2 VS RealWear

Cette section traitera spécialement de l'application prototype « Remote Assist HoloLens 2 Software » utilisée par la personne assistée. Cette dernière requiert et offre les différentes interfaces communes discutées au chapitre « 4.1.3 Layer 2 – Diagramme de composants général (Site) ».

De plus, « Remote Assist HoloLens 2 Software » offre une interface d'annotations supplémentaire à cette interface commune. Cette dernière n'a pas été précédemment considérée comme commune car, certaines fonctionnalités ne peuvent être fournies par tous les dispositifs de déploiement.

En effet, la figure précédente montre que le RealWear HMT-1 n'offre pas la possibilité d'offrir des annotations réalisées par l'utilisateur assisté. Ceci est dû au fait qu'il ne dispose que de reconnaissance vocale et non gestuelle.

7.1.1 Définition de l'architecture du système

Cette partie décrit l'architecture de l'application prototype de la personne assistée sur HoloLens 2. Les outils et bibliothèques de développement servant à la réalisation de ce projet seront présentés avec certains de leurs composants permettant de fournir les différentes interfaces.

Layer 4 : Remote Assist HoloLens 2 Software

Tout d'abord, la figure suivante présente le sous-système représentant l'application HoloLens 2. Cette dernière a été réalisée à l'aide de Unity 3D 2019.4.26.f1 qui dispose encore d'un support à long terme.

Ce sous-système requiert et fournit les différentes interfaces discutées plus tôt à travers ces différents ports. Ici, spécifiquement, les ports servant à la capture d'interactions utilisateurs sont les différents capteurs de l'appareil permettant de faire du suivi de regard ou de la reconnaissance gestuelle. Ensuite, la capture audio et vidéo ainsi que la connexion Internet se font à l'aide d'un microphone, d'une caméra et d'un récepteur Wi-Fi ou Modem LTE 5G.

Les composants internes seront maintenant expliqués afin de comprendre la manière dont les interfaces fournies sont créées.

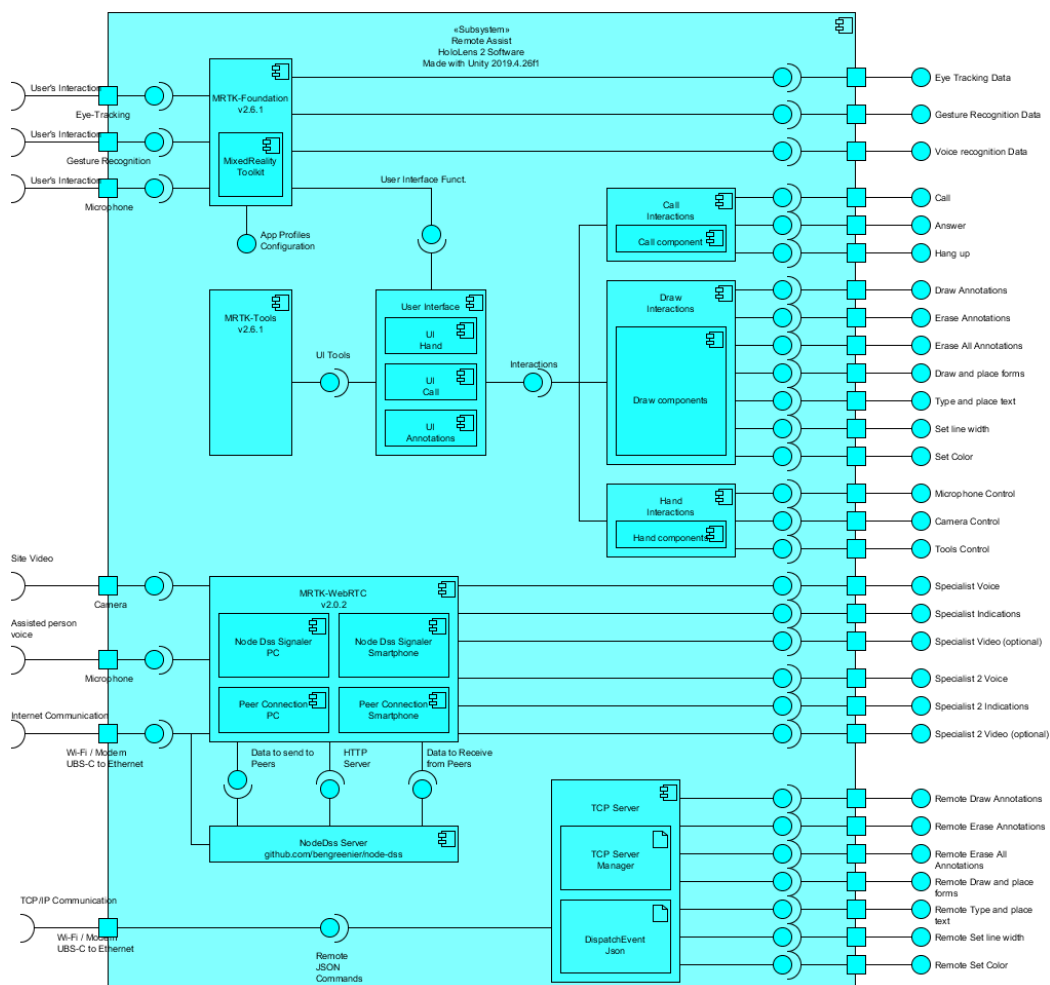


Figure 70 - Layer 4 : Architecture du software HoloLens 2 de la personne assistée

UI et MRTK-Foundation

Parmi les composants présents dans ce software se trouve MRTK-Foundation v2.6.1. Celui-ci est l'ensemble de codes qui permet à l'application d'utiliser des fonctionnalités de réalité mixte. Fourni avec ce paquet, le composant « MixedReality Toolkit » permet de configurer ces différentes fonctionnalités afin d'en tirer le meilleur pour l'application à concevoir. Ensemble, ils fournissent à l'application les données liées aux interactions de l'utilisateur, tels que le suivi du regard, les gestes effectués ou encore des commandes vocales, permettant ainsi de réaliser diverses tâches.

Toutefois, pour réaliser ces différentes tâches, l'utilisateur a besoin d'une UI (User Interface) avec laquelle il peut interagir avec l'application. MixedReality-Toolkit (MRTK) fourni également des composants facilitant la mise en œuvre de cette dernière. Parmi ces composants peuvent se trouver : des boutons, des curseurs, des menus, etc. Dans les références se trouve la documentation à ce sujet afin d'avoir accès au reste des informations^[24].

Dès lors, il est possible de créer les UI désirées. Parmi ces dernières se trouve un menu de la main « UI Hand » qui permet d'ouvrir et d'accéder à d'autres menus. Dans ces menus se trouvent un menu d'appel et un menu d'annotations. Le menu d'appel « UI Call » permet d'accéder aux fonctionnalités d'appels et de fournir les interfaces permettant d'appeler un pair, de répondre à un pair ou de raccrocher un appel. « UI Annotation » permet à l'utilisateur d'accéder aux fonctionnalités d'annotations et de fournir ainsi les interfaces liées à ces dernières. Enfin, dans le menu de la main se trouve un bouton permettant de désactiver les fonctionnalités en cours liées aux annotations, permettant ainsi à l'utilisateur de retrouver ses « mains-libres » pour effectuer diverses actions, tels que désactiver ou activer le microphone et la caméra ou encore manipuler des objets et des options.

Réception d'annotations et de commandes à distances

Un composant important est « TCP Server ». Ce dernier permet à l'HoloLens 2 de jouer un rôle de serveur, lui offrant ainsi la possibilité de voir plusieurs clients s'y connecter. De cette manière, L'HoloLens 2 peut recevoir des messages JSON envoyés par les clients. Cette technique permet aux clients de placer des annotations virtuelles contrôlables à distance, dans l'environnement de la personne assistée pour la guider dans sa maintenance ou sa mise en service.

Communication audio et vidéo.

Un autre composant important est celui permettant de créer une connexion et de communiquer avec ses pairs. Ici, l'appareil peut se connecter à deux spécialistes simultanément. Pour ce faire, MRTK-WebRTC v2.0.2 a été utilisé. Afin d'alléger ce rapport, cette rubrique suggère de se référer au chapitre *Communication audio et vidéo* du prototype PC. En effet, hormis les appareils qui diffèrent, les explications sont les mêmes.

²⁴ MICROSOFT, *Présentation du Mixed Reality Toolkit*

<https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05>
(Consulté le 1^{er} août 2021)

Layer 4 : Interfaces fournies nativement par l'HoloLens 2

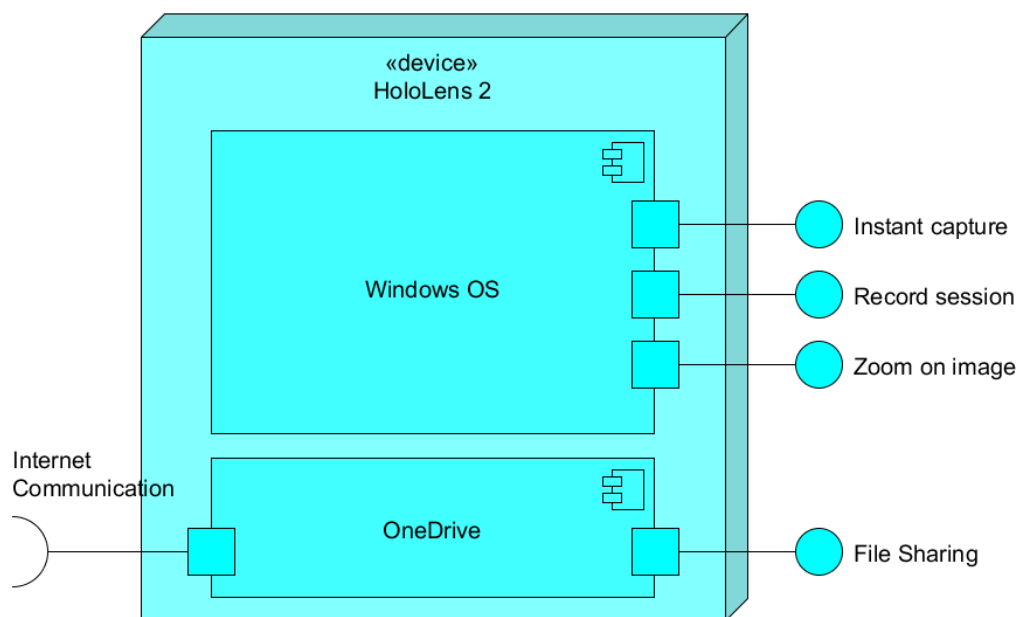


Figure 71 - Layer 4 : Interfaces fournies nativement par l'HoloLens 2

Cette rubrique traite d'un autre élément de cette troisième couche. Parallèlement à l'application de l'HoloLens 2, cet appareil offre nativement des fonctionnalités intéressantes et utiles à ce projet. C'est pourquoi elles sont toutes deux considérées dans ce Layer 3.

La facilité d'accès pendant l'exécution de l'application prototype, ainsi que leur bon fonctionnement de ses fonctionnalités a permis de ne pas se préoccuper de ces dernières durant le développement de ce prototype. En effet, le temps de développement étant restreint, il est compliqué de mettre en œuvre ces fonctionnalités déjà existantes.

En effet, HoloLens 2 offre les capacités de prendre des photos instantanées, de filmer une session ou encore de faire des zooms sur des fichiers. Il est à noter que le RealWear HMT-1 offre également ces interfaces. De plus, pendant l'exécution de l'application prototype, il est possible d'accéder à la plateforme OneDrive (ou autre cloud) permettant de stocker, d'envoyer et de recevoir des fichiers, ce qui peut être fort intéressant dans le cadre d'une assistance à distance. Les différents utilisateurs (spécialistes et personne assistée) partageraient un cloud en commun afin de se partager les différents médias nécessaires. Pour ce faire, l'application ne requiert qu'une connexion Internet, au même titre que pour les communications vidéo ou les annotations à distance.

7.1.2 Définition de l'interface utilisateur

Cette section présentera les différents éléments d'interface utilisateur liés au prototype déployé sur l'HoloLens 2. Les illustrations qui suivront auront été réalisées à l'aide de l'outil en ligne Figma^[25], permettant de réaliser des UI pour la réalité mixte. Ces éléments seront virtuellement superposés dans l'environnement de l'utilisateur.

UI Des fonctionnalités natives

Ici, sera présenté l'interface utilisateur native de l'HoloLens 2, qui offre la possibilité d'utiliser des plateformes externes en parallèle de l'exécution de l'application développée.



Figure 72 - UI Natif : Accès au menu principal

L'utilisateur peut accéder au menu de l'HoloLens 2 en cliquant sur une illustration sur son poigné lorsqu'il le regarde ou en pinçant ces doigts lorsque l'illustration est présente comme le montre la figure ci-dessus.



Figure 73 - UI Natif : Menu Principal

Le menu principal ouvert, l'utilisateur a accès à l'application réalisée dans ce projet et à plusieurs applications « non-immersives », signifiant qu'elles peuvent être ouvertes pendant l'application prototype dite « immersive ». Parmi ces dernières se trouvent Microsoft Edge, OneDrive, la boîte Mail, l'explorateur de fichiers et bien d'autres. L'utilisateur peut installer les applications qu'il souhaite par le biais du Microsoft Store.

²⁵ FIGMA.COM, *Figma Toolkit for MRTK/HoloLens, Windows Mixed Reality*.

<https://www.figma.com/file/ltLag9SxjUIyLQFsp7NNE7/Figma-Toolkit-for-MRTK-%2F-HoloLens%2C-Windows-Mixed-Reality?node-id=116%3A4> (Consulté le 1^{er} août 2021)

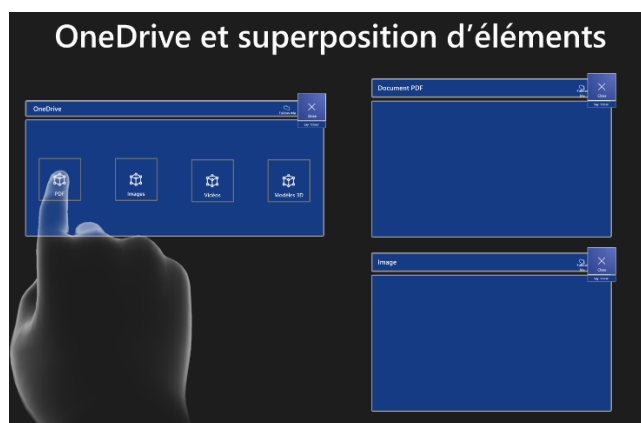


Figure 74 - Exemple de l'utilisation de OneDrive et de la superposition d'éléments

La figure ci-dessus montre un exemple avec la plateforme OneDrive. L'utilisateur peut accéder aux différents fichiers stockés dans le cloud et les ouvrir virtuellement en transparence dans son environnement. Pour d'autres plateformes telles que Microsoft Edge ou l'explorateur de fichier, le principe serait le même.

UI de l'application HoloLens 2

Cette section, quant à elle, présente l'interface utilisateur à développer dans l'application prototype de l'HoloLens 2.



Figure 75 - UI Prototype : Menu de la main

Tout d'abord, comme le montre la figure ci-dessus, l'utilisateur peut faire apparaître un menu en dirigeant la paume de sa main vers lui. Ce menu comporte trois éléments : « Hands-Free », « Draw menu » et « Call Menu ».

Hands free permet à l'utilisateur de désactiver toutes les fonctionnalités liées aux annotations. De la sorte, il peut manipuler des objets, interagir avec des options de l'application ou faire des gestes quelconques sans qu'il ne se passe certaines actions, comme par exemple, des annotations non désirées.

En cliquant sur Draw Menu, l'utilisateur ouvre un autre menu servant à la création d'annotations. En effet, comme le montre l'illustration suivante, celui-ci dispose de plusieurs outils de dessin.

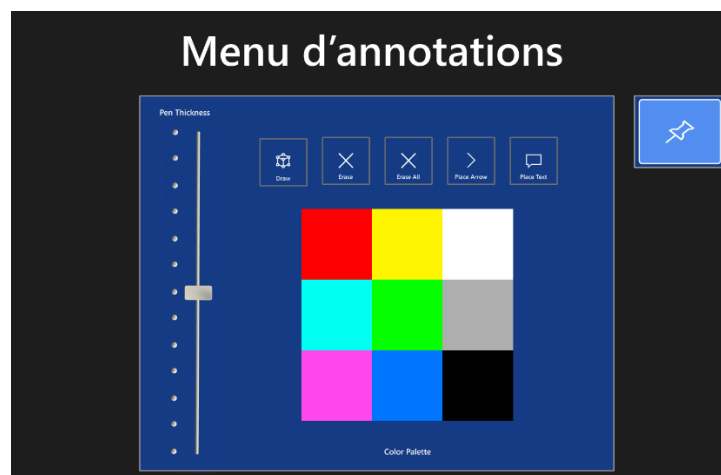


Figure 76 - UI Prototype : Menu d'annotation

A gauche de la figure ci-dessus se trouve un curseur permettant à l'utilisateur de régler l'épaisseur du trait de la ligne de dessin libre. Plus le curseur est vers le haut, plus le trait sera large. Situés au sommet du menu se trouvent cinq boutons permettant d'activer certaines fonctionnalités de dessin.

Le bouton « Draw » active le dessin libre. Lorsqu'il est pressé, une sphère suivrait la main de l'utilisateur. Cette dernière sert de repère, car c'est à sa position que la ligne de dessin sera dessinée lorsque l'utilisateur pincera ses doigts.

Le deuxième bouton nommé « Erase » servirait à effacer des annotations individuellement. Similairement à la sphère de dessin, une croix suivrait la main de l'utilisateur. Lorsque l'utilisateur pincera ses doigts en présence de cette croix, cette dernière effacerait toute annotation subissant une collision avec elle.

Quant à lui, lorsqu'il serait pressé, le bouton « Erase all » effacerait toute annotation présente dans l'environnement de l'utilisateur.

Pour ALPS Automation SA, qui désire avoir la possibilité d'annoter à l'aide de formes, un bouton serait présent pour placer ces dernières. Ici, à l'aide du bouton « Place Arrow », une flèche pourrait être instanciée à l'emplacement pincé par les doigts de l'utilisateur.

De la même manière, un texte d'annotation pourrait être placé et édité par l'utilisateur. C'est à cet effet qu'est présent le dernier bouton « Place Text ».

Au centre de la fenêtre se trouve une palette de couleur permettant de modifier la couleur des différentes annotations.

Enfin, ce menu pourrait être volontairement épinglé à un emplacement dans l'environnement de l'utilisateur, ou pourrait suivre ce dernier lorsqu'il est en mouvement.

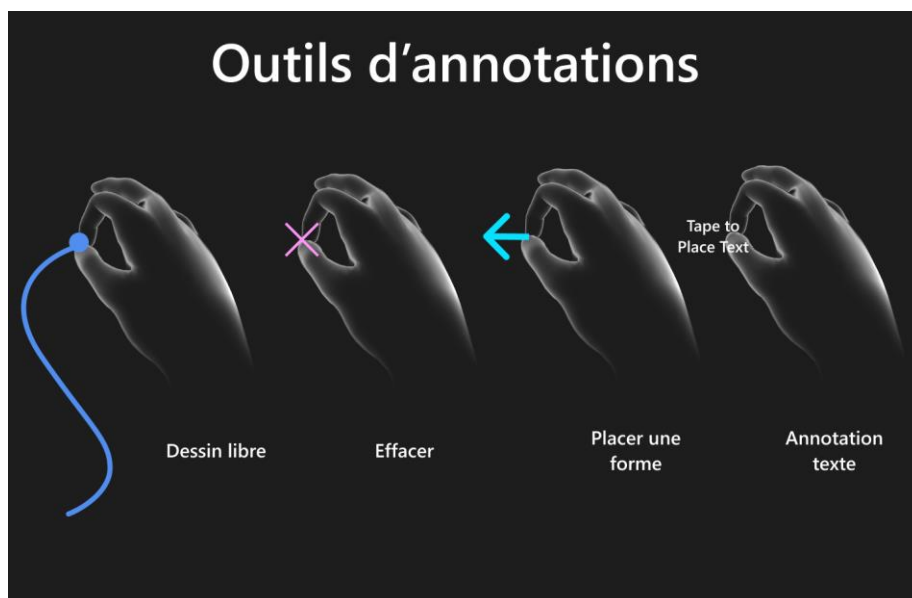


Figure 77 - Outils d'annotations

L'illustration ci-dessus donne une représentation visuelle des outils expliqués auparavant et une idée dont ces derniers seraient utilisés.

Pour terminer cette rubrique, ci-dessous se trouve une représentation du menu d'appel. Ce dernier disposerait de cinq boutons utiles aux appels ainsi que d'un bouton servant à épingler le menu à l'instar du menu d'annotation.

Pour ce prototype, le premier bouton permettrait de créer une communication avec la personne située à son bureau. Le deuxième créerait une communication avec un spécialiste supplémentaire. Dans le cadre de ce projet, ce bouton appellerait un technicien sur son smartphone. Le troisième bouton permettrait de couper la communication avec ses pairs. Enfin, les deux derniers boutons permettraient d'activer ou désactiver le microphone et/ou la caméra de l'HoloLens 2 en cas de besoin.



Figure 78 - UI Prototype : Menu d'appel

7.1.3 Conclusion de la conception HoloLens 2

Ces différents éléments de conception mis en place, il est maintenant possible de commencer l'implémentation du prototype HoloLens 2. L'architecture du projet ainsi que la définition de l'interface utilisateur servent de ligne conductrice pour implémenter correctement les différentes fonctionnalités de ce prototype.

Il est important de relever le fait que certaines fonctionnalités déjà présentes nativement dans l'HoloLens 2 ne seront pas implémentées dans le prototype, car celles-ci s'adaptent très bien aux besoins de ALPS Automation. Au besoin, lors d'une production officielle, il serait tout de même possible d'intégrer ces éléments au prototype.

La modularité a été également mise en avant. Lorsqu'un appareil dispose des différents ports nécessaires à l'implémentation des différentes interfaces, il peut être utilisé pour ce prototype. Toutefois, il devrait être compatible avec la librairie MRTK offrant les fonctionnalités de réalité mixte. En annexe se trouve un tableau indiquant les plateformes compatibles avec cette librairie.

Malheureusement, dû à son manque de reconnaissance gestuelle et de compatibilité avec la librairie MRTK, l'appareil RealWear-HMT1 ne s'adapte pas à la conception du prototype HoloLens 2. Une application conçue avec une autre logique devrait être réalisée pour cet appareil. Le temps de ce projet étant restreint, elle ne sera pas réalisée.

7.2 Implémentation

Cette section traitera de l'implémentation du prototype HoloLens 2. Elle présentera les paquets utilisés, la hiérarchie de composants du projets de façon générale et détaillera certains principes implémentés.

7.2.1 Paquets et bibliothèques externes

▼ Custom	
Mixed Reality Toolkit Extensions	2.6.1 ✓
Mixed Reality Toolkit Foundation	2.6.1 ✓
Mixed Reality Toolkit Standard Assets	2.6.1 ✓
Mixed Reality Toolkit Tools	2.6.1 ✓
MixedReality-WebRTC	2.0.2 ✓
▼ Unity Technologies	
▶ JetBrains Rider Editor	1.2.1 ✓
▶ Test Framework	1.1.24 ↓
▶ TextMeshPro	2.1.4 ↓
▶ Timeline	1.2.18 ✓
▶ Unity Collaborate	1.2.16 ✓
▶ Unity UI	1.0.0 ✓
▶ Visual Studio Code Editor	1.2.3 ✓
▶ Windows Mixed Reality	4.2.3 ✓

Figure 79 - Paquets présents dans le prototype Smartphone

Comme pour le projet PC, ce prototype a besoin de MixedReality-WebRTC 2.0.2 pour réaliser la communication audio et vidéo. De plus, les paquets suivants, provenant de la collection Mixed Reality ToolKit (MRTK), ont été également incorporés au prototype :

- Extensions 2.6.1 : Apporte des fonctionnalités supplémentaires au kit de base.
- Foundation 2.6.1 : Une collection de composants fondamentaux afin de créer des applications de réalité mixte. Il permet également de configurer certains paramètres de l'application, tel que l'accès à la caméra de l'appareil.
- Standard Assets 2.6.1 : Ressources de bases du kit.
- Tools 2.6.1 : Collection d'outils pré-faits à incorporer dans l'environnement. Elle est notamment utilisée pour la création d'UI de réalité mixte.

Le paquet Windows Mixed Reality 4.2.3 doit être également installé afin de pouvoir créer des applications de réalité mixte. Il est à noter que dès la version Unity 2020.1, ce paquet a été abandonné et n'est plus utilisé.

Afin de pouvoir travailler avec les objets JSON, il a fallu installer dans Visual Studio la bibliothèque Newtonsoft.Json via « Manage NuGet Packages » comme expliqué dans l'implémentation du prototype PC.

Les fonctions de texte de base de Unity étant un peu dépassées, TextMeshPro a été importé afin de remplacer ces dernières.

7.2.2 Hiérarchie Unity

La hiérarchie ci-dessous présente les principaux objets instanciés dans la scène. Ces derniers peuvent disposer d'objets « enfants », mais pour des raisons de lisibilité, seuls les objets de « haut-niveau » ont été présentés ici.

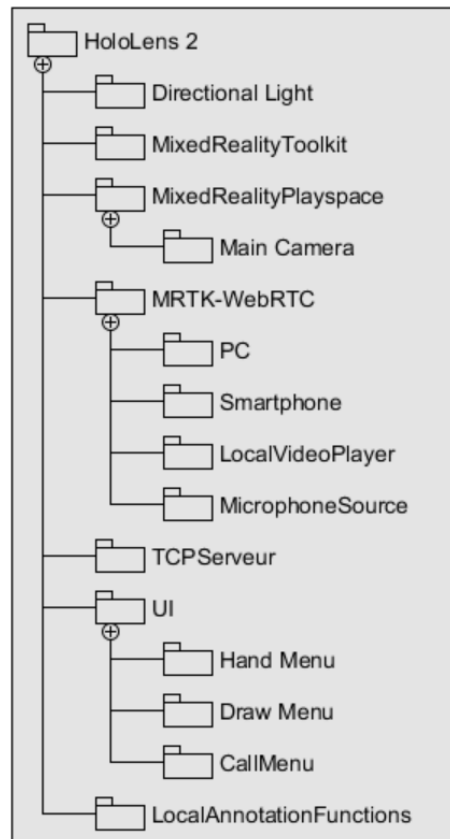


Figure 80 - HoloLens 2 : Hiérarchie des objets Unity

Tout d'abord, dans cette hiérarchie se trouvent les objets « Directional Light » et « Camera ». Ces éléments sont déjà expliqués dans le chapitre « 5. Implémentation générale ».

Un composant important de ce prototype est « MixedRealityToolkit ». C'est dans ce dernier que seront configurés les profils du prototype permettant d'activer ou désactiver certaines fonctionnalités de réalité mixte. Ces configurations seront présentées dans ce chapitre.

Un premier « parent » important est « MRTK-WebRTC ». C'est dans celui-ci que se trouveront les objets liés à la communication vidéo et audio. Dans ce dernier, les objets seront une fois de plus séparés de manière logique. Ensuite, un composant « TCPServeur » disposera des scripts liés à la connexion TCP pour recevoir des commandes à distance. Un autre « parent » est consacré à la réalisation de l'interface utilisateur. Ce dernier est séparé en trois parties : UI de la main, UI d'annotations et UI d'appels. Un dernier « parent » contient toutes les fonctionnalités liées aux annotations locales réalisées par la personne assistée.

7.2.3 Configuration des profils MixedRealityToolkit du prototype HoloLens 2

Une configuration par défaut étant déjà disponible, peu de modifications ont été nécessaires. Pour effectuer des modifications sur les configurations, les profils existants doivent être clonés. Ainsi, il est nécessaire de cloner le profil par défaut afin d'activer la reconnaissance spatiale de l'HoloLens 2 comme le montre la figure ci-dessous.

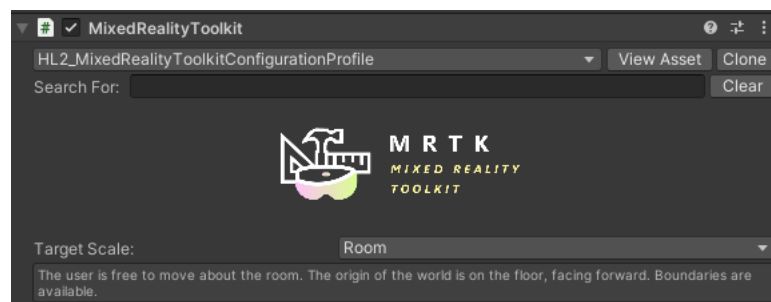


Figure 81 - Clone du profil HL2 par défaut

Dès lors, il est possible de cloner le profil dédié à la configuration de la reconnaissance spatiale et de lui apporter les modifications suivantes afin d'activer cette fonctionnalité.

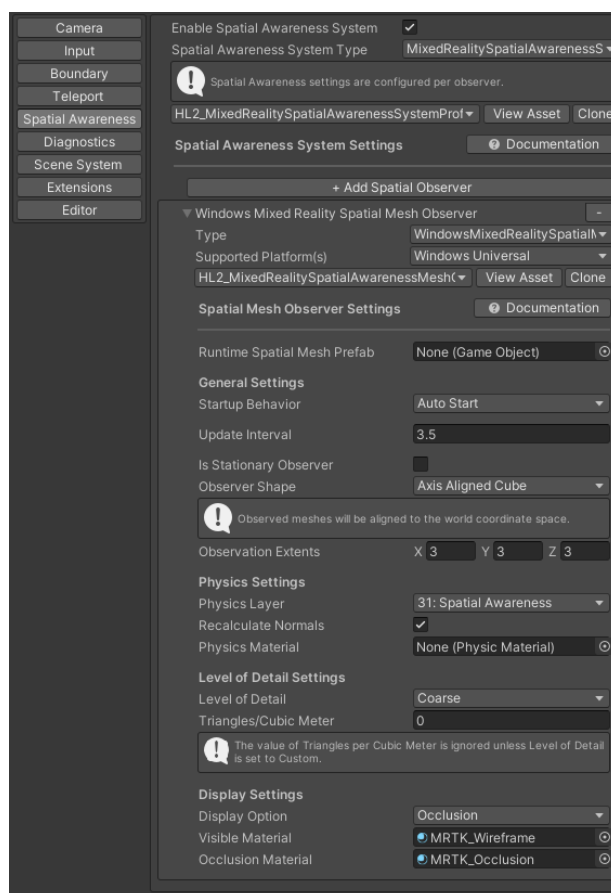


Figure 82 - Clone du profil Spatial Awareness ainsi que ces propriétés

Afin que la reconnaissance spatiale soit présente sans dégrader la vue de l'utilisateur, il a fallu configurer le champ « Display Option » en « Occlusion ».

7.2.4 Implémentation de MRTK-WebRTC

Dans les « parents » PC et Smartphone se trouveront les objets propres à chaque appareil. Ces derniers disposeront chacun des objets suivants : PeerConnection, Signaler, NodeDssSignalerControls, SpecialistVideo et RemoteAudio. De cette façon il est possible de créer des appels multi-participants.

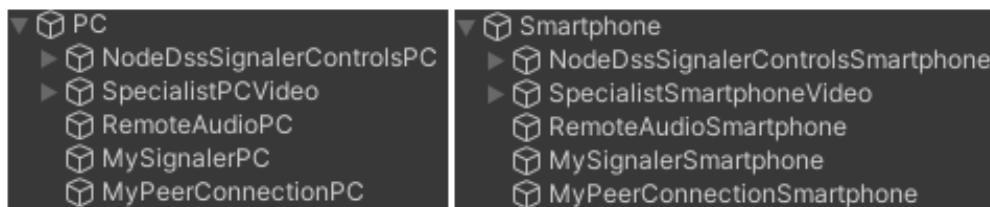


Figure 83 - Enfants de PC et Smartphone

Enfin, « LocalVideoPlayer » et « Microphone Source » font également partie du parent « MRTK-WebRTC ». Ce sont ces composants qui donneront accès à la caméra et au microphone de l'HoloLens 2. En référence se trouve un lien vers le tutoriel¹⁵ utilisé pour implémenter ces composants. De la sorte, ceci ne sera pas décrit ici pour alléger ce rapport. Les informations pour la mise en œuvre des appels multi-participants se trouvent dans le chapitre « 5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC ».

7.2.5 Implémentation des scripts

Ce chapitre décrira les points importants des différents scripts utilisés dans ce prototype. Des dossiers séparent ces derniers selon le schéma ci-dessous.

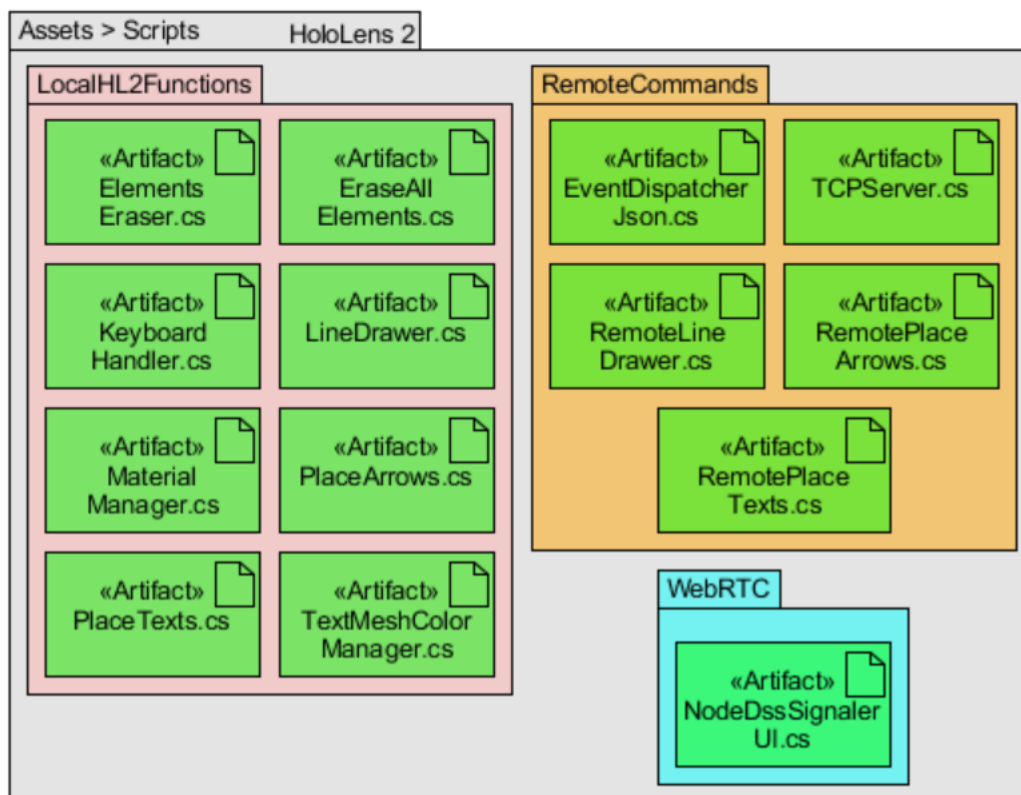


Figure 84 - Répartition des scripts dans le projet Unity PC

LocalHL2Functions Scripts

Dans le dossier « LocalHL2Functions » se trouvent les scripts relatifs aux fonctions d'annotations locales réalisées par la personne assistée. Certaines relations peuvent être réalisées entre les différentes classes de ces scripts.

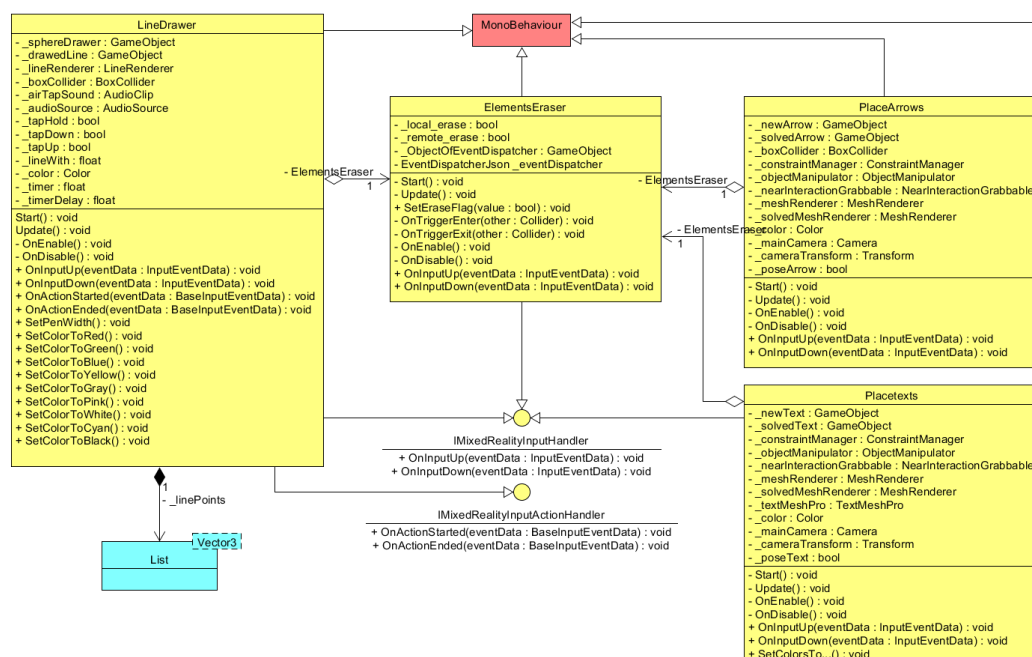


Figure 85 - Diagrammes de classe du dossier LocalHL2Functions

Les classes héritent de MonoBehaviour pour avoir accès aux méthodes de base d'Unity et à l'interface IMixedRealityInputHandler. Celle-ci permet de créer des événements lorsque l'utilisateur fait des actions de « pinch » avec ses doigts comme un click et un relâchement d'un bouton d'une souris. LineDrawer implémente également l'interface IMixedRealityInputActionHandler qui permet de marquer le début et la fin d'une action réalisée par un « pinch ».

Communes à ces classes, certaines méthodes et attributs sont expliqués ci-après avant de passer à l'implémentation individuelle des classes.

Tout d'abord, les méthodes provenant des interfaces IMixedRealityInput ne servent qu'à initialiser des booléens afin de savoir si des actions ont été réalisées ou non par l'utilisateur. De plus, il est nécessaire d'indiquer certaines informations au système dans les méthodes OnEnable() et OnDisable(). Exceptionnellement, quelques lignes de codes seront présentées ci-dessous afin d'avoir un exemple de comment réaliser ceci.

```
private void OnEnable()
{
    // Instruct Input System that we would like to receive all input events of type
    // IMixedRealityInputHandler
    CoreServices.InputSystem?.RegisterHandler<IMixedRealityInputHandler>(this);
    CoreServices.InputSystem?.RegisterHandler<IMixedRealityInputActionHandler>(this);
}

@ Unity Message | 0 references
private void OnDisable()
{
    // This component is being destroyed
    // Instruct the Input System to disregard us for input event handling
    CoreServices.InputSystem?.UnregisterHandler<IMixedRealityInputHandler>(this);
    CoreServices.InputSystem?.UnregisterHandler<IMixedRealityInputActionHandler>(this);
}
```

Figure 86 - Informer le système en cas d'utilisation des interfaces IMixedRealityInput

Les classes disposent également de méthodes afin de changer la couleur de l'annotation à placer. Enfin, dans les différentes méthodes « Start() » les différents attributs sont initialisés.

Parmi les attributs en communs, les trois classes d'annotations possèdent « ElementsEraser » qui sera ajouté comme composant à l'annotation créée afin de la rendre effaçable. « PlaceArrows » et « PlaceText » disposent de trois types d'attributs permettant de manipuler l'annotation instanciée : ObjectManipulator, ConstraintManager et NearInteractionGrabbable.

L'implémentation des scripts de ce dossier sont expliqués ci-dessous.

LineDrawer.cs

Cette classe permet à l'utilisateur de dessiner des lignes dans son environnement en 3D à l'aide de la reconnaissance gestuelle de l'HoloLens 2.

Pour ce faire, un algorithme est exécuté à chaque frame Unity dans la méthode Update(). Celui-ci est présenté ci-dessous.

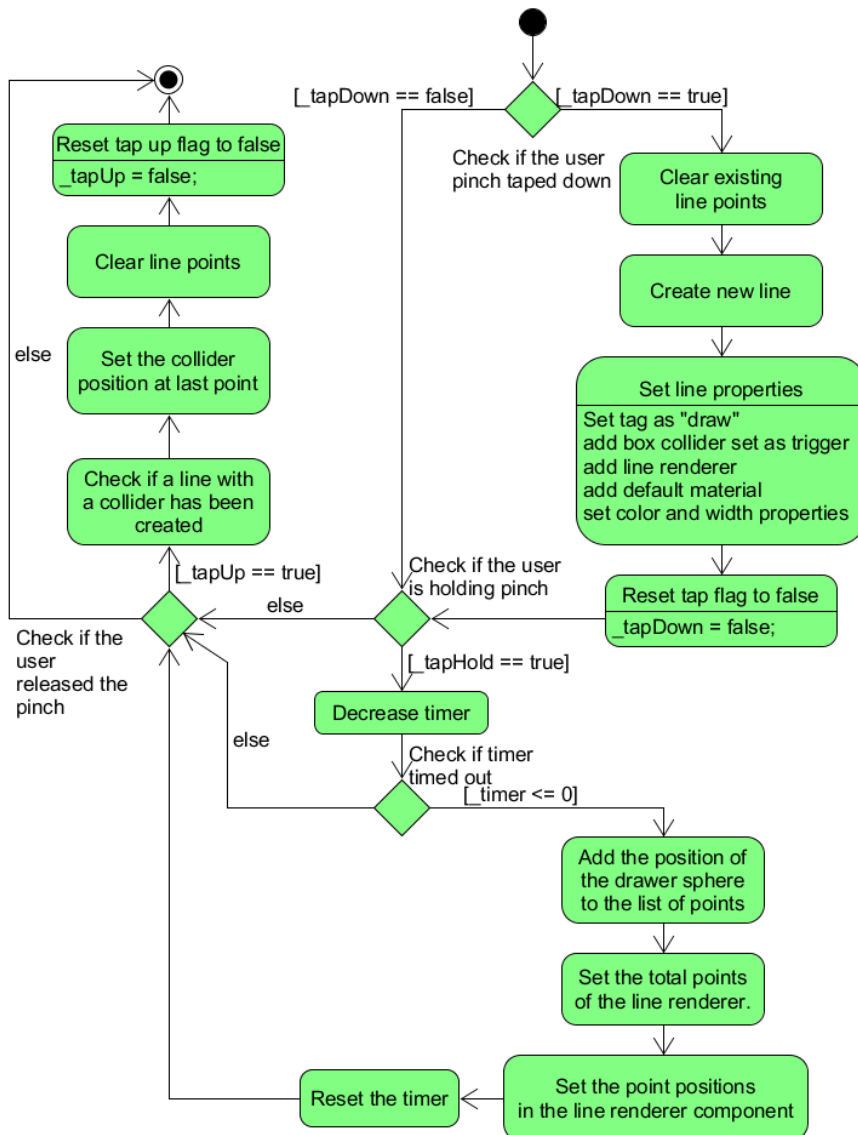


Figure 87 - Algorithme pour créer des lignes de dessin. Méthode Update() dans LineDrawer.cs

Dans cette méthode, les actions de l'utilisateur sont contrôlées via des valeurs booléennes initialisées au préalable par les méthodes fournies par les interfaces IMixedRealityInput.

Si l'utilisateur a « pinch », l'algorithme va nettoyer la liste de points à dessiner afin de commencer un dessin proprement. Ensuite, le nouvel objet qui représentera la ligne de dessin est créé. A ce même objet sont ajoutés une série de composants et des paramètres. L'objet est « tagué » en tant que « Draw » afin d'être effaçable par « EraseAllElements.cs » qui efface tous les objets considérés en tant que dessin. Ensuite, un « collider » est ajouté afin de pouvoir trigger des événements en cas de collision et, ainsi, effacer l'objet. Le composant LineRenderer est également ajouté. C'est ce composant qui contient la ligne de points à dessiner ainsi que les différentes propriétés liées à ceux-ci. Enfin, les matériaux, les couleurs et l'épaisseur des points sont paramétrées. Cela étant fait, la valeur booléenne indiquant qu'un « pinch » a été détecté est réinitialisée à « false ».

Ensuite, si l'utilisateur maintient son « pinch », c'est à ce moment-là que les différents points de la ligne à dessiner seront ajoutés à la liste de points. Il est à noter qu'un « Timer » empêche de dessiner à chaque frame afin de ne pas surcharger le programme de points dessinés aux mêmes positions. Lorsque le timer arrive à zéro, la position de la sphère servant de « stylo » et qui suit la main de l'utilisateur est ajoutée à la liste de points à dessiner. Le nombre de points du composant « LineRenderer » est, alors, mis à jour et la liste de points initialisée au préalable est ajoutée à ce même composant. Dès lors, le timer est réinitialisé.

Dans le cas où l'utilisateur relâche le « pinch », l'emplacement du collider ajouté à la ligne de dessin est ajoutée au même emplacement que le dernier point dessiné. Enfin, la liste de points est vidée et la valeur booléenne de cette partie est remise à false afin de recommencer l'opération.

ElementsEraser.cs

Ce script permet d'effacer les objets qui le possèdent. Ainsi, il est capable d'effacer les annotations créées localement par l'utilisateur de l'HoloLens 2 mais également celles créées à distance par les spécialistes.

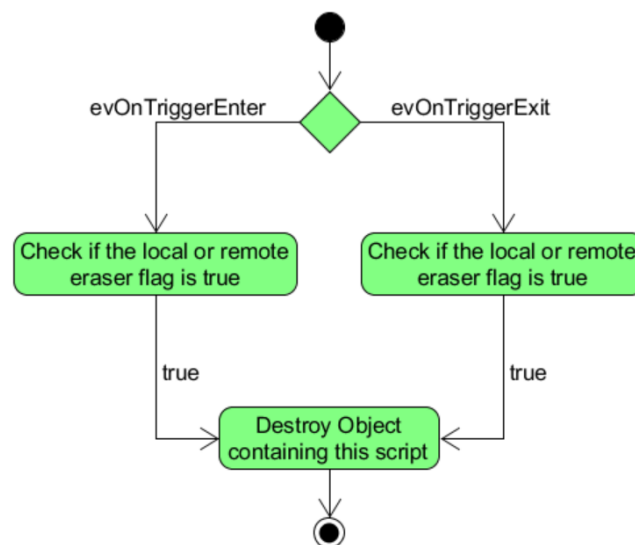


Figure 88 - Algorithme pour effacer une annotation. ElementsEraser.cs

À l'aide des « BoxCollider » ajoutés aux annotations lors de leur créations, ces derniers créent des événements lorsqu'ils entrent en collision avec un autre. Ici, lorsque la « croix » effaçable suivant la main de l'utilisateur entre ou sort du contact de l'annotation, celle-ci est détruite.

EraseAllElements.cs

Ce script permet d'effacer toutes les annotations instanciées dans l'environnement de la personne assistée.

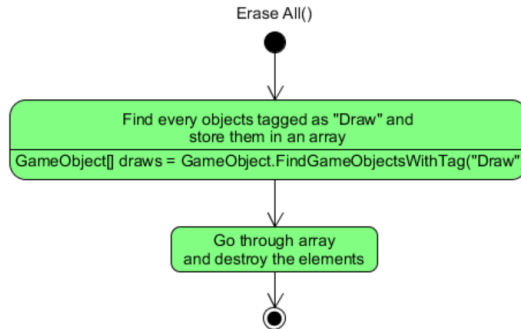


Figure 89 - Algorithme pour effacer toutes les annotations instanciées. Methode EraseAll() de EraseAllElements.cs

À l'aide de la méthode « FindGameObjectsWithTag() » fournie par Unity, il est possible de récupérer toutes les annotations taguées en tant que « Draw ». Ces objets sont ensuite stockés dans un tableau, lequel sera parcouru plus tard et verra ses éléments détruits afin d'effacer les annotations.

PlaceArrows.cs

Ce script est utilisé par l'utilisateur de l'HoloLens 2 afin de placer des flèches 3D dans son environnement.

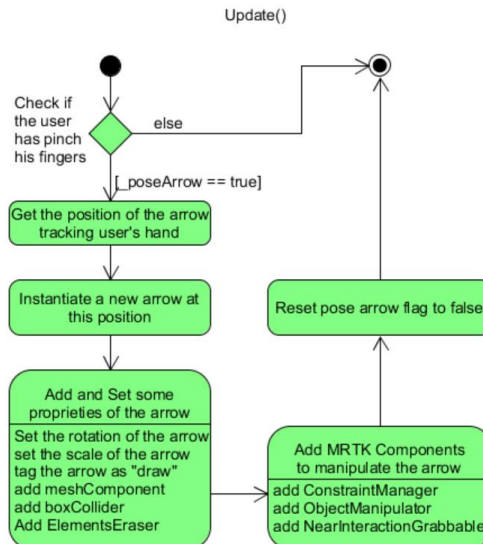


Figure 90 - Algorithme pour instancier des flèches locales. Méthode Update() de PlaceArrows.cs

Un premier contrôle est fait dans la méthode Update() de ce script afin de savoir si l'utilisateur souhaite instancier une nouvelle annotation en flèche. Si c'est le cas, la position de la flèche suivant la main de l'utilisateur est récupérée pour placer la nouvelle flèche à cette position au moment du « pinch ».

Dès lors qu'elle a été instanciée, la rotation de cette dernière ainsi que sa taille sont ajustées. Elle est également taguée en tant que « Draw » afin d'être effaçable. Les composants Mesh, BoxCollider, ElementsEraser ainsi que les composants servant à manipuler la flèche instanciée lui sont ajoutées. Enfin, le « flag » pour poser des flèches est réinitialisé à « false ».

PlaceTexts.cs

Ce script fonctionne selon le même principe que « PlaceArrows.cs », à la différence que le préfabriqué instancié est différent. Ce dernier a également la possibilité de paramétrer la couleur des textes à instancier.

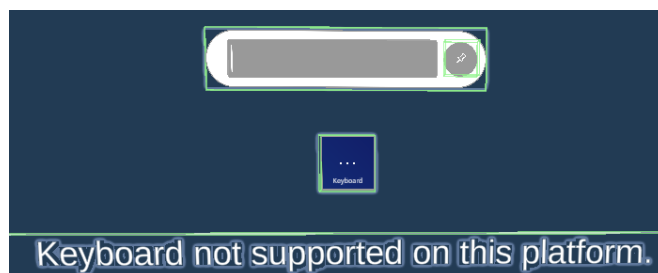


Figure 91 - Prefab de texte à instancier

Ci-dessus se trouve le préfabriqué réalisé dans ce projet pour l'instance d'annotations textes locales. Il est composé d'un champ permettant de visualiser ce qui est tapé au clavier lors de l'écriture de l'annotation, d'un bouton permettant d'ouvrir le clavier de réalité mixte afin de modifier le texte instancié et d'une zone de texte pour l'annotation écrite.

KeyboardHandler.cs

Cette classe donne accès au clavier de réalité mixte de l'HoloLens 2. Pour ce faire, en cliquant sur le bouton « Keyboard » du préfabriqué présenté précédemment, une méthode de ce script est appelée. C'est aussi dans ce script que le « preview » du clavier et le texte du préfabriqué est géré.

MaterialManager.cs

Ce script permet de charger et de changer la couleur des matériaux utilisés par l'objet qui le possède. Les matériaux sont chargés depuis le dossier « Materials\Resource » situé dans les assets du projet.

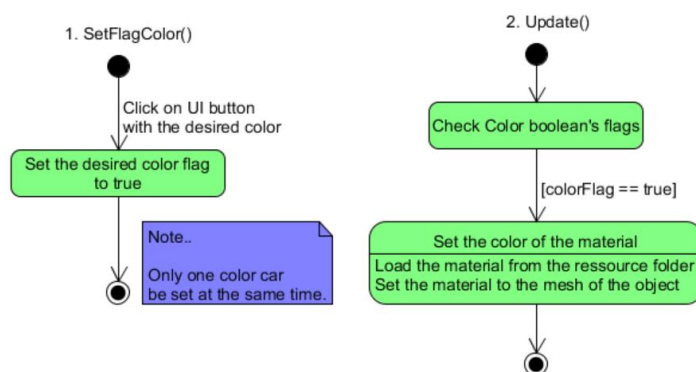


Figure 92 - Changement de couleur d'un matériau. MaterialManager.cs

Pour ce faire, quand ils sont cliqués, les boutons de la palette de couleurs appellent des méthodes permettant d'initialiser des « flags » booléens relatif à la couleur désirée.

Ensuite, dans la méthode `Update()` sont appelées les différentes fonctions nécessaires pour réaliser cette tâche. Ces fonctions ne sont pas appelées dans les méthodes « `SetColor()` » directement car elles ne peuvent qu'être exécutées dans le « `Main Thread` » de Unity. Il faut également veiller à ce qu'un seul flag de couleur ne soit actif à la fois (un objet ne peut pas avoir deux couleurs simultanément).

TextMeshColorManager.cs

Ce script est très semblable à `MaterialManager.cs`. A la différence que les objets changeant de couleurs sont les composants « `TextMeshPro` » des textes instanciés. Une différence supplémentaire est que les méthodes de changement de couleurs des textes n'ont pas besoin d'être appelées dans le « `Main Thread` » de Unity et peuvent être alors directement appelées depuis les boutons de couleurs sans passer par une méthode `Update()` comme pour les matériaux.

RemoteCommands Scripts

Dans ce dossier se trouvent les scripts relatif aux annotations à distance ainsi qu'aux connexions TCP/IP.

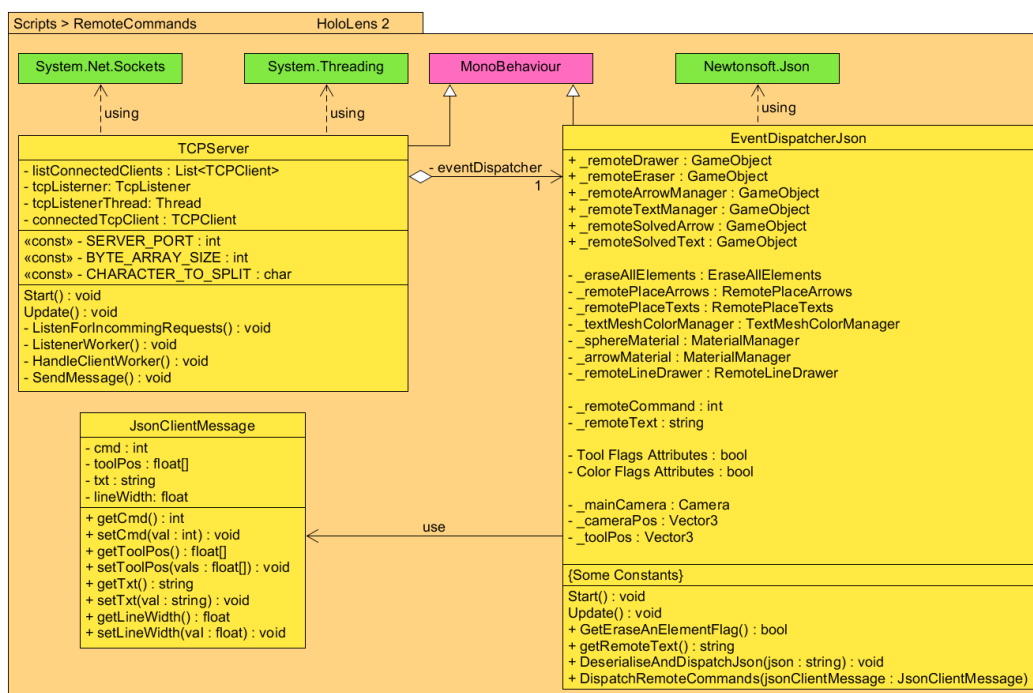


Figure 93 - Relation entre les classes des scripts situés dans RemoteCommands

TCPServer.cs

Ce script permet aux clients TCP de se connecter à l'application de l'HoloLens 2 qui est le serveur TCP de cette connexion et de recevoir leurs messages. Ci-dessous se trouve le principe de fonctionnement de ce script.

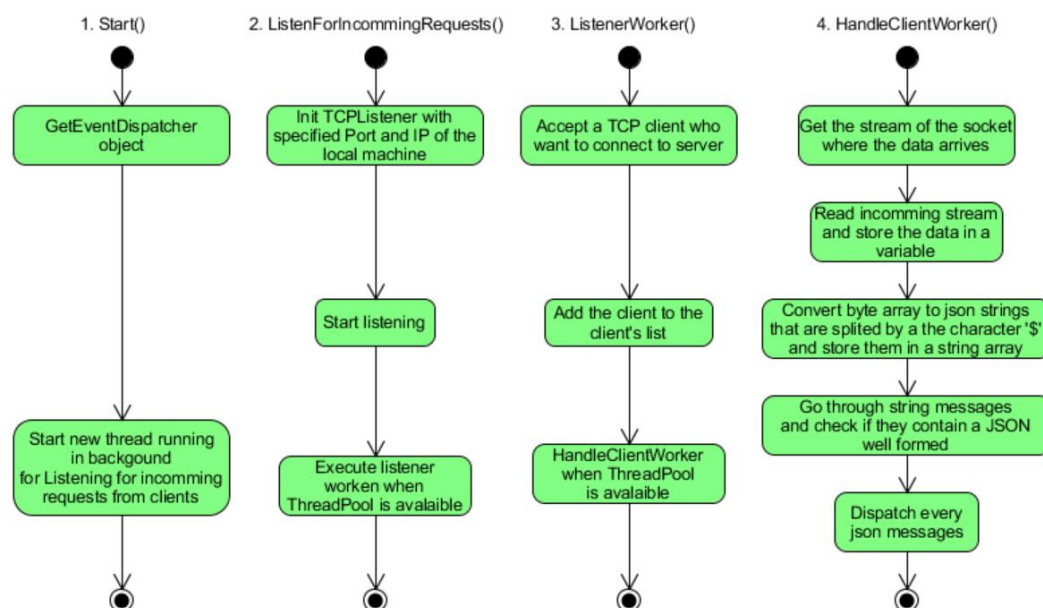


Figure 94 - Principes du script TCPServer.cs

Dans la méthode Start(), l'objet à qui les messages doivent être distribués est récupéré. Ensuite, un thread fonctionnant en arrière-plan est initialisé afin de recevoir les demandes de connexion des potentiels clients TCP.

Pour ce faire, il faut initier le serveur qui écoute si des requêtes arrivent avec une certaine adresse IP et un certain port. Cela étant fait, il peut écouter les tentatives de connexions et les accepter quand le « pool » du thread est disponible. Le serveur accepte alors un client et l'ajoute à sa liste de clients connectés. Ensuite, si le « pool » du thread est disponible, le flux provenant d'un client est récupéré permettant ainsi de lire les données provenant de ce dernier. Ces données sont ensuite converties de bytes en strings. A leur création, les messages JSON se voient ajouter un caractère marquant la fin de ces derniers afin de les séparer entre eux. Ainsi, à l'aide de la méthode « Split() » il est possible de récupérer tous les messages JSON et les stocker individuellement à une place précise d'un tableau de string. Ce tableau est parcouru et chaque message est contrôlé qu'il est bien formé à l'aide de leur caractère de début et de fin (« { » et « } »). Le reste du contrôle est fait avant l'envoi. Enfin, ces messages peuvent être transmis à la classe s'occupant de distribuer ces messages en fonction de leur contenu.

EventDispatcherJson.cs

Dans ce script, les messages JSON sont reçus depuis TCPServer.cs à l'aide de la méthode DeserilizeAndDispatchJson() qui, comme le nom l'indique, désérilise les JSON afin de récupérer ses données et l'envoie dans un dispatcher. Ensuite, le dispatcher distribue les actions à effectuer à l'aide d'un switch case ayant pour paramètre un nombre entier correspondant au premier attribut du JSON reçu. La distribution est faite comme suit :

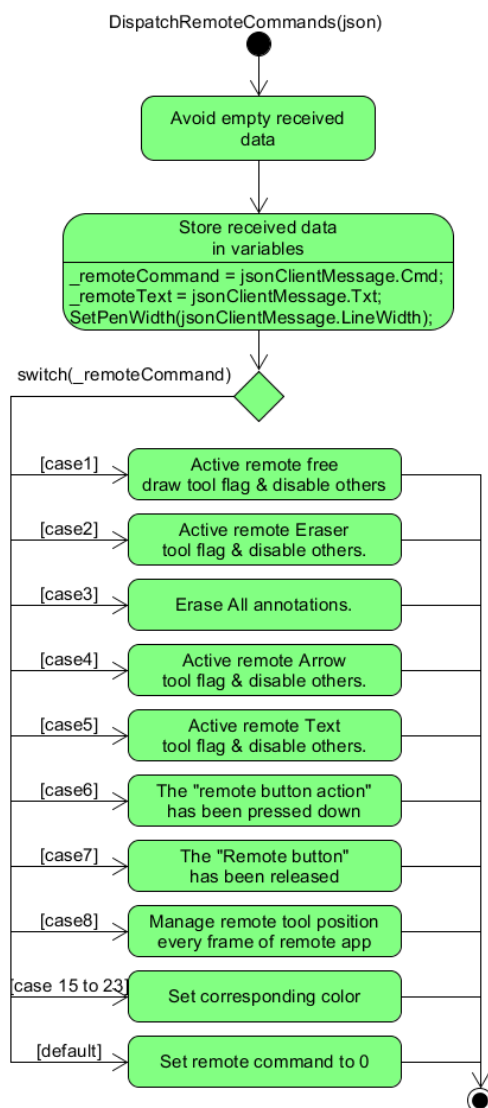


Figure 95 - Algorithme de la méthode DispatchRemoteCommands() de EventDispatcherJson.cs

Comme déjà expliqué précédemment, certaines méthodes doivent être appelées dans le « main thread » de Unity. C'est pourquoi, dans certains cas, seuls des valeurs booléennes sont assignées. Ceci permettra d'effectuer les actions désirées dans la méthode Update() en fonction de la valeurs des différents booléens. Il est à noter que pour réduire la taille de cette illustration, le paramétrage des différentes couleurs ont été regroupés en un seul block. En réalité, il faudrait un block par couleur avec les nombres entiers de 15 à 23. Les valeurs que peut prendre « _remoteCommands » sont celles listées dans le tableau du chapitre « Liste des commandes créées depuis le PC ».

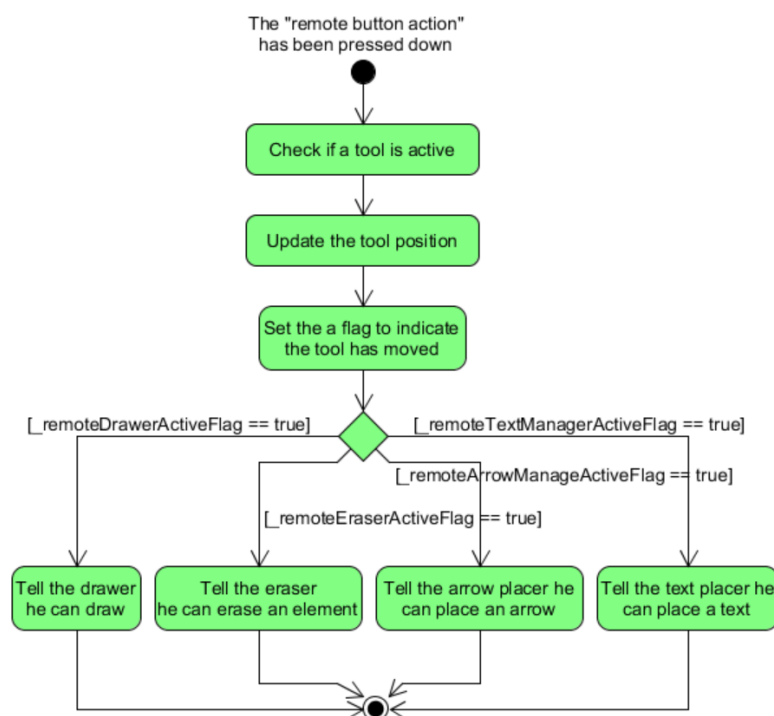


Figure 96 - Algorithme lorsqu'un bouton distant a été pressé. EventDispatcher.Json.cs

Ci-dessus se trouve l'algorithme se trouvant au cas 6 du switch présenté précédemment. Ce code est exécuté lorsqu'un client distant a appuyé une fois sur le bouton de la souris ou sur l'écran tactile du smartphone. D'abord, il est bien contrôlé qu'une fonction d'annotation est active et met la position de l'outil à jour si c'est le cas. Une valeur booléenne est mise à jour lorsque l'outil a changé de position. Ensuite, en dépend de la fonction active, certaines fonctionnalités liées aux annotations sont faites. Majoritairement, les valeurs booléennes de chaque script d'annotation, permettant de savoir si un utilisateur a fait un click, est mise à jour. Ainsi, il est possible de dessiner, d'effacer, de placer des flèches ou des textes.

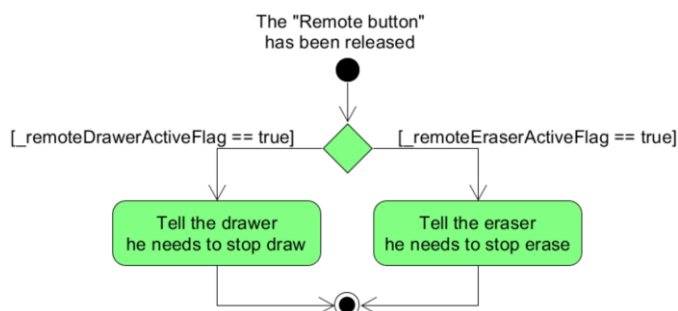


Figure 97 - Algorithme lorsqu'un bouton distant a été relâché. EventDispatcher.Json.cs

Ci-dessus se trouve l'algorithme se trouvant au cas 7 du switch case. Il est exécuté lorsqu'un client distant a relâché le bouton de la souris ou a cessé de toucher l'écran tactile du smartphone. Si un dessin libre se réalisait, le script de dessin doit arrêter de dessiner. Si la fonction d'effacement était active, le script d'effacement doit cesser d'effacer. Pour ce faire, les valeurs booléennes présentes dans ces scripts représentant l'action d'un client doivent être mises à jour.

Des méthodes permettent de mettre à jour les couleurs des outils d'annotations. Les éléments de couleur mis à jour sont : le trait à dessiner, la sphère de dessin suivant la main, les flèches et les textes. Une petite particularité est que la couleur du texte doit être appelée dans Update() dû au « main thread » d'Unity et ses flags doivent être initialisés dans le switch case.

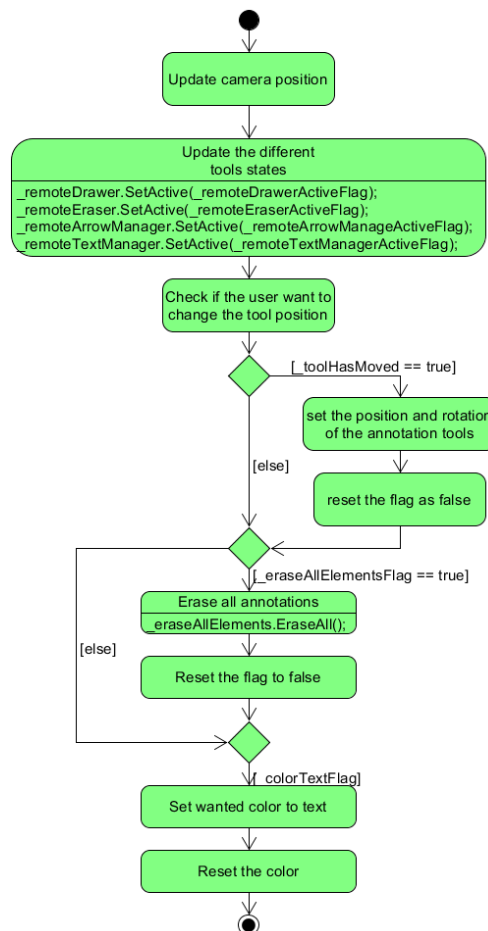


Figure 98 - Algorithme de la méthode Update de EventDispatcherJson.cs

Tout d'abord, dans cet algorithme, la position de la caméra est mise à jour afin de donner correctement la bonne orientation aux annotations. C'est ici que les flags initialisés plus tôt sont utiles. En effet, certaines méthodes, dont SetActive(), ne peuvent pas être appelées n'importe où. C'est pourquoi elles sont appelées ici ayant pour paramètre des flags. De la sorte, les outils d'annotations peuvent être activés ou désactivés. Ensuite, la position et l'orientation des outils peuvent être mis à jour. C'est également dans cette méthode que la fonction permettant d'effacer toutes les annotations par le client est appelée. Enfin, une série de flags permet de mettre à jour la couleur des textes.

Un point important concernant l'emplacement des outils d'annotation est à mettre en lumière. Initialement, les « axes d'annotations » étaient statiques. En effet, le référentiel des clients (spécialistes) étant statiques, les annotations à distances ne suivaient pas la direction de la caméra de la personne assistée. De la sorte, si la personne tournait sur elle-même, les annotations ne seraient plus en face de celle-ci lorsque l'utilisateur distant tentait de dessiner. Pour remédier à ça, les angles de rotation de la caméra de l'HoloLens 2 ont été récupérés et appliqués aux vecteurs des points placés par les spécialistes. En effet, ces points étant en réalité des vecteurs 3D, la rotation leur fait suivre la direction de la caméra.

RemoteLineDrawer, RemotePlaceArrows et RemotePlaceTexts

Ces scripts ne seront pas présentés en détails ici car leur implémentation est similaire à leur script d'annotation local respectifs. La différence avec ces derniers est qu'ils sont pilotés par des valeurs booléennes paramétrées dans le script distribuant les différentes fonctions souhaitées par les spécialiste distants.

WebRTC Scripts

NodeDssSignalerUI.cs

Ce script provenant de la bibliothèque MRTK-WebRTC étant simple de compréhension, il ne sera pas expliqué en détail. Il se trouve ici car il a été modifié de la sorte à pouvoir désactiver la caméra source afin de pouvoir l'utiliser dans une session multi-participants. Cette explication est faite dans le chapitre « 5.2.4 Désactivation de la caméra locale avant de créer un appel ».

7.2.6 Modèles 3D Préfabriqués sur Blender

Afin de servir pour les diverses fonctions d'annotations, des modèles 3D ont été créés sur la plateforme Blender. Ils ont été exportés au format « .fbx » et importés dans Unity.

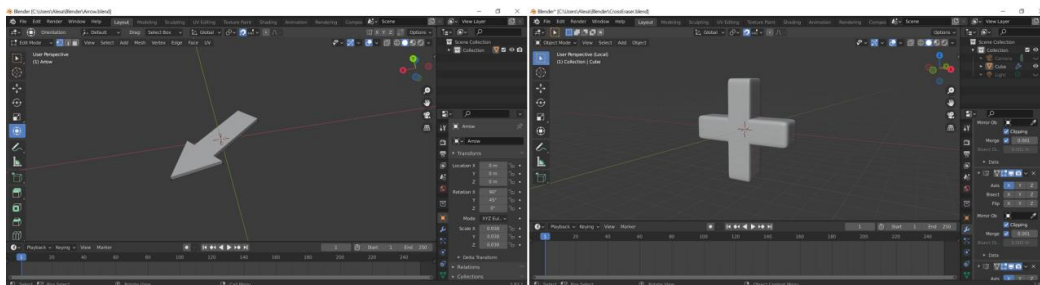


Figure 99 - Modèles 3D créés sur Blender

Ces préfabriqués se trouvent dans le dossier « Assets\Prefabs\3D_Models ». La flèche située à gauche ci-dessus, sert de repère pour l'annotation à l'aide de flèches. Celle-ci traque la main de l'utilisateur et, dès que ce dernier « pinch » avec ces doigts, une même flèche est instanciée à l'emplacement de la flèche suivant la main. A droite se trouve la croix permettant d'effacer des objets lorsqu'elle entre en collision avec ces derniers. Cette croix traque également la main de l'utilisateur à chaque instant.

Pour rappel, ALPS Automation SA souhaite avoir la possibilité de créer des annotations avec des formes prédéfinies. L'exemple du prototype réalisé avec la flèche montre que cette fonctionnalité est fortement envisageable. En effet, il est possible de créer une série de modèles 3D sur Blender et de les importer dans Unity. Ensuite, une fonctionnalité permettrait à l'utilisateur de choisir quel modèle 3D il souhaite instancier en tant qu'annotation. De plus, sur Unity, il est possible de leur assigner différentes couleurs et matériaux.

De plus, cela montre qu'il serait également possible d'importer des modèles 3D ou des modèles CAD tels que des armoires électriques pour mettre en synergie le terrain et le numérique. Dans les annexes se trouve les formats supportés par Unity.

7.2.7 Outils d'annotations Tracking

Ci-dessous sont présentés les différents paramètres nécessaires afin de faire en sorte qu'un objet suive la main de l'utilisateur. Ces configurations ont été faites pour la sphère de dessin, la croix effaçable, la flèche et une zone de texte à placer.

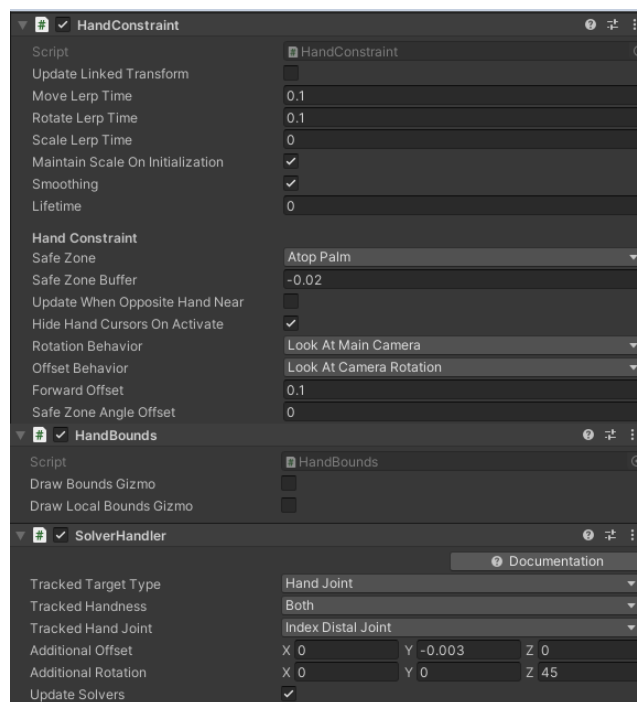


Figure 100 - Composants permettant de faire du tracking d'objet

Ci-dessus se trouve la configuration des composants MRTK permettant de faire du tracking d'objet. Parmi les paramètres importants se trouvent :

- Tracked Target Type : Définit qu'est-ce que l'objet doit suivre. Ici, c'est l'articulation de la main.
- Tracked Handness : Quelle main l'objet doit-il suivre ? Ici, gauche et droite.
- Tracked Hand Joint : Quelle articulation de la main l'objet doit-il suivre ? Ici, l'articulation distal de l'index. (Voir figure-ci dessous)

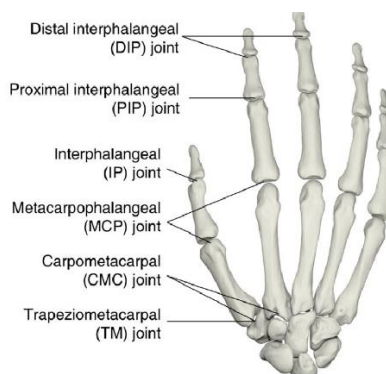


Figure 101 - Articulations de la main

Source : ResearchGate.net

https://www.researchgate.net/figure/Joints-of-the-right-hand-dorsal-view-Note-that-the-terms-trapeziometacarpal-TM-joint_fig1_257045252

- SafeZone : Dans quelle zone doit se situer l'objet ? Ici, devant la paume de la main.

7.2.8 Activation et désactivation des outils d'annotations en local

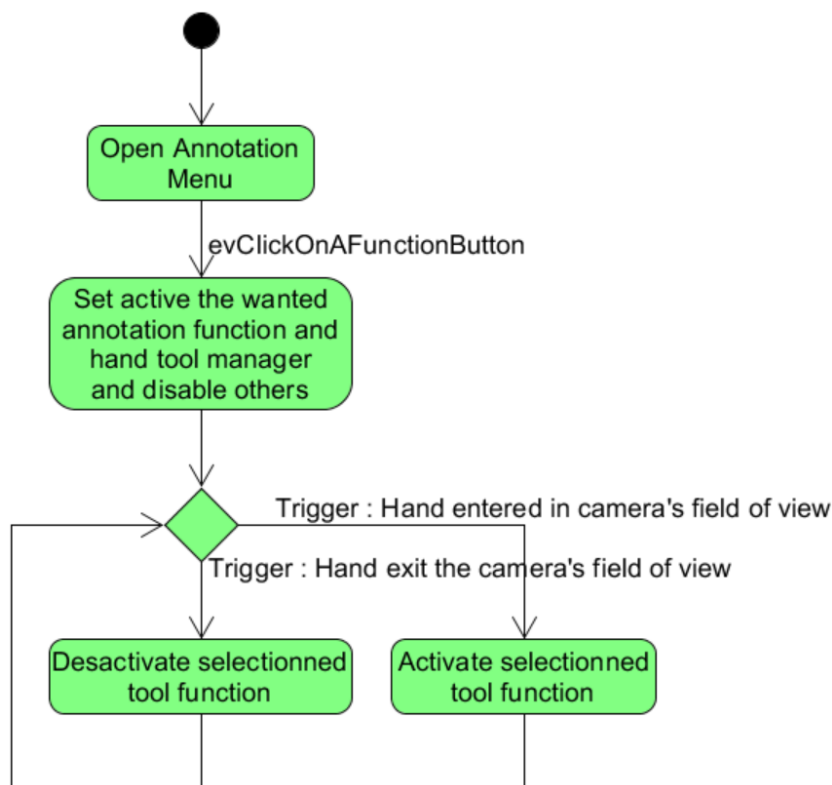


Figure 102 - Algorithme d'activation et désactivations d'outils d'annotations

Ci-dessus se trouve l'algorithme permettant de gérer les outils d'annotations en local. Tout d'abord, l'utilisateur de l'HoloLens 2 peut sélectionner le type d'annotations qu'il souhaite réaliser au moyen de boutons placés dans un menu d'annotations. En sélectionnant un outil, les autres sont automatiquement désactivés afin d'éviter des perturbations. Ensuite, à l'aide de deux fonctions « OnHandActivate() » et « OnHandDesactivate() » du composant HandConstraint présenté plus tôt, il est possible d'activer ou de désactiver les outils d'annotations lorsque les mains de l'utilisateur entrent ou sortent du champ de vision de la caméra de l'HoloLens 2. Ceci évite certains problèmes d'annotations non désirés.

7.2.9 Problème ProximityLight (résolu)

Un « ProximityLight » permet d'éclairer les objets pointés par l'utilisateur. Malheureusement, cette propriété est par défaut limitée à deux « ProximityLight », ce qui causait des problèmes d'interaction avec l'UI. Selon la doc Microsoft, il est possible d'augmenter cette valeur à quatre « ProximityLight ». En référence se trouve la documentation^[26] fournie par Microsoft pour résoudre ce problème.

²⁶ MICROSOFT DOCS, *Lumière de proximité*

<https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtrk-unity/features/rendering/proximity-light?view=mrtrkunity-2021-05> (consulté le 12 août 2021)

7.2.10 Conclusion de l'implémentation HoloLens 2

Dans ce prototype, les fonctionnalités d'appels et d'annotations ont été implémentées. Les fonctionnalités de captures instantanées, de vidéos ainsi que de partage de documents n'ont pas été nécessaires d'implémenter dû au fait qu'elles soient nativement réalisables par l'HoloLens 2 en parallèle de l'application prototype.

Les principaux algorithmes d'annotations distantes et locales ont été présentés. Il a été également montré la manière dont le serveur TCP gère les multi-connexions de clients TCP. Il a également été montré la façon dont les messages JSON reçus des clients sont traités.

7.3 Déploiement de l'application

Ce chapitre présente les différentes configurations Unity afin de déployer ce prototype sur un HoloLens 2.

Tout d'abord, la plateforme de déploiement doit être sélectionnée dans les « Builds Settings ». Ici, « Universal Windows Platform » doit être choisi. Les différents paramètres dans cette fenêtre peuvent être laissés par défaut comme le montre la figure ci-dessous.

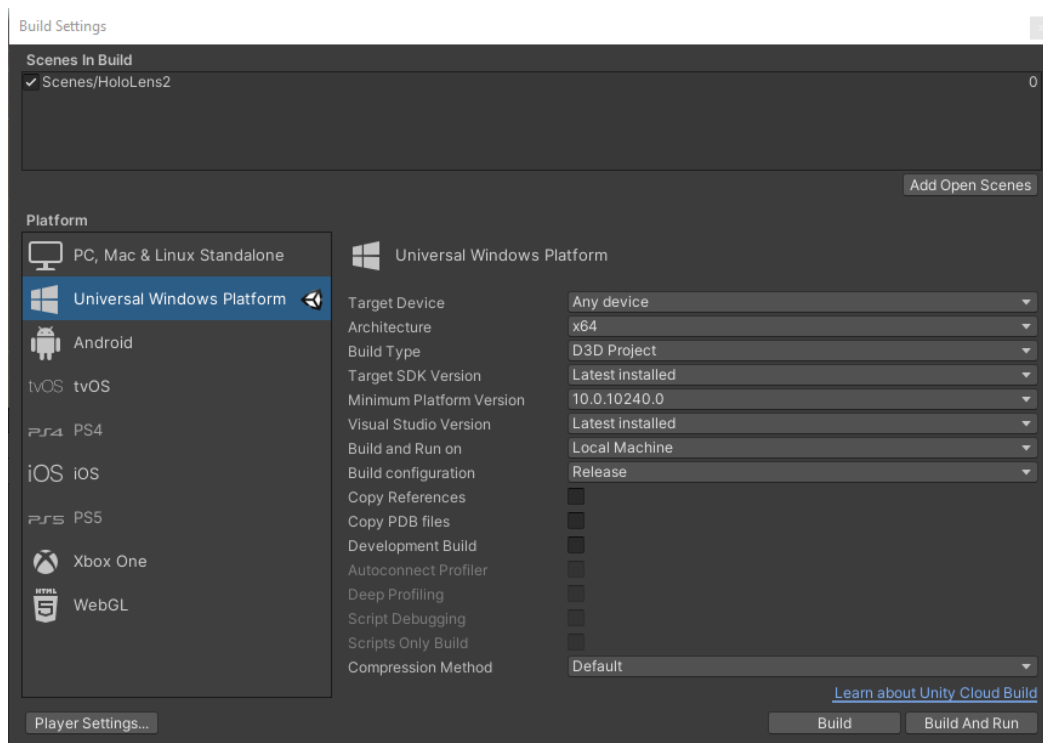


Figure 103 - Build Settings HoloLens 2

Ces modules de déploiement doivent être ajoutés à la version d'Unity installée. Ceci se fait à l'aide de la plateforme « Unity Hub ». (Unity Hub > Installs > Version Unity > Add Modules)

Dans les annexes se trouve une capture d'écran montrant les différentes configurations des modules Unity Hub afin de pouvoir déployer correctement.

Ensuite dans les paramètres de « Player », et l'onglet « Publishing Settings » certaines fonctionnalités doivent être activées. Ces dernières sont présentées dans la figure suivante :



Figure 104 - Fonctionnalités à activer pour un bon fonctionnement de l'application

Afin que l'HoloLens 2 puisse communiquer avec des appareils distants, il est important de cocher les capacités « InternetClient », « InternetClientServer » et « PrivateNetworkClientServer ». De plus, les capacités « WebCam » et « Microphone » doivent être également actives pour donner l'accès de ces périphériques à l'application.

Il est également important de configurer l'onglet « XR Settings » pour un bon fonctionnement de la réalité mixte. Ces propriétés sont illustrées dans l'image suivante :

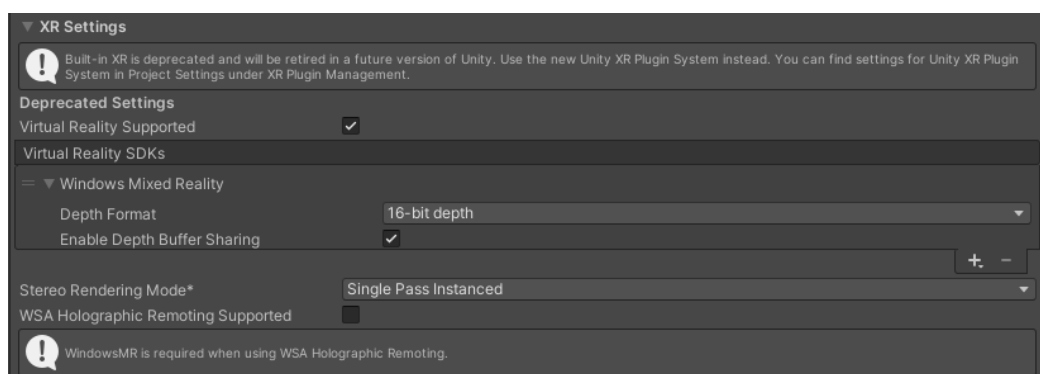


Figure 105 - XR Settings HoloLens 2

Cela étant fait, l'utilisateur peut « Build » son application. A la fin de la compilation, un fichier « .sln » est généré. À l'ouverture de ce dernier, VisualStudio est lancé et permettra de déployer l'application sur l'HoloLens 2.

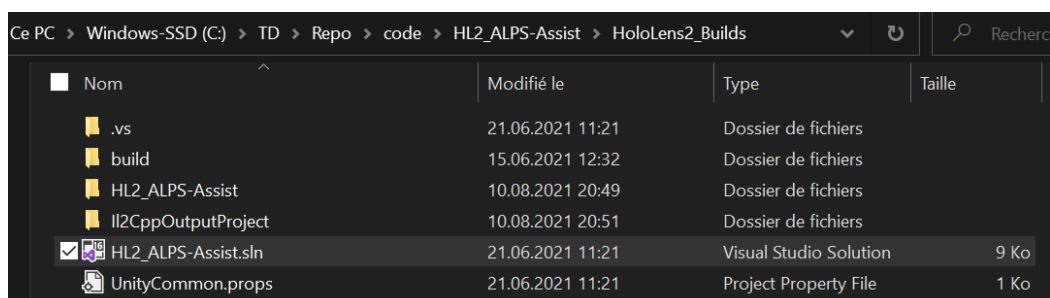


Figure 106 - Génération du fichier .sln

Visual Studio s'étant ouvert, certaines configurations doivent être apportées aux propriétés de cette plateforme. En effet, la méthode de déploiement doit être spécifiée. Pour ce faire, il faut choisir le mode de déploiement ainsi que l'architecture sur laquelle l'application sera déployée. Dans ce projet, le déploiement sur l'HoloLens 2 a toujours été fait via Wi-Fi.

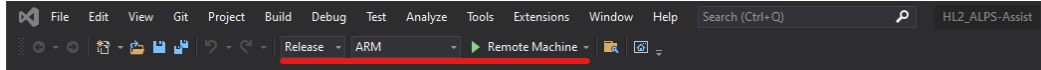


Figure 107 - Configuration du déploiement

Le déploiement étant fait par Wi-Fi, l'adresse IP de l'HoloLens 2 doit être spécifiée dans les configurations de déploiement comme le montre la figure ci-dessous.

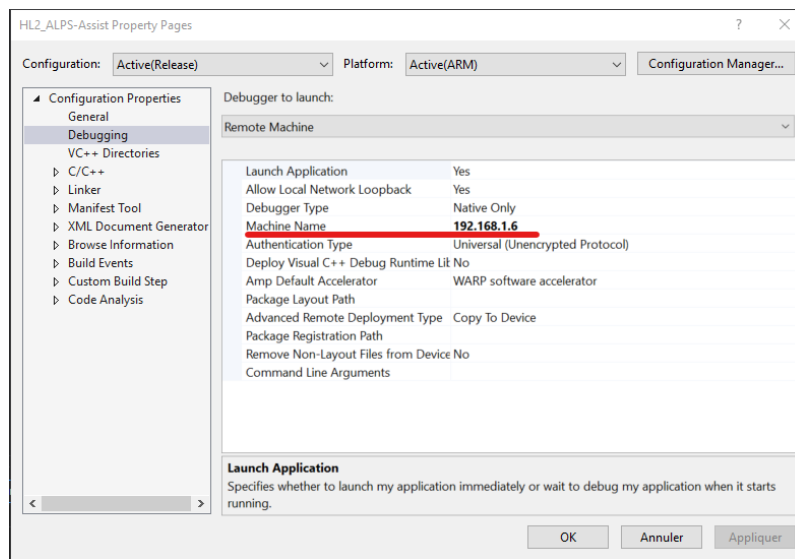


Figure 108 - Configuration de l'adresse IP de l'HoloLens 2 sur Visual Studio

Enfin, il est possible de déployer l'application sur l'HoloLens 2 en cliquant sur « Debug > Start without Debugging ». A la fin de l'opération, l'application se retrouve sur L'HoloLens 2.

Ci-dessous se trouve une photo de l'application déployée.

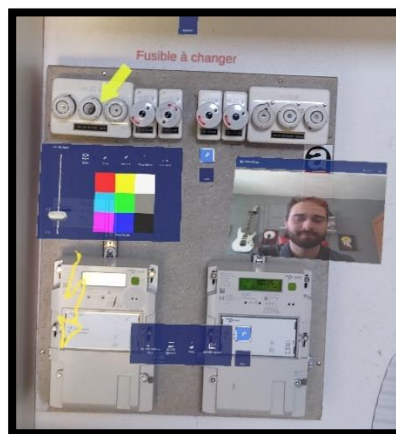


Figure 109 - Application déployée sur l'HoloLens 2 avec certaines menus et annotations présents

7.4 Validation

Ici seront présentés les résultats des différents tests « black-box » effectués pour le prototype HoloLens 2. Ces derniers seront présentés sous forme de tableaux avec les résultats obtenus. Afin de ne pas surcharger la lecture de ce rapport, ces tableaux sont placés en annexes.

Les tests liés à la communication audio et vidéo possèdent un ID « Callx_HL2 », ceux liés aux annotations locales « Annx_HL2 » et ceux liés à L'UI ont un ID « UIx_HL2 ».

Les différents tests ont permis de vérifier le bon fonctionnement du prototype HoloLens 2.

Les résultats ont montré que le prototype HoloLens 2 est capable de créer une communication audio et vidéo avec les prototype Smartphone et PC. De plus, il est également capable de recevoir les appels provenant de ces derniers. L'image vidéo et le son sont correctement transmis et reçus.

Ensuite, les différentes fonctionnalités d'annotations locales montrent également un bon fonctionnement pour la réalisation d'annotations libres, en flèches et en textes. De plus, ces dernières s'effacent correctement.

Les tests des prototypes PC et Smartphone ont permis de contrôler également le bon fonctionnement du serveur TCP ainsi que la réception des commandes à distance. Toutefois, un manque de stabilité concernant les outils distants a été perçu. Parfois, il est arrivé que ces derniers aient disparus.

Un autre test effectué avec cet appareil a été consacré à la reconnaissance vocale. Dans un studio de musique acoustiquement isolé, deux enceintes gauche et droite ont été placées aux extrémités de la pièce. Sur la plateforme « Youtube », une vidéo reproduisant une ambiance sonore industrielle a été lue depuis les enceintes. L'HoloLens 2 était placé entre ces dernières. Le testeur énonçait des commandes vocales à voix haute afin de tester la bonne reconnaissance vocale en dépend du nombre de décibels à son emplacement. Le volume de la vidéo est monté jusqu'à 100 décibels à l'emplacement de l'HoloLens 2. L'appareil a reconnu des commandes telles que « Menu démarrer », « Faire une photo », « Faire une vidéo », « Arrêter la vidéo ». Le volume aurait pu être encore augmenté mais cela ne faisait plus de sens du fait que le testeur n'entendait même plus sa voix et devait crier pour s'entendre. Pour mesurer les décibels, deux décibel mètres ont été utilisés.

Enfin, un dernier test était consacré au partage de fichiers et à la lecture de ces derniers depuis la plateforme OneDrive. Les spécialistes ont partagé différents documents sur OneDrive depuis leur PC, lesquels ont pu être téléchargés et lus sur l'HoloLens 2. La personne assistée a également pu partager des documents sur cette plateforme.

Un chapitre consacré aux tests utilisateurs est présenté plus tard dans le rapport. (cf. 10. Scénario d'ALPS Automation SA et tests d'usabilité)

8. Prototype Smartphone/Tablette : Spécialiste

Ce chapitre traite du développement du prototype se déployant sur un smartphone ou une tablette pour un spécialiste. Ce chapitre sera séparé en quatre parties : Conception, Implémentation, Déploiement et Validation.

8.1 Conception

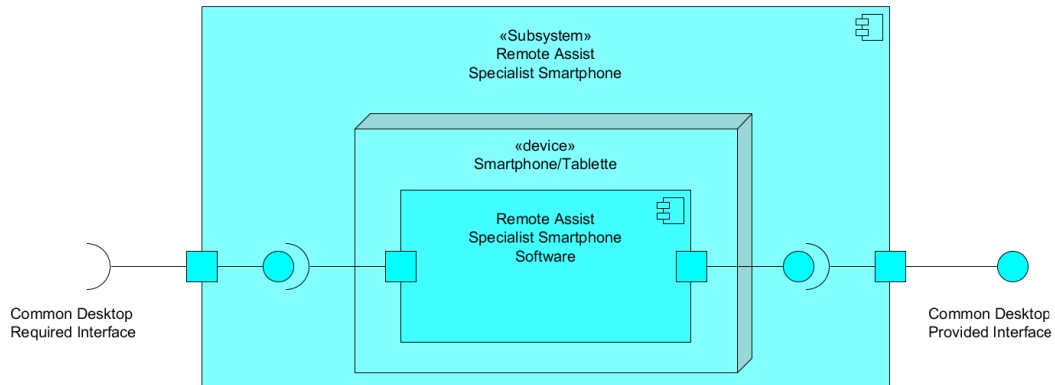


Figure 110 - Layer 3 : Smartphone/Tablette Spécialiste

Cette application prototype requiert et offre les différentes interfaces communes discutées au chapitre « 4.1.2 Layer 2 – Diagramme de composants général (Spécialiste) ».

8.1.1 Définition de l'architecture du système

Cette partie décrit l'architecture de l'application prototype déployée sur un smartphone ou une tablette pour un spécialiste. Les outils et bibliothèques de développement servant à la réalisation de ce projet seront présentés avec certains de leurs composants permettant de fournir les différentes interfaces.

Layer 4 : « Remote Assist Specialist Smartphone Software »

Tout d'abord, la figure suivante présente le sous-système représentant l'application smartphone ou tablette pour une personne assistée. Cette dernière a été réalisée à l'aide de Unity 2D 2019.4.26.f1 qui dispose encore d'un support à long terme.

Ce sous-système requiert et fournit les différentes interfaces communes discutées plus tôt à travers ces différents ports. Ici, spécifiquement, le port servant à la capture d'interactions utilisateurs est l'écran tactile de l'appareil. Ensuite, la capture audio et vidéo ainsi que la connexion Internet se font à l'aide d'un microphone, d'une caméra et d'un récepteur Wi-Fi ou d'une connexion 4G.

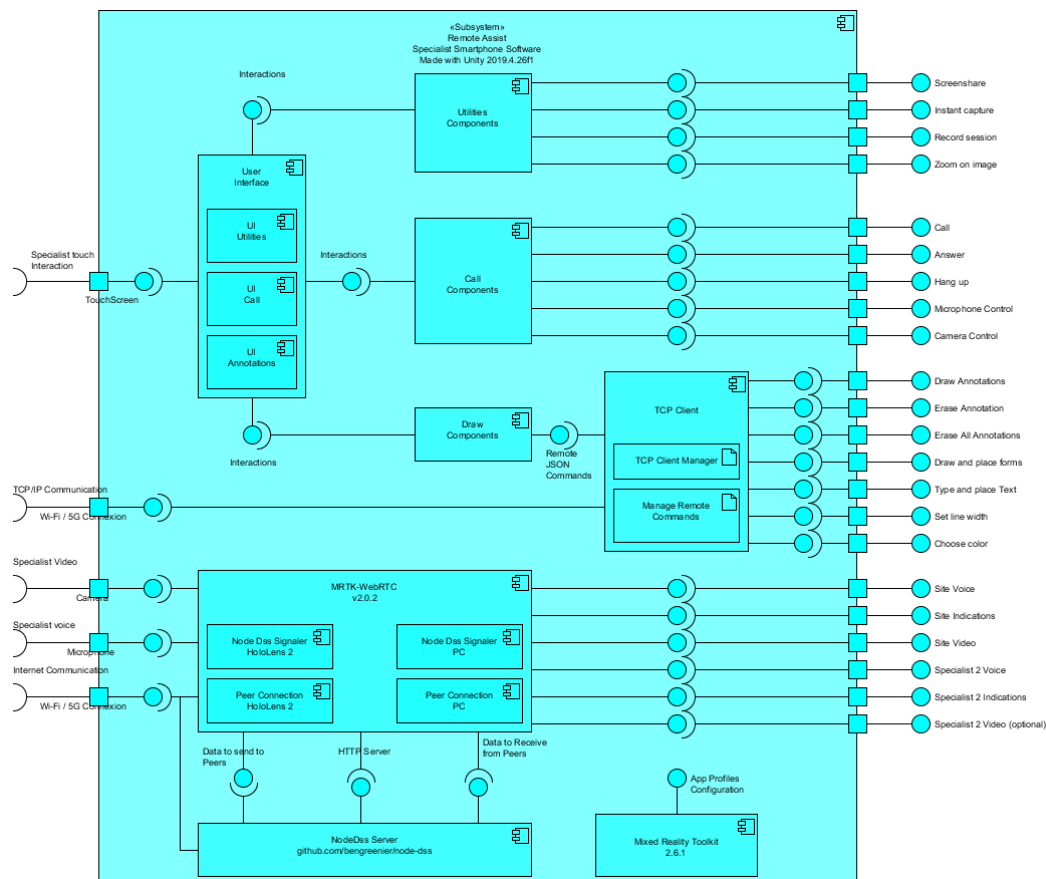


Figure 111 - Layer 4 : Remote Assist Specialist Smartphone Software

Afin d'alléger ce rapport, cette rubrique suggère de se référer au chapitre « Layer 4 – Remote Assist PC Software » qui traite de la conception de l'application PC. En effet, ces deux étant très similaires, la lecture du chapitre proposé ainsi que l'image ci-dessus suffisent à détailler ce concept de façon générale. Des clarifications concernant les différences entre les deux concepts seront tout de même apportées ci-dessous.

Tout d'abord, comme énoncé plus tôt, le schéma précédent montre que le port utilisé pour capturer les interactions de l'utilisateur est l'écran tactile de l'appareil.

A l'instar de l'ordinateur, le smartphone possède un récepteur Wi-Fi pour se connecter à un Internet. En outre, ce dernier peut facilement être équipé d'une carte SIM qui lui fournirait une connexion internet 4G ou 5G.

Ensuite, quelques modifications mineures sont également apportées à la configuration WebRTC. En effet, tout en gardant l'HoloLens 2 comme premier pair, d'autres « Signaler » et « Peer Connection » doivent être implémentés spécifiquement pour un autre appareil pair.

Enfin, pour effectuer certaines configurations au niveau de l'application smartphone concernant les fonctionnalités de réalité mixte, notamment le fait d'accéder à la caméra, un composant « Mixed Reality Toolkit » doit être présent. Il est à noter que ce composant n'était pas nécessaire pour l'application PC mais aussi nécessaire pour l'application HoloLens 2.

8.1.2 Définition de l'interface utilisateur

Ce sous-chapitre présentera l'UI avec laquelle un spécialiste pourra interagir avec son smartphone ou sa tablette. Il est à noter qu'étant donné la petite taille d'écran d'un smartphone et l'importance d'avoir une grande surface dédiée à la diffusion de la vidéo de la personne sur site, la surface de l'UI se verra restreinte

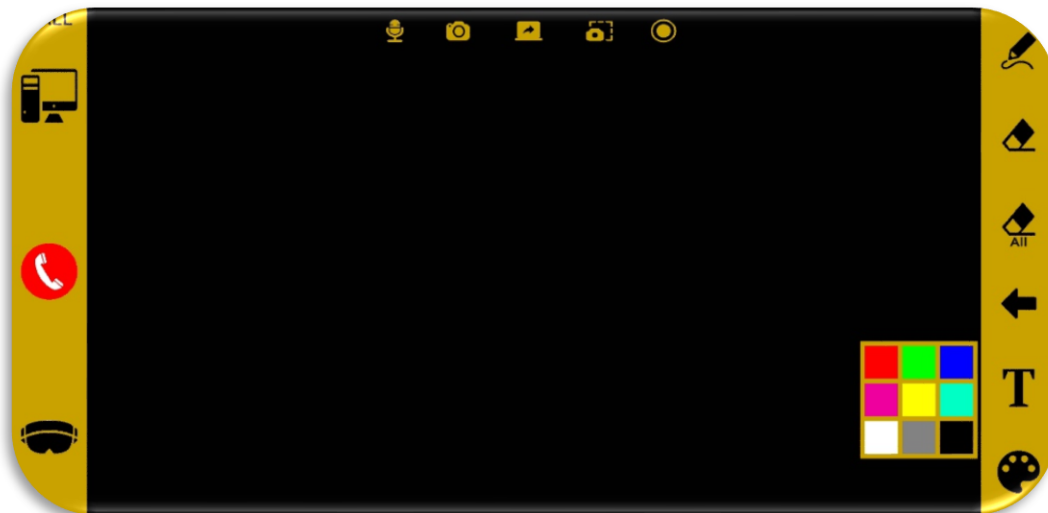


Figure 112 - Interface utilisateur de l'application spécialiste déployé sur smartphone

Tout d'abord, pour obtenir une grande surface de qualité réservée au lecteur vidéo, l'application sera développée en format paysage. De ce fait, toute la surface noire sur la figure ci-dessus accueillera la vidéo de la personne sur le site. Ensuite, dans l'idée d'optimiser un maximum de place, le prototype sera majoritairement réalisé à l'aide d'icônes représentant les diverses fonctionnalités.

Sur la gauche de la figure se trouvent les boutons liés aux appels. Le logo représentant un ordinateur appellerait le spécialiste situé dans son bureau et le logo représentant le casque de réalité augmentée appellerait la personne sur le site. Le bouton rouge situé entre deux servirait à couper la communication avec les pairs distants.

Sur la droite, à l'instar de l'application PC, se trouvent les boutons permettant de réaliser les annotations à distance. De haut en bas, les fonctionnalités suivantes peuvent être activées : dessins libres, effacer une annotation, effacer toutes les annotations, placer une flèche et placer du texte éditable. Tout en bas, se trouve un logo représentant une palette de couleur. En cliquant sur ce dernier, un menu avec neuf couleurs apparaît comme illustré dans la figure précédente. Concernant le changement de taille de trait de dessin, il pourrait être imaginé qu'une zone de l'écran servant de « slider » permettrait de l'ajuster.

Concernant les déplacements horizontaux et verticaux des outils de dessin distants, ils seraient contrôlés par la position à laquelle l'utilisateur appuie sur l'écran à l'aide d'un seul doigt. Ensuite, en utilisant deux doigts et en les rapprochant ou en les éloignant l'un de l'autre, l'utilisateur peut gérer la profondeur des outils.

8.1.3 Conclusion de la conception smartphone spécialiste

Tout comme pour les prototypes précédents, cette conception permet de commencer l'implémentation de l'application smartphone pour un spécialiste. La structure du projet ainsi que la définition de l'interface utilisateur servent de guide pour implémenter correctement les différentes fonctionnalités.

La modularité est également mise en avant. Lorsqu'un appareil dispose des différents ports nécessaires à l'implémentation des différentes interfaces, il peut être utilisé pour ce prototype. C'est pour cela qu'il n'y a pas de distinction entre l'utilisation d'un smartphone ou d'une tablette.

De plus, grâce à MRTK-Foundation et AR Foundation, il est possible de déployer sur Android ou iOS, simplement en faisant certaines configurations dans le projet sans changer les composants de ce dernier.

Pour rappel, étant très similaire à l'application PC, des détails supplémentaires peuvent être trouvés dans les chapitre dédiés à cette dernière. Il est important de rappeler également que l'application smartphone est différente de l'application PC à cause du port d'interaction qui diffère entre ces deux (écran tactile contre clavier et souris) et de la configuration « Mixed Reality Toolkit ».

8.2 Implémentation

Cette section traitera de l'implémentation du prototype smartphone pour spécialiste. Elle présentera les paquets utilisés, la hiérarchie de composants du projet de façon générale et détaillera certains principes implémentés.

8.2.1 Paquets et librairies externes

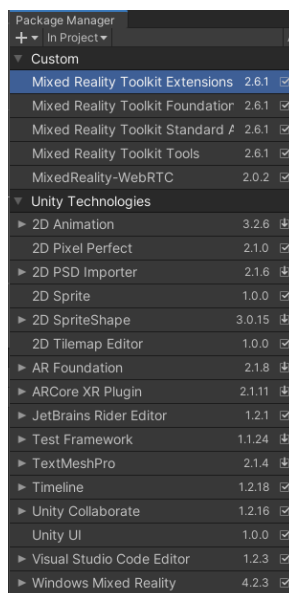


Figure 113 - Paquets importés dans Unity pour le prototype Smartphone (Spécialiste)

A l'instar du projet PC, ce prototype a besoin du paquet MixedReality-WebRTC v2.0.2 pour la communication audio et vidéo. La figure montre que le prototype s'est vu installer les différents paquets de la collection MixedRealityToolkit comme pour le prototype de l'HoloLens 2.

Finalement, hormis Foundation 2.6.1, ces paquets n'ont pas été très utiles à ce prototype spécialiste. Ils auraient été très utiles à la création de réalité mixte pour le prototype smartphone pour la personne assistée. Il est tout de même bien de savoir cela et c'est la raison pour laquelle ils ont été gardés et présentés ici.

Cette librairie fournie par Microsoft a besoin de « AR Foundation », « ARCore XR Plugin » et « Windows Mixed Reality » comme énoncé dans la conception. Ces principes sont expliqués plus en détail dans la conception du prototype smartphone pour personne assistée.

Les fonctions de texte de base de Unity étant un peu dépassées, TextMeshPro a été importé afin de remplacer ces dernières.

Dans le but de travailler avec les fonctionnalités JSON, le paquet Newtonsoft.json a été importé à l'instar du prototype PC.

8.2.2 Hiérarchie Unity

La hiérarchie ci-dessous présente les principaux objets instanciés dans la scène. Ces derniers peuvent disposer d'objets « enfants », mais pour des raisons de lisibilité, seuls les objets de « haut-niveau » ont été présentés ici.

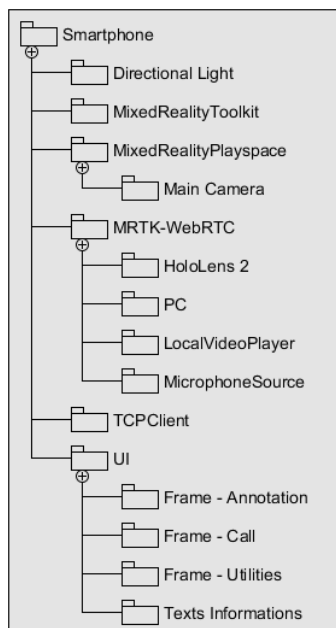


Figure 114 - Smartphone : Hiérarchie des objets Unity

Tout d'abord, dans cette hiérarchie se trouvent les objets « Directional Light » et « Camera ». Ces éléments sont déjà expliqués dans le chapitre « 5. Implémentation générale ».

Tout comme pour l'HoloLens 2, le composant MixedRealityToolkit est présent. C'est grâce à ce dernier que les différentes configurations de réalité mixte peuvent être faites. Ici, il a permis de donner accès à la caméra de l'appareil.

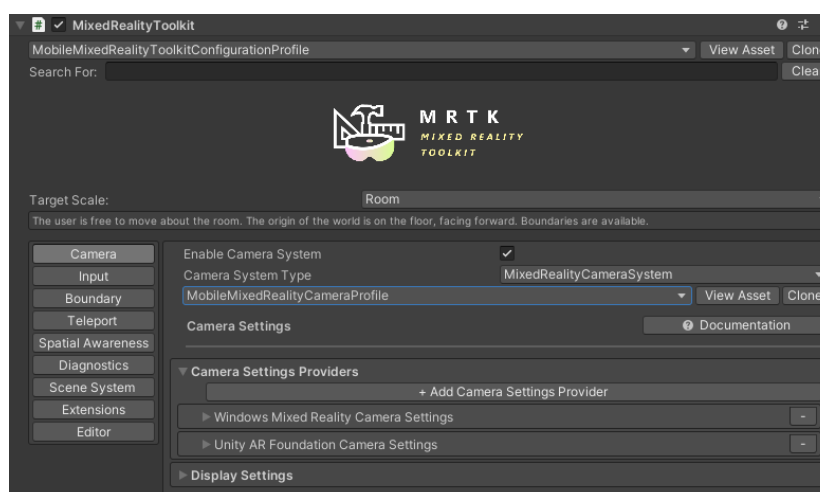


Figure 115 - MixedRealityToolkit Component

Tout comme pour le PC, un « parent » important est « MRTK-WebRTC ». C'est dans celui-ci que se trouveront les objets liés à la communication vidéo et audio. Dans ce dernier, les objets seront une fois de plus séparés de manière logique. Ensuite, un composant « TCPClient » disposera des scripts liés à la connexion TCP pour transmettre des commandes à distance. Enfin, un dernier « parent » est consacré à la réalisation de l'interface utilisateur. Ce dernier est séparé quatre parties : UI d'annotations, UI d'appels, UI de fonctions utiles et les différents textes à afficher sur l'UI.

8.2.3 Implémentation de MRTK-WebRTC

Dans les « parents » HoloLens 2 et PC se trouveront les objets propres à chaque appareil. Ces derniers disposeront chacun des objets suivants : PeerConnection, Signaler, NodeDssSignalerControls, RemoteVideoPlayer et RemoteAudio player. De cette façon il est possible de créer des appels multi-participants.



Figure 116 - Enfants de HoloLens 2 et PC

Enfin, « LocalVideoPlayer » et « Microphone Source » font également partie du parent « MRTK-WebRTC ». Ce sont ces composants qui donneront accès à la caméra et au microphone du smartphone. En référence se trouve un lien vers le tutoriel¹⁵ utilisé pour implémenter ces composants. De la sorte, ceci ne sera pas décrit ici pour alléger ce rapport. Les informations pour la mise en œuvre des appels multi-participants se trouvent dans le chapitre « 5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC ».

8.2.4 Implémentation des scripts

Ce chapitre décrira les points importants des différents scripts utilisés dans ce prototype. Des dossiers séparent ces derniers selon le schéma ci-dessous.

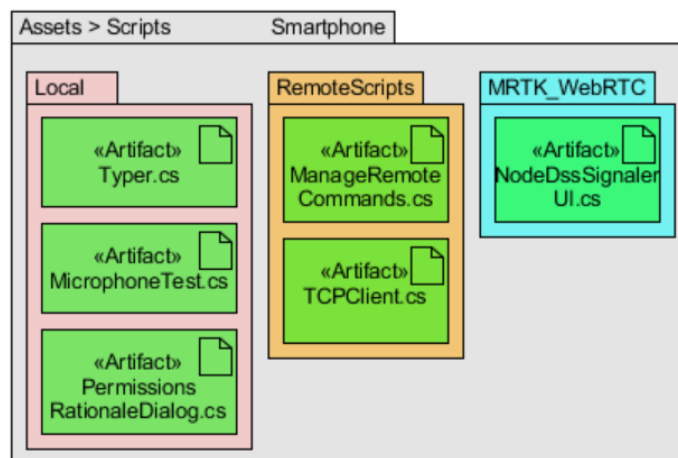


Figure 117 - Répartition des scripts dans le projet Unity Smartphone

Local

Typer.cs

Tout d'abord, bien que les noms soient les mêmes, le contenu de ce script diffère de son équivalent du prototype PC. En effet, l'accès au clavier et aux touches pressés par l'utilisateur est différent. Il sert également à l'écriture d'annotations textes à distance.

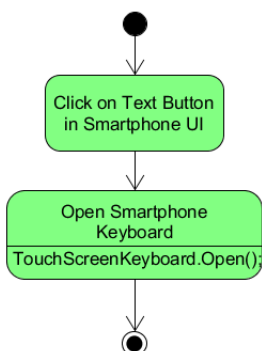


Figure 118 - Typer.cs Smartphone : OpenKeyboard()

Pour ce faire, l'utilisateur clique sur le bouton d'annotation texte sur son smartphone. Dès lors, un évènement est envoyé et ouvre le clavier natif du smartphone à l'aide de la méthode « Open() » de la classe TouchScreenKeyboard^[27] fournie par Unity.

A chaque frame Unity, dans la méthode Update(), le principe suivant est utilisé.

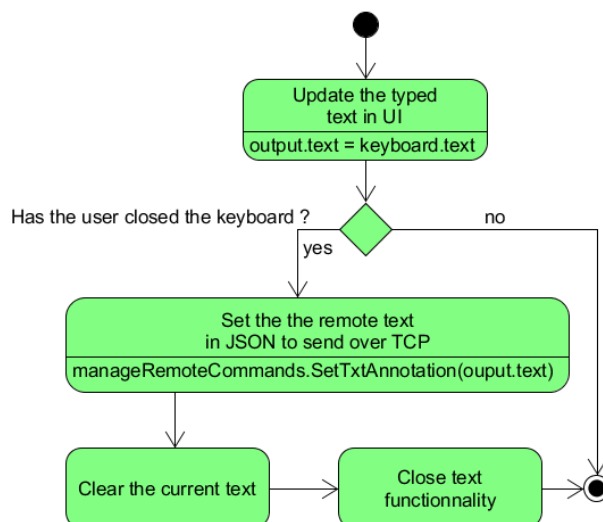


Figure 119 - Typer.cs Smartphone : Update()

Le texte écrit est mis à jour et affiché sur l'écran du smartphone afin de donner un aperçu de ce qui est écrit. Ensuite, si le clavier du smartphone a été fermé, sachant qu'il a été ouvert au préalable, le texte est envoyé à la classe s'occupant de créer les messages JSON afin de l'envoyer dans le flux TCP. Dès lors, le texte est effacé et cette fonctionnalité est fermée dans le but de répéter l'opération.

²⁷ UNITY DOC, TouchScreenKeyboard

<https://docs.unity3d.com/ScriptReference/TouchScreenKeyboard.html> (consulté le 9 août 2021)

MicrophoneTest.cs & PermissionsRationaleDialog.cs

Ces scripts, provenant de la documentation Unity, permettent à l'application de demander à l'utilisateur si elle peut avoir accès au microphone du smartphone. Ceci est réalisé dans le but d'avoir le microphone actif pour la communication WebRTC. En référence se trouve un lien vers cette documentation^[28].

RemoteScripts

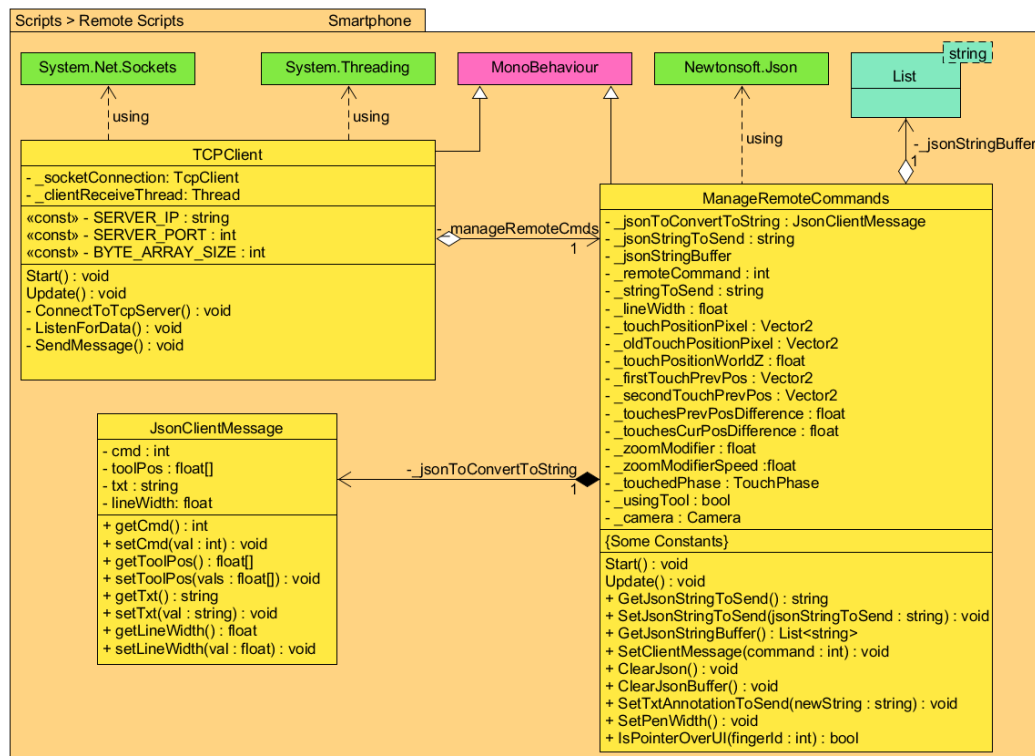


Figure 120 - Smartphone : Diagramme de classes du dossier Remote Scripts

Ce diagramme de classe est très similaire à celui du prototype PC. Ainsi, pour les explications principales, ce paragraphe renvoie au chapitre « TCPClient Scripts ».

Les différences se trouvent dans les attributs servant au déplacement de l'outil d'annotation distant, dans la protection des boutons de l'UI ainsi que dans la méthode Update() de ManageRemoteCommands. En effet, contrairement au prototype PC, l'utilisateur n'utilise pas une souris pour le contrôle de l'application mais interagit avec l'écran tactile avec ses doigts. Ces différences sont présentées ci-après.

²⁸ UNITY DOCUMENTATION, *Requesting Permissions*

<https://docs.unity3d.com/Manual/android-RequestingPermissions.html> (consulté le 12 août 2021)

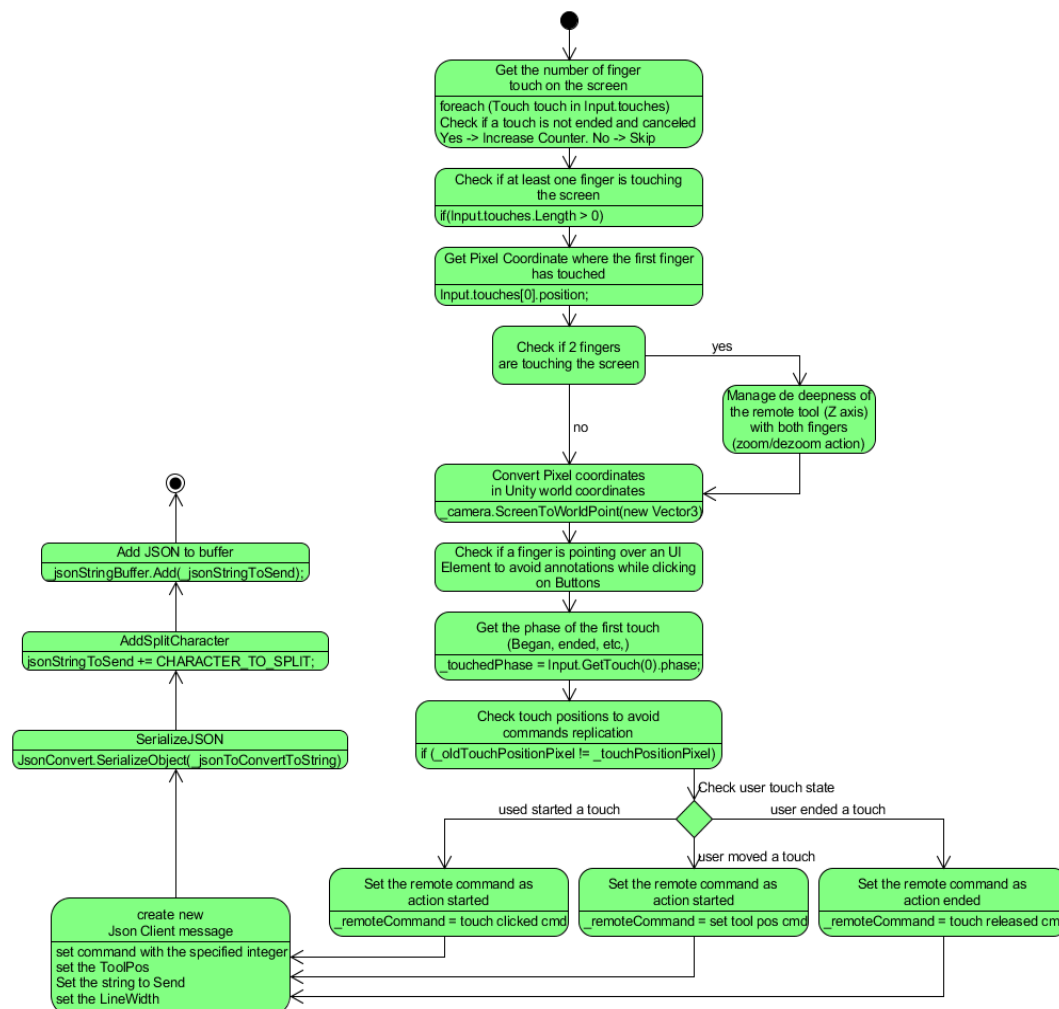


Figure 121 - ManageRemoteCommands.cs Smartphone : Update()

Tout d'abord, afin de créer les divers messages JSON relatifs aux actions souhaitées, il faut récupérer le nombre de doigts touchant l'écran du smartphone. En effet, les actions ne sont pas les mêmes s'il y a un doigt, deux doigts ou aucun doigt sur l'écran tactile. Si au moins un doigt touche l'écran tactile, sa position en pixel est alors récupérée. Ensuite, si le nombre de doigts touchant l'écran était de deux, la profondeur de l'outil distant peut être gérée comme pour la molette de la souris sur le prototype PC. Le fonctionnement de cette technique sera présenté plus tard. Dès que les actions de positionnement ont été réalisées, les coordonnées en pixels sont converties en unités du monde Unity. Dès lors, une méthode permet de vérifier si l'utilisateur a cliqué sur un bouton de l'UI afin d'éviter de placer des annotations non-désirées en cliquant sur celui-ci. Ensuite, grâce aux propriétés fournies par Unity, il est possible de connaître l'état de la touche. Parmi ces états, les suivants ont été utilisés : commencement d'une touche, fin d'une touche, déplacement d'une touche. Ensuite, afin d'éviter de renvoyer des messages à doubles (lorsque la position du doigt n'a pas bougé), un contrôle sur la position de la touche est effectué. Dès lors, en fonction de l'état de la touche, le message JSON est créé avec un nombre entier spécifique à l'action de la touche. Enfin, comme expliqué plusieurs fois dans ce rapport, le JSON est transformé et ajouté à la liste des messages créés.

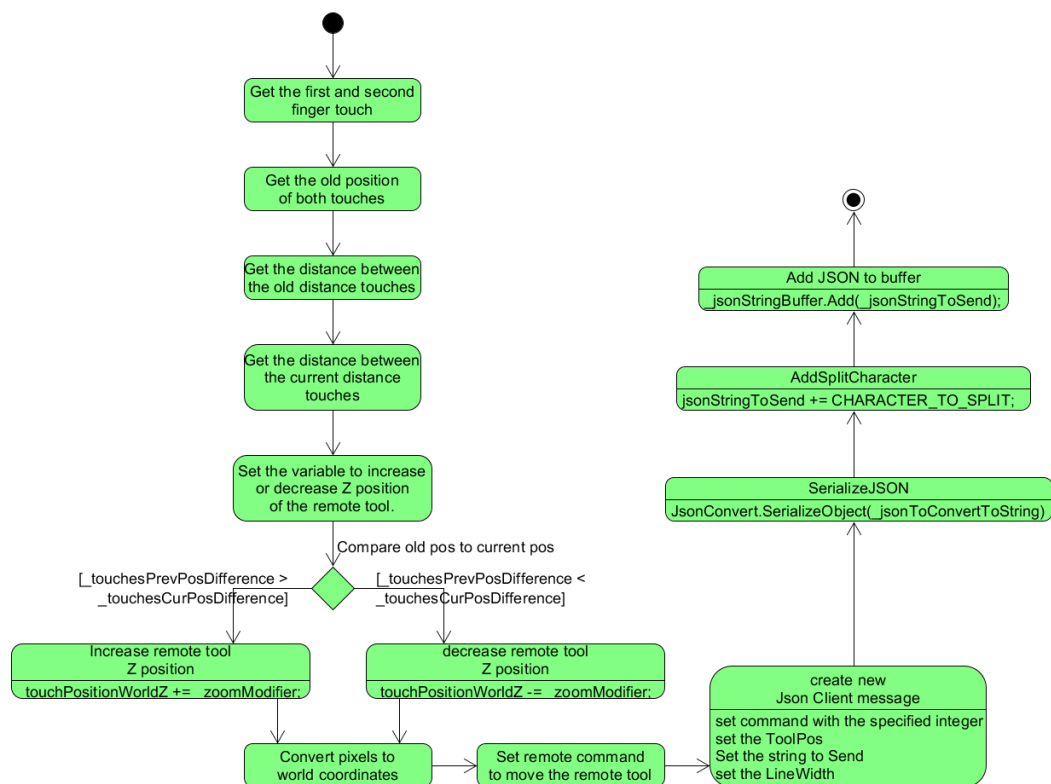


Figure 122 - ManageRemoteCommands Smartphone : in Update(). How to move in Z axis the remote tool

L'algorithme ci-dessus présente la méthode utilisée pour déplacer l'outil distant dans son axe de profondeur (axe Z). Pour ce faire, deux doigts sont nécessaires. L'action à effectuer est la même que la fonction « zoom » et « dézoom » des smartphones standards.

Pour implémenter cette fonctionnalité, il faut récupérer les touches faites par les deux doigts. Ensuite, il est nécessaire de connaître la position de ces doigts à la frame précédente ainsi que la distance entre ces deux. Ensuite, la position actuelle entre les deux doigts est également calculée. Dès lors, il est possible de calculer la variable qui incrémentera ou décrémentera la position en Z de l'outil distant. Si la distance entre les deux doigts était plus grande dans la frame précédente que la distance actuelle, cela signifie que l'action de « dézoom » a été effectuée. Dans ce prototype, cela correspond à incrémenter la position en Z de l'outil distant afin de l'éloigner et d'avoir la sensation de le rapetisser. A l'inverse, cela signifie qu'une action de « zoom » a été effectuée sur l'écran et rapproche ainsi l'outil distant du point de vue de l'utilisateur. Cette position étant fixée, il est alors possible de la convertir en unités du monde Unity et de créer le message JSON comme à son habitude et de l'ajouter au buffer de JSON.

MRTK_WebRTC Scripts

NodeDssSignalerUI.cs

Ce script provenant de la bibliothèque MRTK-WebRTC étant simple de compréhension, il ne sera pas expliqué ici. Il se trouve ici car il a été modifié de la sorte à pouvoir désactiver la caméra source afin de pouvoir l'utiliser dans une session multi-participants. Cette explication est expliquée dans le chapitre dédié à ce problème. (cf. 5.2.4 Désactivation de la caméra locale avant de créer un appel)

Liste des commandes créées depuis le Smartphone

Celles-ci sont exactement les mêmes que celles présentées à l'implémentation du PC. Cette section envoie donc au chapitre « Liste des commandes créées depuis le PC ».

8.2.5 Conclusion de l'implémentation smartphone spécialiste

Dans ce prototype, les fonctionnalités d'appels et d'annotations à distance ont été implémentées. La fonctionnalité de partage d'écran et de contrôle de périphériques n'ont pu être mise en œuvre dû au manque de temps et aux priorités du scénario de ALPS Automation SA.

Les principaux algorithmes ont été présentés. Il a été également montré la manière dont la création de commandes à distance est réalisée et a pu montrer comment l'ajout ou suppression de nouvelles fonctionnalités serait simple.

Etant très similaire au prototype PC, les informations non présentées dans ce chapitre se trouvent dans le celui dédié à l'implémentation du PC.

8.3 Déploiement de l'application

Ce chapitre présente les différentes configurations Unity afin de déployer ce prototype sur un smartphone.

Tout d'abord, la plateforme de déploiement doit être sélectionnée dans les « Builds Settings ». Ici, ne possédant qu'un appareil Android, cette plateforme a été choisie. Ensuite, l'appareil cible doit être branché à l'ordinateur via USB et, la champ « Run Device » doit sélectionner cet appareil. Le reste des valeurs peuvent être laissées par défaut. De plus, il faut sélectionner les scènes Unity à compiler.

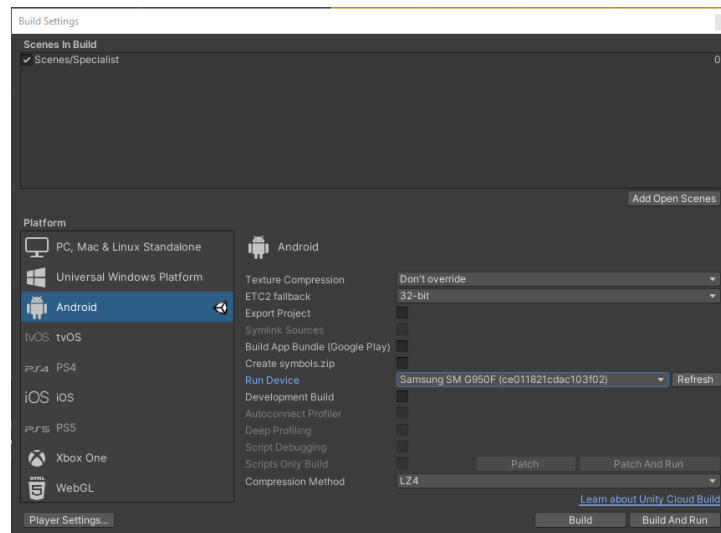


Figure 123 - Build Settings Android

Ces modules de déploiement doivent être ajoutés à la version d'Unity installée. Ceci se fait à l'aide de la plateforme « Unity Hub ». (Unity Hub > Installs > Version Unity > Add Modules)

Dans les annexes se trouve une capture d'écran montrant les différentes configurations des modules Unity Hub afin de déployer correctement. De plus, des captures d'écran montrant les différentes configurations de « Player Settings » y sont également présentes.

Enfin, en cliquant sur « Build And Run », l'application sera compilée et déployée sur le smartphone.

Ci-dessous se trouve une figure montrant l'application déployée sur un Android.



Figure 124 - Application déployée sur Android avec certaines fenêtres et annotations présentes

8.4 Validation

Ici seront présentés les résultats des différents tests « black-box » effectués pour le prototype smartphone. Ces derniers seront présentés sous forme de tableaux avec les résultats obtenus. Afin de ne pas surcharger la lecture de ce rapport, ces tableaux sont placés en annexes.

Les tests liés à la communication audio et vidéo possèdent un ID « Callx_Smartphone » et les tests liés aux annotations ont un ID « Annx_Smartphone ».

Les résultats ont montré que le prototype smartphone est capable de créer une communication audio et vidéo avec le prototype PC et HoloLens 2. De plus, il est également capable de recevoir les appels provenant de ces derniers. L'image vidéo est transmise mais le son semble rencontrer des difficultés. Cela semble être dû à un problème de compatibilité entre MRTK et Smartphone.

Ensuite, les différentes fonctionnalités d'annotations montrent également un bon fonctionnement. Cependant, l'image vidéo est saccadée lors des appels multi-participants et rend ces dernières compliquées.

Via le flux TCP, les messages JSON parviennent au serveur TCP et, de la sorte, le spécialiste peut faire toutes sortes d'annotations ainsi que d'en modifier les options. Cependant, à certains emplacements, l'annotation n'est pas dessinée à l'emplacement du doigt de l'utilisateur. Une calibration serait donc nécessaire.

Ce prototype fonctionne moins bien que les autres, il a une grande marge d'amélioration. Le temps à disposition n'ayant pas permis de perfectionner ce prototype, il restera dans l'état actuel. Toutefois, dans le chapitre d'amélioration, des solutions seront apportées. Une solution pourrait être de créer l'application avec « Android Studio » et utiliser des bibliothèques adaptées.

Un chapitre consacré aux tests utilisateurs est présenté plus tard dans le rapport. (cf. 10. Scénario d'ALPS Automation SA et tests d'usabilité)

9. Prototype Smartphone/Tablette : Personne assistée

Ce chapitre traite de la conception du prototype se déployant sur un smartphone ou une tablette pour une personne nécessitant une assistance à distance.

A la suite du scénario écrit par ALPS Automation SA, mettant en avant une application smartphone pour un spécialiste, le prototype mobile pour la personne assistée ne sera pas réalisé par faute de temps. En effet, le scénario et les protocoles de tests ont priorisé l'utilisation du smartphone pour les spécialistes.

Une solution de conception sera tout de même présentée dans ce chapitre. Ceci permettrait d'avoir une base solide déjà conçue dans le cas où le projet viendrait à être repris et continué.

9.1 Conception

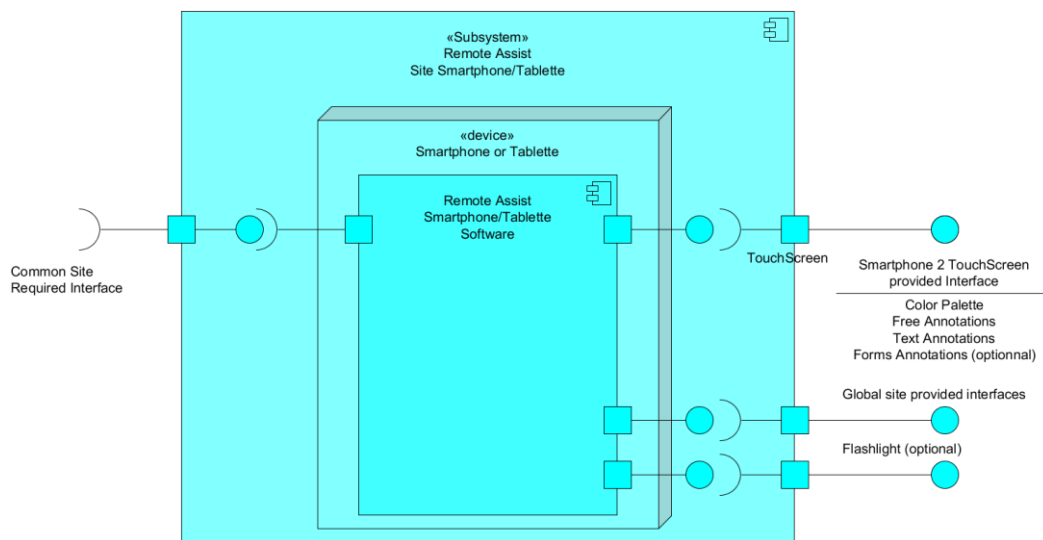


Figure 125 - Layer 3 : Smartphone/Tablette Assisted person

Le prototype « Remote Assist Smartphone/Tablette Software », utilisé par une personne nécessitant une assistance, requiert et offre les différentes interfaces communes discutées au chapitre « 4.1.3 Layer 2 – Diagramme de composants général (Site) ».

Comme le montre la figure ci-dessus, il offre les interfaces liées aux annotations au moyen de son écran tactile. Pour rappel, le RealWear-HMT1 n'offrait pas ces dernières, c'est pourquoi elles ne sont pas considérées comme communes à toutes les plateformes conçues ici. De plus, étant conçu pour être déployé sur smartphone ou tablette, il serait possible d'accéder à la lumière flash de ces derniers pour éclairer la zone d'intervention et obtenir une meilleure image vidéo.

9.1.1 Définition de l'architecture du système

Cette partie décrit l'architecture de l'application. Les outils et bibliothèques de développements servant à la réalisation de ce projet seront présentés avec certains de leurs composants permettant de fournir les différentes interfaces.

Layer 4 : « Remote Assist Specialist Smartphone Software »

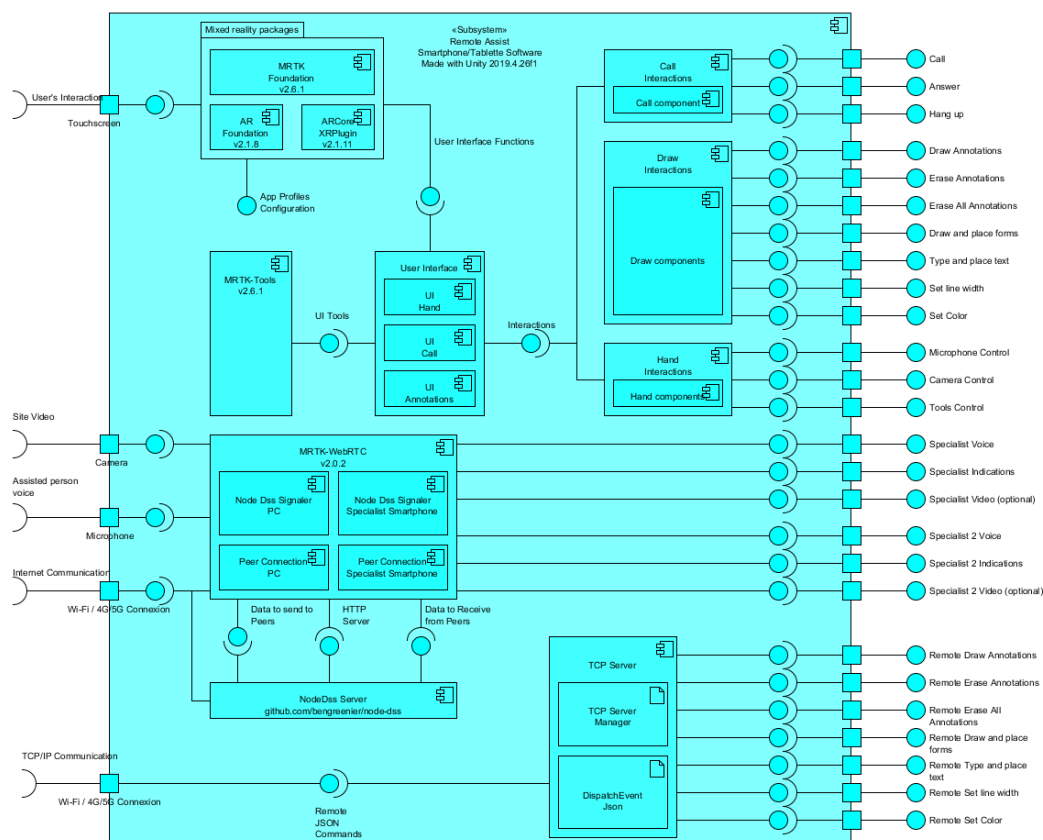


Figure 126 - Layer 4 : Remote Assist specialist smartphone or tablette software

Afin d'alléger ce rapport, cette rubrique suggère de se référer au chapitre « Layer 4 : Remote Assist HoloLens 2 Software » qui traite de la conception de l'application HoloLens 2. En effet, ces deux étant très similaires, la lecture du chapitre proposé ainsi que l'image ci-dessus suffisent à synthétiser ce concept de façon générale. Des clarifications concernant les différences entre les deux concepts seront tout de même apportés ci-dessous.

Tout d'abord, il faut savoir que la collection de librairies MRTK est également intégrable sur mobile. Pour ce faire, l'application a également besoin des paquets AR Foundation.

AR Foundation est une collection permettant de créer des applications de réalité augmentée de manière multiplateforme sur Unity, en fournissant des interfaces à utiliser pour les développeurs Unity. Ensuite, pour utiliser cette collection sur un appareil cible, il est nécessaire d'intégrer certains paquets distincts pour les plateformes cibles prises en charge par Unity tel que ARCore XR Plugin pour Android ou ARKit XR Plugin pour iOS. A titre d'exemple, dans l'illustration précédente, la plateforme Android a été considérée. Pour que ces paquets fonctionnent correctement entre eux, il faut veiller à ce que les versions de ces derniers soient compatibles. Pour exemple, avec la version Unity 2019.4.26f1 et MRTK v2.6.1, il faut utiliser la version 2.1.8 de AR Foundation et la version 2.1.11 de ARCore XR Plugin.

Un lien^[29] dans les références renvoie sur une page internet avec les diverses configurations pour les autres plateformes compatibles.

Un autre point à mentionner est la configuration MRTK-WebRTC. Lors de l'implémentation de ce prototype, le développeur doit veiller à configurer correctement les ID des appareils pairs dans le « Signaler » et dans le « Peer Connexion ».

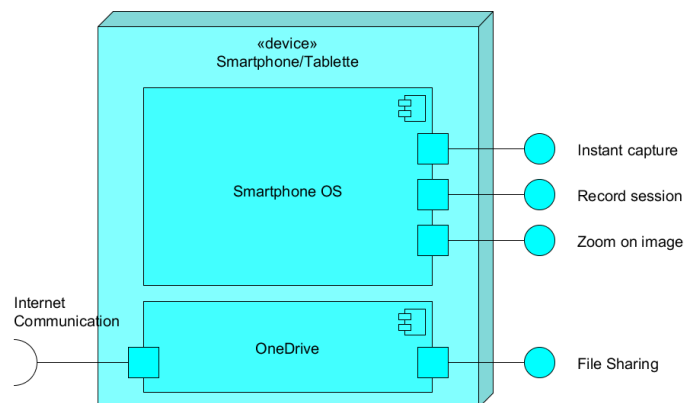


Figure 127 - Smartphone Native Interfaces

Certaines interfaces sont fournies nativement par les smartphones ou les tablettes. Typiquement, la fonctionnalité de capture d'écran permettant de faire des captures instantanées est présente sur la majorité des appareils mobiles. De plus, les plus récents, ont la capacité d'enregistrer leur écran, leur permettant d'enregistrer une session de communication avec les spécialistes. De même, les zooms de caméra sont également bien connus de ces appareils. Concernant le partage de documents, les smartphones et les tablettes peuvent également ouvrir plusieurs applications en parallèle, ce qui leur permettrait d'accéder à des plateformes cloud pour récupérer et partager des fichiers.

9.1.2 Conclusion de la conception smartphone personne assistée

Pour conclure, ce chapitre a fourni des éléments clés liés à la conception pour une utilisation du prototype par une personne nécessitant une assistance à l'aide de son smartphone ou de sa tablette. Comme énoncé auparavant, ce modèle ne sera pas réalisé et testé par faute de temps, mais offre une base pour le faire si le projet venait à être repris.

Les fonctionnalités de réalité mixte peuvent être réalisées à l'aide de MRTK ainsi que de AR Foundation. De nouveau, il faut veiller à ce que les versions soient compatibles entre elles en vérifiant ces dernières sur la documentation de Microsoft.

De plus, il faut être conscient qu'un smartphone ne peut offrir la même qualité d'utilisation qu'un HoloLens 2 et serait plutôt utilisé dans le cas d'interventions non-planifiées. De ce fait, ce concept pourrait être considéré comme version « dégradée » de l'HoloLens 2 en offrant les fonctionnalités de bases mais étant limité d'une quelconque manière.

²⁹ MICROSOFT, Déploiement de MRTK sur Android et iOS (AR Foundation)
<https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/supported-devices/using-ar-foundation?view=mrtkunity-2021-01>

10. Scénario d'ALPS Automation SA et tests d'usabilité

10.1 Contexte

Afin de tester les prototypes d'assistance à distance réalisés dans le cadre de ce travail de Bachelor et de recueillir un retour d'expérience utilisateur, un scénario impliquant trois participants a été imaginé.

Prototypes testés :

- Application pour une personne assistée sur HoloLens 2.
- Application pour un spécialiste sur PC.
- Application pour un technicien sur smartphone.

Ces prototypes étaient connectés au réseau Internet de ALPS Automation SA via leur réseau Wi-Fi.

Cette journée de test s'est déroulée dans l'atelier de fabrication d'ALPS Automation SA et s'articulait autour d'une armoire électrique en cours de fabrication afin de disposer de différents composants électrotechniques semblables à ceux qui pourraient être rencontrés lors d'une situation réelle d'assistance à distance.



Figure 128 - Armoire électrique du scénario

Durant cette journée, trois équipes de trois participants ont joué ce scénario. Ces participants faisaient partie des collaborateurs de ALPS Automation SA et provenaient de différents départements de manière à couvrir un large panel de sensibilités.

10.1.1 Acteurs impliqués

- Un automaticien de l'usine sur site (COL-SITE)
- Un automaticien d'ALPS Automation SA au bureau (COL-ALPS)
- Un technicien spécialiste de l'équipement testé (COL-TECH)

10.2 Tests des différents prototypes

Afin de tester les prototypes, certains prérequis étaient nécessaires :

- Connexion des appareils au réseau Wi-Fi d'ALPS Automation.
- Configuration des adresses IP des appareils dans les différentes applications prototypes.
- Tests de connectivité entre les différents appareils.

De plus, une journée de « pré-test » avait été nécessaire afin de vérifier que la connexion entre les appareils se faisaient de manière correcte. De plus, elle a permis de planifier cette journée de scénario.

Afin que les participants ne soient pas déboussolés, des séances de prises en mains et d'explications ont été faites avant chaque session de tests.

10.2.1 Interfaces entre les participants

Acteur	Appareil	Application
COL-SITE	HoloLens 2	Prototype
COL-ALPS	PC	Prototype
COL-TECH	Smartphone	Prototype

Tableau 4 - Interfaces entre les participants du scénario

A la base, l'appareil RealWear-HMT1 devait être également testé dans le scénario avec l'application TeamViewer AR Assist (Pilot). Malheureusement, ne disposant que d'une version d'essai, celle-ci était limitée et les annotations disparaissaient quelques secondes après les avoir posées. De ce fait, seul une prise en main de l'appareil ainsi qu'un test de confort a été testé sur ce dernier.

En tant qu'observateurs et guides, l'étudiant de ce TB, le superviseur ainsi que l'expert étaient également présents.

10.2.2 Déroulement du scénario de test

Le scénario était composé de 19 tâches à réaliser par les différents participants. La liste des tâches a été placée en annexe dans le but d'alléger ce rapport.

10.3 Questionnaire de feedback pour les participants

A la fin de chaque session, les participants ont rempli un questionnaire de « feedback » basé « System Usability Scale » (SUS). Ce dernier est placé en annexe.

10.3.1 SUS

Les informations sur le « System Usability Scale » ci-dessous proviennent du site myfeedback.com^[30].

Le SUS est un questionnaire de 10 questions conçu pour être à la fois simple et rapide. Il permet de déterminer le niveau de satisfaction des utilisateurs d'un service, ce qui se prête bien aux prototypes développés ici.

Questions posées lors de ce scénario

Les 10 questions de retour posées aux utilisateurs se trouvent dans le questionnaire de feedback placé en annexe. Elles étaient posées sous forme d'affirmation à chaque utilisateur pour leur prototype testé.

Les participants avaient le choix entre cinq réponses possibles allant de « Pas du tout d'accord » à « Tout à fait d'accord » selon le modèle ci-dessous :



Figure 129 - Réponses SUS

Calcul du score SUS

Le score SUS est compris entre 0 et 100. Pour le calculer, la méthode suivante est utilisée.

- Pour les questions impaires, à consonnance positive, la réponse donnée par le répondant doit être soustraite d'un.
- Pour les questions paires, à consonnance négative, le score est égal à cinq moins la réponse donnée par le répondant.
- Une fois le total calculé, le score est multiplié par 2,5.

Ayant plusieurs répondants, une moyenne a été faite à chaque score pour chaque appareil.

³⁰ MYFEELBACK.COM, *Comment utiliser le SUS pour mesurer l'expérience utilisateur ?*
<https://www.myfeedback.com/fr/blog/questionnaire-system-usability-scale-experience-client>
(Consulté le 17 août 2021)

La figure suivante illustre une échelle de satisfaction afin de donner un sens au score obtenu.

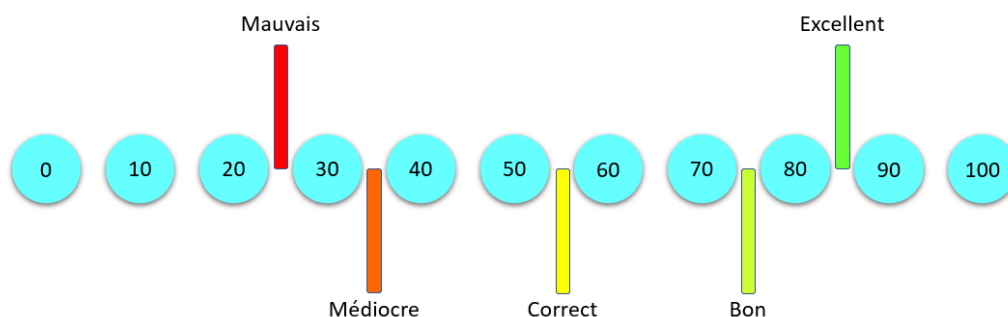


Figure 130 - Echelle de satisfaction

En général, il est estimé qu'un score est :

- Mauvais en dessous de 25.
- Médiocre en dessous de 35.
- Correct à partir de 50.
- Bon à partir de 75.
- Excellent à partir de 85.

De plus, certaines recherches^[31] ont relevé qu'un score au-dessus de 68 serait considéré comme étant au-dessus de la moyenne.

Ici, il est à noter que ces score ne sont pas un indicateur de performance. Ils servent plutôt de première indication par rapport à l'état actuel du proof-of-concept. Ils pourraient être considérés comme évaluation formative en décelant certains problèmes d'usabilité et non pas en tant qu'évaluation sommative. Afin d'avoir un score significatif, l'expérience devrait être réalisée avec un plus grand nombre de testeurs pour chaque appareil, ce qui n'est pas le but à ce stade du projet.

³¹ USABILITY.COM, *System Usability Scale (SUS)*

<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (Consulté le 18 août 2021)

10.4 Résultats obtenus pour le prototype de l'HoloLens 2

En annexe se trouvent deux tableaux liés aux feedback du prototype HoloLens 2. Un premier tableau présente les résultats direct délivrés par les testeurs et, le deuxième présente le score obtenu après le calcul SUS présenté précédemment. Dans ce dernier se trouve également le score total calculé pour chaque utilisateur ainsi que la moyenne de ces derniers.

Ce prototype a obtenu un score SUS de 66,67 faisant de lui un prototype assez correct.

Pour les points qui suivent, il est à relever que les appareils n'avaient jamais été testés auparavant et qu'une simple prise en main de 15 minutes a été faite avant les tests.

10.4.1 Points positifs appréciés

- Les capacités et fonctionnalités de l'HoloLens 2 ont été appréciées.
- La vision des éléments en 3D et la possibilité de les manipuler.
- La possibilité de faire en sorte que les fenêtres suivent l'utilisateur.
- Les écrans 3D rendant le prototype immersif.
- La prise de photos à l'aide des commande vocales.
- « L'outil semble être abouti. »
- Utilisation intuitive

10.4.2 Points négatifs relevés

- Les fonctions avec les doigts peu « responsives ».

10.4.3 Quelques remarques relevées

- « C'est un appareil de niche, cela ne peut être utile que dans de rares occasions et la plupart des informations peuvent être transmises plus facilement. », Testeur COL-SITE 1.
- « Continuer à creuser, c'est l'avenir ! », Testeur COL-SITE 3.

10.4.4 Quelques suggestions apportées

- Avoir un bouton pour retrouver/rappeler les fenêtres non-visibles.
- Possibilité de fixer une position relative entre les écrans. Les écrans suivent le point de vue de l'utilisateur tout en maintenant leur position.
- Possibilité de regrouper les fenêtres par « groupes » et les déplacer ensemble.

10.4.5 Problèmes rencontrés

- Problème de surchauffe au passage du premier groupe. Une piste dit qu'il serait dû au long enregistrement vidéo tournant en arrière-plan.
- Le menu d'annotation a disparu lors d'une session et n'était pas récupérable.

10.5 Résultats obtenus pour le prototype du PC

Les mêmes tableaux de scores que pour l'HoloLens 2 sont placés en annexes avec les scores obtenus pour le prototype PC. Il est à noter qu'une personne supplémentaire a souhaité donner son avis sur le prototype HL2 car le sien, qui était le smartphone, n'a pu être testé.

Ce prototype a obtenu un score SUS de 67,5 faisant de lui un prototype assez correct également. Ce score est le plus haut parmi les trois prototypes.

Pour les points qui suivent, il est à relever que les appareils n'avaient jamais été testés auparavant et qu'une simple prise en main de 15 minutes a été faite avant les tests.

10.5.1 Points positifs appréciés

- Technologie de pointe avec un côté ludique.
- Facilité d'utilisation et de prise en main.
- Interactivité (Textes, flèches, voix)

10.5.2 Points négatifs relevés

- Stabilité de l'image lors des annotations car la caméra de la personne assistée bouge.
- Gestion de la profondeur en 3D des outils d'annotations.

10.5.3 Quelques remarques relevées

- « C'est difficile avec un produit en développement mais je dirais que c'est l'axe de profondeur des outils d'annotations qui est à améliorer », Testeur COL-ALPS 2.
- « Attention aux protocoles et aux flux autorisés car il y a des grandes contraintes chez les clients sur ces sujets-là. », Testeur COL-ALPS 3.

10.5.4 Quelques suggestions apportées

- Besoin d'une fonction « snapshot » pour être indépendant du porteur du casque et travailler sur une image figée pour gagner en stabilité.
- Afficher un icône de chargement lorsque la caméra est « freeze » avant de se rafraîchir.
- Possibilité de réinitialiser la position des outils 3D lorsqu'ils ne sont plus visibles.
- Possibilité de changer l'apparence des outils d'annotations lorsqu'ils passent derrière un élément du décor.
- Ajouter des formes d'annotations supplémentaires en plus des flèches.
- Rendre les textes annotés encore plus lisibles.
- Avoir une liste des annotations instanciées avec la possibilité de les effacer depuis cette liste.
- Pouvoir partager l'écran depuis le PC.
- Possibilité de déplacer les annotations après les avoir placées.

10.5.5 Problèmes rencontrés

- Annotations libres compliquées dû au mouvement de la caméra de la personnes assistée qui causait une image non stable.
- Perte des outils d'annotations dans l'environnement 3D de la personne assistée.
- Pas de possibilité de zoom ce qui rend certaines valeurs du système de mise en service illisible.
- Curseur de dessin libre et textes pas très lisible.

10.6 Résultats obtenus pour le prototype du smartphone

Les mêmes tableaux de scores que pour les autres prototypes sont placés en annexes avec les scores obtenus pour le smartphone. Il est à noter qu'une personne en moins a donné son feedback car son prototype smartphone n'a pas pu se connecter aux autres appareils lors de la première session.

Ce prototype a obtenu un score SUS de 50 faisant de lui un prototype passable voir peu fiable. Il a obtenu le score le plus bas entre les différents prototypes, ce qui peut être expliqué par son peu de temps de développement.

10.6.1 Points positifs appréciés

- Avoir la vue sur le site.
- Possibilité de faire des annotations depuis un smartphone.

10.6.2 Points négatifs relevés

- Lenteur et latence (dû au prototype en phase de développement).
- Semble moins adapté dû à la petite taille de l'écran.
- Résolution insuffisante.
- Positionnement des annotations

10.6.3 Quelques remarques relevées

- « Les outils doivent être hyper réactifs pour être utilisables et fonctionnels dans l'environnement professionnel. », Testeur COL-TECH 3.

10.6.4 Quelques suggestions apportées

- Travailler sur une image figée pour gagner en stabilité.
- Pouvoir déplacer les annotations après les avoir placées.
- Ajouter des formes d'annotations supplémentaires en plus des flèches.
- Rendre les textes annotés encore plus lisibles.
- Avoir une liste des annotations instanciées avec la possibilité de les effacer depuis cette liste.
- Pouvoir recalibrer les axes d'annotations.

10.6.5 Problèmes rencontrés

- Annotations libres compliquées dû à la latence qu'a engendré l'incapacité de MRTK-WebRTC à s'adapter à des sessions multi-participants.
- Dans la première session, le smartphone ne parvenait pas à se connecter au serveur node.js. (Portée Wi-Fi ? Réseau Wi-Fi invité limité ?)
- Image vidéo très saccadée.

10.6.6 Conclusion

Pour conclure, cette journée de scénario a démontré l'importance de faire des tests avec des utilisateurs d'un domaine ciblé. En effet, en développant le prototype et en ne le testant que dans un bureau, certains problèmes ne sont pas mis en lumière et peuvent ne pas être perçus.

Ici, le terrain a démontré que les prototypes ne sont pas encore prêts pour être déployés officiellement dans le milieu professionnel. Toutefois, un grand potentiel a été perçu et les participants seraient d'accord de les utiliser si les produits venaient à être abouti. Les participants ont également eu de la facilité à utiliser les fonctionnalités opérationnelles présentes et les trouvaient simples et intuitives bien que des ajustements soient nécessaires. Ci-dessous se trouve le score SUS final des trois prototypes.

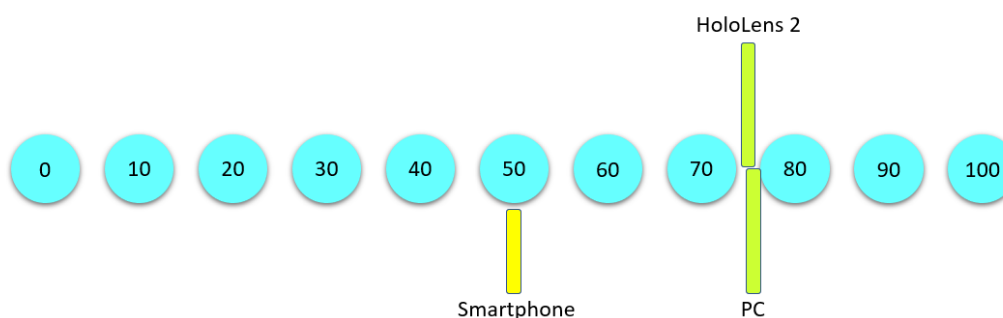


Figure 131 - Score final SUS des trois prototypes

L'application smartphone a été la moins appréciée dû à sa latence et à son image saccadée qui rendaient les annotations à distance très compliquées. Ensuite l'HoloLens 2 et le PC ont été bien appréciés et leur potentiel a été repéré. Pour rendre le PC et l'HoloLens 2 vraiment utilisable avec ces prototypes, la stabilisation de l'image, une fonctionnalité de « snapshot » ainsi qu'une meilleure gestion de l'axe de profondeur des outils devraient être améliorées.

En cas de continuité de développement, d'autres phases de tests supplémentaires comme ce scénario devraient être faites. Pour aller un peu plus loin et apporter du feedback supplémentaire, il serait intéressant de réaliser ces tests dans des salles différentes et d'utiliser les fonctions vocales car, ici, les micros étaient désactivés pour éviter l'écho indésirable.

Enfin, bien que le RealWear-HMT1 avec TeamViewer Pilot n'ait pu être testé, les utilisateurs étaient d'accord pour accorder à l'HoloLens 2 un meilleur confort long terme.

11. Conclusion Générale

Pour terminer ce rapport, une conclusion finale résumera les étapes importantes du rapport, comparera les résultats obtenus avec les objectifs initiaux et rassemblera des suggestions relatives aux améliorations à apporter à ce projet afin que d'éventuels nouveaux développeurs aient des pistes potentielles.

11.1 Résumé du projet

Les paragraphes qui suivront résumeront ce travail de Bachelor. Il s'est déroulé en quatre grandes étapes qui seront décrites ci-dessous. De plus, quelques points importants du développement seront relevés.

La première était consacrée à la recherche sur l'utilisation de la réalité augmentée dans l'industrie 4.0 ainsi qu'aux prises en mains de technologies liées à cette dernière. Il en a été ressorti que la réalité augmentée a le potentiel d'apporter énormément de ressources aux entreprises et que ces dernières doivent suivre l'ère de cette industrialisation 4.0 si elles ne veulent pas se faire dépasser par leurs concurrents. Plusieurs avantages ont été ressortis. Parmi ceux-là figurent : le gain de productivité, la facilitation de l'apprentissage ou encore la mise en synergie du numérique et du terrain. Toutefois, les études ont démontré que certaines industries ne sont pas encore prêtes à utiliser la réalité augmentée à cause de certaines limites telles que : les ressources financières, les infrastructures informatiques limitées ou encore l'accès à une connexion internet de qualité dans certains lieux.

La deuxième étape était consacrée à l'analyse du cahier des charges d'ALPS Automation ainsi qu'à l'étude des dispositifs et services de RA pour la réalité augmentée. L'étude a conclu que le meilleur dispositif dans ce cas de figure était l'HoloLens 2 et, parmi les services existants d'assistance à distance, la plateforme « Remote Assist » développée par Microsoft est celle qui se rapprocherait le plus des exigences de ALPS Automation.

La troisième étape était dédiée au développement des différents prototypes d'assistance à distance pour ALPS Automation SA. Ils sont déployés sur trois plateformes : PC, HoloLens 2 et smartphone. Les spécialistes utiliseraient les prototypes PC et smartphone, alors que la personne assistée utiliserait l'HoloLens 2. ALPS Automation SA souhaiterait également fournir à la personne assistée une application smartphone d'assistance à distance pour le cas d'une intervention non-planifiée. Malheureusement, le temps à disposition n'a pas permis l'implémentation de cette dernière. Cependant, dans ce rapport se trouve un chapitre dédié à la conception de ce prototype afin de donner des pistes aux nouveaux développeurs.

Pour relever quelques points importants liés au développement, la norme WebRTC a été utilisée afin de permettre une communication audio et vidéo entre les différents prototypes. Ensuite, une connexion TCP/IP permet aux appareils de s'échanger des messages JSON afin d'effectuer des actions à distance. Ces actions sont liées aux annotations virtuelles en 3D dans l'environnement de la personne assistée.

Utilisant l'appareil HoloLens 2, certaines fonctionnalités souhaitées par ALPS Automation SA étaient déjà présentes nativement dans cet appareil. Ainsi, il n'a donc pas été nécessaire d'implémenter les fonctionnalités de capture d'images et de vidéos, de partage de fichiers ou de mise en transparence de fichiers dans l'environnement de la personne assistée. Il est à noter qu'actuellement, pour le partage de fichiers, une plateforme de cloud est utilisée par les utilisateurs

Enfin, la dernière étape concernait les différents tests d'entrées/sorties ainsi que des tests utilisateurs. Ils ont été effectués afin de vérifier le fonctionnement des prototypes. Certains problèmes liés à la communication audio et vidéo ont été rencontrés. De plus, une calibration de l'écran du smartphone ainsi qu'une stabilisation de l'image provenant du site devraient être également nécessaire afin d'améliorer la précision d'annotation à distance depuis cet appareil. Ces problèmes, ainsi que des solutions pour y remédier, seront exposés dans le chapitre dédiés aux problèmes rencontrés et aux recommandations. Un chapitre consacré aux retours des tests d'usabilité présente également des perspectives d'améliorations.

11.2 Comparaison avec les objectifs initiaux

Pour rappel, ce projet avait deux objectifs principaux et deux objectifs secondaires. Le premier principal était consacré à la recherche sur les capacités et limites des dispositifs de RA et des services existants. Cet objectif a été réalisé entre la première et la deuxième partie du rapport comme énoncé dans le résumé de la page précédente. De plus, l'étude de la réalité augmentée dans l'industrie 4.0 et des différentes technologies existantes a permis de réunir suffisamment de connaissances pour passer au deuxième objectif principal. De plus, des mini-démonstrateurs avec Unity, Unreal Engine et Mesh ont été présentés et accomplissent ainsi l'un des objectifs secondaires.

Le deuxième objectif principal consistait à concevoir, développer et tester un proof-of-concept pour le cas d'ALPS Automation SA. Cet objectif a été atteint en réalisant trois prototypes déployés sur PC, HoloLens 2 et smartphone. ALPS Automation SA a fourni un scénario à jouer avec des utilisateurs finaux permettant d'accomplir un autre objectif secondaire. Ce scénario a donné un « fil conducteur » et a permis de se focaliser sur certains points et de prendre des décisions concernant ce projet.

En effet, le temps à disposition étant limité, certaines questions se sont posées durant la réalisation de ce travail : « Faut-il se focaliser uniquement sur un prototype et l'optimiser à son maximum ? », « Faut-il se focaliser sur le côté spécialiste ou le côté personne assistée ? », etc. Avec ce scénario, la décision prise a été de « valider la boucle » entre un spécialiste PC et smartphone et une personne assistée connectés simultanément, permettant ainsi d'explorer tous les horizons du système et d'avoir un premier retour global de ALPS Automation SA. Ceci a l'avantage de donner un aperçu de ce qui est réalisable entre les différents acteurs. L'inconvénient est qu'avec cette méthode, aucun prototype n'est complètement terminé. En effet, le temps ayant été réparti entre plusieurs prototypes, certains points nécessitant un grand temps de réalisation n'ont pu être priorisés et, de ce fait, n'ont pas été implémentés. Néanmoins, la majorité des fonctionnalités ont été implémentées permettant ainsi d'atteindre le deuxième objectif.

11.3 Retour sur la planification du projet

Au commencement du projet, une planification a été réalisée. Elle a été un élément clé de l'organisation en permettant de se situer temporellement durant la réalisation du projet. Globalement, ce planning a été respecté mais certains détails se sont passés autrement. Dans les annexes se trouvent le planning initial et le planning réel.

Ci-dessous se trouvent les différences entre les deux plannings.

- Durée de conception plus longue que prévue. Un chevauchement plus long entre la conception et l'implémentation est perçu.
- Les tests des prototypes ont commencé plus tôt. En effet, certains tests ont pu être réalisés en parallèle de l'implémentation, ce qui cause un chevauchement entre l'implémentation et la vérification.
- L'implémentation a été étirée à cinq jours supplémentaires.
- Une plage pour la mise en forme du rapport durant 4 jours a été ajoutée.
- La date du jour du scénario a été mise à jour et ne dure qu'une journée.
- Les mini-démonstrateurs facultatifs de Unity, UE4 et Mesh ont été réalisés durant la prise en main des technologies.

Finalement, le planning montre qu'il a été nécessaire de consacrer une grande partie du temps à l'étude et aux recherches initiales. La question se demandant si prendre autant de temps pour cela était utile pourrait se poser. En fin de compte, cela a permis de bien cerner le problème et de comprendre les enjeux de la réalité augmentée, ce qui fût nécessaire dans ce proof-of-concept. De plus, si de nouveaux développeurs venaient à reprendre ce travail, ils disposeraient de l'essentiel dans ce rapport, leur permettant d'éviter de consacrer trop de temps à ce sujet et d'en gagner sur le développement.

11.4 Problèmes rencontrés

Lors de la réalisation des prototypes, certains problèmes rencontrés méritent d'être énoncés dans ce chapitre. L'idée étant d'apporter des pistes de solutions à des erreurs, si celles-ci venaient à être reproduites.

11.4.1 Erreur de connexion HoloLens 2

Lors de l'implémentation de MRTK-WebRTC sur l'HoloLens 2, ce dernier ne parvenait pas à accéder au serveur Node.js. Une semaine de développement a été perdue à cause de cette erreur, car il a fallu la résoudre.



Figure 132 - Erreur de connexion de l'HoloLens 2 au serveur Node.js

Alors que cette erreur persistait sur l'HoloLens 2, elle n'était pas présente sur les prototypes PC et smartphone. Plusieurs tentatives de débogage ont été élaborées, telles que l'ouverture de ports ou la désactivation du pare-feu mais ce, en vain.

Finalement, un nouveau projet Unity a été créé en important une seconde fois tous les paquets MRTK-WebRTC, en développant le prototype de communication et en le redéployant sur l'HoloLens 2 pour vérifier si l'erreur persistait. Cette fois, la connexion s'est bien établie.

Il semblerait que l'erreur vienne du nombre de caractères maximum des chemins des répertoires du projet Unity. Selon Microsoft :

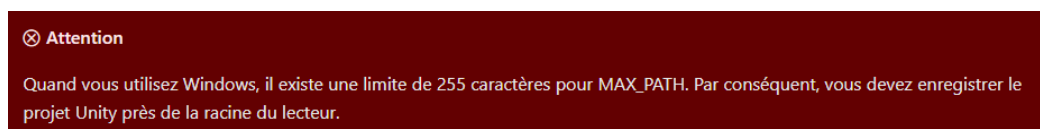


Figure 133 - Cause de l'erreur de la connexion HL2 et du serveur Node.js

source : MICROSOFT DOCS, Initialisation de votre projet et déploiement de votre première application
<https://docs.microsoft.com/fr-fr/windows/mixed-reality/develop/unity/tutorials/mr-learning-base-02?tabs=openxr> (consulté le vendredi 13 août 2021)

En effet, le projet Unity étant dans plusieurs dossiers (répertoire Git compris), possédait un long chemin dépassant cette limite du nombre de caractères. Lors de l'importation des différents paquets, certains fichiers aux noms trop grands se voyaient corrompus. Certaines dépendances étaient alors perdues, causant ainsi divers problèmes.

Les projets Unity ont été déplacés, les noms des sous-dossiers et répertoires Git ont été raccourcis jusqu'à ce qu'un compteur de caractère en ligne^[32] en indique moins de 255 pour le chemin le plus long. De la sorte, l'erreur a été résolue.

³² CHARACTERCOUNTERONLINE.COM
<https://www.charactercountonline.com/fr/> (consulté le 13 août 2021)

11.4.2 Mise en œuvre des appels multi-participants

Comme expliqué au chapitre « 5.2 Mise en œuvre des appels multi-participants avec MRTK-WebRTC », la librairie utilisée pour la norme WebRTC n'était pas faite pour des communications multi-participants. Cependant, c'est actuellement la seule qui fournit ces outils pour les plateformes « Universal Windows Platform » comme l'HoloLens 2. Il a fallu apporter quelques modifications à cette librairie afin d'établir une telle collection. Bien qu'elle soit établie, quelques problèmes de latences peuvent être perçus à partir de trois participants et il faut, parfois, faire deux tentatives d'appels afin que l'image vidéo soit correctement transmise. Dans les perspectives sera présentée une solution à ce problème qui n'a pas eu le temps d'être implémentée ici.

11.4.3 Pare-feu bloquant la connexion Android

Un autre problème a été rencontré lorsque l'application Android (TCP Client) tentait de se connecter à la plateforme Unity (TCP Server) via une connexion TCP. En effet, le pare-feu Windows bloquait la connexion du smartphone et empêchait ainsi de tester et de déboguer les erreurs directement sur la plateforme Unity.

Afin de résoudre ce problème, une nouvelle règle a été ajoutée dans le pare-feu Windows comme le montre la figure ci-dessous.

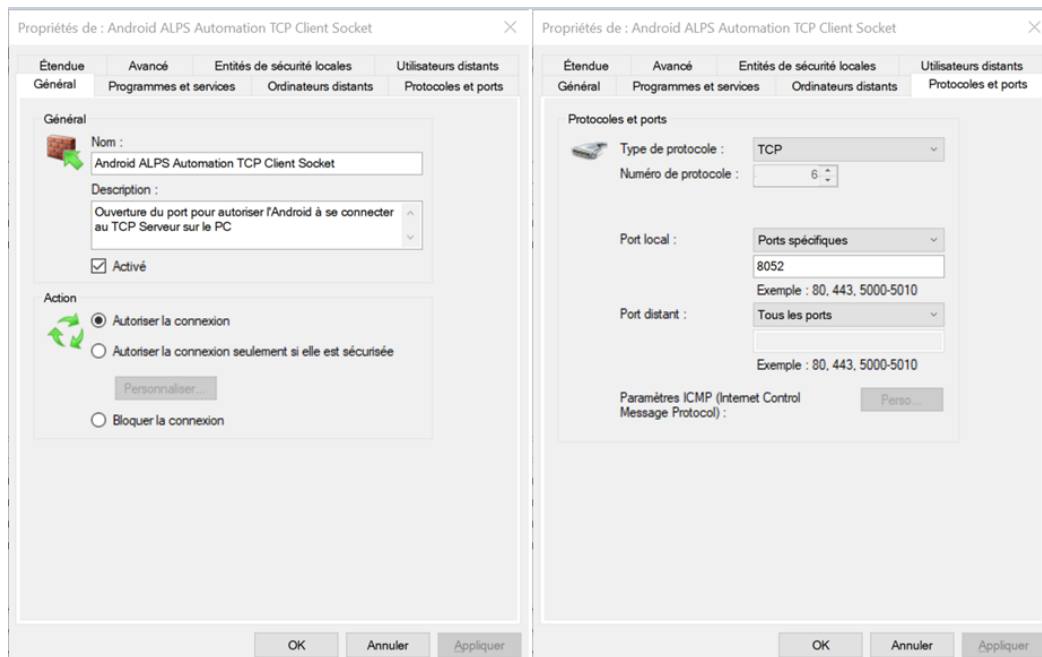


Figure 134 - Création d'une nouvelle règle dans le pare-feu pour autoriser la connexion TCP provenant du smartphone.

Dans cette règle, il fallait autoriser la connexion du smartphone en ouvrant le port 8052 (port utilisé pour la connexion TCP des prototypes) pour un protocole TCP. Depuis, la connexion TCP est établie correctement.

Il est à noter que lorsque le serveur TCP était déployé sur l'HoloLens 2, ce problème n'a pas été rencontré.

11.4.4 Transfert des coordonnées statiques 2D (PC, smartphone) aux coordonnées dynamiques 3D (HL2).

Dans les premières phases des prototypes, lorsqu'un spécialiste dessinait depuis son écran, les coordonnées étaient correctement transmises à l'environnement 3D de la personne assistée, du moment que celle-ci ne se déplaçait pas et regardait toujours droit devant elle. Cependant, si elle se déplaçait ou effectuait des tours sur elle-même, le référentiel de l'HoloLens 2 changeait, alors que celle des appareils des spécialistes ne changeaient pas. Ceci impliquait que les annotations n'étaient pas dessinées dans la direction du regard de la personne assistée, ce qui peut être embêtant dans une mise en service où l'utilisateur sur le site d'intervention peut être amené à beaucoup se déplacer.

Finalement, la géométrie vectorielle a permis de résoudre ce problème. Les coordonnées envoyées via TCP pouvant être considérées comme des vecteurs, il suffisait d'appliquer des angles à chaque axe (X, Y et Z) afin de leur faire effectuer une rotation égale à celle de la caméra de l'HoloLens 2. De la sorte, au moment de leur création, les annotations se retrouveront toujours dans la direction de l'utilisateur.

11.4.5 Faible précision des annotations smartphone

Ce problème est lié à l'application smartphone du spécialiste. Une certaine calibration devrait être faite à partir de la taille de l'écran de ce dernier. En effet, les annotations ne sont pas exactement dessinées là où le spécialiste a cliqué avec son doigt. Le temps n'ayant pas permis de résoudre ce problème, dans les perspectives, quelques pistes seront présentées afin de le résoudre.

11.4.6 Microphone smartphone

Un autre problème causé par la librairie MRTK-WebRTC est la compatibilité avec le smartphone. Plusieurs éléments ont été mis en place afin de parvenir à effectuer la connexion comme expliqué dans la phase de développement. Bien que le script permette d'autoriser l'accès au Microphone du smartphone, il arrive que ce dernier ne s'active pas durant la communication WebRTC.

11.4.6 Problème d'utilisation de JSON avec Newtonsoft.json

Malgré les paquets Newtonsoft.json installés dans les différents projets Unity, l'utilisation de ces derniers n'étaient pas fonctionnels. Il fallait spécifier dans le fichier link.xml les lignes suivantes.

```
<linker>
  <assembly fullname="Your.Dto.Package.Name" preserve="all" />
  <assembly fullname="System.Core">
    <type fullname="System.Linq.Expressions.Interpreter.LightLambda" preserve="all" />
  </assembly>
</linker>
```

Figure 135 - Lignes de code permettant de résoudre le problème de fonctionnement de JSON.
(Source : <https://stackoverflow.com/questions/16359628/json-net-under-unity3d-for-ios>)

11.5 Perspectives

Ces prototypes ont permis d'explorer les horizons de l'assistance à distance à l'aide de la réalité augmentée. N'ayant pu se focaliser sur un seul prototype et de l'optimiser, ces derniers possèdent une très grande marge d'amélioration et un grand potentiel. Ici, seront présentées quelques suggestions à propos des améliorations à apporter aux prototypes.

11.5.1 Configuration d'un serveur sécurisé pour la norme WebRTC

Comme expliqué lors de la conception, le serveur Node.js mis à disposition dans ce prototype n'est pas sécurisé car les données sont transmises en texte clair sur http.

Il faudrait donc implémenter un serveur STUN/TURN sécurisé. Dans les références se trouvent quelques liens^[33] ^[34] contenant des pistes d'implémentation de ce dernier.

11.5.2 Utilisation d'un serveur Web pour les connexions multi-participants

Dans ce rapport, il a été très souvent mentionné que MRTK-WebRTC n'était pas fait pour des connexions multi-participants. Toutefois, les modifications apportées ont permis d'établir une telle connexion mais étant limitée pour une production officielle.

Une meilleure solution serait l'utilisation d'un serveur centralisé SFU (Selective Forward Unit), avec lequel chaque appareil disposerait d'un seul « Peer Connection ». L'architecture pourrait être considérée comme « étoile » plutôt qu'en « triangle ». De la sorte, les flux audio et vidéo ne sont transmis qu'une fois au serveur et c'est ce dernier qui s'occupe ensuite de les partager aux autres pairs. Les SFU réduisent à la fois la bande passante et la charge de l'encodeur et peut également fonctionner comme serveur de signalisation. A titre d'exemple, Janus WebRTC Server^[35], développé par Meetecho, pourrait être, ici, une solution.

Pour information, WinRTC semble aller dans la bonne direction pour le développement d'appels WebRTC pour UWP mais est encore en cours de développement. Il serait donc intéressant de garder un œil sur cette technologie.

³³ MEDIUM.COM, *What is WebRTC and How to setup STUN/TURN Server for WebRTC Communication?*

<https://medium.com/av-transcode/what-is-webrtc-and-how-to-setup-stun-turn-server-for-webrtc-communication-63314728b9d0> (consulté le 14 août 2021)

³⁴ HTML5ROCKS.COM, *Build the backend services needed for a WebRTC app*

<https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/> (consulté le 14 août 2021)

³⁵ JANUS.CONF.MEETECHOCOM, *Janus: the general purpose WebRTC server*

<https://janus.conf.meetecho.com/> (consulté le 14 août 2021)

11.5.3 Précision des annotations Smartphone

Comme expliqué dans les problèmes rencontrés, une calibration des coordonnées smartphones devrait être faite afin de garder une bonne précision d'annotation. Une solution qui supprimerait ce problème, serait le contrôle des outils à distance à l'aide de « Joysticks » ou de boutons sur l'UI du smartphone. En effet, au lieu de cliquer dans une zone de l'écran et de faire apparaître une annotation au mauvais emplacement, l'utilisateur pourrait, à l'aide des commandes dans l'UI, déplacer les annotations et les mettre à la place désirée.

11.5.4 Autre méthode pour les annotations à distance

Dans ce projet, les annotations à distance sont contrôlées à l'aide de clients TCP ainsi que leur messages JSON envoyés dans un flux et gérés par le serveur TCP.

Une autre solution serait d'envoyer le « mesh » (mailles) de l'environnement perçu par l'HoloLens 2 aux appareils des spécialistes, qui pourraient alors placer des annotations à distance en ayant pour repère une structure de la pièce de la personne assistée. Ce mesh serait récupérable depuis le « Windows Device Portal^[36] », il pourrait être ensuite importé dans Unity, modifié et retransmis.

A titre d'exemple, une vidéo^[37] très parlante a été trouvée sur la plateforme Youtube et pourrait ainsi servir d'inspiration à la réalisation de cette tâche. Elle peut être retrouvée dans les références.

Une autre piste serait également l'utilisation de Spatial Anchors^[38].

11.5.5 Création d'outils individuels propre à chaque spécialiste connectés

Actuellement, utilisant les mêmes numéros de commandes envoyés par TCP, les différents spécialistes contrôlent les mêmes outils d'annotations.

Une solution serait d'ajouter un attribut au message JSON contenant un ID utilisateur. Ainsi, le contrôle des outils pourrait être créé et contrôlé en fonction de cet ID.

11.5.6 Adresses IP Dynamiques

Il serait intéressant d'implémenter un système permettant d'entrer les adresses IP des prototypes manuellement au démarrage des applications ou de les automatiser pour les connexions WebRTC et TCP/IP. Ceci permettrait d'éviter de devoir faire ces changements dans le code et de devoir les redéployer sur les appareils.

³⁶ MICROSOFT DOCS, *Using the Windows Device Portal*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/platform-capabilities-and-apis/using-the-windows-device-portal> (consulté le 14 août 2021)

³⁷ YOUTUBE.COM, *Fata Morgana – VIRTUAL DEMO*
<https://www.youtube.com/watch?v=WSuW6MiruiQ> (consulté le 14 août 2021)

³⁸ MICROSOFT AZURE, *Spatial Anchors*
<https://azure.microsoft.com/en-us/services/spatial-anchors/#overview> (consulté le 14 août 2021)

11.5.7 Quelques suggestions supplémentaires

Avec plus de temps, certains éléments des prototypes auraient pu être améliorés pour l'expérience utilisateur. Ci-dessous se trouve une liste avec quelques points d'améliorations et des nouvelles idées.

- Ajouter des sons lorsque des actions sont effectuées afin d'avoir un retour.
- Pouvoir effacer la dernière annotation à l'aide d'une commande comme « CTRL-Z ».
- Mettre une option « Hide All » permettant de simplement cacher les annotations au lieu de les supprimer complètement.
- Ajouter des commandes vocales. Par exemple pour récupérer une fenêtre lorsqu'elle est perdue.
- Ajouter des nouveaux gestes de mains reconnaissables par l'HoloLens 2, spécifiques pour effectuer certaines fonctions.
- Améliorer la lisibilité des outils à distance et des annotations perçues depuis l'appareil spécialiste. Par exemple en ajoutant des bordures, des surbrillances ou des bandes d'arrière-plans.
- Création d'un mesh spécifique à chaque annotation libre pour une suppression plus simple des annotations. (Actuellement, seul un box collider est utilisé.)
- Ajout d'une légère indication de l'emplacement du box collider pour faciliter la suppression.
- Cacher le bouton « Keyboard » pour éditer les annotations textes lorsque l'utilisateur est loin de ces dernières et le faire apparaître quand il en est proche.
- Possibilité de changer la rotation de certains outils par les spécialistes avant de les placer à distance (et après).

Bien d'autres suggestions se trouvent dans le chapitre dédié aux retours expérience des testeurs finaux. (cf. 10. Scénario d'ALPS Automation SA et tests d'usabilité)

11.6 Conclusion personnelle

Pour clore ce rapport, moi, l'auteur de celui-ci, aimerait exprimer quelques mots.

Tout d'abord, ce travail de Bachelor m'a permis de découvrir des technologies fascinantes. En effet, mes premiers pas avec la réalité augmentée et l'HoloLens 2 auront commencé ici. J'ai été impressionné par ce que la technologie d'aujourd'hui est capable de faire et me réjouis, en tant que développeur, de continuer à apporter mes connaissances. Je suis également curieux de voir jusqu'où seront capable d'aller la réalité augmentée et mixte et ne manquerais pas de suivre cela de près.

Parmi les nouvelles technologies découvertes se trouvent la plateforme Unity et le langage C#, voir même Unreal Engine. J'ai pris un grand plaisir à travailler avec ces dernières et je continuerai bien volontiers à les utiliser.

Enfin, je m'estime très chanceux d'avoir eu la possibilité de réaliser ce travail de diplôme et j'en suis très reconnaissant.

12. Bibliographie

12.1 Références

- [1] VISIATIV SOLUTIONS, *Industrie 4.0 : définition et mise en œuvre vers l'usine connectée*
<https://www.visiativ-solutions.fr/industrie-4-0> (Consulté le 17 mai 2021)
- [2] WIKIPEDIA (EN), *Ivan Sutherland*
https://en.wikipedia.org/wiki/Ivan_Sutherland (Consulté le 17 mai 2021)
- [3] FINANCES ONLINE, *Technology statistics, Augmented Reality*
<https://financesonline.com/technology-statistics> (Consulté le 17 mai 2021)
- [4] VISIATIV SOLUTIONS, *6 bénéfices de la réalité augmentée dans l'industrie*
<https://www.visiativ-solutions.ch/realite-augmentee-dans-l-industrie> (Consulté le 17 mai 2021)
- [5] VISIATIV SOLUTIONS, *6 bénéfices de la réalité augmentée dans l'industrie*
<https://www.visiativ-solutions.ch/realite-augmentee-dans-l-industrie> (Consulté le 17 mai 2021)
- [6] AUDROS, *La réalité augmentée industrielle au service de l'industrie 4.0*
<https://www.audros.fr/realite-augmentee-industrie> (Consulté le 17 mai 2021)
- [7] FINANCESONLINE, *20 Current Augmented Reality Trends & Prediction for 2021/2022 and Beyond, 11. AR Remote Assistance*
<https://financesonline.com/augmented-reality-trends> (Consulté le 17 mai 2021)
- [8] UNITY, *Réalité augmentée*
<https://unity.com/fr/unity/features/ar> (Consulté le 18 mai 2021)
- [9] UNITY, *Unity Store*
<https://store.unity.com/fr/compare-plans> (Consulté le 18 mai 2021)
- [10] Unity Learn, *Placing and Manipulating Objects in AR*
<https://learn.unity.com/tutorial/placing-and-manipulating-objects-in-ar#> (Consulté le 4 juin 2021)
- [11] Unreal Engine, *Unreal Products*
<https://www.unrealengine.com/en-US/custom-license> (Consulté le 21 mai 2021)
- [12] Circuit Stream, *Introduction to AR Development with Unreal Engine 4 — Build an Image Tracker App under 90 Minutes!*
<https://www.youtube.com/watch?v=yQKJkGenTnA> (Consulté le 20 mai 2021)
- [13] SMÍD, Antonín, *Comparison of Unity and Unreal Engine*, Czech Technical University in Prague, 2017. Chapter 6, Results and Comparison, p. 41 – 60
- [14] MICROSOFT, *1. Introduction to the MRTK tutorials*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/tutorials/mr-learning-base-01> (Consulté le 7 juin 2021)
- [15] MICROSOFT.GITHUB.IO, *Hello, Unity World!*
<https://microsoft.github.io/MixedReality-WebRTC/manual/unity/helloworld-unity.html>
 (Consulté le 7 juin 2021)
- [16] GITHUB.COM, *AgoraIO-Community/AgoraUWP*
<https://github.com/AgoraIO-Community/AgoraUWP> (Consulté le 5 août 2021)
- [17] MICROSOFT, *Spectator View pour HoloLens et HoloLens 2*
<https://docs.microsoft.com/fr-ch/windows/mixed-reality/develop/platform-capabilities-and-apis/spectator-view> (Consulté le 5 août 2021)
- [18] GITHUB.COM, *Microsoft/winrtc*
<https://github.com/microsoft/winrtc> (Consulté le 5 août 2021)

- [19] WIKIPEDIA, *Session Description Protocol*
https://fr.wikipedia.org/wiki/Session_Description_Protocol (Consulté le 4 août 2021)
- [20] GITHUB.COM, *beengrenier/node-dss*
<https://github.com/bengreenier/node-dss> (Consulté le 4 août 2021)
- [21] NODEJS, Download for Windows (x64)
<https://nodejs.org/en/> (Consulté le 4 août 2021)
- [22] JSON.ORG, *Introducing JSON*
<https://www.json.org/json-en.html> (Consulté le 5 août 2021)
- [23] MICROSOFT DOCS, *Welcome to the Mixed Reality Feature Tool*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/welcome-to-mr-feature-tool> (Consulté le 13 août 2021)
- [24] MICROSOFT, *Présentation du Mixed Reality Toolkit*
<https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05> (Consulté le 1er août 2021)
- [25] FIGMA.COM, *Figma Toolkit for MRTK/HoloLens, Windows Mixed Reality*.
<https://www.figma.com/file/ltLag9SxjUIyLQFsp7NNE7/Figma-Toolkit-for-MRTK-%2F-HoloLens%2C-Windows-Mixed-Reality?node-id=116%3A4> (Consulté le 1^{er} août 2021)
- [26] MICROSOFT DOCS, *Lumière de proximité*
<https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/features/rendering/proximity-light?view=mrtkunity-2021-05> (Consulté le 12 août 2021)
- [27] UNITY DOC, *TouchScreenKeyboard*
<https://docs.unity3d.com/ScriptReference/TouchScreenKeyboard.html> (Consulté le 9 août 2021)
- [28] UNITY DOCUMENTATION, *Requesting Permissions*
<https://docs.unity3d.com/Manual/android-RequestingPermissions.html> (Consulté le 12 août 2021)
- [29] MICROSOFT, Déploiement de MRTK sur Android et iOS (AR Foundation)
<https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/supported-devices/using-ar-foundation?view=mrtkunity-2021-01> (consulté le 12 août 2021)
- [30] MYFEELBACK.COM, *Comment utiliser le SUS pour mesurer l'expérience utilisateur ?*
<https://www.myfeelback.com/fr/blog/questionnaire-system-usability-scale-experience-client> (Consulté le 17 août 2021)
- [31] USABILITY.COM, *System Usability Scale (SUS)*
<https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (Consulté le 18 août 2021)
- [32] CHARACTERCOUNTERONLINE.COM
<https://www.charactercountonline.com/fr/> (Consulté le 13 août 2021)
- [33] MEDIUM.COM, *What is WebRTC and How to setup STUN/TURN Server for WebRTC Communication?*
<https://medium.com/av-transcode/what-is-webrtc-and-how-to-setup-stun-turn-server-for-webrtc-communication-63314728b9d0> (Consulté le 14 août 2021)
- [34] HTML5ROCKS.COM, *Build the backend services needed for a WebRTC app*
<https://www.html5rocks.com/en/tutorials/webrtc/infrastructure/> (Consulté le 14 août 2021)
- [35] JANUS.CONF.MEETECHOCOM, *Janus: the general purpose WebRTC server*
<https://janus.conf.meetecho.com/> (Consulté le 14 août 2021)

[36] MICROSOFT DOCS, *Using the Windows Device Portal*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/platform-capabilities-and-apis/using-the-windows-device-portal> (Consulté le 14 août 2021)

[37] YOUTUBE.COM, *Fata Morgana – VIRTUAL DEMO*
<https://www.youtube.com/watch?v=WSuW6MiruiQ> (Consulté le 14 août 2021)

[38] MICROSOFT AZURE, *Spatial Anchors*
<https://azure.microsoft.com/en-us/services/spatial-anchors/#overview> (consulté le 14 août 2021)

12.2 Liens supplémentaires utiles

HoloLens 2 : Cannot connect to destination host – bad access
<https://github.com/microsoft/MixedReality-WebRTC/issues/784>

[Question] Is it possible to create a communication between 3 devices with MRTK-WebRTC
<https://github.com/microsoft/MixedReality-WebRTC/issues/769>

SLACK, *HoloDevelopers*
[Holodevelopers.slack.com](https://holodevelopers.slack.com)

REDDIT.COM, *Microsoft HoloLens*
<https://www.reddit.com/r/HoloLens/>

MICROSOFT DOCS, *RECOMMENDED SETTINGS FOR UNITY*
<https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/recommended-settings-for-unity>

UNITY DOCUMENTATION, *Order of execution for event functions*
<https://docs.unity3d.com/Manual/ExecutionOrder.html>

13. Acronymes

- RA : Réalité augmentée
- UE : Unreal Engine
- UE4 : Unreal Engine 4
- SDK : Software development kit
- UI : User interface (Interface utilisateur)
- HL2 : HoloLens 2
- JSON : JavaScript Object Notation
- MRTK : Mixed Reality Toolkit
- SUS : System Usability Scale
- VS : Visual Studio
- VS2019 : Visual Studio 2019

15. Annexes

Tout d'abord, le répertoire des projets Unity présentés dans ce rapport ainsi que les différents documents (code, schémas, etc.) liés à ce travail de Bachelor peuvent se retrouver en cliquant sur le lien GitLab ci-dessous. Dans le ReadMe de ce répertoire se trouve un guide pour les utilisateurs souhaitant déployer et tester les prototypes sur leurs propres machines.

- <https://gitlab.hevs.ch/SPL/bachelorthesis/2021-ar4ra/td-augmented-reality-for-remote-assistance>

Ensuite, les différentes annexes citées dans ce rapport se trouveront dans les pages qui suivent.

A. Code de l'Application Unity réalisé dans la prise en main

```

SpawnableManager.cs
Assets > Scripts > SpawnableManager.cs > SpawnableManager > Update()
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.XR.ARFoundation;
0 references
5  public class SpawnableManager : MonoBehaviour
6  {
7      [SerializeField] ARRaycastManager m_RaycastManager;
3 references
8      List<ARRaycastHit> m_Hits = new List<ARRaycastHit>();
1 reference
9      [SerializeField] GameObject spawnablePrefab;
10
11     Camera arCam;
2 references
12     GameObject spawnedObject;
7 references
13
14     // Start is called before the first frame update
0 references
15     void Start()
16     {
17         spawnedObject = null;
18         arCam = GameObject.Find("AR Camera").GetComponent<Camera>();
19     }
20
21     // Update is called once per frame
0 references
22     void Update()
23     {
24         if(Input.touchCount == 0)
25         {
26             return;
27         }
28
29         RaycastHit hit;
30         Ray ray = arCam.ScreenPointToRay(Input.GetTouch(0).position);
31

```

```

32     if(m_RaycastManager.Raycast(Input.GetTouch(0).position, m_Hits))
33     {
34         if(Input.GetTouch(0).phase == TouchPhase.Began && spawnedObject == null)
35         {
36             if(Physics.Raycast(ray, out hit))
37             {
38                 if(hit.collider.gameObject.tag == "Spawnable")
39                 {
40                     spawnedObject = hit.collider.gameObject;
41                 }
42                 else
43                 {
44                     spawnPrefab(m_Hits[0].pose.position);
45                 }
46             }
47         }
48         else if(Input.GetTouch(0).phase == TouchPhase.Moved && spawnedObject != null)
49         {
50             spawnedObject.transform.position = m_Hits[0].pose.position;
51         }
52         if(Input.GetTouch(0).phase == TouchPhase.Ended)
53         {
54             spawnedObject = null;
55         }
56     }
57 }
58
59 1 reference
60 private void spawnPrefab(Vector3 spawnPosition)
61 {
62     spawnedObject = Instantiate(spawnablePrefab, spawnPosition, Quaternion.identity);
63 }
64

```

B. Tableau comparant Unity et Unreal Engine^[13]

Fonctionnalités Plateformes de déploiement	Unity	Unreal Engine
Desktop	Windows PC, macOS, WebGL, Linux.	Windows PC, macOS, Linux, HTML5.
Mobile	Windows Phone, iOS, Android, Tizen.	Android, iOS.
Console	Xbox One, Xbox Series X, Wii U, PlayStation 4, PlayStation 5, PlayStation Vita, Nintendo Switch, Google Stadia.	PlayStation 4, PlayStation 5, Xbox One, Xbox Series X, Nintendo Switch.
VR/AR	SteamVR/HTC Vive, Oculus Rift, Google VR/Daydream, Samsung Gear VR, Microsoft Hololens, Playstation VR, Magic Leap One, OSVR.	SteamVR/HTC Vive, Oculus Rift, Google VR/Daydream, Samsung Gear VR, Microsoft Hololens, Playstation VR, Magic Leap One, OSVR.
TV	Android TV, tvOS, Samsung Smart TV.	tvOS
Editeur	Unity Editor	Unreal Editor, VR Editor for HTC VIVE
Fonctionnalités spéciales	Deux fenêtres pour la vue de la scène et la vue du jeu	Posséder et éjecter, simuler, navigateur de contenu
Langages de programmation	C#	C++, Blueprint visual scripting
Engine Code	Possibilité d'ajouter un plug-in GitHub à Unity	Source Code C++ disponible via GitHub
Formats supportés		
Image	psd, jpg, png, gif, bmp, tga, tiff, iff, pict, dds, exr, hdr	psd, jpg, png, exr, dds, exr, hdr, tga, pcx, float, bmp, ies
Audio	mp3, ogg, aiff, wav, mod, it, sm3	mp3, wav, wma, wave, snd, caf, cdda, aiff, aif, au
Vidéo	mov, avi, asf, mpg, mpeg, mp4	mov, avi, mp4, wmv, aac
3D Models	fbx, .dae (Collada), .3ds, .dxf, c4d, jas, lxo, blend, skp	fbx, obj, srt (speedtree)
Rendering	Shading différencié	Shading différencié
Matériaux	Basé sur la physique, Unity Standard shader, tessellation shader, mobile shaders	Basé sur la physique, Blueprint éditeur de matériaux, tessellation, Matériaux en couches, instances des

		matériaux, translucidité éclairée, modèle d'ombrage subsurf
Eclairage	Directionnel, point, spotlight, lumière de zone	Directionnel, point, spotlight, lumière de zone, lumière du ciel, IES light profiles, Rect Lights
Ombrages	Ombres dures et douces en temps réel	Ombres dures et douces en temps réel, ombres en cascade, ombres à distance,
Illumination globale (IG)	IG Basée sur l'image, IG précalculée en temps réel, Baked IG	IG Basée sur l'image, Baked IG
Réflexions/Réfractions	Sondes de réflexion	Sondes de réflexion, réflexion « screen space », réfraction
Particules PostProcess	Particules basées sur des courbes simples Effets avec assets supplémentaires (p. ex. : pile de post-traitement)	Système de particules en cascade avec particules GPU Effets faisant partie du moteur (AA, Bloom, DoF, etc.), support pour les matériaux de post-traitement.
Animation	Éditeur de courbes d'animation, Animations squelettiques, Mélange formes, poids des animations, événements aux animations, State Machine et transitions	Éditeur de courbes d'animation, jeu d'outils d'animation Persona, machines à états, animation basée sur la physique, plans d'animation, cibles de morphing
Physiques	Corps rigides, collisions, articulations, tissu, collisionneur de roues, physique matériaux, PhysX	Corps rigides, collisions, matériaux physiques, intégration APEX, PhaT (éditeur de physique de maillage squelettique), physique des véhicules, PhysX
Outils de cinématique	-	Matinee cinematic toolset, outil de séquençage
Terrain et feuillage	Moteur de terrain simple, peinture de texture, éditeur d'arbres, plugins pour la construction de monde	Outils très avancés de peinture de paysages et outils de peinture de feuillage dans l'éditeur.
UI	Objets de l'interface utilisateur sur le Canvas, IMGUI	Widgets et blueprints
IA	NavMesh, NavAgents, Path finding	Arbres comportementaux, Caractère contrôleurs de personnages, NavMesh, Path finding
Optimisations	Unity Profiler (CPU, GPU, mémoire, physique), support LOD, présélections de qualité	Profilage GPU/CPU, enregistrement des statistiques dans un fichier, LOD hiérarchique, génération automatique de LOD, Optimisation viewmodes

C. Plateformes et appareils pris en charge par MRTK-Unity

Plateforme	Appareils pris en charge
OpenXR (Unity 2020.3.8+)	Microsoft HoloLens 2 Casques Windows Mixed Reality
Windows Mixed Reality	Microsoft HoloLens Microsoft HoloLens 2 Casques Windows Mixed Reality
Oculus (Unity 2019.3 ou ultérieur)	Oculus Quest
OpenVR	Casques Windows Mixed Reality HTC Vive Oculus Rift
Suivi de la main Ultraleap	Ultraleap Leap Motion Controller
Mobile	iOS et Android

Figure 136 - Plateformes et appareils pris en charge par MRTK-Unity

Source : <https://docs.microsoft.com/fr-ch/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05>

D. Tests Black-box PC

Test ID	Call0_PC
Description	Ce test vérifie que le prototype peut appeler les appareils pairs.
Pattern	Blackbox
Prescription	Le serveur node.js tourne, les trois applications prototypes sont en marche, l'utilisateur PC essaie d'appeler les appareils pairs.
Personnel	Alexandre Figueiredo
Résultats	La communication audio et vidéo est établie.
Conclusions	Test réussi.
Mesures	Call0_PC étant réussi, passer à Call1_PC.

Tableau 5 - Tableau 3 - Test Call0_PC

Test ID	Call1_PC
Description	Ce test vérifie que le prototype peut recevoir des appels des appareils pairs.
Pattern	Blackbox
Prescription	Le serveur node.js tourne, les autres prototypes tentent d'établir une communication avec ce prototype.
Personnel	Alexandre Figueiredo
Résultats	La communication audio et vidéo est établie.
Conclusions	Test réussi.
Mesures	Call1_PC étant réussi, passer à Ann1_PC.

Tableau 6 - Test Call1_PC

Test ID	Ann1_PC
Description	Ce test vérifie qu'il est possible d'activer des outils d'annotations à distance.
Pattern	Blackbox
Prescription	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur essaie alors d'activer les outils à distances en pressant les boutons de l'UI sur son PC.
Personnel	Alexandre Figueiredo
Résultats	Les outils de dessin libre, d'effacement, de placement de flèches et de textes ont été activés correctement à distance.
Conclusions	Test réussi.
Mesures	Ann1_PC étant réussi, passer à Ann2_PC.

Tableau 7 - Test Ann1_PC

Test ID	Ann2_PC
Description	Ce test vérifie qu'il est possible de déplacer les outils d'annotations à distance.
Pattern	Blackbox
Prescription	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur essaie alors de déplacer les outils à distances à l'aide de la souris.
Personnel	Alexandre Figueiredo
Résultats	Les outils de dessin libre, d'annotations avec des flèches et de textes ont été déplacés correctement.
Conclusions	Test réussi.
Mesures	Ann2_PC étant réussi, passer à Ann3_PC.

Tableau 8 - Test Ann2_PC

<i>Test ID</i>	Ann3_PC
<i>Description</i>	Ce test vérifie qu'il est possible dessiner librement et placer des flèches à distance.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur essaie alors de placer les diverses annotations à distances à l'aide des clicks de sa souris ayant activés ces fonctionnalités au préalable à l'aide des boutons de l'UI.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les dessins libres, et les annotations avec des flèches ont été correctement réalisés.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann3_PC étant réussi, passer à Ann4_PC.

Tableau 9 - Test Ann3_PC

<i>Test ID</i>	Ann4_PC
<i>Description</i>	Ce test vérifie qu'il est possible de placer des annotations textes à distance tapées au clavier de l'ordinateur
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur sélection l'outils texte, il écrit le mot « test », tape la touche « Enter » pour valider le mot et le place dans l'environnement à l'aide de la souris.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Le mot « test » a correctement été envoyé et placé à distance dans le prototype de l'HoloLens 2.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann4_PC étant réussi, passer à Ann5_PC.

Tableau 10 - Test Ann4_PC

<i>Test ID</i>	Ann5_PC
<i>Description</i>	Ce test vérifie qu'il est possible de changer la couleur des outils et des annotations de l'HoloLens 2 à distance depuis le PC.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur sélectionne alors les différents outils distants, fait des annotations et change de couleurs au moyen de la palette de couleur du prototype PC.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les différents outils ainsi que les nouvelles annotations changeaient de couleur en respectant celle choisie depuis la palette de couleur du PC.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann5_PC étant réussi, passer à Ann6_PC.

Tableau 11 - Test Ann5_PC

<i>Test ID</i>	Ann6_PC
<i>Description</i>	Ce test vérifie qu'il est possible de changer l'épaisseur du trait de dessin à distance.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur sélectionne l'outil de dessin libre et tente de modifier sa taille à l'aide du slider de l'UI PC dû à cet effet.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	L'épaisseur des annotations libres à correctement changé.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann6_PC étant réussi, les test PC sont terminés.

Tableau 12 - Test Ann6_PC

E. Tests Black-box Smartphone

<i>Test ID</i>	Call0_Smartphone
<i>Description</i>	Ce test vérifie que le prototype peut appeler les appareils pairs.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le serveur node.js tourne sur le PC, les trois applications prototypes sont en marche, l'utilisateur Smartphone essaie d'appeler les appareils pairs.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	La communication audio et vidéo est établie. Cependant, il arrive que la voix ne soit pas toujours transmise.
<i>Conclusions</i>	Réussi quelques fois, mais il semble qu'un problème de compatibilité entre MRTK et Smartphone empêche d'accéder au micro de l'appareil.
<i>Mesures</i>	Des tentatives de debug ont été tentées mais en vain. Le temps ne permettant pas de continuer, des suggestions sont apportées dans ce rapport. Les tests continuent avec Call1_Smartphone

Tableau 13 - Tableau 3 - Test Call0_Smartphone

<i>Test ID</i>	Call1_Smartphone
<i>Description</i>	Ce test vérifie que le prototype peut recevoir des appels des appareils pairs.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le serveur node.js tourne sur le PC, les autres prototypes tentent d'établir une communication avec ce prototype.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	La communication audio et vidéo est établie. Pareil que précédemment le son n'est pas toujours transmis
<i>Conclusions</i>	Réussi quelques fois, mais il semble qu'un problème de compatibilité entre MRTK et Smartphone empêche d'accéder au micro de l'appareil.
<i>Mesures</i>	Mêmes mesures que pour Call0_Smartphone. Les tests continuent avec Ann1_Smartphone

Tableau 14 - Test Call1_PC

<i>Test ID</i>	Ann1_Smartphone
<i>Description</i>	Ce test vérifie qu'il est possible d'activer des outils d'annotations à distance.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes Smartphone et HoloLens 2 sont en marche. L'utilisateur essaie alors d'activer les outils à distances en pressant les boutons de l'UI Smartphone.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les outils de dessin libre, d'effacement, de placement de flèches et de textes ont été activés correctement à distance.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann1_Smartphone étant réussi, passer à Ann2_Smartphone.

Tableau 15 - Test Ann1_Smartphone

<i>Test ID</i>	Ann2_Smartphone
<i>Description</i>	Ce test vérifie qu'il est possible de déplacer les outils d'annotations à distance.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes Smartphone et HoloLens 2 sont en marche. L'utilisateur essaie alors de déplacer les outils à distances à l'aide de l'écran tactile.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les outils de dessin libre, d'annotations avec des flèches et de textes ont été déplacés mais un calibrage est nécessaire car l'emplacement des outils ne correspond pas tout à fait à l'emplacement du doigt sur l'écran tactile.
<i>Conclusions</i>	Test réussi mais calibrage nécessaire.
<i>Mesures</i>	Ann2_Smartphone étant réussi (calibrage nécessaire), passer à Ann3_Smartphone.

Tableau 16 - Test Ann2_Smartphone

<i>Test ID</i>	Ann3_Smartphone
<i>Description</i>	Ce test vérifie qu'il est possible dessiner librement et placer des flèches à distance.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes Smartphone et HoloLens 2 sont en marche. L'utilisateur essaie alors de placer les diverses annotations à distances à l'aide de touches sur l'écran tactile, ayant activés ces fonctionnalités au préalable à l'aide des boutons de l'UI.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les dessins libres, et les annotations avec des flèches ont été correctement réalisés.
<i>Conclusions</i>	Test réussi. (Calibrage lié au test Ann2_Smartphone nécessaire)
<i>Mesures</i>	Ann3_Smartphone étant réussi, passer à Ann4_Smartphone.

Tableau 17 - Test Ann3_Smartphone

<i>Test ID</i>	Ann4_Smartphone
<i>Description</i>	Ce test vérifie qu'il est possible de placer des annotations textes à distance tapées au clavier du smartphone
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes Smartphone et HoloLens 2 sont en marche. L'utilisateur sélectionne l'outil texte, il écrit le mot « test », tape la touche « Enter » pour valider le mot et le place dans l'environnement à l'aide d'une touche sur l'écran tactile.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Le mot « test » a correctement été envoyé et placé à distance dans le prototype de l'HoloLens 2.
<i>Conclusions</i>	Test réussi. (Calibrage lié au test Ann2_Smartphone nécessaire)
<i>Mesures</i>	Ann5_Smartphone étant réussi, passer à Ann6_Smartphone.

Tableau 18 - Test Ann5_Smartphone

<i>Test ID</i>	Ann6_Smartphone
<i>Description</i>	Ce test vérifie qu'il est possible de changer la couleur des outils et des annotations de l'HoloLens 2 à distance depuis le Smartphone.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Les prototypes PC et HoloLens 2 sont en marche. L'utilisateur sélectionne alors les différents outils distants, fait des annotations et change de couleurs au moyen de la palette de couleur du prototype Smartphone.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les différents outils ainsi que les nouvelles annotations changeaient de couleur en respectant celle choisie depuis la palette de couleur du Smartphone.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann6_Smartphone étant réussi, les tests smartphones sont terminés

Tableau 19 - Test Ann6_PC

F. Tests Black-box HL2

<i>Test ID</i>	Call0_HL2
<i>Description</i>	Ce test vérifie que le prototype peut appeler les appareils pairs.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le serveur node.js tourne, les trois applications prototypes sont en marche, l'utilisateur HL2 essaie d'appeler les appareils pairs.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	La communication audio et vidéo est établie.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Call0_HL2 étant réussi, passer à Call1_HL2.

Tableau 20 - Test Call0_HL2

<i>Test ID</i>	Call1_HL2
<i>Description</i>	Ce test vérifie que le prototype peut recevoir des appels des appareils pairs.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le serveur node.js tourne, les autres prototypes tentent d'établir une communication avec ce prototype.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	La communication audio et vidéo est établie.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Call1_HL2 étant réussi, passer à Ann1_HL2.

Tableau 21 - Test Call1_HL2

<i>Test ID</i>	Ann1_HL2
<i>Description</i>	Ce test vérifie qu'il est possible d'accéder aux différents UI de l'application HoloLens 2.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	L'app HoloLens 2 est en marche. L'utilisateur regarde alors la paume de ça main. Un menu devrait apparaître avec des boutons. En cliquant sur ces boutons, d'autres UI devraient apparaître.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	En regardant la paume de sa main, un menu est apparu avec des boutons. La pression de ces boutons a correctement activé les fonctionnalités souhaitées
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann1_HL2 étant réussi, passer à Ann2_HL2.

Tableau 22 - Test Ann1_HL2

<i>Test ID</i>	Ann2_HL2
<i>Description</i>	Ce test vérifie que l'activation et la désactivation des outils d'annotations est correctement réalisée lors de la pressions de boutons ou lors de l'apparition/disparition de la main de l'utilisateur dans le champ de vision de la camera.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur essaie alors d'activer les fonctionnalités d'annotations à l'aide du menu d'annotations. Si elles s'activent, il contrôle ensuite qu'elles sont désactivées lors de la disparition de la main de l'utilisateur et réactivé lors de sa réapparition.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les outils d'annotations ont correctement été activés et désactivés au moment voulus.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann2_HL2 étant réussi, passer à Ann3_HL2.

Tableau 23 - Test Ann2_HL2

<i>Test ID</i>	Ann3_HL2
<i>Description</i>	Ce test vérifie que les outils d'annotations suivent correctement la main de l'utilisateur lorsqu'ils sont actifs.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur essaie alors d'activer une fonctionnalité d'annotation et bouge sa main dans différentes directions afin de contrôler que les outils d'annotations suivent correctement sa main.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	La sphère de dessin, la croix effaçable, la flèche ainsi que la zone de texte ont correctement suivi la main de l'utilisateur.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann3_HL2 étant réussi, passer à Ann4_HL2.

Tableau 24 - Test Ann3_HL2

<i>Test ID</i>	Ann4_HL2
<i>Description</i>	Ce test vérifie qu'il est possible de placer les diverses annotations et de les effacer.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur sélectionne alors les outils et les placent les un après l'autre pour vérifier qu'ils sont correctement placés.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les différents outils ont pu être correctement placés et effacés. De plus, le clavier de l'Hololens 2 a fonctionné correctement.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann4_HL2 étant réussi, passer à Ann5_HL2.

Tableau 25 - Test Ann4_HL2

<i>Test ID</i>	Ann5_HL2
<i>Description</i>	Ce test vérifie qu'il est possible de changer la couleur des outils et des annotations de l'HoloLens 2.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens est en marche. L'utilisateur sélectionne alors les différents outils, fait des annotations et change de couleurs au moyen de la palette de couleur du menu d'annotation.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les différents outils ainsi que les nouvelles annotations changeaient de couleur en respectant celle choisie.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann5_HL2 étant réussi, passer à Ann6_HL2.

Tableau 26 - Test Ann5_HL2

<i>Test ID</i>	Ann6_HL2
<i>Description</i>	Ce test vérifie qu'il est possible de changer l'épaisseur du trait de dessin.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur sélectionne l'outil de dessin libre et tente de modifier sa taille à l'aide du slider du menu d'annotation dû à cet effet.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	L'épaisseur des annotations libres a correctement changé.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	Ann6_HL2 étant réussi, passer à UI1_HL2

Tableau 27 - Test Ann6_HL2

<i>Test ID</i>	UI1_HL2
<i>Description</i>	Ce test vérifie que les fenêtres dotées du suivi de mouvement de la tête suivent correctement l'utilisateur
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur ouvre alors les fenêtres étant capable de suivre l'utilisateurs (menu d'annotation, menu d'appel et lecteurs vidéo distants). Il se déplace et contrôle que les fenêtres le suivent correctement
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les fenêtres ont correctement suivi l'utilisateur.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	UI1_HL2 étant réussi, passer à UI2_HL2

Tableau 28 - Test UI1_HL2

<i>Test ID</i>	UI2_HL2
<i>Description</i>	Ce test vérifie que les fenêtres dotées du suivi de mouvement de la tête puissent être capable d'être fixées à un emplacement.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur ouvre alors les fenêtres étant capable de suivre l'utilisateurs (menu d'annotation, menu d'appel et lecteurs vidéo distants). Il clique sur le bouton permettant de fixer l'emplacement de ces fenêtres et ce met en mouvement. Ce test est réussi si les fenêtres restent sur place.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les fenêtres sont correctement restées à l'emplacement fixe désiré.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	UI2_HL2 étant réussi, passer à UI3_HL2

Tableau 29 - Test UI2_HL2

<i>Test ID</i>	UI3_HL2
<i>Description</i>	Ce test vérifie que la fonctionnalité main libre fonctionne correctement
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur fait des annotations et clique sur le bouton « main libre ». Les fonctionnalités d'annotations devraient être désactivées.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les fonctionnalités ont correctement désactiver les fonctionnalités
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	UI3_HL2 étant réussi, passer à UI4_HL2

Tableau 30 - Test UI3_HL2

<i>Test ID</i>	UI4_HL2
<i>Description</i>	Ce test vérifie que le slider du menu annotation change correctement la taille de la ligne de dessin.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur varie la position du slider du menu annotation et fait des lignes de dessin. Le test est réussi si les lignes de dessins changent en dépend de la position du slider. (haut = épais)
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les lignes de dessins ont correctement changé.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	UI4_HL2 étant réussi, passer à UI5_HL2

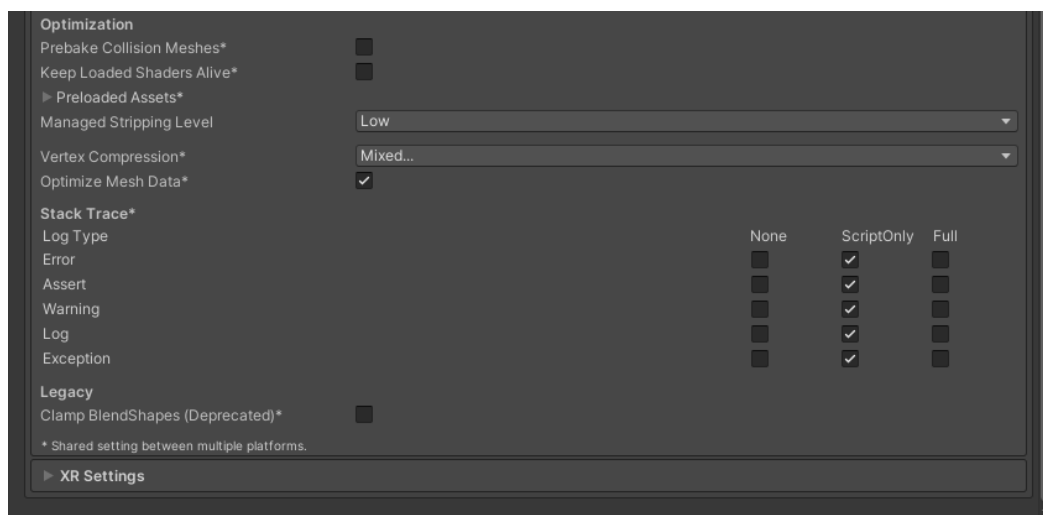
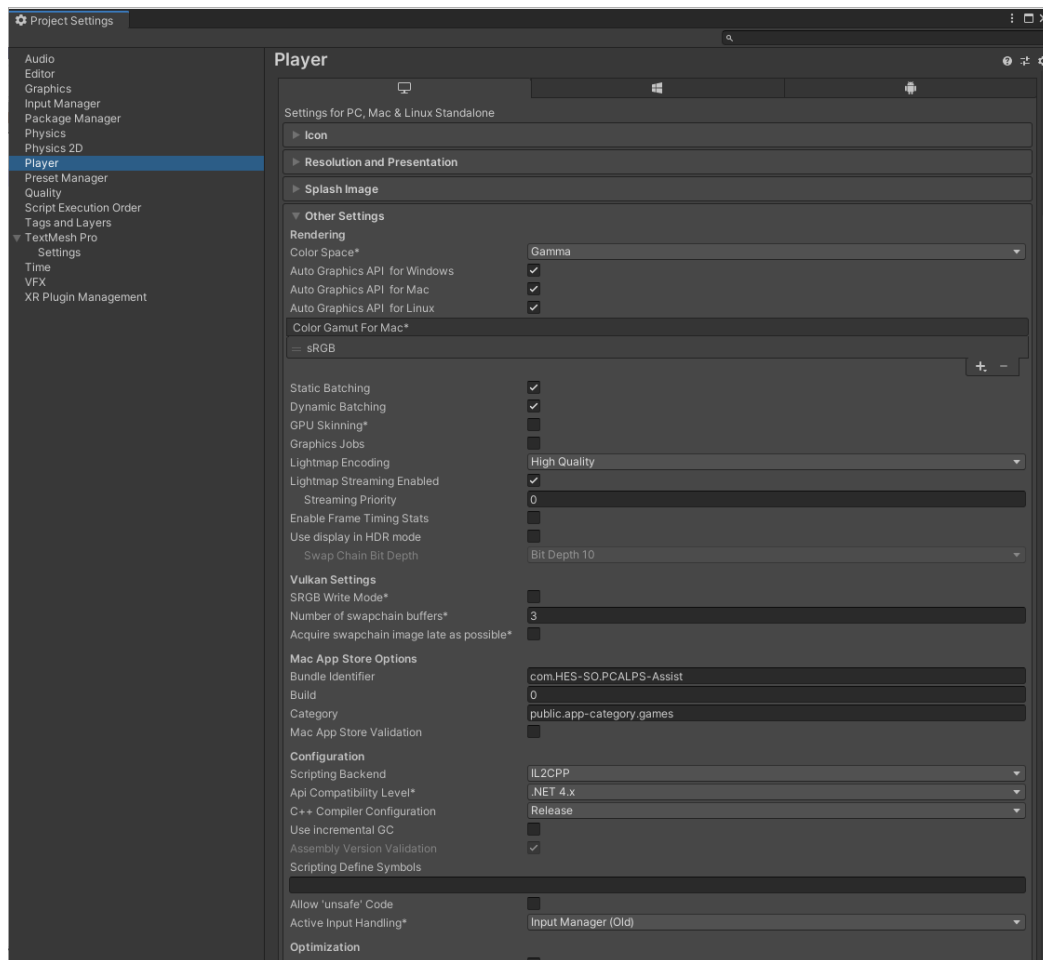
Tableau 31 - Test UI4_HL2

<i>Test ID</i>	UI5_HL2
<i>Description</i>	Ce test vérifie que les annotations instanciées peuvent être manipulées par les main de l'utilisateur.
<i>Pattern</i>	Blackbox
<i>Prescription</i>	Le prototype HoloLens 2 est en marche. L'utilisateur place alors des annotations flèches et textes. Il sélectionne alors la fonctionnalité « main-libres ». Il essaie ensuite de les déplacer avec ces mains, de changer leur rotation et de les agrandir.
<i>Personnel</i>	Alexandre Figueiredo
<i>Résultats</i>	Les annotations instanciées ont correctement réagi.
<i>Conclusions</i>	Test réussi.
<i>Mesures</i>	UI5_HL2 étant réussi, les tests HL2 sont terminés

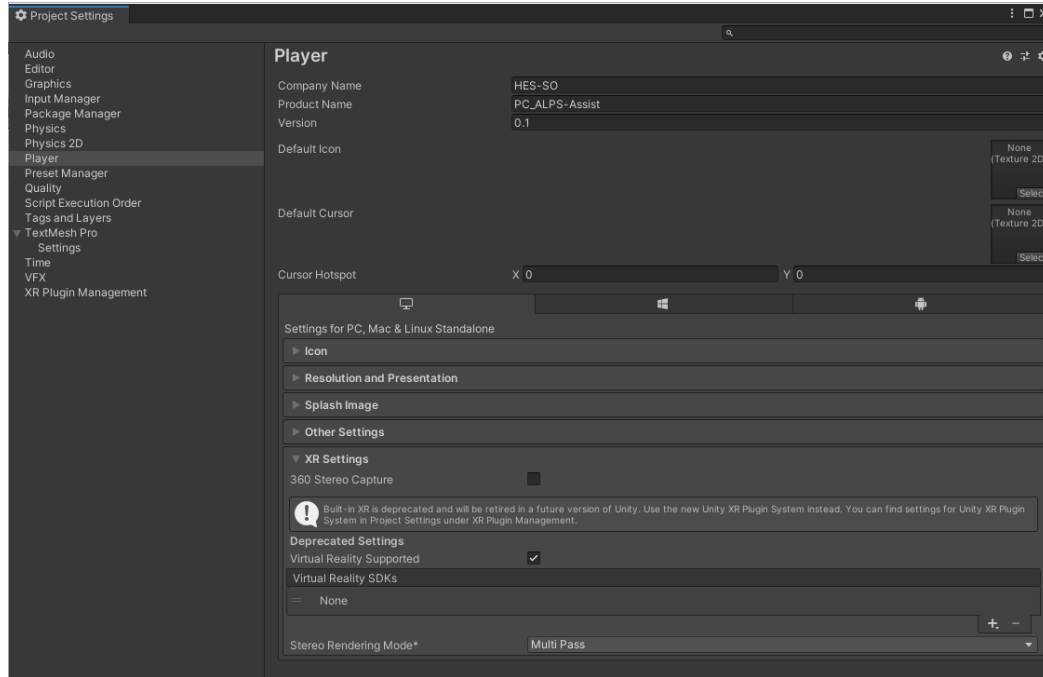
Tableau 32 - Test UI5_HL2

G. Configuration du « Player » pour déployer sur PC

G.1 Other Settings

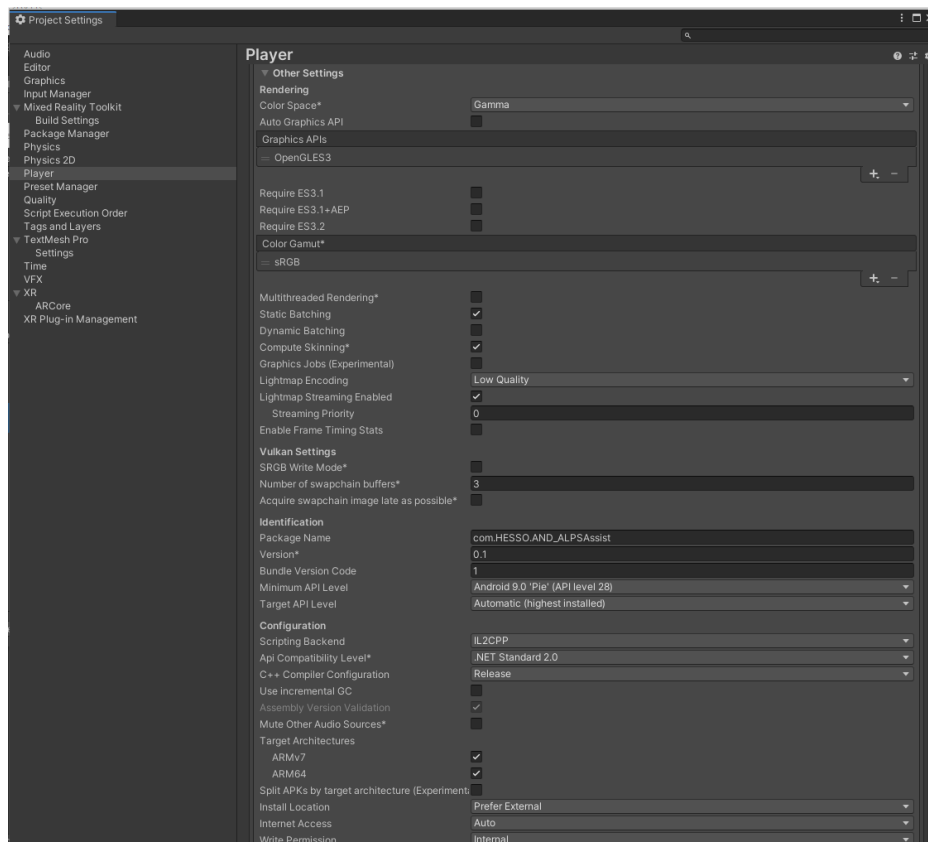


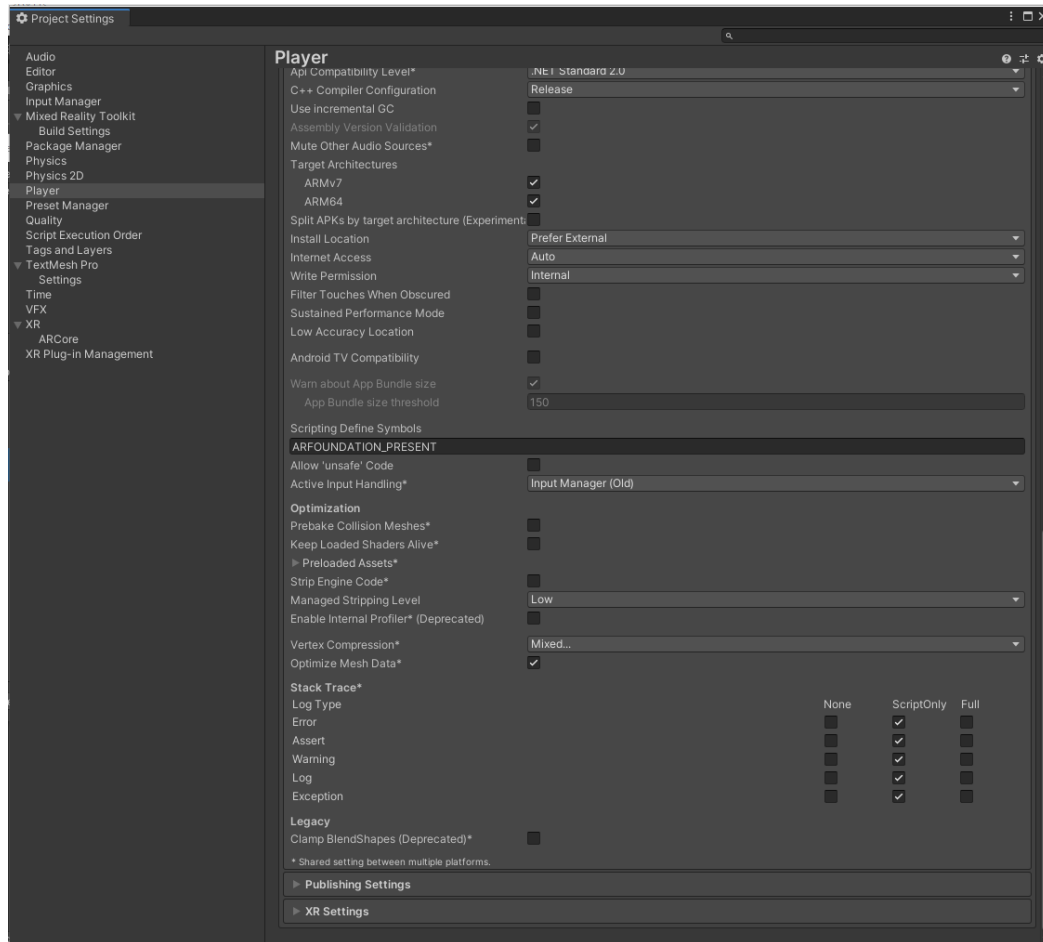
G.2 XR Settings



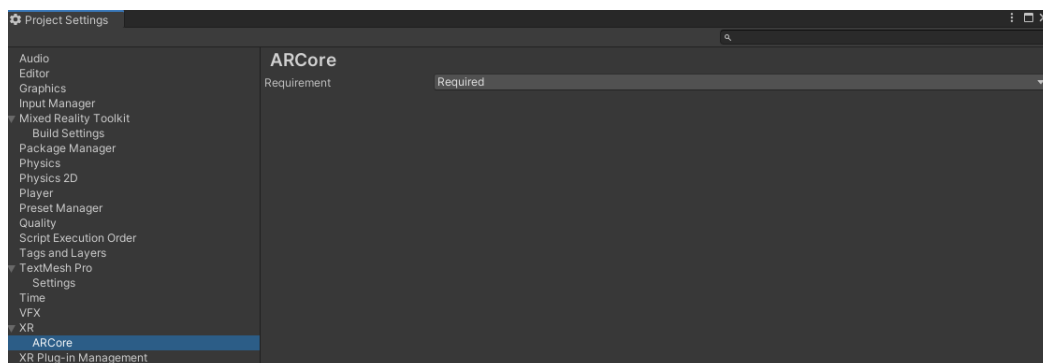
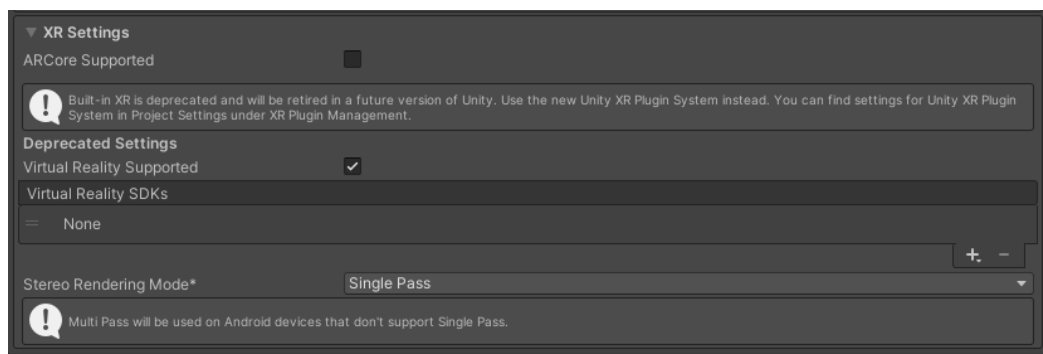
H. Configuration du « Player » pour déployer sur smartphone

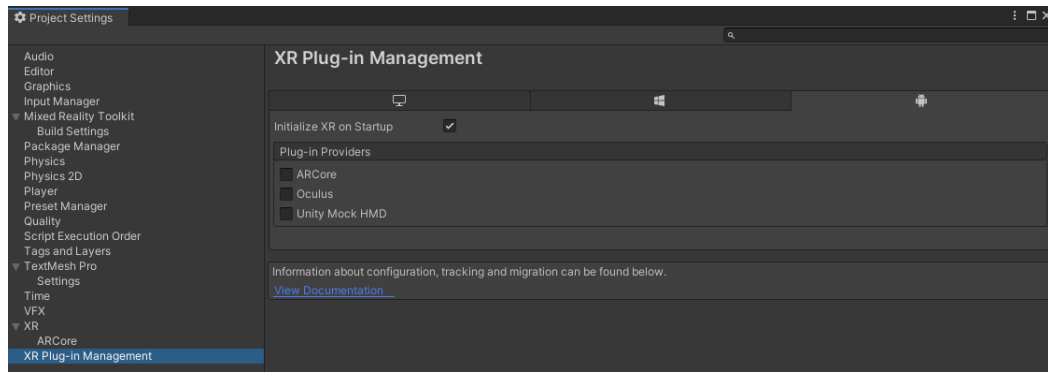
H.1 Other Settings





H.2 XR Settings





I. Modules de déploiements installés dans Unity

Add Modules			
Platforms			
✓	Android Build Support	Installed	1.1 GB
✓	Android SDK & NDK Tools	Installed	3.0 GB
✓	OpenJDK	Installed	70.5 MB
✓	Universal Windows Platform Build Support	Installed	2.1 GB
□	WebGL Build Support	252.5 MB	921.4 MB
✓	Windows Build Support (IL2CPP)	Installed	358.1 MB

J. Scénario de test des prototypes

HES-SO Valais

Sion

TD 2021

Remote assistance

-


Scénario de test du prototype



Chemin des cibleries 3 . CP 288 . CH-1896 Vouvry . Tel. +41 24 482 06 50 . Fax +41 24 482 06 51 . info@alpsautomation.ch . www.alpsautomation.ch


13 août 2021

Page : 1/5

	Remote assistance with AR Scenario test prototype	<i>Doc: FOR018 - TD HEVS</i> <i>Remote Assistance -</i> <i>Scénario test prototype v0.2</i> Rev: v0.1
---	--	---

SOMMAIRE

1. TRAÇABILITE DU DOCUMENT.....	3
1.1 HISTORIQUE	3
1.2 VERIFICATION	3
1.3 DISTRIBUTION.....	3
1.4 GLOSSAIRE.....	3
1.5 OBJECTIF	3
1.6 DOCUMENTS DE REFERENCE	3
2. CONTEXTE.....	4
2.1 PERSONNES IMPLIQUEES	4
3. TEST DU PROTOTYPE	4
3.1 PREREQUIS.....	4
3.2 INTERFACES ENTRE LES PARTICIPANTS	4
3.3 DEROULEMENT DU SCENARIO DE TEST	4

	Remote assistance with AR Scenario test prototype	<i>Doc: FOR018 - TD HEVS</i> <i>Remote Assistance -</i> <i>Scénario test prototype v0.2</i>
		Rev: v0.1

1. TRAÇABILITE DU DOCUMENT

1.1 HISTORIQUE

Date de modification	Type de modification	Modifié par	Version
30 juin 2021	Création	AGR	v0.1
13 août 2021	Adaptations pour tests à l'atelier FAB	AGR	v0.2

1.2 VERIFICATION

	Nom	Fonction	Date	Signature
Préparé par	AGR	Expert TD	13 août 2021	
Vérifié par				
Approuvé par				

1.3 DISTRIBUTION

Où	Qui
Externe	Alexandre Figueiredo Francesco Carrino
Interne	

1.4 GLOSSAIRE


CF : [DOC003 – Liste des abréviations](#)

1.5 OBJECTIF

L'objectif de ce document est de décrire un scénario permettant de tester le prototype d'assistance à distance développé par Monsieur Alexandre Figueiredo dans le cadre de son travail de diplôme au sein de la HES Sion.

1.6 DOCUMENTS DE REFERENCE

Nom	Type	Description

	Remote assistance with AR Scenario test prototype	<i>Doc: FORD18 - TD HEVS</i> <i>Remote Assistance -</i> <i>Scénario test prototype v0.2</i> Rev: v0.1
---	--	--

2. CONTEXTE

Afin de tester le prototype d'assistance à distance réalisé par Monsieur Figueiredo dans le cadre de son travail de diplôme, un scénario impliquant trois participants a été imaginé. Il se déroule dans l'atelier de fabrication d'ALPS Automation à Vouvry et s'articule autour d'une armoire électrique en cours de fabrication afin de disposer de différents composants électrotechniques semblables à ceux qui pourraient être rencontrés lors d'une situation réelle d'assistance à distance. Dans ce scénario, il n'y aucune intention de réaliser des actions à buts concrets (mise en route/dépannage/test) mais l'on peut partir du principe qu'il s'agit d'une assistance durant une phase de mise en service d'une nouvelle armoire électrique. L'unique objectif est de tester les fonctionnalités du prototype d'assistance à distance et recueillir des retours d'expérience des participants.

Le prototype utilisant un casque de réalité augmentée HoloLens 2, le scénario sera joué une deuxième fois avec les lunettes RealWear HMT-1 et TeamViewer Pilot afin de pouvoir comparer différentes solutions matérielles et logicielles. Pour le test, les casques, laptops et smartphones communiquent via le WiFi ALPS. Le scénario de test sera joué avec 3 équipes distinctes composées de collaborateurs ALPS de différents départements de manière à couvrir un large panel de sensibilités.

2.1 PERSONNES IMPLIQUEES

- Un automaticien de l'usine sur site (COL-SITE)
- Un automaticien d'ALPS automation au bureau (COL-ALPS)
- Un technicien spécialiste de l'équipement testé (COL-TECH)
- Alexandre Figueiredo : étudiant réalisant le travail de Bachelor (ETU)
- Francesco Carrino : superviseur du travail de Bachelor (SUP)
- Alain Gruet : expert du travail de Bachelor (EXP)

2.1.1 FICHIERS/PROGRAMMES

- Schémas électriques.
- Documentation technique.

3. TEST DU PROTOTYPE

3.1 PREREQUIS

- Installation des applications sur les appareils.
- Connection des appareils au WiFi ALPS.
- Tests de connectivité entre les différents appareils.
- Prise en main du casque de réalité augmentée par COL-SITE.

3.2 INTERFACES ENTRE LES PARTICIPANTS

Personne	Hardware	Application
COL-SITE	HoloLens 2/RealWear HMT-1	Prototype
COL-ALPS	PC	Prototype
COL-TECH	Smartphone	Prototype

ETU, SUP et EXP observent les interactions entre les participants qui jouent le scénario de test et les guident au besoin.

3.3 DEROULEMENT DU SCENARIO DE TEST

1. COL-SITE et COL-ALPS lancent l'application « Remote Assist » sur leur appareil respectif. Lorsque la connexion entre le laptop et le casque est établie et que le flux vidéo du site est visible sur l'application PC, COL-ALPS informe COL-SITE que l'assistance à distance peut débuter.
2. Il est décidé que COL-SITE utilise la couleur orange pour effectuer ses annotations et que COL-ALPS utilise le bleu.
3. COL-SITE fait remarquer à COL-ALPS qu'il manque une étiquette sur un relais 230V et dessine un cercle à l'emplacement auquel celle-ci devrait se trouver.
4. COL-ALPS partage le schéma électrique avec COL-SITE (**emplacement à spécifier**) et lui indique la désignation de l'élément dont l'étiquette manque (page 40, 70K9). COL-ALPS place

ALPS AUTOMATION	Remote assistance with AR Scenario test prototype	Doc: FOR018 - TD HEVS Remote Assistance - Scénario test prototype v0.2 Rev: v0.1
---------------------------	--	---

- une annotation avec la désignation du composant « 70K9 » dont l'étiquette manque à l'emplacement correspondant.
- COL-SITE supprime le cercle précédemment tracé puis capture une image qui contient le relais « 70K9 » ainsi que sa désignation annotée.
 - COL-ALPS demande à COL-SITE de regarder la partie inférieure de l'armoire électrique puis trace un cercle bleu autour des disjoncteurs « 60F4 », « 70F2 » et « 70F6 » et lui demande de basculer les trois interrupteurs dans la position « 0-OFF ».
 - COL-ALPS demande à COL-SITE de se munir d'un tournevis, puis diriger son regard à l'intérieur de l'armoire sur la droite à mi-hauteur où se trouvent les thermostats. COL-ALPS place une flèche bleue qui pointe sur le bouton rotatif rouge du thermostat « 70B6 » et demande à COL-SITE de régler la température à 25°C tout en annotant « 25 » à proximité du bouton rotatif.
 - COL-SITE capture une image du thermostat « 70B6 » qui montre que la température a été correctement réglée.
 - COL-ALPS demande à COL-SITE de remettre les trois interrupteurs des disjoncteurs « 60F4 », « 70F2 » et « 70F6 » dans la position « 1-ON ».
 - COL-ALPS demande à COL-SITE de regarder les afficheurs analogiques sur la face avant de la porte de l'armoire électrique et démarrer un enregistrement vidéo.
 - COL-ALPS entoure le bouton-poussoir « ACQUITTEMENT DEFAULT » d'un cercle bleu et demande à COL-SITE de le presser, puis vérifier que les LEDs « Défaut Majeur » et « Défaut Mineur » sont éteintes.
 - L'installation ne fonctionne malheureusement pas encore. COL-SITE place une flèche en direction des bornes X290 #001 & #002 ainsi qu'une annotation « ORDRE EXTÉRIEUR DE CHARGE » et informe COL-ALPS qu'aucun câble n'y est raccordé.
 - COL-ALPS répond que les premiers tests peuvent être réalisés sans ce signal mais demande à COL-SITE quand le câble sera raccordé.
 - COL-SITE ne dispose pas de l'information, et invite COL-TECH à rejoindre la discussion.
 - Lorsque COL-TECH est connecté via son smartphone, COL-SITE lui explique la situation et lui demande quand le câble du signal « ORDRE EXTÉRIEUR DE CHARGE » sera raccordé.
 - COL-TECH répond que cette tâche sera terminée jeudi midi et ajoute l'annotation « 19.08.2021 12:00 » en vert à côté de la flèche en direction des bornes.
 - COL-SITE capture une image du bornier et des annotations.
 - COL-TECH mentionne cependant que le disjoncteur électronique « 80F3 » qu'il désigne au moyen d'un cercle est défectueux. Il ajoute une flèche entre un carton au fond de l'armoire qui contient le nouveau disjoncteur et celui défectueux.
 - COL-ALPS et COL-SITE décident de faire une pause. COL-SITE met fin à l'enregistrement vidéo puis supprime toutes les annotations.

K. Formulaire d'autorisation pour l'utilisation de données de feedback

Formulaire d'autorisation pour l'utilisation des données feedback

Les données des utilisateurs de test resteront entièrement anonymes et privées.

→ **A renseigner lisiblement.**

Je soussigné(e) (nom et prénom) :

Adresse :
.....

Email :

Téléphone :

Autorise toute personne liée au travail de diplôme « Augmented Reality for remote assistance », réalisé par la Haute école d'ingénierie du Valais en collaboration avec l'entreprise ALPS Automation SA à utiliser mes données de « feedback » concernant les prototypes testés aujourd'hui. En conséquence, les informations recueillies par les formulaires de « feedback » pourront être mentionnés dans le rapport de projet final.

Finalité des collectes d'informations :

Les données collectés via les formulaires de feedback seront utilisées en vue de :

- Obtenir le ressenti des utilisateurs concernant les prototypes développés dans le cadre de ce projet.
- Obtenir des recommandations d'amélioration venant de personnes pouvant potentiellement utiliser les prototypes si une mise en production officielle venait à se faire.

Le responsable de traitement : Alexandre Figueiredo

Date / Lieu :

Signature :

→ Questionnaire de Feed-back au dos de la feuille.

Questionnaire de feedback

Veuillez cocher le prototype testé :



PC



HoloLens 2
(casque)



Smartphone

Veuillez cocher les bulles qui vous semblent le plus approprié. Veuillez noter que ce n'est pas un produit fini et étant en cours de développement. Si le prototype est finalisé et fourni par l'entreprise...

1. Je pense que je l'utiliserais fréquemment pour des assistances à distances.



Pas du tout
d'accord



Pas d'accord



Ni d'accord ni
pas d'accord



D'accord



Tout à fait
d'accord

2. Je pense que le service est inutilement complexe.



Pas du tout
d'accord



Pas d'accord



Ni d'accord ni
pas d'accord



D'accord



Tout à fait
d'accord

3. Je pense que le service est facile d'utilisation.



Pas du tout
d'accord



Pas d'accord



Ni d'accord ni
pas d'accord



D'accord



Tout à fait
d'accord

4. Je pense que je vais devoir faire appel au support technique pour pouvoir utiliser ce service.



Pas du tout
d'accord



Pas d'accord



Ni d'accord ni
pas d'accord



D'accord



Tout à fait
d'accord

5. Je trouve que les fonctionnalités du service sont bien intégrées.



Pas du tout
d'accord



Pas d'accord



Ni d'accord ni
pas d'accord



D'accord



Tout à fait
d'accord

Suite à la page
suivante.

6. Je trouve qu'il y a beaucoup trop d'incohérences dans ce service.

1

Pas du tout d'accord

2

Pas d'accord

3

Ni d'accord ni pas d'accord

4

D'accord

5

Tout à fait d'accord

7. Je pense que la plupart des gens apprennent très rapidement à utiliser le service.

1

Pas du tout d'accord

2

Pas d'accord

3

Ni d'accord ni pas d'accord

4

D'accord

5

Tout à fait d'accord

8. Je trouve le service vraiment très lourd à utiliser.

1

Pas du tout d'accord

2

Pas d'accord

3

Ni d'accord ni pas d'accord

4

D'accord

5

Tout à fait d'accord

9. Je me suis senti très confiant en utilisant ce service.

1

Pas du tout d'accord

2

Pas d'accord

3

Ni d'accord ni pas d'accord

4

D'accord

5

Tout à fait d'accord

10. J'ai dû apprendre beaucoup de choses avant de pouvoir utiliser ce service.

1

Pas du tout d'accord

2

Pas d'accord

3

Ni d'accord ni pas d'accord

4

D'accord

5

Tout à fait d'accord

→ **Remarques ou suggestions (optionnel)**

Qu'avez-vous apprécié le plus ?

.....

Qu'avez-vous apprécié le moins ?

.....

Si vous avez des recommandations, si vous trouvez qu'il manque des fonctionnalités ou si vous voulez apporter des informations supplémentaires, n'hésitez pas.

.....

Merci pour votre participation !

L. Résultats des tests d'usabilité obtenus et calculés

Résultats Obtenus										
HoloLens 2	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Testeur 1	2	3	2	2	4	2	3	3	4	2
Testeur 2	4	2	3	2	4	2	5	2	4	5
Testeur 3	4	2	4	2	4	2	4	2	4	2

Résultats calculés											
HoloLens 2	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
Testeur 1	1	2	1	3	3	3	2	2	3	3	57,5
Testeur 2	3	3	2	3	3	3	4	3	3	0	67,5
Testeur 3	3	3	3	3	3	3	3	3	3	3	75
										Moyenne :	66,6666667

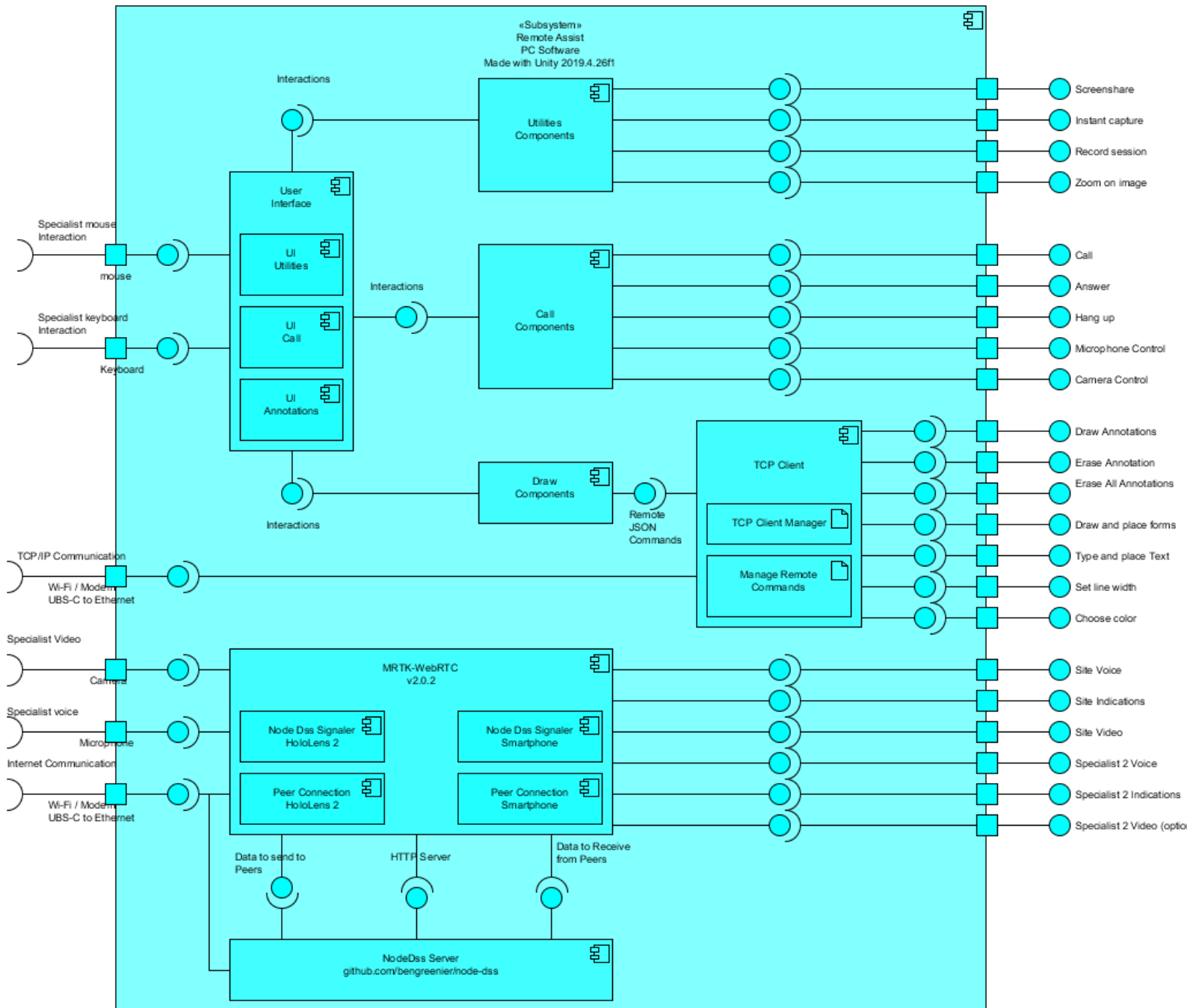
Résultats Obtenus										
PC	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Testeur 1	3	1	3	4	4	1	3	2	3	1
Testeur 2	3	2	4	2	3	2	4	2	4	2
Testeur 3	4	2	4	4	4	2	4	2	4	4
Observateur	3	2	4	2	4	2	2	3	3	2

Résultats calculés											
PC	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
Testeur 1	2	4	2	1	3	4	2	3	2	4	67,5
Testeur 2	2	3	3	3	2	3	3	3	3	3	70
Testeur 3	3	3	3	1	3	3	3	3	3	1	65
Observateur	2	3	3	3	3	3	1	2	2	3	62,5
										Moyenne :	67,5

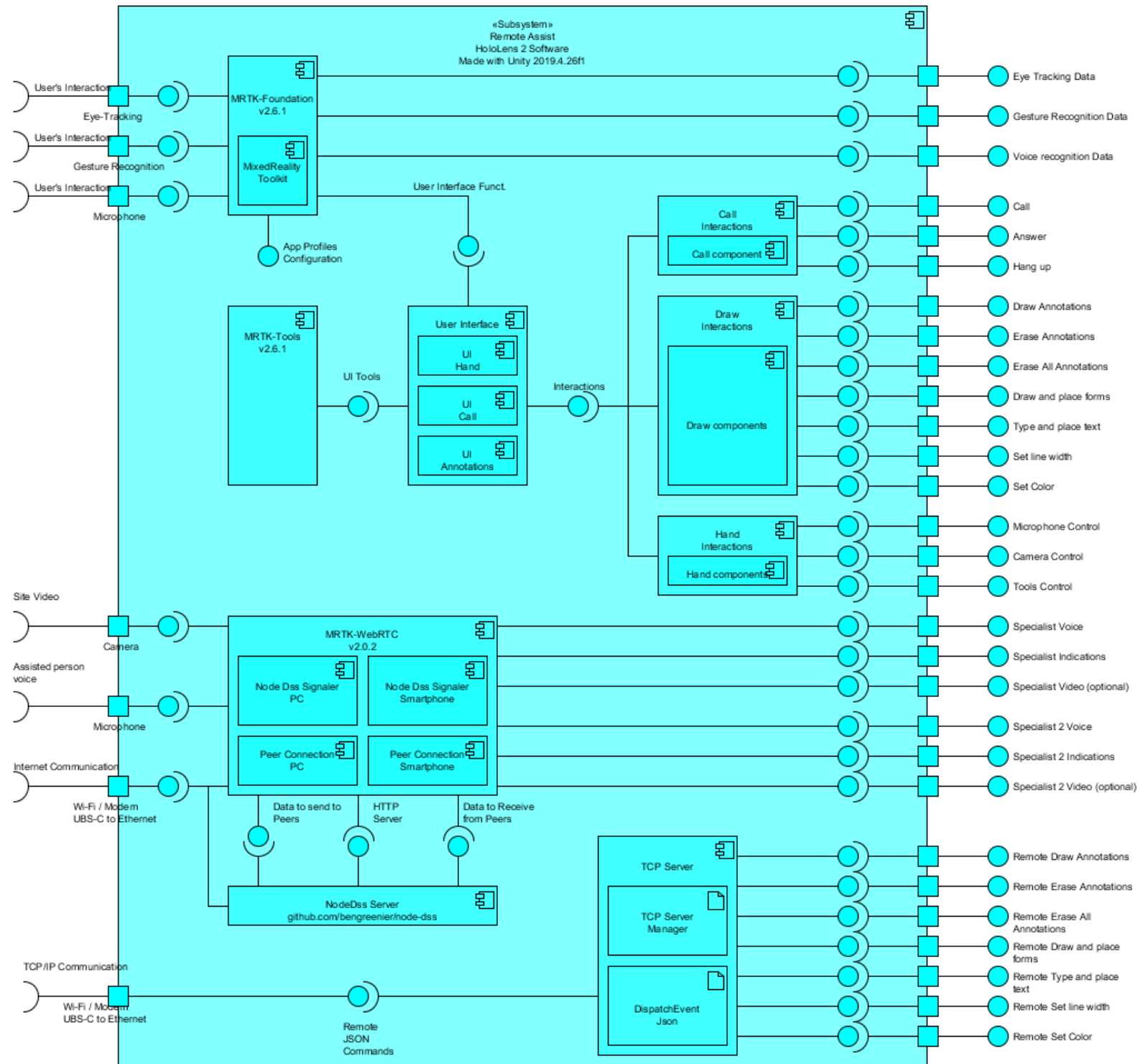
Résultats Obtenus										
Smartphone	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
Testeur 1	4	2	3	1	3	1	5	1	4	2
Testeur 2	2	2	2	4	3	4	2	3	2	3

Résultats calculés											
Smartphone	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
Testeur 1	3	3	2	4	2	4	4	4	3	3	80
Testeur 2	1	3	1	1	2	1	1	2	1	2	37,5
										Moyenne :	50

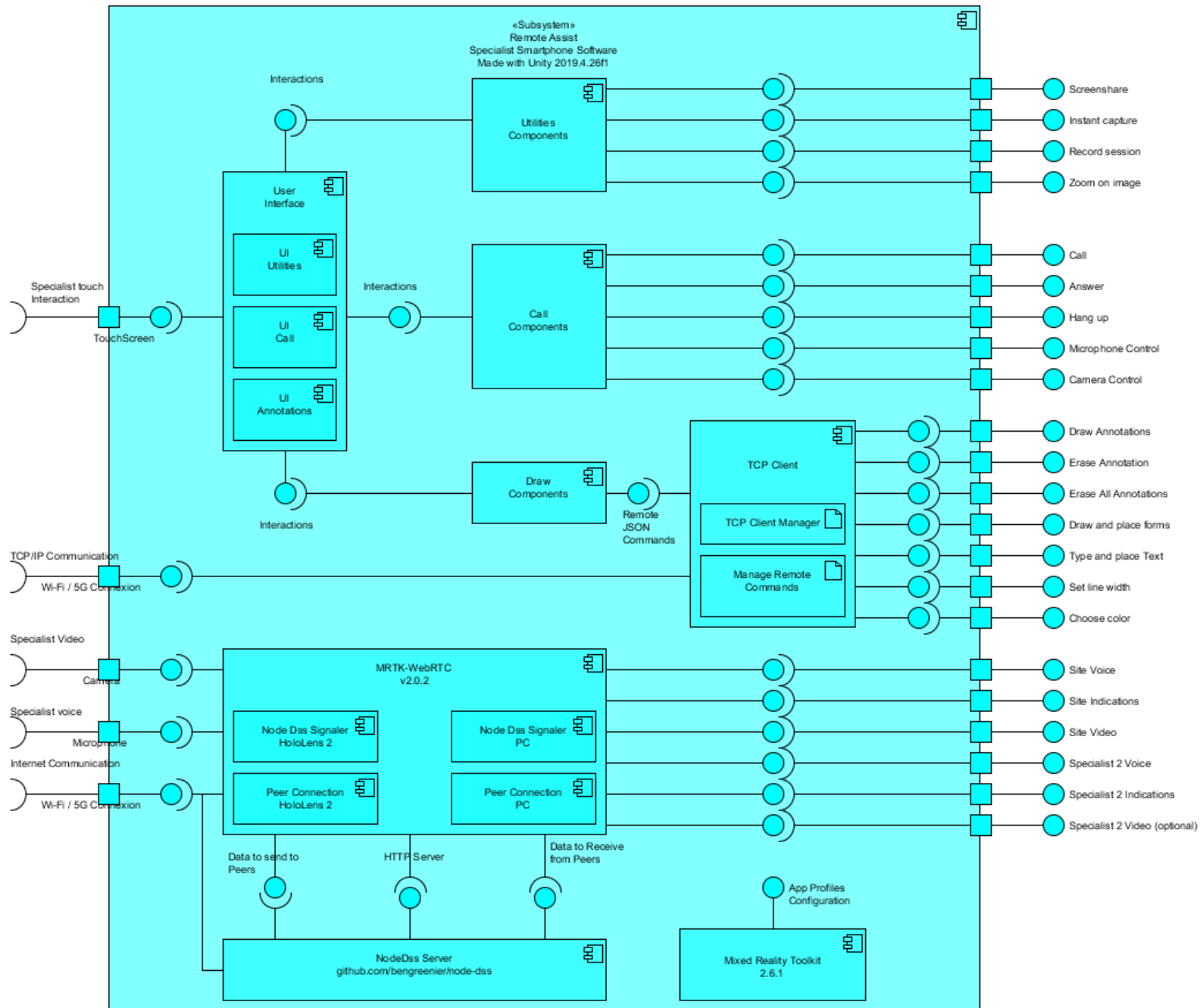
M. Architecture du software PC Spécialiste



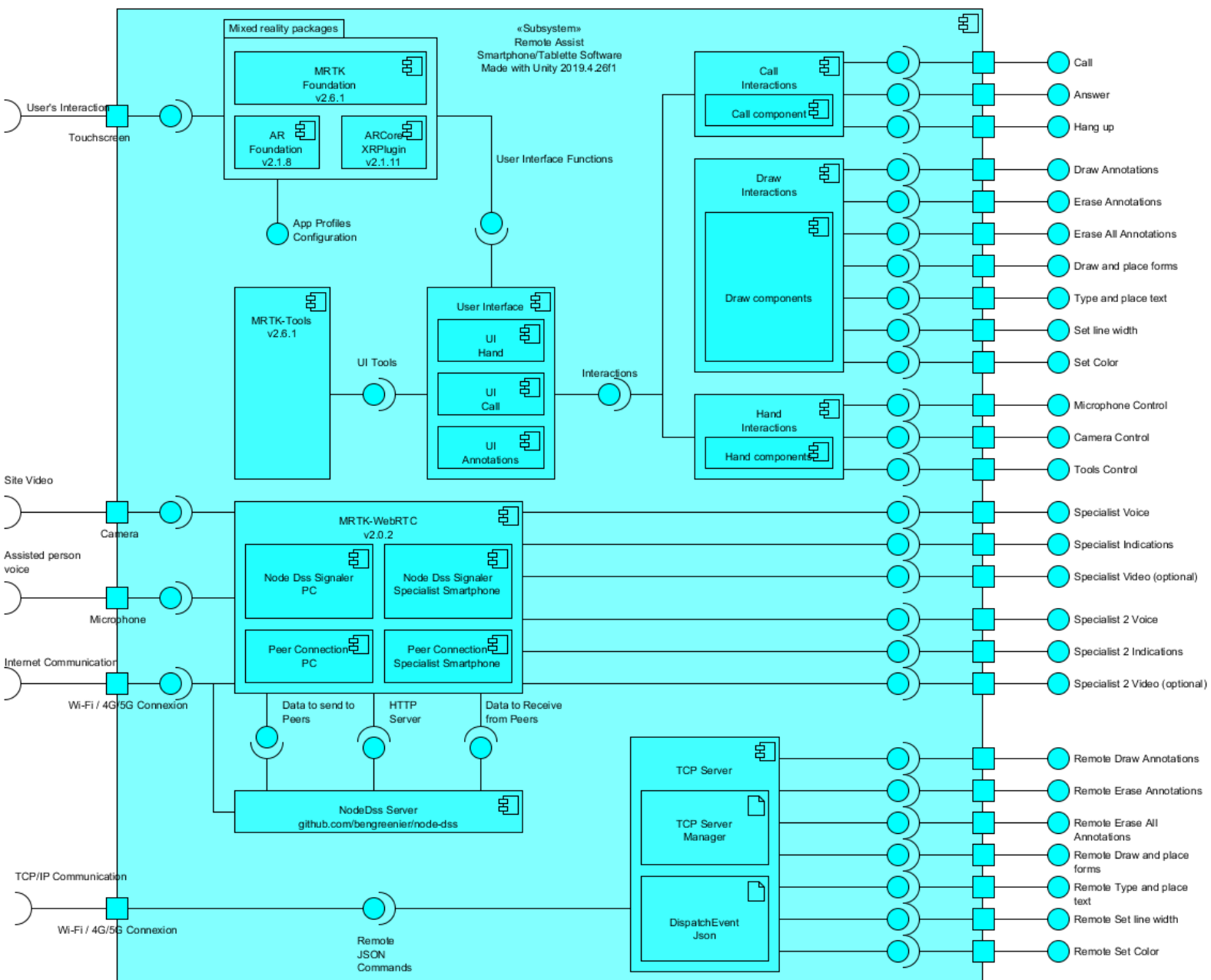
N. Architecture du software HoloLens 2 de la personne assistée



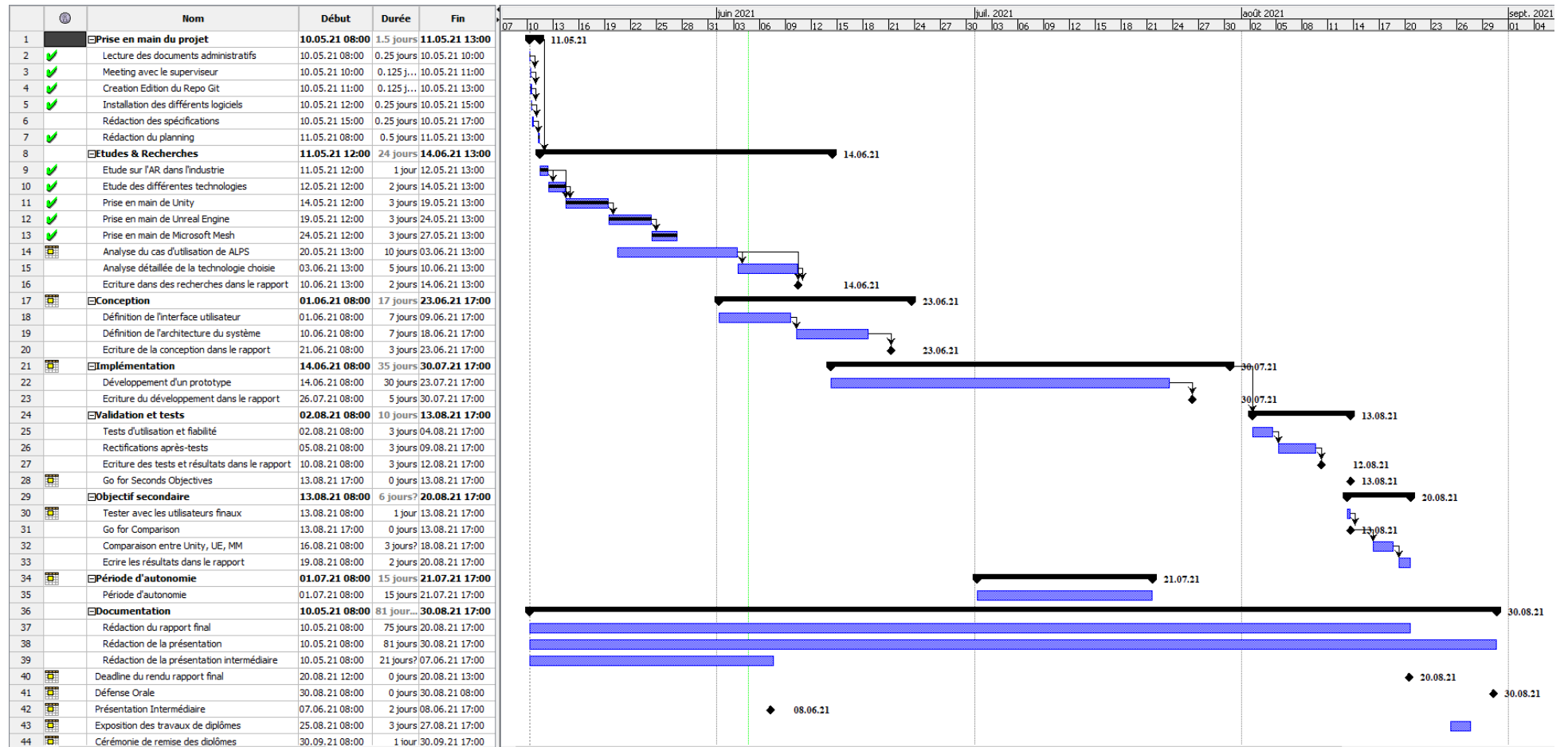
O. Layer 4 : Remote Assist Specialist Smartphone Software



P. Layer 4 : Remote Assist specialist smartphone or tablette software



Q. Planning initial



R. Planning Réel

