

# **CONCEVOIR UNE APPLICATION DE RÉALITÉ MIXTE AFIN DE METTRE EN VALEUR DES PROJETS PATRIMONIAUX**

AVEC LE CASQUE MICROSOFT HOLOLENS

**Mémoire de Master Design Global**  
Spécialisation Architecture, Modélisation et Environnement

**Gabriel Cramoisy**  
Sous la direction de Damien Hanser et Gilles Halin

Soutenu le 13 Septembre 2018







# SOMMAIRE

<b>REMERCIEMENTS</b>	<b>9</b>
<b>AVANT-PROPOS</b>	<b>11</b>
<b>INTRODUCTION : ORIGINE DU MÉMOIRE</b>	<b>13</b>
Mise en situation	15
Besoins	
Mise en valeur des projets	16
Réponse aux besoins	17
Application	
Utilisation finale	17
<b>NIVEAU 1 : ENTRONS DANS UNE NOUVELLE RÉALITÉ</b>	<b>21</b>
Introduction	23
Une Autre Réalité	
La Réalité Virtuelle (VR) : 100% Virtuel - 0% Réel	25
La Réalité Augmentée (AR) : 30% Virtuel - 70 % Réel	25
La Réalité Mixte (MR) : 50% Virtuel - 50% Réel	26
L'origine du casque mixte	26
Le Matériel	
L'ensemble des composants	28
Les formats supportés par le casque	29
Le Fonctionnement	
Le fonctionnement général	30
Les points faibles	30
Les points forts	32
L'état de l'art de l'HoloLens	33
Possibilités	35
Actualités	37
L'avenir de la technologie	38

## **NIVEAU 2 : UTILISATION DU CASQUE** **41**

---

Introduction	43
HoloLens	
L'utilisation dans les détails	45
Les Espaces	47
Les gestes reconnus	48
La reconnaissance vocale	48
Travail en Groupe	49
Le spatial mapping	49
Un ressenti personnel	50
Quelques expérimentations sur le fonctionnement	50
Unity	
Présentation du logiciel	52
Présentation de l'interface	53
Programmation	
Les langages	53
L'apprentissage du langage	53
API	54
Le code	54
Le fonctionnement d'un script	55

## **NIVEAU 3 : APPLICATION DU CRAI** **57**

---

Introduction	59
L'explication de Unity	
La préparation de l'application	61
La vérification de l'application sur l'HoloLens	61
Les caméras	61
La création de l'hologramme	62
Lumière	62
Le Canvas	62
Les gestes	63
Le viseur/Gaze	63
Le Spatial Mapping	63
L'ancrage	64
HoloToolKit	65
Vuforia	
L'installation de l'application	65
La mise à jour de la bibliothèque	66
Fonctionnalités	
La connaissance des besoins	68
Quelques difficultés	
L'optimisation de l'application	68
Une superposition	69
La mise en valeur d'un projet patrimonial	69

<b>NIVEAU 4 : UTILISATION DE L'APPLICATION</b>	<b>71</b>
Introduction	73
Les fiches des options	
Le Menu	74
Importer	76
Supprimer	76
Déplacer	78
Environnement	80
Mode Visite	82
Mode Immersif	84
Superposition	86
Couches	88
Coupes	90
<b>NIVEAU 5 : ENTRONS DANS LE CODE</b>	<b>93</b>
Introduction	95
La programmation de l'application	
Les Gestes	97
Le Menu	98
La coupe	99
L'importation	99
<b>LA CONCLUSION</b>	<b>101</b>
<b>LA BIBLIOGRAPHIE</b>	<b>106</b>
<b>LES ANNEXES</b>	<b>109</b>





## REMERCIEMENTS

Je tiens, tout d'abord, à remercier Gilles Halin, Directeur du CRAI et les membres du laboratoire CRAI, qui m'ont accueilli et ont bien voulu tester l'application. Leurs conseils m'ont permis de la faire évoluer.

Je tiens également à remercier Pascal Humbert, Chercheur au CRAI, d'avoir partagé ses connaissances avec moi. Il m'a permis de surmonter les difficultés rencontrées dans le programme.

Je souhaite remercier, Adrien Coppens, intervenant à la conférence ENCCA'18. Son expérience sur l'HoloLens m'a été d'une grande aide.

Je tiens également à remercier Damien Hanser, Enseignant à l'école d'Architecture de Nancy et Chercheur au CRAI, pour son encadrement.

Enfin, je souhaite remercier mes amis, mes proches, et ma famille, pour leurs conseils et leur soutien.



## AVANT-PROPOS

Avant de commencer à proprement parler du mémoire, j'aimerais déterminer quelques points de vocabulaire.

**Kit Developers** : Il s'agit d'une version utilisable et fonctionnelle d'un projet, sans être la version finale de celui-ci. Cette phase est utile pour les développeurs et permet de perfectionner les logiciels ou les applications du futur produit.

**Chunk** : C'est une portion d'un élément plus grand. Diviser un objet en Chunk permet des calculs plus rapides. Ils peuvent être chargés indépendamment.

**Frame** : Une *Frame* est une image, un instant permettant de calculer l'ensemble des données d'un programme. Lorsque nous parlons de 60 FPS : (*frames per second*), cela signifie qu'il y a 60 images qui sont calculées et qui s'affichent en une seconde.

**Normale** : La normale d'un plan est le vecteur perpendiculaire à ce dernier.

Pour se connecter au casque :  
Identifiant : HoloLensCRAI  
Mot de passe : CRAI\*microsoft

L'application se trouve sur le serveur dans le dossier à mon nom.



# **INTRODUCTION**

**ORIGINE DU  
MÉMOIRE**



## MISE EN SITUATION

---

Actuellement, je pratique un stage de recherche dans le laboratoire dépendant de l'école d'architecture: le CRAI (Centre de Recherche en Architecture et Ingénierie). Ce laboratoire fait partie d'un regroupement appelé le MAP (Modèles et simulations pour l'architecture et le patrimoine). Les bureaux de Nancy sont l'un des quatre laboratoires de recherche en France. *“Le MAP-CRAI conduit des travaux de recherche dans les domaines des modèles, des méthodes et des outils numériques relatifs à la conception architecturale, et l'étude du bâti patrimonial. Ces recherches ont pour caractéristique commune de s'appuyer sur des expérimentations concrètes et des applications au domaine du bâtiment et de l'architecture. Elles débouchent majoritairement sur la production de logiciels et le développement d'outils informatiques de simulation et d'assistance à la conception. Le laboratoire accueille des doctorants en « sciences de l'architecture » de l'école doctorale IAEM de l'UDL et des stagiaires de Masters recherche et professionnel.”*<sup>[1]</sup>

Plusieurs thématiques sont abordées dans le CRAI :

- La recherche patrimoniale
- Activités collectives et maquette numérique
- Modélisation et fabrication numérique multi-échelle
- Approches pédagogiques et professionnelles
- La visualisation et interaction dans les espaces virtuels

Le laboratoire de recherche appartient au CNRS (Centre National de Recherche Scientifique) et est attaché à l'École Nationale d'Architecture de Nancy. Un cursus est proposé reliant ces deux services : le master AME (Architecture, Modélisation et Environnement)<sup>[2]</sup>. Il s'agit de développer les connaissances des étudiants sur les nouvelles technologies en lien avec l'architecture. Personnellement étudiant dans ce cursus, il m'a été donné la possibilité de travailler avec des technologies comme les imprimantes 3D, les scanners laser, ou encore l'utilisation de logiciels comme Unity ou Blender. Le stage au CRAI augmente cette liste technologique avec le casque de réalité mixte.

L'objectif de ce mémoire est de répondre à cette dernière thématique. Il est intéressant de noter qu'il s'intègre au contexte global du laboratoire. D'autre part, nous nous interrogerons sur la mise en valeur des projets patrimoniaux réalisés au sein de celui-ci. Bien que ce sujet soit moins exposé dans ce mémoire, le casque permet un travail en collectivité et en collaboration à distance. Enfin, ce mémoire vise à faire comprendre ce qu'est la technologie proposée par Microsoft et montrer son fonctionnement. Il est possible d'apporter une autre manière de travailler sur le casque, une manière qui peut interroger une autre thématique.

En parallèle, je me suis intéressé, pour mon projet de fin d'étude, à la réhabilitation et l'extension du château d'eau de la gare de Metz. Les projets patrimoniaux sont des moyens de mettre en valeur un bâtiment existant, tout en lui apportant une contemporanéité. Il est intéressant de pouvoir valoriser ces projets avec le casque HoloLens. Il est possible de mettre l'avant projet et après projet ce qui est important dans un projet de patrimoine, afin de voir l'impact du projet sur son environnement. Nous pouvons ajouter des éléments si nous en avons besoin. Il est pratique que ce soit mon projet, afin de voir ce qu'il manquait lors de la conception. En intégrant le CRAI, j'ai eu la possibilité de faire un stage sur une nouvelle technologie d'immersion : le casque de réalité mixte est une technologie récente et unique. C'est une opportunité d'avoir un accès à ce type de technologie, qui n'est pas encore accessible au public.

---

1 Citation du site internet du Laboratoire de Recherche : CRAI [en ligne]. Disponible sur <<http://www.map.cnrs.fr/>> [consulté le 02/03/2018]

2 Le site internet du master AME [en ligne]. Disponible sur <<http://www.crai.archi.fr/wp-ame/>> [consulté le 15/09/2017]

Le casque HoloLens a été acheté récemment par le CRAI. Ils avaient besoin d'une personne qui puisse appréhender cette technologie. J'ai pu la comprendre et développer un projet avec celle-ci. C'est un produit peu connu du grand public et souvent confondu avec les casques de réalité virtuelle. En début de stage, j'ai testé les limites et appris à m'en servir. Comme c'est un casque de réalité mixte, l'intérêt est de montrer des projets virtuels fabriqués par le laboratoire. Au cours des années, plusieurs projets ont été réalisés, qu'il s'agisse de projets d'étudiants en AME ou des chercheurs du laboratoire.

## BESOINS

---

### Mise en valeur des projets

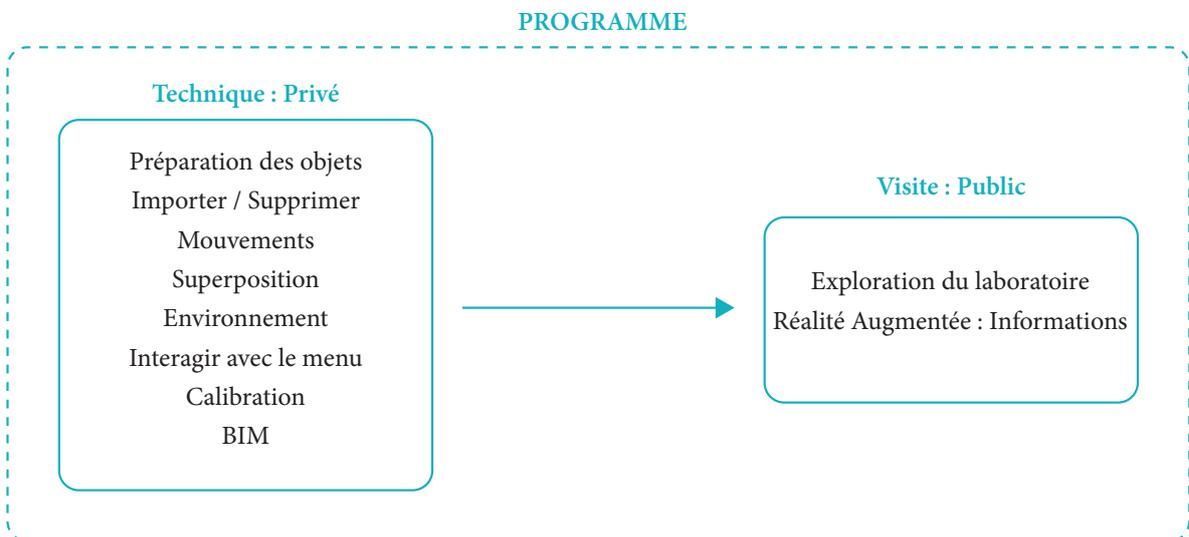


Fig.1 Schéma des besoins

Suivant les thématiques du laboratoire, l'idée est d'avoir un lieu de regroupement des connaissances assimilées durant le stage. Nous avons l'espace du laboratoire qui permet la mise en valeur des travaux réalisés dans le bureau de recherche.

Nous voulons que l'ensemble des objets soient visibles lorsque nous sommes dans le laboratoire et qu'ils soient posés sur les tables ou les murs. Une contrainte est d'avoir les modélisations qui ne bougent pas de place lorsque nous quittons l'application. Ensuite, elles doivent être déplaçables pour permettre des interactions, tout en prenant compte l'environnement, ce qui est une spécificité du casque de réalité mixte.

Pour mettre en évidence les projets et expliquer ce qu'il en est, il serait intéressant d'afficher des informations sur les objets présents dans l'application, afin d'informer le type d'objet que nous avons et pourquoi il a été réalisé. Le fait de montrer quelque chose que nous ne voyons pas sans le casque ouvre une vision sur une autre réalité. De plus, il faut prendre en compte le visiteur, qui n'a pas d'expérience dans cette technologie. Une majorité des nouveaux utilisateurs n'arrivent pas à faire les gestes d'interaction correctement. Il est aussi difficile d'expliquer comment le casque fonctionne sans voir ce que la personne fait. Il faut par conséquent simplifier le parcours pour les futurs visiteurs du CRAI et limiter leurs interactions.

L'espace du laboratoire est constitué de murs, de tables et d'étagères. Il permet une variété de types d'objets à installer. Ces éléments peuvent être utilisés pour poser les projets virtuels du laboratoire dans l'espace. Comme

dit précédemment, le laboratoire produit une multitude d'objets numériques sur plusieurs thématiques. Afin de mettre en valeur ces éléments de types différents, simples ou plus complexe, des assemblages ou encore des objets animés, l'application doit être capable de les regrouper dans un même espace. Les objets 3D conçus au laboratoire sont : le plan relief de Verdun, des statues, des projets d'étudiants, des vidéos et des sépulcres.

## Réponse aux besoins

Lors d'échanges avec Gilles HALIN, nous avons eu trois idées dans le développement de l'application.

La première permettrait d'utiliser les technologies présentes dans le laboratoire : la table tactile. Un projet a été conçu sur cette table en rapport avec le BIM. L'HoloLens permet de poser des objets sur la table et de se déplacer dans ce modèle 3D. Ce modèle serait, donc, projeté à la fois sur la table en plan, sur un écran en perspective et en volume grâce au casque. Comme il y a un rapport avec le BIM, il serait intéressant d'ajouter des informations sur le modèle ou des liens vers des pages internet, afin d'avoir une interaction mixte.

La seconde idée serait de mettre en valeur un projet patrimonial : le plan relief de Verdun. Il existe en maquette dans un musée et il a été reconstitué en virtuel au laboratoire. En ayant le modèle 3D, il est possible de montrer ce projet de différentes manières :

- superposer avec le plan ancien en deux dimensions
- mettre des informations sur chaque îlot
- s'immerger et se déplacer à l'échelle du projet dans la ville

Le travail étant déjà fait en amont, il suffit de mettre les liens dans l'HoloLens. En relation avec cette idée, il est possible de mettre en place l'exposition du laboratoire avec d'autres modèles. Il serait envisageable d'aménager les bureaux en mettant de véritables étagères, afin d'y poser des éléments virtuels.

La dernière idée serait d'importer mon Projet de Fin d'Études (PFE) qui a un niveau de détail différent de ce qui est fait dans le laboratoire. Il reprendrait les caractéristiques du modèle BIM dont nous avons parlé dans la première idée. Le fait de travailler avec un fichier que j'ai créé, a l'avantage de pouvoir être changé plus facilement et mettre en place une hiérarchie souhaitée dans le modèle. Je sais ce que je veux valoriser dans ce projet d'architecture patrimonial. Il est plus constructif d'adapter le modèle pour qu'il soit exploitable avec les autres modèles 3D. Toutefois, la modélisation reste différente des autres. Il faudrait ouvrir l'application pour un grand échantillon de modèles, sans que nous ayons besoin de les modifier.

## APPLICATION

---

### Utilisation finale

En réponse à ces idées, l'application doit permettre une interaction avec les modèles 3D du laboratoire. Afin de mettre en valeur ces modèles, l'application doit créer l'ambiance d'une exposition qui montre les travaux réalisés. Créer une seule application permet de regrouper toutes les connaissances accumulées durant le stage. Il sera ensuite possible de trouver plus facilement une information technique. Cette application donne aussi les notions primaires de l'ensemble des possibilités de la programmation pour un casque HoloLens. Pour une meilleure utilisation future, la plupart des fonctionnalités ont été retranscrites dans celle-ci. Cette application réunie une grande variété de fonctions basiques. Ces options pourront être utilisées pour une autre thématique du laboratoire.

Toutefois, l'application doit être simple et doit permettre de se rendre compte des utilisations possibles de la technologie pour l'architecture. Même si la technologie change dans les versions futures, car elle n'est

qu'un prototype, les bases seront posées. Un nouveau modèle de casque HoloLens est déjà prévu pour 2019. L'application permet d'interagir avec un projet architectural. Ces premières fonctionnalités prennent en compte les besoins d'un projet d'architecture, mais elles s'adaptent au public qui l'utilise. Il faut que l'application soit flexible sur les fichiers que nous pouvons importer, afin d'accueillir un grand nombre d'objets.

Dans la mise en valeur du patrimoine, il est intéressant de comparer l'existant avec les nouvelles constructions. Ainsi, il est possible de voir cette différence lorsque nous superposons les deux phases du projet. Le casque peut fonctionner dans l'espace existant. Il est alors possible de faire apparaître à l'échelle 1:1 ce qui est modifié. De plus, les enjeux et l'analyse historique révélés avec ces projets peuvent être mis en valeur durant une exposition publique.

Pour des questions de temps, il n'a pas été possible de concevoir une application en connexion avec le projet BIM sur la table tactile. Néanmoins, l'application ouvre la possibilité de superposition d'un modèle 3D sur cette table. Même s'il n'y a pas d'interactions entre les différentes technologies, il est possible de manipuler le modèle superposé.

Dans une application future, il pourrait être facile de concevoir des programmes en lien direct avec les logiciels, comme le fait le laboratoire du Mons<sup>[3]</sup> avec son importation Grasshopper ou avec l'utilisation de Blender<sup>[4]</sup>.

---

3 COPPENS Adrien,PARAMETRIC MODELLING IN IMMERSIVE ENVIRONMENTS, conférence du 12 juillet 2018, ENCCA

4 Chaîne Youtube u9vata : Blender meets Hololens 2 - upgrades to HoloRenderer [en ligne]. Disponible sur < <https://www.youtube.com/watch?v=b5wflLPtigY> > [consulté 17/08/2018]





# **NIVEAU 01**

**ENTRONS DANS UNE  
NOUVELLE RÉALITÉ**



## INTRODUCTION

---

Dans ce chapitre, nous allons voir de quelle technologie nous parlons et dans quel domaine elle se place.

La Réalité Virtuelle, la Réalité Augmentée et la Réalité Mixte sont regroupées sous le nom de *Cross Reality* (XR). Ces technologies ont fait leur apparition au même moment que les ordinateurs. Pourtant, elles deviennent connues par le grand public dans les années 2000.



## UNE AUTRE RÉALITÉ

---



Fig.2 Qu'est ce que la VR AR MR, <sup>[5]</sup>

---

### La Réalité Virtuelle<sup>[6]</sup> (VR) : 100% Virtuel - 0% Réel

Cette technologie<sup>[7]</sup> est la première à voir le jour. Elle débute en 1950, mais est exploitée seulement dans les années 90. Morton Heilig est le précurseur de la Réalité Virtuelle en créant le premier cinéma immersif, en 1962, appelé le Sensorama. Ce cinéma utilisable par une seule personne, il diffuse cinq courts métrages immersifs. Les sièges vibrent et des ventilateurs simulent l'environnement des scènes du film présentées. Cette technologie est imposante.

C'est en 1968 que le premier casque "Epée de Damoclès" voit le jour grâce à Ivan Sutherland. Il s'agit d'un casque lourd et peu confortable avec des mouvements limités, mais il a permis d'inspirer de nombreux créateurs. Par la suite, le Dataglove est utilisé par l'Armée avec la NASA. Durant les années 90, la technologie est majoritairement utilisée dans les jeux vidéo et reste aujourd'hui le secteur majeur de la Réalité Virtuelle. L'année 1995 est une année importante. Les iGlasses de Virtual I-O, le Cybermaxx par Victormaxx, le Virtual Boy par Nintendo et le VFX-1 sont développés par Forte Technologie.

Cette technologie plonge l'utilisateur dans un monde complètement virtuel en trois dimensions. Elle permet aussi de s'immerger dans les vidéos ou photographies 360°. Aujourd'hui, les casques VR les plus connus sont : le HTC Vive de HTC 2016, l'Occulus Rift de Facebook 2016, le PS VR de Sony 2016 et CAVE de Visbox.

### La Réalité Augmentée (AR) : 30% Virtuel - 70 % Réel

Cette technologie est une augmentation de l'environnement réel qui nous entoure. C'est un ajout d'informations soit en 2D, soit en 3D et peuvent interagir ou s'ajouter sur l'environnement. C'est une seconde couche virtuelle superposée sur le support (lunettes, ordinateurs ou smartphone). Avec la Réalité Augmentée, l'utilisateur ne peut pas interagir directement avec les objets, à moins d'avoir une interaction sur l'écran ou un élément extérieur (exemple : un *controller*). C'est cette différence qui ne fait pas entrer le casque HoloLens dans la classe de Réalité Augmentée. Aujourd'hui, les technologies les plus connues de l'AR sont les Google Glass et les Smartphones.

---

5 Images de BHUSHAN JAGWANI Valendu, informaticien [en ligne]. Disponible sur <<https://www.quora.com/What-is-the-difference-between-virtual-reality-augmented-reality-and-mixed-reality>> [consulté le 06/07/2018]

6 KRESS Bernard C. and CUMMINGS William J. , Towards the Ultimate Mixed Reality Experience : HoloLens Display Architecture Choices. Microsoft Corporation, Redmond, WA, USA, publié le 02 Juin 2017.

7 TURQUIER Amandine, Article du Magazine de la Réalité Virtuelle [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/lhistoire-realite-virtuelle>> [consulté le 22/02/2018]

## La Réalité Mixte<sup>[8]</sup> (MR) : 50% Virtuel - 50% Réel

Cette technologie est plus récente. Nous retrouvons des éléments virtuels dans un environnement réel. Les objets virtuels peuvent interagir avec le monde qui nous entoure. A la différence de la réalité augmentée, les objets sont intégrés sous forme d'hologrammes dans le monde réel. Il est donc possible d'interagir avec les éléments, de tourner autour ou de les modifier. Contrairement à la Réalité Virtuelle, la Réalité Mixte est plutôt tournée vers les professionnels et moins dans les jeux vidéos.

Les avis<sup>[9]</sup> sont partagés sur le terme à employer et nous allons expliquer pourquoi.

Microsoft emploie le terme de Réalité Mixte alors que Milgram<sup>[10]</sup> parle de cette technologie comme de la Réalité Augmentée. Nous reprendrons le terme du constructeur et nous parlerons, tout au long de ce mémoire, de Réalité Mixte. Nous pouvons, toutefois noter, que la définition ne correspond pas totalement au casque HoloLens.

La définition de Milgram n'est pas totalement fautive, mais compte tenu de la date de création, celle-ci mériterait d'être mise à jour. En effet, les casques de Réalité Mixtes n'existaient pas.

Aujourd'hui nous trouvons :

- le Meta 2 : alternative directe de l'HoloLens. Ces lunettes sont vendues sous forme de Kit développeurs.
- l'ASUS HC102 avec une possibilité de basculement entre le monde virtuel et le monde réel
- les Windows Mixed Reality
- son principal concurrent Magic Leap<sup>[11]</sup>

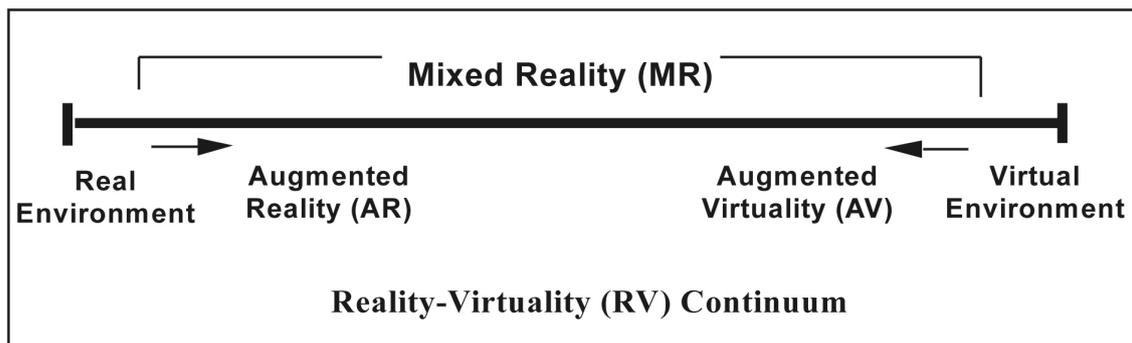


Fig.3 Tableau de P. MILGRAM sur le continuum de la réalité virtuelle. <sup>[7]</sup>

## L'ORIGINE DU CASQUE MIXTE

L'HoloLens a été développé par Microsoft<sup>[12]</sup> et commercialisé en mars 2016. Nous allons commencer par rappeler comment cette société est arrivée à fabriquer le premier casque de Réalité Mixte commercialisable.

Microsoft a été fondé par deux étudiants du MIT, Bill Gates et Paul Allen, en 1975. Ils se lancent dans les

8 LABBE Pierrick, Article du Magazine de la Réalité Virtuelle [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/realite-mixte-definition-exemples-0106>> [consulté le 20/02/2018]

9 CHALON René, Thèse sur la "Réalité mixte et travail collaboratif : IRVO, un modèle de l'interaction homme-machine" [en ligne]. Disponible sur <<https://tel.archives-ouvertes.fr/tel-00152230/document>> [consulté le 19/07/2018]

10 MILGRAM Paul, Augmented Reality : A class of displays on the reality-virtuality continuum, Japon, 1995

11 LABBE Pierrick, Article du Magazine de la Réalité Virtuelle [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/hologrammes-magic-leap-realite>> [consulté le 22/02/2018]

12 Page de présentation de l'HoloLens par Microsoft [en ligne]. Disponible sur <<https://www.microsoft.com/fr-fr/hololens>> [consulté le 20/02/2018]

premiers systèmes d'exploitation, les débuts de Windows. Microsoft n'est pas seulement la société qui a fabriqué le système d'exploitation. En effet, en passant par les Smartphones, Microsoft a depuis longtemps un pied dans le monde virtuel avec sa console de salon : la Xbox. L'entreprise a agrandi son domaine d'action en aboutissant dans les casques de réalité virtuelle. Microsoft finit par innover avec le Microsoft HoloLens. Ce qui nous amène, aujourd'hui en 2018, à tester la technologie de Réalité Mixte.



Fig.4 Photographie de Alex KIPMAN avec le casque HoloLens. Source : <techtrends.tech>

Le casque de réalité mixte est le premier dans son genre commercialisé. Il a été conçu par Microsoft et plus précisément par Alex Kipman<sup>[13]</sup>. Il est spécialiste dans la perception d'intelligences artificielles et la MR. Sorti du R.I.T en 2001, il rejoint Microsoft dans la foulée. Il est l'inventeur du Kinect, du *motion controller* et de l'HoloLens. Ce dernier est sélectionné pour le prix de l'inventeur 2018. Alors que le casque de Réalité Mixte voit le jour, tous les autres constructeurs se concentrent sur l'influence importante de la Réalité Virtuelle. Ce casque entre VR et AR, permet, que ce soit bien ou non, un pas en avant dans la technologie.

Microsoft pense que l'hologramme est l'avenir et la prochaine évolution informatique. Un des avantages de ce casque, en comparaison avec d'autres, est qu'il a une utilisation dans de nombreux domaines. De plus, il est autonome, maniable et contient son propre ordinateur.

UWP (universal windows Platform) est la plateforme universelle pour Windows. L'HoloLens fait partie de ce continuum. Elle permet de regrouper l'ensemble de ces informations depuis les différentes applications. Le livre<sup>[14]</sup> d'Ayan Chatterjee permet d'en apprendre plus sur son fonctionnement.

---

13 KIPMAN Alex, chaîne youtube Epofilms [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=Unj0kHT2XWk>> [consulté le 20/02/2018]

14 CHATTERJEE Ayan, Building Apps for the Universal Windows Platform, Apress, Royaume-Unis, 2017

## LE MATÉRIEL

---

### L'ensemble des composants

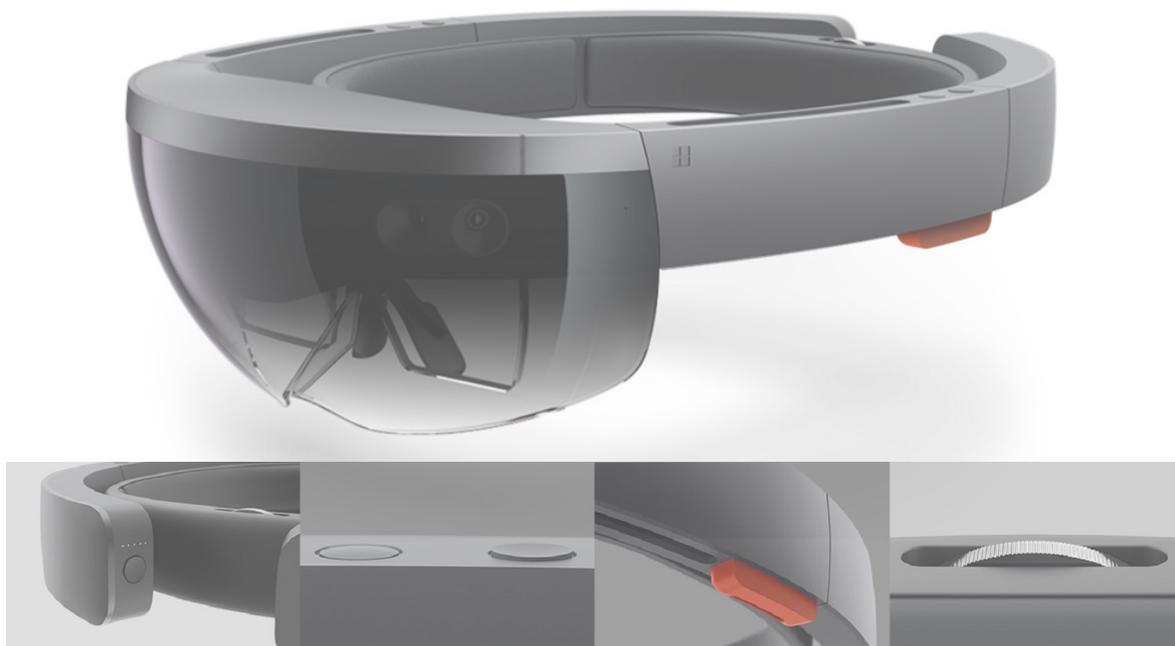


Fig.5 Photographie de l'HoloLens, Détails des éléments du casque. Source : variety.com et microsoft.com

Lorsque nous décortiquons le casque, nous obtenons quelques informations sur sa composition. Le matériel étant protégé, nous n'avons pas l'ensemble des informations le concernant. Il faut savoir par équivalence que l'HoloLens est une grosse tablette actuelle, un ordinateur autonome qui fonctionne sous Windows 10 avec une architecture x86. Son fonctionnement sous Windows reprend les mêmes icônes et les mêmes navigations grâce au UWP. Il contient des caméras, des détecteurs, des gyroscopes ou encore des hauts parleurs.

D'un poids extrêmement léger de 579g, le casque est agréable à utiliser et à porter. Afin de faire l'ensemble des calculs, nous avons le CPU (*Central Processing Unit*) Puce Intel Atom Cherry Trail de 1GHz. Le casque a besoin d'une capacité de calcul graphique. C'est pour cela que nous retrouvons un GPU (*Graphics Processing Unit*). Afin d'améliorer l'expérience, nous trouvons également un HPU 1.0 (*Holographic Processing Unit*). Il est conçu spécialement pour le casque. Pour stocker l'ensemble des applications et des hologrammes, nous avons une mémoire vive de 2 Go Vive, 1Go pour le GPU, 1Go pour le HPU et une mémoire Flash de 64 Go.<sup>[15]</sup>

Pour capter tous les mouvements de la tête, nous retrouvons deux capteurs de mouvements. Une Unité de Mesure de l'inertie avec un gyroscope et un magnétomètre sur les parties latérales du casque. Afin de repérer l'environnement, il y a deux doubles caméras<sup>[16]</sup>. Celles-ci mesurent le relief, ce qui permet le placement des objets. Pour afficher ces objets holographiques, le système d'affichage a un affichage dit HD (720p) en 2,5M de points lumineux en 16:9. Il n'y a pas de comparaison de résolution avec les écrans plus classiques. Afin de capter les gestes et la profondeur, l'HoloLens contient 4 caméras à LED infrarouges. La technologie utilisée dans l'HoloLens reprend celle de la caméra Kinect<sup>[17]</sup>. Pour finir nous pouvons trouver au centre de la visière, un capteur de luminosité à coté de la caméra 2MP pour enregistrer ce que l'utilisateur voit. Afin de faire fonctionner

---

15 RUBINO Daniel, Article : Here are the full processor, storage and RAM specs [en ligne]. Disponible sur <<https://www.windowscentral.com/microsoft-hololens-processor-storage-and-ram>> [consulté le 20/02/2018]

16 Page explicative sur la caméra de Microsoft [en ligne]. Disponible sur <<https://docs.microsoft.com/fr-fr/windows/mixed-reality/locatable-camera>> [consulté le 03/02/2018]

17 PECUCHET Bertrand, Rapport sur la kinect, Lille, 2013 [en ligne]. Disponible sur <<http://iut-gmp.univ-lille1.fr/fichiers/LPVI/RapportFinalkinect.pdf>> [consulté le 03/02/2018]

tout cela, le casque à deux batteries qui sont découpées en trois sections. Elles se trouvent à l'arrière de celui-ci dans les parties qui peuvent s'écarter.



Fig.6 Photographie du casque HoloLens démonté. Source : Google Image

### Les formats supportés par le casque

Pour avoir un ordre d'idée de la puissance, nous allons prendre l'exemple de l'application 3D Viewer Beta

- Format FBX (maximum FBX release 2015.1.0)
- Entre 5KB et 500MB
- Polygone seulement, pas de surfaces subdivisées ou NURBs
- Les textures doivent être intégrées dans le fichier FBX
- En images : JPEG ou PNG, BMP (24-bit RGB true-color), TGA images (24-bit RGB et 32-bit RGBQ true-color)

- La taille maximale est de 2048 x 2048
  - Une diffuse map, une normale map, une réflexion par maillage
  - Si l'alpha de la texture diffuse est inférieure à 50%, il y a un rejet des pixels
- Pour plus d'infos, voir la page <sup>[18]</sup>

Dans un cas général, tout dépend de l'utilisation de l'application. Plus il y a d'objets à importer dans l'application, plus les fichiers devront être légers pour que l'application ne lague pas. Si l'environnement réel est pris en compte, cela demande une plus grande puissance de calcul. Il faudra gérer cette demande d'information supplémentaire. Si nous développons une application Unity, les formats de fichiers sont des formats compatibles avec Unity. Il est possible d'intégrer des fichiers \*.blend de Blender directement dans l'application Unity.

Nous n'avons pas de connaissances particulières sur Unity. La taille maximale des fichiers ou des textures dépend de la capacité du logiciel. Il est possible d'importer un fichier directement depuis Onedrive<sup>[19]</sup>. L'application est directement accessible depuis le menu de l'HoloLens. Montrer l'application qui peut le faire .

18 Page utilisation de la 3D de Microsoft [en ligne]. Disponible sur <<https://support.microsoft.com/fr-fr/help/13766/hololens-using-3d-viewer-on-hololens>> [consulté le 03/02/2018]

19 LIANG Yining, Tutoriel pour importer un fichier depuis One Drive [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=ReE-PetXVh0&t=1s>> [consulté le 10/03/2018]

## LE FONCTIONNEMENT

---

### Le fonctionnement général<sup>[20]</sup>

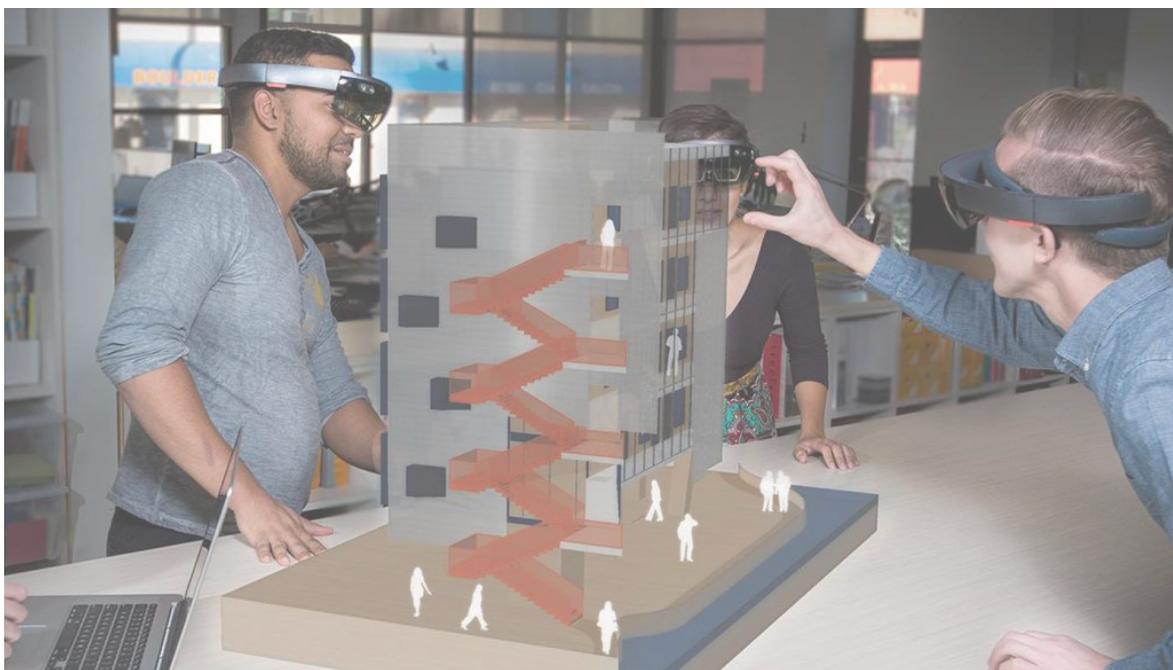


Fig.7 Photographie montrant des architectes autour d'un hologramme. Source : <geospatialmedia.s3.amazonaws.com>

Le casque de Réalité Mixte est composé d'une visière transparente. Les objets virtuels sont projetés sur celle-ci. Avec toute une série de capteurs infrarouges, la technologie détecte nos mouvements, notre emplacement avec l'environnement. Il est possible de circuler entre nos objets 3D, de projeter des écrans et d'interagir directement avec tout ce qui nous entoure. Ce casque est autonome et permet de fonctionner sans aucun ordinateur ou programme extérieur. Il est possible de se déplacer dans le monde réel sans aucun câble branché, ni d'élément à poser sur les murs pour repérer les mouvements. Comme le monde réel reste visible, il n'y a pas risque de se cogner, de trébucher ou de casser quelque chose.

De même pour les objets que nous importons dans l'environnement, ces derniers prennent en compte les éléments réels. Il n'est donc pas possible de traverser une table. Par contre, il est possible de cacher un élément derrière celle-ci. C'est cette spécificité qui fait de ce casque un outil de Réalité Mixte. Il y a une immersion moitié virtuelle et moitié réelle. Lorsque nous allumons l'HoloLens, il scanne son environnement et reconnaît où nous sommes. Nous retrouvons les éléments (dans l'application Holograms) qu'on avait placés précédemment. Cette fonctionnalité est pratique pour ne pas recommencer à importer nos objets virtuels, continuellement lorsque nous redémarrons le casque. Ensuite la technologie de capture gestuelle a été reprise du Kinect de la Xbox.

### Les points faibles

L'utilisation du casque devient désagréable sur des longues durées. Cela peut engendrer des maux de têtes chez les utilisateurs. Ce problème apparaît lorsque nous utilisons des objets réels avec le casque sur les yeux. Il y a un filtre (le casque) qui se met entre nous et l'objet réel. À ce moment là, l'œil fait des allers-retours entre l'objet et les lunettes. Il est conseillé d'avoir des utilisations courtes. Après mes propres expérimentations de l'appareil, il n'y a pas de problèmes à ce niveau. Néanmoins, les utilisations sont brèves lors du développement de l'application. À aucun moment, il n'y a besoin d'une utilisation prolongée.

---

20 LAMELOT Matthieu, Article d'un test HoloLens [en ligne]. Disponible sur <<https://www.tomshardware.fr/articles/test-micro-soft-hololens,2-2602.html>> [consulté le 20/02/2018]



Fig.8 Réalité du champs de vision du casque HoloLens. Source : donwar.fr

L'angle de vision de 40 degrés est trop petit et coupe la projection holographique. Cet angle de vision a été très largement critiqué par l'ensemble des utilisateurs et est le plus gros point négatif du casque. Microsoft prévoit de changer cela pour la prochaine version.

Le curseur est au centre du regard et est dépendant des mouvements de la tête. Sûrement le seul moyen d'avoir un curseur fonctionnel pour le casque seul (pour ne pas avoir un autre appareil dans les mains, comme une manette). Ceci peut-être problématique dans des jeux ou dans les menus. Ce moyen de fonctionnement apporte vite une fatigue lorsque nous souhaitons entrer un mot sur le clavier virtuel. Il faut bouger la tête avec précision sur chaque lettre.

Les gestes pour interagir avec l'environnement ou les hologrammes peuvent également devenir fatigants. L'inconvénient vient du fait que ce ne sont pas des gestes naturels dans un effort prolongé. Les gestes ne sont pas intuitifs, il faut les apprendre pour faire fonctionner l'appareil. De plus, si nous sommes trop proches ou trop loin et si l'angle de la main n'est pas le bon, les gestes ne sont pas reconnus. Pour finir, il manque une sensation dans les mains lorsque nous interagissons avec le virtuel.

Lorsqu'il y a une forte lumière qui traverse la visière du casque, qu'il s'agisse d'une fenêtre ou d'une lampe, nous ne voyons plus les hologrammes. Cette addition d'inconvénients, diminue l'immersion lors des utilisations.

Nous pouvons ajouter que le prix est élevé, donc il y a moins de ventes. Ceci engendre directement moins de développements de particuliers et une démocratisation moindre. Lorsque les professionnels développent leurs applications, elles sont privées et sont plus difficiles d'accès.

Le temps d'autonomie est assez court, mais il peut être suffisant pour certain cas. Après quelques temps d'utilisation, le temps d'autonomie n'est pas un problème grave. Lors du développement il est possible de rester branché au secteur pour les utilisations proches et le débrancher pour les déplacements. Les utilisations de l'appareil ne sont souvent pas longues. Le problème peut être présent si nous jouons à un jeu ou pour une exposition d'une journée, si nous prenons l'exemple d'une journée porte ouverte au CRAI.

La précision de l'HoloLens est un inconvénient. Par exemple, il ne peut pas prendre en compte tous les détails d'une maquette. Nous avons une forme trop aléatoire. Il devient impossible de différencier une maquette d'autre chose. Il faut donc trouver un moyen pour reconnaître celle-ci.

## Les points forts

Pour un premier casque de Réalité Mixte, il y a des concepts qui fonctionnent et qui prouvent que cette technologie a des avantages par rapport aux autres. Il est connu que les casques de Réalité Virtuelle rendent malade. Bien que cela ait changé par rapport aux débuts, l'HoloLens n'a pas ce problème. Ceci est dû au fait d'avoir des hologrammes et de voir en permanence l'environnement. Il est promis à un meilleur avenir car cette technologie ne coupe pas les personnes du reste du monde.

Dans l'éducation, un élève peut étudier le fonctionnement d'un mécanisme complexe (mécanique automobile), peut travailler à l'échelle 1:1 (architecture) ou peut travailler sur des éléments dangereux en toute sécurité (électricité). Microsoft mise sur la collaboration à distance de l'HoloLens.

Le casque offre une large ouverture dans de nombreux domaines. Il n'est pas seulement bloqué dans les jeux vidéos. Ceux-ci sont peut-être la dernière préoccupation de Microsoft. L'HoloLens est vraiment un produit dédié pour les entreprises. Cependant, les jeux attisent la curiosité des particuliers et le rendent malgré tout attractif. La possibilité de faire ses propres applications reste accessible pour les particuliers.

La haute résolution des écrans permet l'affichage de textes lisibles. Ce qui est moins le cas de la VR où la résolution n'est pas optimale.

Alors que les détails ne sont pas précis, les mesures globales le sont. Lorsque l'HoloLens scanne son environnement, les éléments de la pièce se superposent avec le maillage virtuel. Il y aura des problèmes sur les arêtes des objets mais pas les zones planes.

Le poids de l'HoloLens est faible : 579g. Ce qui est appréciable lors d'une longue utilisation.

Pour finir, le casque reste maniable, sur de grandes distances. En effet, le casque n'a pas besoin d'être obligatoirement branché. Ainsi, nous sommes totalement autonomes dans les déplacements et avons les mains libres grâce à la reconnaissance gestuelle.

	AVANTAGES	INCONVENIENTS
VISION	Vision de l'environnement Textes lisibles	Champs de vision faible Mauvaise gestion de la luminosité
RESSENTI APRES UTILISATION	Casque léger Aucune nausée	Maux de tête lors d'utilisation prolongée Fatigue
CURSEUR	Prise en compte de la forme des objets 3D	Fixe au centre de l'écran
GESTES	Mains libres	Gestes peu naturels Gestes parfois non reconnus Manque de précision
ACHAT	Rentable pour les entreprises	Prix élevé pour les particuliers
AUTONOMIE	Possibilité de laisser le casque branché Maniable sur de longues distances	Autonomie courte

Fig.9 Tableau récapitulatif du casque HoloLens

## L'état de Part de l'HoloLens<sup>[21]</sup>

La technologie de la Réalité Mixte ouvre plus de possibilités que les autres réalités. Les cas d'applications sont plus importants, et touchent plus de domaines. Comme les autres casques, nous retrouvons les jeux vidéos avec des jeux comme Minecraft<sup>[22]</sup>. Il s'agit d'un jeu qui appartient à Microsoft. Une partie du jeu a été spécialement développée pour la technologie du casque. Ensuite, nous avons quelques jeux spécifiques à l'HoloLens : RoboRaid, un jeu de combat avec un robot ; Fragments, il faut découvrir des scènes de crimes pour être un détective ; HoloTour, pour explorer certaines villes du Monde ; Young Conker, qui est un mini jeu pour les enfants. Même la manière de faire un jeu pour le casque change par rapport aux autres technologies. La plupart de ces jeux prennent en compte l'environnement réel, les tables ou les murs.

Nous avons une variété d'utilisation dans l'architecture déjà existante. Des logiciels qui sont connus pour les architectes, comme Revit ou Sketchup, ont déjà développé leurs applications. Celles-ci sont payantes et peuvent être limitées dans les fonctionnalités. L'utilisation directe de Revit sur le terrain pourrait faire gagner du temps sur les chantiers pour les architectes. Une fois la 3D alignée au bâtiment existant, nous pourrions voir, par exemple, les trous qu'il faut percer ou si les dimensions sont exactes. Il y a une application qui permet de voir un projet BIM<sup>[23]</sup>.

Nous pouvons importer nos fichiers personnels grâce à 3D Viewer Beta. Néanmoins, les fonctionnalités après l'importation sont limitées. Nous n'avons pas parlé de l'application : HoloLive3D. Nous pouvons, directement sur le site, nous projeter les travaux à faire ou en cours. Il est aussi possible de montrer les canalisations et vérifier si elles ne croisent pas d'objets sur leur chemin. Avec cette application, nous pouvons anticiper les défauts d'un projet.

Dans le domaine industriel, nous avons beaucoup d'applications développées. Dans un premier temps, nous avons Skype<sup>[24]</sup> qui permet d'afficher l'application sur un mur. Il permet aussi le travail à distance. Une personne formée peut prendre en charge la réparation en n'étant pas sur place. Il est possible pour elle, de mettre en valeur ou d'annoter des objets qu'elle voit dans le casque. Elle donne des indications pour que le technicien sur place puisse intervenir. Grâce au casque, l'intervention peut être contrôlée en temps réel. L'HoloLens peut ainsi faire changer les formations ou l'éducation à distance.

Dans la mécanique, nous avons des sociétés comme Volvo (Automobile), Volkswagen (Automobile), Ford<sup>[25]</sup>(Automobile), ThyssenKrupp (Ascenseurs) ou encore la NASA qui utilisent le casque. Ils l'utilisent pour le design de leurs pièces ou pour améliorer le confort des ouvriers sur les chaînes de montage. Volvo utilise les lunettes pour une Réalité Augmentée.

Le domaine médical<sup>[26]</sup> a lui aussi utilisé cette technologie, et a réalisé les premières opérations avec succès. Nous retrouvons des utilisations dans le *Trading*<sup>[27]</sup> : application qui permet d'afficher différents écrans pour les besoins de la profession.

Pour terminer, il existe une application de création d'objets : HoloStudio. Elle n'est pas orientée pour les professionnels, mais cette ébauche pourrait devenir intéressante si elle se développe.

---

21 PIROTTE Jeremy, Article : [Test] Microsoft HoloLens : le casque de réalité augmentée de Microsoft [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/test-microsoft-hololens>> [consulté le 02/03/2018]

22 Chaîne Youtube Kotabu : Minecraft HoloLens demo at E3 2015 [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=xgakdcEzVwg>> [consulté le 20/02/2018]

23 Chaîne Youtube Synchro Software : Introduction to HoloSITE, utilisation application HoloLens pour un projet BIM [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=Pdl0mLKTmy4>> [consulté le 20/02/2018]

24 Chaîne Youtube Microsoft HoloLens : Microsoft HoloLens: Skype [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=4QiGYtd3qNI>> [consulté le 20/02/2018]

25 BERGOUNHOX Julien, [Interview] Comment Ford utilise la réalité virtuelle pour mieux concevoir ses voitures [en ligne]. Disponible sur <<https://www.usine-digitale.fr/article/interview-comment-ford-utilise-la-realite-virtuelle-pour-mieux-concevoir-ses-voitures.N562253>> [Consulté le 01/03/2018]

26 Chaîne Youtube AP-HP : Opération de l'épaule réalisée avec un casque de réalité mixte [en ligne]. Disponible sur <[https://www.youtube.com/watch?time\\_continue=4321&v=xUVMeib0qek](https://www.youtube.com/watch?time_continue=4321&v=xUVMeib0qek)> [consulté le 22/02/2018]

27 Site 8Ninths de l'Université de Valence, travaux sur la réalité augmentée [en ligne]. Disponible sur <<http://8ninth.com/>> [consulté le 22/02/2018]



Fig.10 Photographies issues d'applications HoloLens. Source : Google Image

Ces applications sont déjà installées dans le casque :

- Windows Store : sert à avoir les différentes applications pour l'HoloLens
- Windows Feedback : l'application d'échange d'informations
- Setting : nous l'avons déjà vu pour les paramètres
- Microsoft Edge : le navigateur internet
- Photos : permet de prendre ou de voir les photographies prises
- Calibration : pour adapter les paramètres en fonction de la personne qui utilise le casque
- Learn Gesture : permet d'apprendre les gestes de l'HoloLens
- Holograms : l'application qui permet d'intégrer des éléments dans le monde par défaut.

Nous trouvons d'autres applications sur le Windows Store. Elles sont pour la plupart gratuites et permettent d'interagir, de modifier ou de contempler le monde de la Réalité Mixte.

Toutes ces possibilités ouvrent les champs possibles à beaucoup d'applications et c'est pour cela que l'HoloLens est mis en avant. Par rapport à ses concurrents, l'utilisation est différente. Ce qui est problématique, c'est le fait de n'avoir aucun concurrent direct pour faire une comparaison de la technologie. Il est possible qu'en 2019 de nouveaux casques apparaissent, comme le casque Magic Leap. Ce casque est en développement et se veut accessible pour le grand public.

Il est préférable pour Microsoft de permettre à la communauté HoloLens de pouvoir concevoir ses propres applications. Le fait de faire ces développements ouvre des possibilités et augmente l'utilisation du matériel sans l'aide de Microsoft. Ouvrir une technologie permet un développement plus rapide pour une technologie. Nous remarquons souvent le cas dans les logiciels OpenSource.

## Possibilités

Un des avantages majeurs d'une autre Réalité Mixte est l'absence de gravité. Ceci ouvre des options impossibles dans l'environnement réel. Nous pouvons penser à l'exploration d'un projet montrant les différentes couches le constituant. Il est possible d'enlever l'une de ces couches. Celles-ci peuvent être éclatées sur la hauteur afin d'être séparées les unes des autres. Montrer les différents éléments d'un projet permet de mieux comprendre certains aspects spécifiques de celui-ci. Quand nous avons seulement la structure, nous montrons la construction d'un projet, sa capacité à être modulable ou les zones plus ouvertes. Cela peut montrer aussi les faiblesses et peut permettre de les corriger.

Pour un projet patrimoine, il est important de montrer la différence entre l'avant et l'après réhabilitation. Nous pouvons le montrer facilement avec le casque, en intégrant plusieurs phases d'un même projet. Pour reprendre ce qui est dit précédemment, nous pouvons mettre une phase de projet par calque.

En cas d'agrandissement ou rétrécissement de l'objet, l'échelle peut être calculée automatiquement. Ensuite nous pouvons nous placer dans cet objet pour être en immersion et nous déplacer.



*Fig.11 Perspective du PFE patrimoine, Gabriel Cramoisy*

Nous pouvons superposer la maquette réelle avec la maquette virtuelle, afin de faire une sorte de calibration ou de synchronisation. Si nous n'avons réalisé qu'une seule partie de la maquette réelle, il est possible de la finaliser virtuellement. Pour reprendre le projet patrimoine de la ville de Metz, nous avons un élément marquant du projet : le château d'eau. Il devient intéressant de faire simplement cet élément en réel et d'implanter le reste du projet en virtuel. Lors de phase de construction, le projet change toutes les semaines. Avoir une 3D qui représente le projet est moins long que de faire une maquette, surtout avec les logiciels qui font les 3D en même temps que les plans.



Fig.12 Rendu Blender du laboratoire en scanner par l'HoloLens

Nous avons la possibilité de travailler sur un logiciel comme Revit<sup>[28]</sup> ou Sketchup<sup>[29]</sup> afin d'explorer le projet. Nous pouvons faire un lien avec Blender pour des modifications en temps réel entre le logiciel et le casque<sup>[30]</sup>. En poussant le concept plus loin, nous pouvons avoir une chaîne de création. Un casque HoloLens branché à Blender lors d'une réunion sur un projet, offre la possibilité de changer celui-ci soit avec le casque, soit sur Blender en temps réel. Une fois que le modèle convient, il est possible de l'imprimer.

En poussant l'interopérabilité entre le programme et le casque, il deviendrait possible d'établir des points de vues dans l'HoloLens et de sortir le rendu photo-réaliste du logiciel.

Dans un projet d'architecture, au moment de le présenter, nous établissons un discours face à un écran. Le temps étant court pour parler d'un travail de plusieurs mois, nous ne pouvons pas tout montrer. En imaginant que tous les membres du jury portent un casque, nous pourrions rendre une présentation plus interactive. Il est possible d'avoir un discours autour du projet numérique, et que chacun puisse l'explorer à sa manière tout en suivant ce qu'il se passe durant la présentation. Il serait possible de mettre en évidence des éléments du projet, en reprenant les possibilités énumérées plus tôt. Sur un même projet, nous pourrions avoir différentes échelles : commencer avec l'échelle de la ville, situer le projet, puis se rapprocher jusqu'aux détails. Cependant, une technologie pareille demande une compréhension totale de l'ensemble du projet de la part de l'architecte. Un autre avantage de la technologie est qu'un membre du jury venant d'un autre pays, avec un casque, puisse participer à la présentation sans se déplacer.

---

28 Chaîne Youtube VisualLife : Revit to HoloLens [en ligne]. Disponible sur < <https://www.youtube.com/watch?v=Wxfhctei0-4> > [consulté le 10/03/2018]

29 Chaîne Youtube Sketchup : SketchUp Viewer for Hololens Demonstration [en ligne]. Disponible sur < <https://www.youtube.com/watch?v=dmpoCjz0Yc0> > [consulté le 10/03/2018]

30 Chaîne Youtube u9vata : Blender meets Hololens 2 - upgrades to HoloRenderer [en ligne]. Disponible sur < <https://www.youtube.com/watch?v=b5wflLPtigY> > [consulté 17/08/2018]

## Actualités

Le premier casque de Réalité Mixte commercialisé a beaucoup fait parler de lui. Il est un sujet que tout le monde veut aborder, tester ou acheter. Malgré le grand nombre d'articles, ils se répètent sur les faiblesses et les points forts d'une telle technologie. Il faut savoir que durant mon stage, plusieurs mises à jour ont considérablement fait évoluer le casque. Une première en mars 2018 a permis de corriger une partie des bugs connus, puis une seconde en août 2018 qui était plus importante. Nous avons un changement d'interface, des scans plus légers, mieux organisés et plus précis. Cette dernière mise à jour est arrivée durant la programmation de l'application, ce qui n'a pas eu d'incidences, mais a permis un travail plus fluide.

Avant de commencer réellement nous pouvons trouver quelques livres qui ont été écrits sur le casque de MR. Nous avons *Develop Microsoft Holograms Apps Now* de Allen G. Taylor<sup>[31]</sup>, *Building Apps for the Universal Windows Platform* de Ayan Chatterjee<sup>[32]</sup>. De manière générale, ils reprennent les éléments de création d'une application et ce qui est développé sur le site de Microsoft, et Unity.

Nous allons voir ce que les articles de presse disent de ce casque HoloLens :

Microsoft Holograms : le casque de réalité augmentée de Microsoft <sup>[33]</sup> :

L'article présente les différentes caractéristiques du casque. Les points forts sont la qualité et facilité d'interaction. De plus, l'installation initiale est rapide et simple d'utilisation, environ 5min. Néanmoins, le champs de vision fait vite parler de lui car il est trop petit et bloque l'immersion. Le son compense ce point négatif, grâce à un haut parleur sur chaque oreille. Il y a une bonne immersion dans l'environnement. Le son se déplace en même temps que nos mouvements.

Avec son prix élevé, il est peu accessible pour les particuliers qui n'y trouvent pas un grand intérêt. Bien que le casque soit volumineux, il reste léger et agréable sur la tête. Cortana (commande vocale de Microsoft) est réactive, mais est inutilisable en français.

Dans l'expérimentation, le casque est agréable à utiliser du fait qu'il n'est pas de connexion filaire. Nous avons une liberté dans les mouvements et de plus, il a une autonomie de 2 à 3 heures en fonctionnement. De part les capteurs du Microsoft HoloLens, il y a une grande performance en terme de qualité des mouvements de têtes et des déplacements dans l'environnement. Les applications volent autour de nous et font oublier le casque. L'HoloLens n'a pas de rapport direct avec la concurrence. Il est dans une autre catégorie et il est donc difficile de se faire une idée d'un bon rapport qualité/prix ou d'avoir un avis sur sa performance.

100% Windows : à quoi sert Holograms ? <sup>[34]</sup> :

Nous avons comme intervenant dans cette vidéo, Michel Rousseau qui est consultant chez Microsoft France. Le Microsoft HoloLens est un PC autonome qui ne coupe pas du monde réel. Il scanne l'environnement avec des capteurs qui calculent en permanence l'environnement. Windows pense que c'est l'évolution des PC. Il n'y aura plus besoin d'écrans. En ce qui concerne l'interaction : il a une utilisation de 2h30. Nous pouvons garder les lunettes de vue si nous en avons. L'HoloLens a le même système d'exploitation que le Windows PC et est compatible avec l'ensemble Microsoft. Tout le monde cherche à développer son casque de réalité virtuelle.

La résolution n'est pas calculable mais est comparée à du 720p par écran.

---

31 G.TAYLOR Allen, *Develop Microsoft Holograms Apps Now*, Apress, Royaume-Uni, 2016

32 CHATTERJEE AYAN, *Building Apps for the Universal Windows Platform*, Apress, Royaume-Uni, 2017

33 PIROTTE Jeremy, Article : [Test] Microsoft Holograms : le casque de réalité augmentée de Microsoft [en ligne]. Disponible <<https://www.realite-virtuelle.com/test-microsoft-holograms>> [consulté le 01/03/2018]

34 SOREL François, *100% Windows : à quoi sert Holograms ?* [en ligne]. Disponible sur <<https://www.01net.com/mediaplayer/video/100percent-windows-a-quoi-sert-holograms-950139.html>> [consulté le 01/03/2018]

Le consultant explique le point de vue de Microsoft et vend la technologie comme celle de demain, utilisable dans tous les domaines, dans les jeux vidéos, l'éducation internationale et l'industrie. Il énumère différentes entreprises qui utilisent le casque. Le Falcon (Avion de ligne) est construit virtuellement et il n'existe plus de prototype réel. La technologie permet de faire des prototypes qui ne coûtent pas chers et qui sont totalement virtuels. La première construction est fonctionnelle. La Nasa l'utilise pour les expérimentations pour Mars, ou encore Facebook qui lance quelques innovations avec le Facebook live. Ils comparent les hologrammes avec ceux de Star Wars en parlant de concept d'holoportation<sup>[35]</sup>. La technologie proposée par Microsoft est simple d'utilisation et permet le développement des applications pour n'importe qui. Comment marche la programmation des applications, sur Unity ? C'est facile !

[Hola HoloLens: Reg man gets face time with Microsoft's holographic headset](#)<sup>[36]</sup> :

Cet article a été publié avant la sortie de l'HoloLens. Ils comparent le casque avec l'Oculus Rift qui est plus avantageux économiquement. Néanmoins, Oculus nous rend statique et aveugle au monde qui nous entoure. Microsoft a créé un produit qui permet une simplicité de circulation autour des objets et de bouger la tête. L'interface reste simple d'utilisation, même dans le contrôle de l'HoloLens grâce aux gestes, aux yeux ou la voix. Nous retrouvons le son en stéréo qui reste un point fort de la technologie et ils rajoutent un bon design et une batterie avec une longue durée de vie. Dans cet article, ils ont testé les applications : HoloStudio, Minecraft-type Game, Skype et Mars. Cette application de la NASA permet la reconstitution de la planète à l'aide des données de Curiosity Mars. Cette dernière est une bonne expérience avec une superbe vision. Dans l'ensemble, l'HoloLens est une bonne expérience et le produit a de l'avenir.

[J'ai testé l'HoloLens de Microsoft et le futur n'est pas pour tout de suite](#)<sup>[37]</sup> :

Cette fois l'article est publié un an après la sortie du produit. Celui-ci est plus critique que les autres. L'HoloLens donne des maux de têtes et le casque est inconfortable au bout d'un certain temps, bien qu'il ne donne pas la nausée. Ils comparent l'HoloLens à la technologie des Google Glass. Ils parlent rapidement des hologrammes, de la reconnaissance de certains gestes. Ils ont utilisé l'application Skype durant leur expérience. Par contre, il y a une prise en main rapide et naturelle de l'appareil. Malgré le relevé de tout ces points négatifs, ils sont convaincus de la technologie proposée.

## L'avenir de la technologie

Bien que certaines personnes soient septiques<sup>[38]</sup> du casque, Microsoft se projette<sup>[39]</sup> déjà dans l'avenir de sa technologie avec sa nouvelle version de l'HoloLens. Il est vrai que les possibilités sont grandes. Dans un futur, nous pourrions imaginer des casques supportant des logiciels comme ArchiCAD ou TwinMotion. Il serait possible de travailler directement dans le casque pour construire un projet d'architecture. Il est déjà autonome

---

35 Chaîne Youtube I3D Past Projects : holoportation: virtual 3D teleportation in real-time (Microsoft Research) [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=7d59O6cfaM0>> [consulté 01/03/2018]

36 THOMSON Iain, Article : Hola HoloLens: Reg man gets face time with Microsoft's holographic headset [en ligne]. Disponible sur <[https://www.theregister.co.uk/2015/01/23/hands\\_on\\_with\\_the\\_hololens\\_or\\_how\\_facebook\\_wasted\\_2bn\\_on\\_oculus/](https://www.theregister.co.uk/2015/01/23/hands_on_with_the_hololens_or_how_facebook_wasted_2bn_on_oculus/)> [consulté le 22/02/2018]

37 RONFAUT Lucie, Article : J'ai testé l'HoloLens de Microsoft et le futur n'est pas pour tout de suite [en ligne]. Disponible sur <<http://www.lefigaro.fr/secteur/high-tech/2016/10/17/32001-20161017ARTFIG00001-j-ai-teste-l-hololens-de-microsoft-et-le-futur-n-est-pas-pour-tout-de-suite.php>> [consulté le 20/02/2018]

38 Chaîne Youtube Windows : Envisioning the Future with Windows Mixed Reality [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=2MqGrF6JaOM>> [consulté le 20/02/2018]

39 KIPMAN Alex, A Futuristic Vision of the Age of Holograms [en ligne]. Disponible sur <[https://www.ted.com/talks/alex\\_kipman\\_the\\_dawn\\_of\\_the\\_age\\_of\\_holograms/discussion?la](https://www.ted.com/talks/alex_kipman_the_dawn_of_the_age_of_holograms/discussion?la)> [consulté le 20/02/2018]

pour certains logiciels. Avec le développement des ordinateurs cloud, l'HoloLens n'aurait pas besoin d'une grande capacité de calcul. Nous pouvons, aujourd'hui, retrouver le futur dans notre casque avec la vision de Terminator<sup>[40]</sup>.

---

40 MICROSOFT, Article : Building the Terminator Vision HUD in HoloLens [en ligne]. Disponible sur <<https://blogs.windows.com/buildingapps/2017/03/06/building-terminator-vision-hud-hololens/#HpM7CKKhT6khyrZH.97>> [consulté le 20/02/2018]



# **NIVEAU 02**

UTILISATION DU  
CASQUE



## INTRODUCTION

---

Dans ce chapitre nous allons parler des différentes manières d'utiliser le casque HoloLens.

Nous venons de voir avec les articles de presses la face visible de l'HoloLens, mais nous pouvons rappeler que Microsoft mise sur la popularisation et sa facilité d'utilisation. C'est pour cela que dans les prochaines parties nous allons montrer comment entrer au cœur du casque. C'est-à-dire, la partie que Microsoft veut bien nous laisser entrevoir. Nous verrons les caractéristiques, les paramètres et plus tard nous entrerons dans le code afin d'y développer l'application.



# HOLOLENS

## L'utilisation dans les détails

Nous allons commencer par la partie interne au casque. Sur le menu, nous voyons en haut, la connexion au réseau Wi-Fi, puis l'heure avec le niveau de batterie et enfin le niveau sonore. Lorsque nous avons le menu ouvert, au centre, nous avons les liens directs vers les applications. Dans la partie basse, nous trouvons le mode photo et vidéo. Il faut cliquer sur l'icône de l'application, la placer dans le monde et elle se lance automatiquement. Chaque application s'ouvre dans une nouvelle fenêtre qu'il faut forcément placer dans le monde initial.

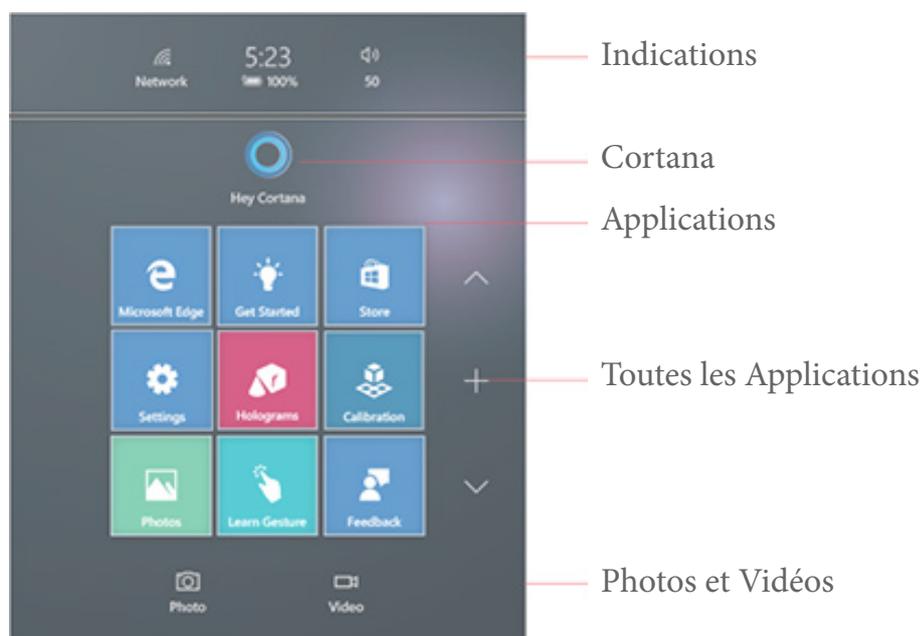


Fig.13 Photographie du menu de l'HoloLens. Source : <realitycheck3000.com>

Ce qui nous intéresse est l'application *Setting*. Elle permet d'en apprendre un peu plus du casque. Nous allons seulement voir les éléments qui nous intéressent d'un point de vue technique : les éléments utiles pour l'application et comprendre la technologie.

< *Windows Setting / System / Storage* > : il s'agit de l'espace de stockage de 60GB des applications que l'on peut mettre sur l'HoloLens. Une partie du stockage est dédiée au scan de l'environnement. C'est cet espace qui est enregistré dans la mémoire et qui permettra la reconnaissance du lieu dans lequel nous sommes. Il est possible de récupérer cet espace.

< *Windows Setting / system / Utilities* > : il est pratique pour réinitialiser les paramètres de l'appareil, la sensibilisation ou la calibration de l'appareil. Normalement il faut calibrer le casque pour toutes les personnes qui vont l'utiliser. Cette calibration va jouer sur la profondeur de champs de vision des yeux et l'écartement. L'HoloLens va corriger automatiquement l'alignement.

< *Windows Setting / Devices / Bluetooth* > : il est possible de connecter quelques appareils par bluetooth afin d'augmenter l'interaction comme un clavier ou une souris. Par défaut Microsoft met dans son pack HoloLens, un Clicker qui permet simplement de faire un clic pour valider ou de faire un geste tenu afin de pouvoir se déplacer dans l'espace comme pour la main. Les *Motions Controller* prévus pour la VR ne peuvent pas être connectés sur le casque. Néanmoins, il est possible de connecter les manettes *Xbox Wireless Controller S*.

< *Windows Setting / Network and Internet/ Wi-Fi* > : dans cette section, il est possible de modifier le réseau. Nous pouvons récupérer, en cliquant sur : *hardware properties*, l'IPv4 qui permet de se connecter.

Il y a la possibilité de récupérer plus d'informations sur l'HoloLens en le connectant à un ordinateur. Pour cela, il existe deux possibilités d'atteindre le *Windows Device Portal*. La première est de brancher le casque par USB, puis ouvrir un navigateur à l'adresse 127.0.0.1:10080 (l'adresse locale).

La méthode la plus fiable est par Wi-Fi, en se connectant sur le navigateur à l'adresse IPv4 de l'HoloLens, elle commence par 10.0.1. ... . Elle peut varier si le réseau change. Il faut, par conséquent que le casque et l'ordinateur soient connectés sur le même réseau Wi-Fi.

Le point suivant est contraignant lorsque nous voulons partager une expérience avec des personnes qui n'ont pas accès à un casque. Lorsqu'il est nécessaire de montrer des interactions avec des éléments, un projet, ou le fonctionnement d'une application à un public, nous avons besoin de savoir ce que l'utilisateur voit dans son casque. Microsoft a prévu sur le *Windows Device Portal* (WDP), dans l'onglet <Views / Mixed Reality Capture>, un accès à ce que l'utilisateur perçoit. A l'aide de la caméra centrale, nous voyons l'environnement qui est superposé avec les hologrammes. Nous avons donc, la retranscription de ce qu'il se passe dans le casque. Néanmoins, deux contraintes entrent en jeu et enlèvent complètement l'interaction entre l'utilisateur et le public.

Premièrement, la caméra ne se trouve pas au niveau des yeux de l'utilisateur. Ce détail crée un décalage lorsque nous montrons une interaction précise. Dans l'application, lorsque nous voulons superposer les objets virtuels et les objets réels, il faut pointer un coin de l'objet. Lorsque nous le montrons à l'écran, le coin ne se trouve pas sous le curseur.

Deuxièmement, la retranscription de la vidéo a un décalage de quelques secondes. L'expérience n'a pas montré une amélioration de ce point négatif.

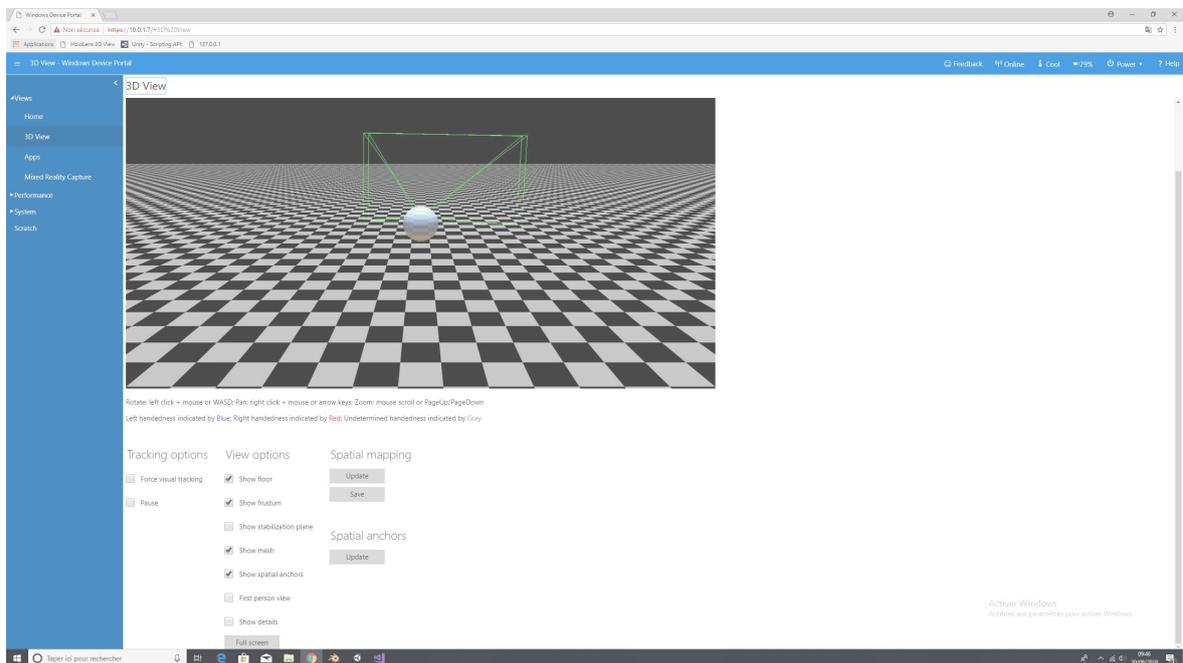


Fig.14 Le Windows Device Portal (WDP) : interface de l'HoloLens. Capture d'écran personnelle

Nous arrivons dans l'interface de Microsoft HoloLens et à partir de celui-ci, il est possible de contrôler les paramètres du casque. Les premières options, regroupées dans View, permettent de gérer les paramètres principaux des lunettes ainsi que tout ce qui va être créé ou perçu par l'utilisateur. Nous retrouvons la visualisation de l'environnement capturé par la caméra centrale, avec la possibilité d'afficher les hologrammes, ou bien toutes les données mesurées par l'ensemble des capteurs.

Il existe une fenêtre "3D View" plus technique qui apporte toutes les données captées par l'HoloLens. Lorsque nous sélectionnons *Show Details* : nous pouvons repérer la position de la tête, le nombre de mains visibles ou les vecteurs. Il y a aussi une fenêtre de visualisation. Celle-ci montre l'HoloLens et son champs

de capture de 0.5m à 4m. Il est possible d'afficher l'environnement réel scanné qui nous entoure. Ce qui est intéressant avec cet environnement est que nous pouvons le récupérer en \*.obj. Il n'est pas possible de récupérer l'ensemble du maillage en une seule fois. Lorsque nous cliquons sur Update, seule la partie proche qui nous entoure se met à jour. Des volumes plus lointains apparaissent. Il peut s'agir des parties visibles s'il y a un trou dans le maillage.

Pour finir, nous avons d'autres onglets qui permettent de paramétrer l'HoloLens. Nous pouvons voir dans < *System / Preferences* >, la valeur spécifique à chaque utilisateur. En millimètre, elle représente le point focal. Avec l'onglet *Performances*, il est possible de connaître toutes les informations en temps réel du casque.

## Les Espaces

Lorsque nous allumons le casque, nous remarquons un chargement. L'HoloLens charge à ce moment là l'environnement. C'est ce que nous appelons : *Space*. Il est créé dès lors que l'HoloLens n'en trouve pas d'existant. Si le casque n'en reconnaît pas, alors qu'il le devrait, il faut se déplacer. Ces espaces se sauvegardent au fur et à mesure que nous avançons dans le monde réel ou que nous ajoutons un objet. Si nous sommes déconnectés, nous pouvons charger la dernière version de notre *Space*. Il est possible de récupérer l'espace en format \*.obj. (Dans la connexion USB ouvrir *3D View / Spatial Mapping : Update* puis *Save*).

Lorsqu'il scanne l'environnement, il existe des surfaces qu'il ne prend pas en compte. Il ne voit pas toutes les matières transparentes ou réfléchissantes. Nous retrouvons ce problème dans la plupart des Scanners.

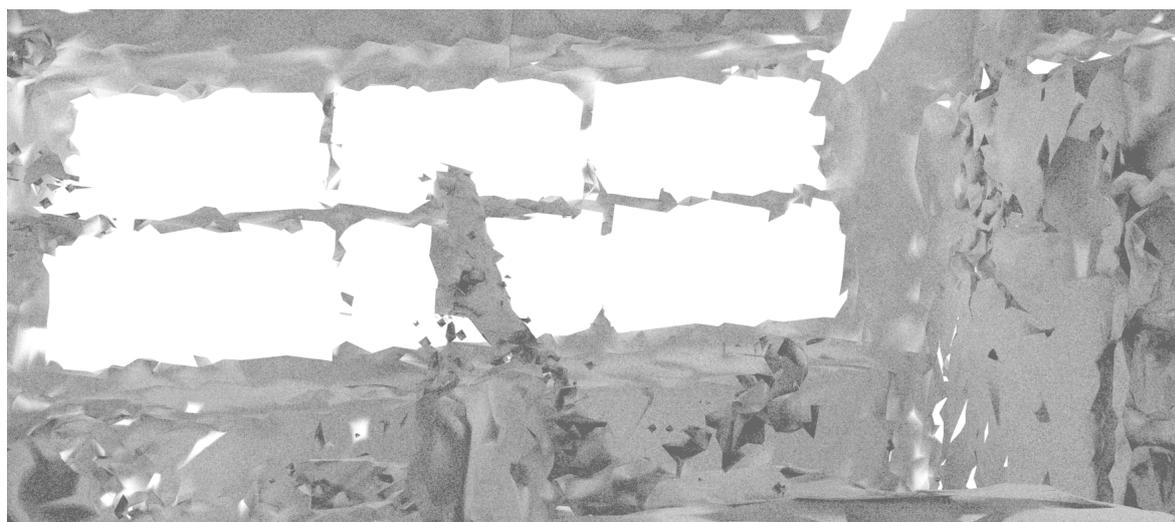


Fig.15 Image d'une fenêtre du CRAI issue du scan HoloLens, gris : maillage. Rendu Blender

Afin de fonctionner de façon optimale, le casque a besoin d'être connecté à un réseau Wi-Fi. Dans le wifi *advanced options*, il y a l'information adresse IP qui permet de pouvoir se connecter à l'appareil à partir de n'importe quel PC *for developers*. Il est également possible de faire fonctionner le casque sans Wi-Fi.

Lorsque l'HoloLens n'est pas connecté à un réseau, il est limité dans ses actions. Il entre dans le mode limité, c'est-à-dire que certaines applications ne fonctionnent plus (ou avec des capacités réduites) et il faut charger manuellement les *Spaces*. Il y aura une zone limitée lorsque l'on se déplacera dans notre environnement.

Malgré le prix de l'HoloLens, certains bugs connus peuvent apparaître. Des solutions simples<sup>[41]</sup> peuvent être appliquées pour la plupart des cas. Tout au long des mises à jour, les bugs disparaissent. Il y a une nette amélioration de l'environnement et la rapidité d'exécution dans la dernière mise à jour.

41 L.Bastien, Article : Microsoft HoloLens : solutions aux principaux bugs et problèmes [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/microsoft-hololens-problemes-bugs>> [consulté le 01/03/2018]

## Les gestes reconnus

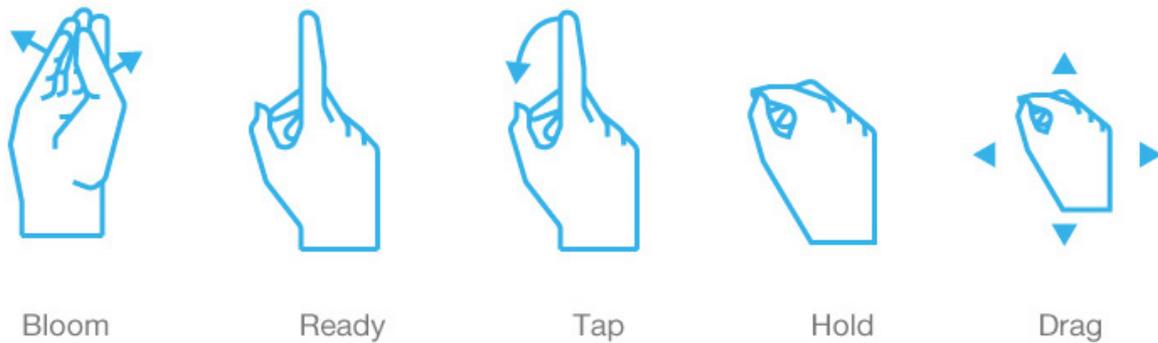


Fig.16 Dessin montrant les gestes utilisables pour le casques HoloLens Source : <learn.fologram.com>

Pour pouvoir interagir avec les hologrammes, il faut bien sûr obtenir les informations sur les entrées de l'utilisateur. Cette partie paraît simple, mais ne l'est pas forcément. Nous pouvons récupérer les coordonnées de la main de l'utilisateur ainsi que ses mouvements. C'est l'une des dernières choses que j'ai appris à faire sur Unity. Le problème est que les notices que l'on trouve sur internet ne sont pas exactes et le fait de devoir les adapter relève plus de connaissances que de simplement comprendre un programme. Avec le peu d'informations, et sans l'aide de Pascal Humbert, je n'y serai pas parvenu. Une fois ces données récupérées, il est possible de superposer un objet sur notre main pour voir son déplacement. Toutefois dans notre cas, il aurait été difficile d'utiliser cette manipulation. Dans une autre application nous aurions pu prendre en compte ces coordonnées afin de déplacer les objets avec ces mouvements.

Voici les manipulations par défaut :

Bloom : Ouverture du menu, peut importe où l'on se trouve, dans n'importe quelle application. Le bloom activé, il ramènera dans le menu initial de l'HoloLens.

Ready : Attente d'une instruction et permet de détecter la main.

Tap : Sélection classique

Hold : Prise en main de l'objet

Drag : Déplacement d'un objet

Nous pouvons noter qu'il existe une application permettant d'utiliser les deux mains.<sup>[42]</sup>

## La reconnaissance vocale

La reconnaissance vocale permet d'élargir les gestions d'entrée du casque. Cette fonctionnalité se fait avec Cortana, l'intelligence artificielle de Microsoft. Le problème de cette technologie est la langue. Il faut parler en anglais pour se faire comprendre. De plus, il faut que l'accent soit présent pour un bon fonctionnement. Pour une utilisation grand public de l'application, la reconnaissance vocale n'est pas activée. Afin que l'application soit utilisable par des francophones. Afin de corriger l'accent français sur les mots anglais, il faudrait, par exemple, programmer une reconnaissance du mot "tree" (arbre) pour le mot "three" (trois).

42 ODOM Jason, 3D Live Tiles, Two-Handed Manipulation & More Finally Coming to the HoloLens 03/03/2018 [en ligne]. Disponible sur < <https://hololens.reality.news/news/dev-report-3d-live-tiles-two-handed-manipulation-more-are-finally-coming-holo-lens-0183241/> > [consulté le 08/03/2018]

## Travail en Groupe

Le casque offre une possibilité de travailler à plusieurs sur un même projet. Dans le cas de plusieurs utilisateurs HoloLens, un avatar apparaît. Il est possible de travailler à distance en collaboration et de concevoir des projets à plusieurs ou de simplement montrer les modifications d'un projet. Microsoft mise une partie de sa recherche dans la partie collaborative de l'HoloLens. Aujourd'hui, avec les dernières mises à jour, les applications par défaut permettent de simplifier ces travaux, notamment par l'échange d'informations visuelles, ou l'interaction entre différents utilisateurs. Deux applications spécifiques<sup>[43]</sup> au casque pour les entreprises sont disponibles pour la collaboration. Dont une qui est totalement dédiée à l'aide collaboratif *Microsoft Remote Assist*. Cette application permet d'avoir une assistance à distance. Un professionnel peut donner des informations sur la manière d'intervenir sur un élément technique.

## Le spatial mapping<sup>[44]</sup>

On a déjà vu comment obtenir ce maillage depuis l'interface de l'HoloLens : *Windows Device Portal* (WDP). Mais dans les détails, nous allons voir comment cela fonctionne exactement. Dans un premier temps et d'après les éléments composants le casque, l'environnement est capté à l'aide des caméras de profondeur. Il faut par conséquent être dans un espace assez lumineux pour que les caméras captent cet environnement. De plus, l'espace doit être assez grand pour avoir le recul nécessaire par rapport aux objets qui nous entourent. L'HoloLens est "perdu" quand nous sommes à une cinquantaine de centimètres de l'objet à scanner. Si l'espace est perdu, il faudra revenir sur nos pas et retrouver le maillage de la pièce.

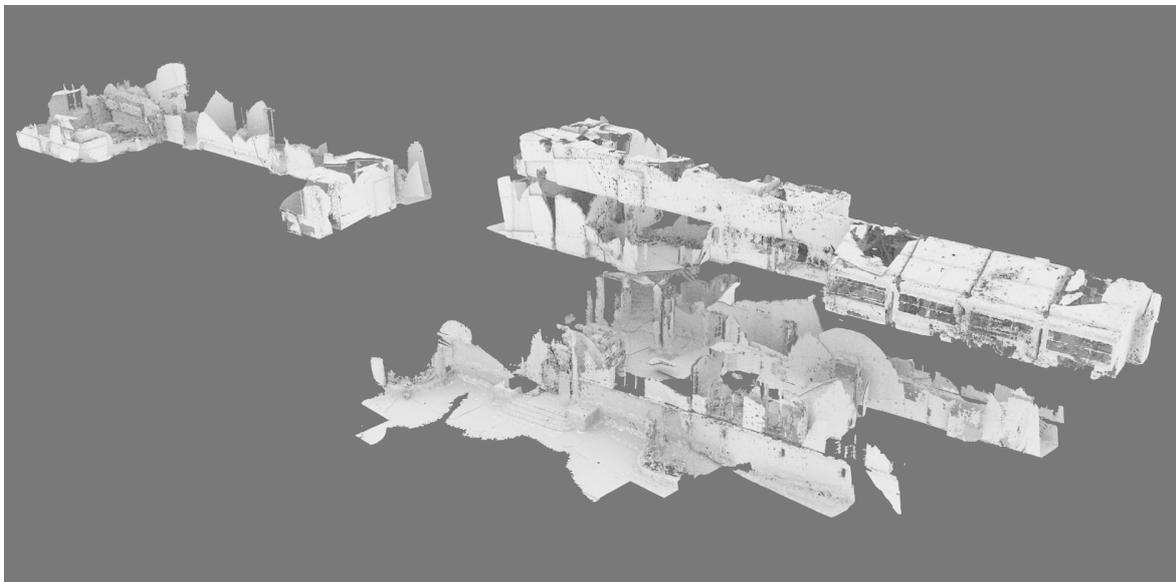


Fig.17 Le scan de l'école, Rendu Blender

L'HoloLens scanne l'espace qui nous entoure et le sauvegarde dans la mémoire interne du casque. Il aura toujours besoin de trouver un environnement lorsque nous l'allumons ou lorsque nous nous déplaçons. Si l'environnement n'existe pas, il le créera. Bien entendu, si l'espace existe il prolongera l'environnement déjà scanné. Il fera le lien et inclura ce scanne dans la globalité de l'ancien. Il faut cependant avoir le lien entre les

---

43 MICROSOFT, Preview new mixed reality apps for HoloLens [en ligne]. Disponible sur <<https://docs.microsoft.com/en-us/HoloLens/hololens-public-preview-apps>> [consulté le 19/07/2018]

44 MICROSOFT, Spatial mapping [en ligne]. Disponible sur <<https://docs.microsoft.com/fr-fr/windows/mixed-reality/spatial-mapping>> [consulté le 19/07/2018]

différents éléments. Une fois que ce scan est fait, nous pouvons le récupérer. Il faut savoir que pour de grands espaces il n'est pas possible de récupérer l'ensemble des données en une seule fois. Dans un premier temps, au moment où nous cliquons sur le bouton *Update* (dans le WDP/View), l'HoloLens prend la position du casque et fait un rayon autour de lui de 15 à 20m. Nous pouvons voir qu'il fragmente l'espace en *Chunk* qu'il va charger au fur et à mesure. Il affichera les éléments que l'utilisateur pourrait voir à travers les objets proches de lui. Par exemple, lors d'un scan, si le maillage proche contient un trou, et que nous avons préalablement scanné ce qu'il y a derrière ce trou, il sera possible de le voir.

## Un ressenti personnel

Je vais exposer mon ressenti personnel du 20 Février 2018, durant ma première prise en main du casque HoloLens. Lorsque nous commençons à utiliser l'appareil, nous aimerions qu'il fonctionne du premier coup car nous sommes surpris de ce que nous voyons dans le casque. Les premières pensées nous poussent à vouloir essayer d'interagir avec les objets.

Pour une réelle première utilisation de l'HoloLens, il est difficile de comprendre le fonctionnement et d'intégrer les gestes. Un simple clic du doigt n'est pas facile à entreprendre : il faut la bonne inclinaison et bien marquer le geste. Ce n'est pas la même manière de faire qu'avec une souris, où il suffit d'appuyer sur un bouton. Le *Air Tap* n'est pas palpable et lorsqu'il ne marche pas, nous ne savons pas pourquoi. Ce premier ressenti est confirmé lorsque qu'une personne extérieure veut utiliser le casque. Il devient plus simple d'utiliser le *Clicker* (une manette) alors qu'il fonctionne de la même manière. Une fois que nous avons le geste et que nous savons que le pointeur suit les mouvements de la tête, la prise en main se fait rapidement et nous comprenons aisément l'ensemble et le fonctionnement général du casque. Nous arrivons à naviguer simplement dans les différentes interfaces. Nous sommes dans un fonctionnement similaire à un Smartphone.

En revenant sur la visée de la tête, ce geste n'est pas intuitif et nous voulons attraper l'objet directement avec la main. Là aussi, ce ressenti est confirmé avec une personnes extérieures. Le fait de cliquer à coté pour que ce soit le pointeur qui soit pris en compte dérange les premières minutes d'utilisation. Il est peut-être possible de prendre le centre du casque et la position de la main, afin de tracer un vecteur. Quand ce vecteur croise un objet, l'objet est visé.

## Quelques expérimentations sur le fonctionnement

Afin de comprendre un peu mieux le fonctionnement du casque par rapport au stockage de l'environnement, nous allons faire quelques expériences. Celles-ci, étant réalisées avant la mise à jour sont obsolètes. Auparavant, les espaces de stockages étaient séparés et pouvaient être sélectionnés indépendamment. Maintenant il n'y a plus que la zone de stockage. Néanmoins, ne trouvant pas de réponses exactes concernant l'endroit où le *Spatial Mapping* est stocké, ces expériences donnent des pistes. Est-ce que l'environnement s'enregistre dans la mémoire du casque ou sur un *Cloud* ? Je me suis posé cette question parce que le casque cherche tout le temps à avoir une connexion pour enregistrer ce que l'utilisateur fait. Cette première approche donne aussi une idée de la taille de stockage des éléments scannés.

Voici l'état avant agrandissement de la zone scannée (seulement le laboratoire) :

Données Menu Spaces du 22/02/2018  
Dernière utilisation CRAI Privé  
*Space 1* Size 374MB (Active)  
*Space 2* Size 38MB  
Storage 10.3GB sur 58.2GB

Voici l'état après l'agrandissement de la zone scannée (descendre dans le couloir du premier étage)

Données Menu Spaces du 02/03/2018  
Dernière utilisation ASUS\_X008D  
Space 1 Size 374MB (Active)  
Space 2 Size 38MB  
Storage 10.5GB sur 58.2GB

Le résultat n'est pas logique. Nous avons bien une augmentation de la mémoire globale de 0.2GB mais aucun des deux *Spaces* n'a changé. Nous avons marché dans le *Space 1* qui est activé. Ce chiffre n'a jamais changé. Après la mise à jour d'août, il n'est plus possible de voir les *Spaces* scannés. Il y a juste un accès à la taille de la mémoire globale de l'HoloLens.

Nous avons vu précédemment qu'il était possible de voir le maillage de l'environnement que l'HoloLens scanne. Nous allons voir qu'il est aussi possible de le sauvegarder et de l'utiliser dans des logiciels 3D. Pour revenir sur la mise à jour d'août, le maillage est plus net et plus léger après un nouveau scan. Nous remarquons que le fichier a un tiers de poids en moins, quand nous l'ouvrons dans blender par exemple. Nous remarquons que c'est le nombre de faces qui change. Cette mise à jour permet des simplifications du maillage. Nous voyons d'ailleurs qu'il y a moins de superpositions inutiles entre chaque *Chunk*. La précision de calcul est ainsi plus importante et permet de mieux déterminer les formes : un poteau carré ressemblera moins à un poteau rond qu'auparavant. Cette légèreté de fichier permet de récupérer la puissance de calcul pour autre chose et d'être aussi plus rapide dans les exécutions.

En scannant l'intégralité de l'école nous avons 0.6Gb de données supplémentaires sur l'espace de stockage. Il est possible de réaliser ce scan en une seule fois en se déplaçant dans le bâtiment. Mais il existe des zones inaccessibles, en tant qu'utilisateur, comme le toit, les hauts des façades ou les éléments en hauteur. Afin de récupérer l'intégralité de ce scan, il est nécessaire de se déplacer dans toute la structure et de faire un *Update* tous les 15m environ. Si le scan est réalisé en une seule fois, lors de l'import du fichier 3D dans un logiciel, les éléments s'assembleront automatiquement. Si nous scannons une partie supplémentaire, le point central de l'ensemble changera et les éléments constituant la 3D seront décalés.

Une autre expérimentation a été réalisée sur la dimension des objets existants. Lorsque nous prenons la table tactile, la valeur mesurée en 3D est située entre 1.35m et 1.45m de long et 0.9m et 1m de large. Tandis que la valeur réelle est de 1.4m de long et 0.95m de large. Nous avons alors une précision de plus ou moins 10cm. C'est assez difficile d'être précis dans la prise de mesures. Si nous passons plus de temps à scanner, la précision augmenterait. Cette différence de mesure est dû à la triangulation des objets. En effet, avec le casque les petites surfaces ne sont pas reconnues et nous avons une perte de précision pour les arêtes. Des mesures précises sont toutefois possibles si l'objet est assez grand, comme un carton ou une pièce à vivre.

Une dernière expérimentation a été faite pour la détection de la main. Celle-ci est reconnue seulement dans une certaine position. Si la main fait une rotation d'un angle de plus de 45 degrés, quel que soit le sens, le geste n'est plus détecté. Cette contrainte rend l'utilisation de la fonctionnalité difficile.

Lorsque nous venons sur le WDP et voulons la vision du casque, il y a un retardement à l'affichage sur l'ordinateur par rapport à ce qu'il se passe sur l'HoloLens. Le défaut vient peut-être du niveau de chargement de la batterie. Pour cela, j'ai testé l'application à trois niveaux de batterie différents. La différence de vitesse peut également provenir du type de connexion.

Voici les temps de réponses de la vidéo lorsque nous sommes connectés en Wi-Fi avec l'ordinateur :

- Premier essai ( 30% batterie ) : entre 3 et 4s
- Deuxième essai (50% batterie) : moins d'une seconde
- Troisième essai (100% batterie) : entre 3 et 4 s

Le résultat est incohérent, ainsi nous savons que la batterie n'impacte pas la vitesse de visualisation. Il est possible que cette variation dépende du débit internet. Voici le résultat pour une connexion USB :

- Pour le premier test (50% batterie): un peu plus de 1s
- Pour le second test (50% batterie) : entre 3 et 4 s
- Pour le troisième test (40% batterie) : entre 3 et 4 s

Peu importe la façon dont nous branchons l'appareil, la retranscription prend du retard par rapport au direct. Cette rubrique manque de fluidité pour une bonne utilisation. Si nous voulions faire une présentation avec un visuel, le retard sur le discours ne serait pas agréable. Lorsque nous testons l'application avec Unity la retranscription est instantanée, alors qu'il existe une latence sur le WDP. Nous pouvons en déduire que celle-ci vient de la transmission des données vidéos, malgré une qualité de 720p.

## UNITY

### Présentation du logiciel

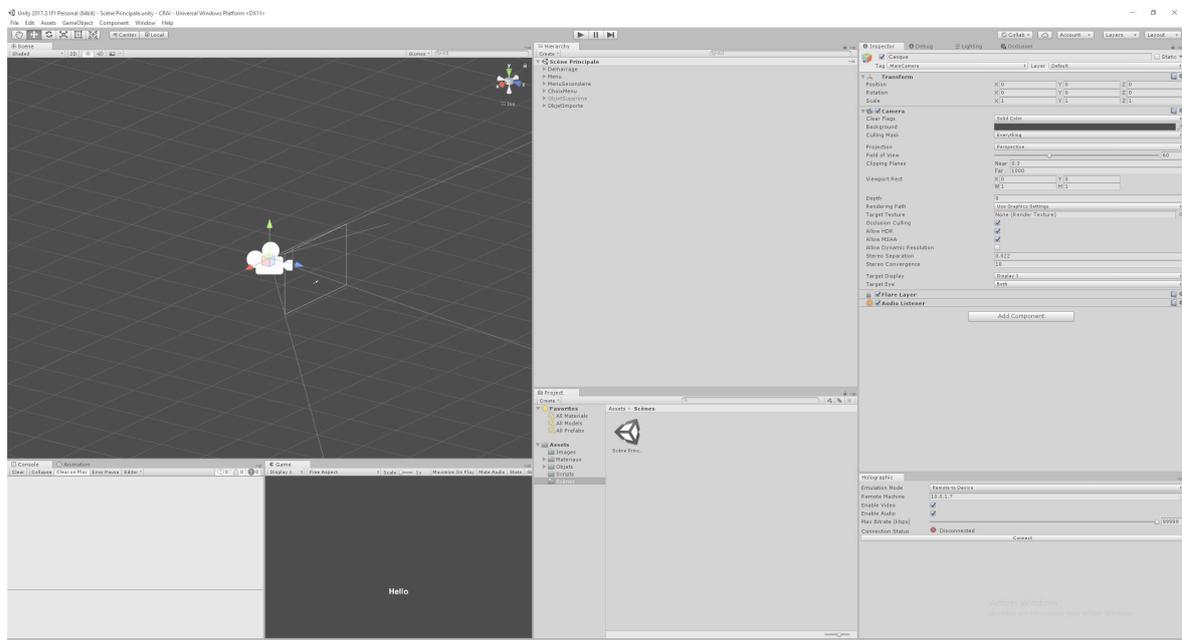


Fig.18 Image de l'interface

Unity est un moteur de jeux développé par *Unity Technologies*. C'est un logiciel gratuit si les bénéfices réalisés avec celui-ci sont en dessous de 100 000\$ par an. Unity est un logiciel multi-plateforme. Il permet de faire aussi bien un jeu sur Smartphone ou sur IOS ou encore sur Windows. Nous pouvons donc créer une application ou un jeu pour l'HoloLens car il fonctionne sous Windows. Le fait d'avoir un grand panel de plateformes, rend

Unity plus simple car ses fonctions sont plus proches de l'utilisateur. Comme Unity est un logiciel gratuit, il existe beaucoup de tutoriels<sup>[45]</sup> qui permettent de comprendre son fonctionnement. Ce logiciel permet de créer les applications pour l'HoloLens. Une grande partie de ces applications sont développées sur Unity. Nous pouvons, alors, interagir avec un élément sur l'écran, faire apparaître quelque chose en fonction du temps, de nouveaux gestes, ou encore d'autres fonctionnalités. Les applications de l'HoloLens semblent illimitées.

## Présentation de l'interface

Unity se divise en plusieurs fenêtres. La scène permet de voir visuellement en 3D ou en 2D, ce que nous sommes en train de développer. Tous les éléments importés dans la scène sont classés dans l'onglet *hierarchy*, avant de pouvoir les importer dans notre *scene*. Les objets ou les images qui font parti de l'application doivent être placés dans la fenêtre *project*. Cette fenêtre est une visualisation des dossiers composants le projet global. Il est possible de voir cette hiérarchisation dans les dossiers ordinateur de notre projet dans son *asset*. Une fois que nous voulons essayer notre application, nous pouvons voir le résultat dans la fenêtre *game*. Dans *inspector*, nous avons la fiche descriptive d'un objet. Il s'agit du regroupement de l'ensemble des interactions possibles sur un objet du projet.

Il est possible de faire énormément d'interactions graphiques avec Unity<sup>[46]</sup>, mais il y arrive un moment où nous ne pouvons pas nous passer de la programmation en code. Cette phase permet d'améliorer l'expérience de notre jeu ou de notre application. La plupart des interactions programmées reprennent les fonctionnalités de l'onglet inspector.

# PROGRAMMATION

---

## Les langages

Afin d'atteindre l'ensemble des possibilités du logiciel Unity, il est indispensable de passer par la programmation. Deux langages de programmation s'offrent à nous : J# (*J-Sharp*) et C# (*C-Sharp*). Il s'agit de deux dérivés du langage Java et du langage C, connus de tous les programmeurs. Le langage C d'origine est de bas niveau (c'est-à-dire qu'il est plus proche de la machine) et le langage Java est orienté objet (plus simple d'utilisation). Microsoft les a repris et simplifiés pour être de plus haut niveau (c'est-à-dire, plus proche de l'utilisateur). Ce qui fait d'eux des langages accessibles même pour des débutants. C'est un avantage pour les outils qui utilisent ces langages car il permettent aux utilisateurs expérimentés, ou non, de faire avancer à leur manière une nouvelle technologie. Lorsque Microsoft lance l'HoloLens, il ouvre ce casque au public et donne les outils pour son développement. Nous utiliserons le langage C# dans l'ensemble de ce document.

## L'apprentissage du langage

Avec quelques notions de programmation en langage C et C++. L'apprentissage du C# n'est pas insurmontable. De plus, il existe de nombreux tutoriels accessibles gratuitement sur internet<sup>[47]</sup>. Ils reprennent les bases de la programmation et certains vont un peu plus loin. Une fois que les bases sont acquises, nous pouvons facilement entrer dans la programmation proprement dite. Unity est ouvert au public et a une documentation détaillée de toutes les fonctionnalités programmables dans les *scripts* : nous appelons cela l'API.

---

45 Chaîne Youtube Mark Sinnathamby : HoloLens Development using Unity and Visual Studio 2015 [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=yDm9tryDiIY>> [consulté le 20/02/2018]

46 Chaîne Youtube Thought Experiment : #HoloLens - Tutorial #1 - Getting Started [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=OYx4qIqi0oI>> [consulté le 12/04/2018]

47 HILAIRE Nicolas, Apprenez à développer en C# [en ligne]. Disponible sur <<https://openclassrooms.com/fr/courses/1526901-apprenez-a-developper-en-c>>. [consulté le 02/03/2018]

Malgré cette documentation poussée, il y a des limites lorsque nous entrons dans les nouvelles technologies tel que l'HoloLens. Sur Unity, nous devons aller dans la section XR (l'ensemble des réalités). A ce stade, il est rare de trouver des exemples de codes. Il y a la documentation explicative de la fonction mais sans suite approfondie. Ensuite pour améliorer nos connaissances sur des sujets spécifiques, il est possible de rechercher sur internet directement ce dont nous avons besoin et trouver des morceaux de programmes déjà faits. La résolution de problèmes peut aussi se faire via des forums. Le casque étant commercialisé depuis plusieurs années, la plupart des cas sont traités. Malgré tout, nous pouvons réussir à faire une application simple.

Malheureusement, il faut avoir des connaissances plus approfondies pour l'HoloLens. C'est un environnement très spécifique avec ses fonctions spécifiques. Celles-ci permettent de reconnaître l'environnement, les gestes ou encore les interactions. L'HoloLens a été vendu en grande partie aux entreprises. Les documentations sur le produit sont rares, mais ils en existent certaines comme des forums spécifiques ou des tutoriels, avec par exemple l'HoloToolkit.

## API

Une API<sup>[48]</sup> (*Application Programming Interface*) est l'interface de programmation de l'application. Il s'agit de montrer l'ensemble des fonctions et des classes utilisables dans la programmation d'une application. En ce qui concerne l'usage pour le casque, nous avons l'information de la fonction, avec l'explication de celle-ci.

Nous pouvons retrouver facilement des détails ou des exemples pour comprendre le programme. Généralement, nous avons cette fonctionnalité dans HoloToolkit, qu'il faut retrouver, afin de la comprendre. Lorsque nous avons peu de connaissances dans la programmation, il devient difficile de programmer, puis avec l'expérience, nous commençons à comprendre les subtilités de l'API et sa manière de fonctionner. Au final avec de la persévérance, les programmes sont écrits plus rapidement.

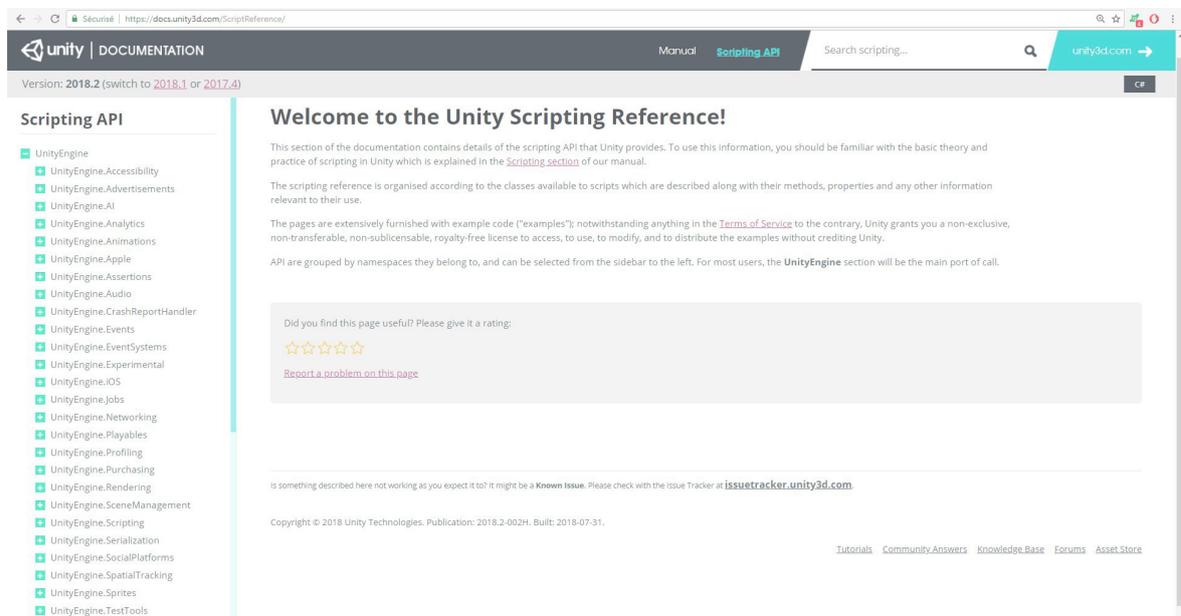


Fig.19 Page Internet de l'API de Unity . Source : <docs.unity3d.com>

## Le code

48 API de Unity [en ligne]. Disponible sur <https://docs.unity3d.com/ScriptReference/> [consulté le 12/04/2018]

Pour simplifier, le code est une liste d'instructions envoyées à l'ordinateur pour qu'il puisse savoir comment travailler. Les lignes sont lues les unes après les autres. Chacune impacte l'ensemble du code. Une fois que la totalité des codes sont lus, il s'agit d'une *Frame*. Plus les codes sont longs à exécuter, plus la *Frame* est longue et plus l'application sera lente. Il est important d'avoir un code simple et concis pour que l'application soit le plus fluide possible. Nous développerons cela par la suite.

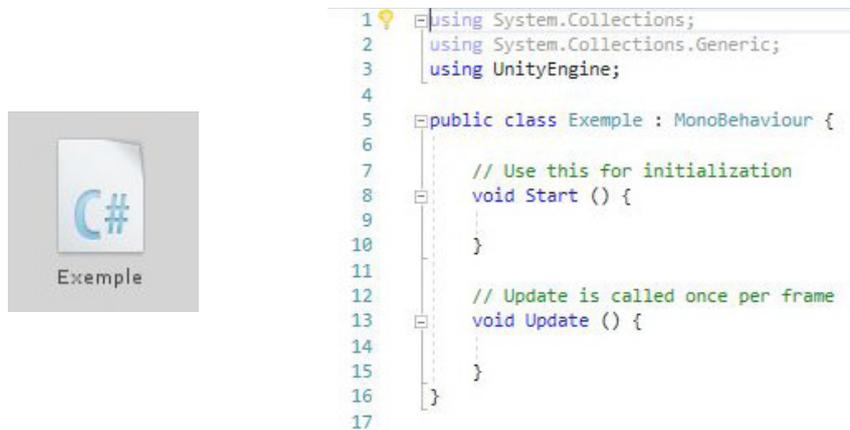


Fig.20 Icône du script "Exemple" dans Unity et son code par défaut

## Le fonctionnement d'un script

Un script commence toujours par l'importation des bibliothèques. Celles-ci sont définies par *Using*. C'est là que sont stockées les fonctions que l'on utilise dans le code. Dans l'API de Unity, il s'agit de la hiérarchie située sur la gauche. Notre script commence par une *Class*. Le nom est important car il doit reprendre le même que le script, afin que ce dernier fonctionne.

Par défaut nous avons les fonctions suivantes :

Start()

Elle permet lors d'un démarrage de lire le code qui se trouve entre accolades. Il est important de savoir qu'il n'est pas possible, sans éteindre l'application ou d'appeler la fonction, de lire une deuxième fois ce code automatiquement, même si nous désactivons puis réactivons le script.

La deuxième fonction présente est :

Update()

Il s'agit d'une boucle continue tant que le script est activé. Le code entre accolades reprendra au début à chaque *Frame*. Il faut faire attention à ce que nous mettons dans cette fonction, car elle peut vite engendrer une latence dans le programme.

Pour l'instant, nous n'allons pas rentrer dans les détails de la programmation. Il faut savoir qu'il existe des tutoriels qui expliquent le fonctionnement et restent accessibles. Nous allons commencer par quelques fonctions ou méthodes spécifiques à l'HoloLens dans Unity. Il existe une utilisation simple qui permet d'afficher le message entre parenthèse dans la console de Unity :

Debug.Log()



# **NIVEAU 03**

APPLICATION DU  
CRAI



## INTRODUCTION

---

Nous créerons, ici, une application pour l'HoloLens à partir de Unity. Nous allons voir l'implantation d'objets, en partant d'un cube interne à Unity avec un matériau uni jusqu'aux îlots de Verdun avec des textures.

Ensuite, nous allons voir comment interagir avec les hologrammes. Cette application intègre plusieurs fonctions qui permettent de les déplacer, les couper ou les intégrer dans l'environnement existant.

Enfin, nous ferons en sorte que les objets restent dans la position donnée, même en quittant l'application.



Avec le recul, il est important de penser au fonctionnement d'une application en avance, afin de comprendre la hiérarchie globale. Il faut se poser les bonnes questions : que se passe-t-il lorsque nous voulons une option ? Quelles interactions voulons nous avec les objets ? Comment les afficher ?

Ayant travaillé la première partie de mon stage en alternance, j'ai assimilé les connaissances sur Unity à ce moment là. Durant cette partie, l'application était une succession d'interactions, sans aucune cohérence globale. C'est pour cela que j'ai recommencé l'application afin de corriger les erreurs. Je n'avais pas les connaissances au début de la programmation. Il y a toujours une autre manière d'arriver au même résultat avec un code. Pour cela, aujourd'hui, avec plus d'expérience, je ne recommencerais pas l'application de la même manière.

Pourquoi ne pas avoir utilisé une application déjà existante du casque ? La plupart des applications ont un concept global et sont spécifiques à certaines utilisations. Le programme souhaité par le CRAI est un regroupement de plusieurs applications déjà existantes. Celui qui se rapprocherait le plus de ce qu'un architecte aurait besoin, c'est l'application Sketchup Viewer.

## L'EXPLICATION D'UNITY

---

### La préparation de l'application

Pour commencer un travail<sup>[49]</sup>, nous allons créer un nouveau fichier Unity, afin de concevoir une application pour l'HoloLens. Dans un premier temps, il faut changer quelques paramètres :

Dans < Edit / ProjectSetting / Player > puis < windows / XR Settings > sélectionner *Virtual Reality Supported*. Ensuite, il faut aller à < windows / Publishing Settings >, il est possible de changer le nom et la version de l'application. Ce qui nous intéresse est la dernière partie : *Capabilities*. Il faut sélectionner *InternetClient*, *WebCam* et *SpatialPerception*, pour que l'application fonctionne. Si nous avons une application qui fonctionne avec un appareil extérieur, il faudra activer, *Bluetooth* et si nous voulons utiliser la reconnaissance vocale, il faudrait activer, *Microphone*. Pour la qualité, < Edit / Project Setting / Quality >, il est conseillé de la mettre en *Fastest*.

### La vérification de l'application sur l'HoloLens

Avant toutes choses, il est important de savoir comment nous pouvons tester notre application. Afin de la vérifier, nous pouvons nous connecter en direct. Pour cela, il faut installer une application qui s'appelle *Holographic Remoting Player (HRP)* sur l'HoloLens. Il faut ajouter, la fenêtre *Holographic* en allant dans < Window / Holographic Emulation >. En se connectant sur Unity avec l'adresse IPv4, celle-ci s'affiche dans le casque lorsque nous lançons l'application *HRP*. Nous arrivons à voir le projet dans le casque quand nous lançons notre programme.

Il est possible de voir le projet sur un ordinateur avec l'émulateur HoloLens. Toutefois, il n'a pas été possible d'essayer cette fonctionnalité car l'installation ne marche pas sur le PC, probablement à cause de *Bootcamp*.

### Les caméras

Pour avoir une application fonctionnelle, il n'y a pas besoin d'une grande préparation et cela relève d'un peu plus de logique. Pour symboliser le casque HoloLens, il faut ajouter une caméra et la placer en (0,0,0) dans la scène. Utiliser d'autres coordonnées peut engendrer des problèmes, surtout sur la superposition du *Spatial Mapping* qui se décalera. Une chose à savoir : les écrans qui produisent les hologrammes, ne projettent que de la

---

49 Chaîne Youtube NDC Conferences : Developing Object Recognition for HoloLens - Stephen Carter [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=LxEl7DN5uRg>> [consulté le 12/04/2018]

couleur et du blanc. Le noir est produit en l'absence de couleurs. C'est pour cela que le *Background* doit être en noir afin que celui-ci devienne transparent lors de l'utilisation. Enfin pour avoir une visualisation plus réaliste de la perspective et se rapprocher du réel, il faut mettre le < *clipping Planes / Near* > à 0.85 (m) dans la caméra.

## La création de l'hologramme

Les hologrammes ne sont pas une chose que nous pouvons paramétrer. Il s'agit des objets virtuels importés dans Unity. Il suffit d'ajouter un objet en face de la caméra, en évitant de le mettre au même niveau que celle-ci. Afin de mieux reconnaître les formes des différents volumes importés dans la scène, il faut mettre de la lumière. Pour plus de réalisme, il est possible d'ajouter les textures sur les objets tout en gardant les caractéristiques maximales. En terme d'objets, il est possible d'ajouter des textes, des formes ou importer nos propres fichiers 3D. Il est possible d'afficher l'ensemble des éléments d'Unity. Néanmoins, il faut faire attention aux caractéristiques des objets pour que l'HoloLens ne sature pas.

## Lumière

Plusieurs possibilités s'offrent à nous. La première est de mettre une lumière dans l'environnement afin de produire une ombre et mieux déterminer les volumes.

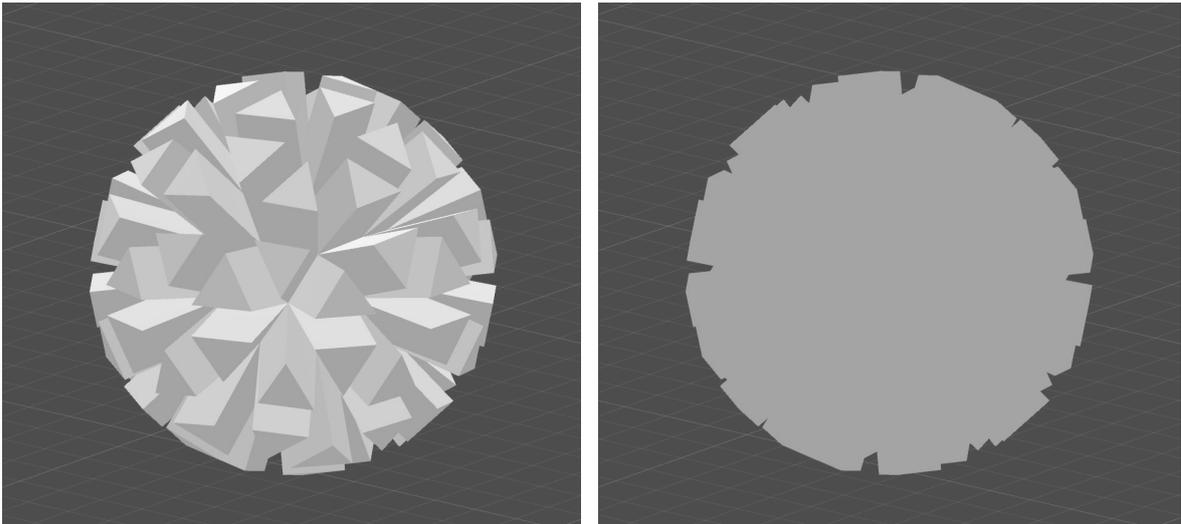


Fig.21 Objet avec et sans lumière. Source : Images personnelles de Unity

La seconde est de mettre une émission de couleurs sur les matériaux. Ainsi nous n'aurons pas d'ombres sur les objets. Cela peut être pratique dans les menus afin de ne pas avoir de changements de couleurs lors du déplacement de celui-ci.

La dernière possibilité est de mettre en supplément une *Directional Light*, dans la fenêtre < *Lighting* >, < *Environment Lighting / Source Color / Ambient Color* > mettre une couleur proche du blanc et < *Ambient Mode* > en *Realtime*.

## Le Canvas

C'est un plan qui se place devant la caméra, utile pour les menus ou pour afficher des informations, tel que du texte. Il permet d'interagir avec l'utilisateur. Ce *Canvas* sera le lieu de visualisation de l'augmentation de la réalité. Il est important de mettre le plan *Canvas* à 0.8m de distance afin de ne pas loucher.

Afin d'attacher un *Canvas* sur la caméra, il faut insérer la caméra dans le *Canvas* à < *Canvas / Render Mode / Render Camera* >. Ensuite il faut mettre < *Canvas Scaler / UI Scale Mode* > sur *Scale With Screen Size*. Il

faut faire attention pour l'utilisation d'un *Canvas* avec l'HoloLens. Lorsque nous plaçons les éléments sur celui-ci, les objets placés dessus suivent le regard et deviennent non sélectionnables. Ils bougeront toujours en même temps que le curseur central de vision. C'est pour cela que les menus de l'HoloLens ne sont pas sur le Canvas. Cela permet une interaction plus facile.

Les éléments placés sur le *Canvas* peuvent changer de dimensions entre le moment où nous testons l'application sur Unity et le moment où nous l'installons sur l'HoloLens. Lors de l'enregistrement sur le WDP d'une vidéo, nous constatons cette modification de taille.

Nous avons eu l'occasion d'expérimenter plusieurs couleurs sur ce *Canvas* : le rouge, le vert, le bleu, le noir, le blanc et le gris. Nous pouvons confirmer que le noir devient transparent. Le gris tend vers la transparence. Le bleu laisse entrevoir ce qu'il se passe à l'arrière. Le vert laisse un peu moins passer la lumière. Le rouge et le blanc sont les couleurs qui marchent le mieux. Cependant s'il y a trop de lumière dans la pièce, nous ne voyons plus grand chose. Regarder une fenêtre rend l'expérience illisible. D'ailleurs, lorsque nous scannons et récupérons la 3D de l'espace qui nous entoure, le vitrage n'apparaît pas.

Pour finir, nous pouvons noter que pour une plus grande immersion, un fond avec plusieurs couleurs et des mouvements, fonctionne mieux.

## Les gestes<sup>[50]</sup>

Pour reconnaître les gestes que nous pouvons faire avec l'HoloLens, il faut obligatoirement passer par un script. Ce script spécifique est expliqué dans son intégralité dans la dernière partie de ce mémoire, lorsque nous aborderons le Code.

## Le viseur/Gaze<sup>[51]</sup>

La création de ce viseur permet de visualiser le point central de l'écran que nous avons dans le monde virtuel. Sans celui-ci, il n'est pas possible de se repérer et il devient difficile de savoir avec précision ce que nous regardons. Il s'agit d'un objet 3D, un tore (un peu aplati), qui suit la normale de la caméra. Celui-ci, avec l'aide d'un *script*, se placera sur la face que l'utilisateur vise. Ce *script* lui permet de détecter les surfaces et la direction du casque, afin qu'il interagisse en même temps que nos mouvements de tête.

Pour plus de compréhension, ce curseur change de couleur en fonction du geste réalisé.

## Le Spatial Mapping<sup>[52]</sup>

Nous avons beaucoup parlé de l'environnement qui entoure l'utilisateur. Nous allons voir comment le mettre en place dans notre application. Dans un premier temps, nous allons le récupérer en créant un *empty* qui sera utilisé comme lieu de stockage. Pour cela, il faut ajouter deux composants sur cet *empty* : un *Spatial Mapping Renderer* et un *Spatial Mapping Collider*. Le *Renderer* est là pour afficher le maillage produit par l'HoloLens. Il est possible de modifier la précision. Il n'est pas conseillé de le changer car l'application sera plus lente. Nous pouvons toutefois changer le rayon de calcul de cette visualisation ou modifier le temps de rafraîchissement. Tout en bas de cette section, il est possible de changer les matériaux du maillage. Le *Spatial Mapping Occlusion* permet de rendre le maillage noir. Ainsi, les objets présents, passeront devant les hologrammes si c'est réellement le cas, et le maillage ne se verra pas car il est noir. Le *Collider* permet d'avoir un environnement solide afin que les objets ne passent pas au travers. Les mêmes paramètres sont présents.

---

50 MICROSOFT, MR Input 211: Gesture [en ligne]. Disponible sur < <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-211> > [consulté le 22/03/2018]

51 MICROSOFT, MR Input 210: Gaze [en ligne]. Disponible sur < <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-210> > [consulté le 22/03/2018]

52 MICROSOFT, MR Spatial 230: Spatial mapping [en ligne]. Disponible sur < <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-230> > [consulté le 22/03/2018]

## L'ancrage

L'environnement est toujours scanné lors de l'allumage de l'HoloLens. Il est possible ou non de l'afficher lorsque nous utilisons l'application. Néanmoins, pour que les éléments virtuels se positionnent dans l'environnement et qu'ils restent là où nous les posons, il faut créer des ancrages. Ces *Anchor* sont des points de coordonnées locaux, par rapport à l'environnement, qui sont sauvegardés sur les objets importés. Si nous déplaçons l'objet, son ancien point d'ancrage doit être, dans un premier temps, supprimé afin de le remplacer par le nouveau point. Cette fonctionnalité permet de pouvoir retrouver les éléments virtuels à la position enregistrée à chaque fois que l'on revient dans l'application.

Il faut que tous les noms d'objets soit uniques, afin d'y assigner un point d'ancrage spécifique. Ces points d'ancrages sont stockés dans un *Store* et s'enregistrent lorsque nous quittons l'application. Pour que les éléments de l'environnement reprennent la position que nous leur avons donnée précédemment, il faut recharger le *Store* au démarrage de l'application. Les points d'ancrages se mettent sur les objets et permettent d'avoir un repère que l'HoloLens devra replacer exactement par rapport à l'environnement. Nous pouvons tester cette fonctionnalité simplement dans Unity, en relançant l'application après l'avoir éteinte. Il n'y a pas besoin d'installer l'application sur le casque pour constater les effets des ancrages.

## HOLOTOOLKIT

---

Afin de faciliter certains codes ou d'importer des objets, une bibliothèque a déjà été faite pour l'HoloLens sur Unity. HoloToolkit reprend les codes des tutoriels Microsoft. Nous retrouvons un des outils pour le curseur, pour la maîtrise des gestes ou encore l'environnement. Avec des exemples, des explications du code, l'HoloToolkit permet de développer toutes les fonctionnalités de l'HoloLens facilement lorsque nous avons des notions de programmation. Les interactions entre Unity et l'HoloLens ne peuvent se faire que par la programmation en C#.

## VUFORIA

---

Vuforia est une application qui permet de reconnaître des objets, des images, des cylindres, des boîtes ou des textes. Une fois que l'élément est reconnu, l'application peut interagir avec lui. Nous avons la position de l'objet reconnu et pouvons ainsi lier un objet 3D à une image. Par la suite, nous pouvons donner différentes informations sur cet objet et les afficher. L'avantage de Vuforia, est qu'il s'intègre dans Unity et offre de nombreuses possibilités, comme le propose l'AR Controller<sup>[53]</sup>. Il s'agit d'un papier avec un dessin reconnaissable par l'application. Celui-ci affiche deux boutons. En les cachant avec le doigt, le geste est reconnu comme un clic.

Cependant, pour avoir une application utilisable, il faut avoir une clef d'activation. Puis il faut créer le fichier pour que l'HoloLens puisse reconnaître l'objet ou l'image. Pour chaque nouvel objet à reconnaître, il faut le paramétrer à nouveau sur le site de Vuforia. Pour un résultat plus simple, il faudrait pouvoir créer l'interaction à partir d'éléments développés dans Unity. Pour les objets complexes, au lieu de reconnaître l'ensemble des composants, nous pouvons prendre qu'un seul marqueur que nous réutilisons pour toutes les utilisations.

Toutefois, il y a trois contraintes supplémentaires dans l'utilisation de l'application :

La première est qu'il faut installer<sup>[54]</sup> l'ensemble de Vuforia dans notre programme, ce qui l'alourdit. Il

---

53 Chaîne Youtube Kostas Pap : Vuforia AR Controller [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=020UtR57uP8>> [consulté le 10/05/2018]

54 API de Unity : Vuforia [en ligne]. Disponible sur <<https://docs.unity3d.com/Manual/vuforia-sdk-overview.html>> [consulté le

existe l'API<sup>[55]</sup> pour utiliser l'application.

La deuxième est le ralentissement de l'application car elle doit scanner les images reçues par la caméra pour détecter les objets. Et enfin, il est possible d'avoir des décalages dans les objets virtuels superposés sur les objets reconnus.

## L'installation de l'application

Pour créer l'application, aller dans < File / Built Setting >. Dans un premier temps, il faut insérer les Scènes dans *Scenes In Build* en les glissant. Puis, sélectionner *Windows Universal Platform*, nous pouvons cliquer sur le bouton *Switch Platform* afin que cette plateforme soit par défaut. Il faut, ensuite, changer quelques paramètres.

*Target Device* : HoloLens.

*Build Type* : XAML

*Visual Studio* : Version Latest installed

*Build and Run on* : Local Machine

*Unity C# Projects* : Cocher

Après le chargement du *build* dans Unity, il faut ouvrir le fichier *Microsoft Visual Studio Solution* (un fichier \*.sln) qui se trouve dans le dossier désigné pour le *build*. Une fois ouvert, dans la hiérarchie, nous cliquons droit sur le projet et faisons < Store / Créer des packages d'application... >. La création d'un *package* permet de pouvoir importer et installer l'application sur l'HoloLens. Le format souhaité pour l'application HoloLens est \*.appx avec l'ajout des dépendances. Suivre les instructions et créer le *package*. Il est possible de créer, ou non, une application disponible directement sur le *Windows Store*. Nous n'avons pas besoin que notre application soit téléchargeable sur le *Windows Store*. Pour rappel, l'architecture de l'HoloLens est x86. Il faut donc créer le *package* d'installation en x86. Une fois créé, aller sur le *Windows Device Portal* de l'HoloLens.

Pour installer l'application sur le casque, nous allons sur < App / Add / Installed App > :

- cocher "Install app package from local storage"
- glisser le fichier que l'on trouve dans < projet2 / APPLI / appPackage > dans la dernière version (Nom du projet \_ version\_ bebug\_test)
- prendre le fichier \*.appxbun

Pour installer les dépendances :

- cocher "I want to specify optional package "
- cliquer sur "Next"
- glisser les deux dépendances dans x86 en \*.appx
- cliquer "Next" puis installer

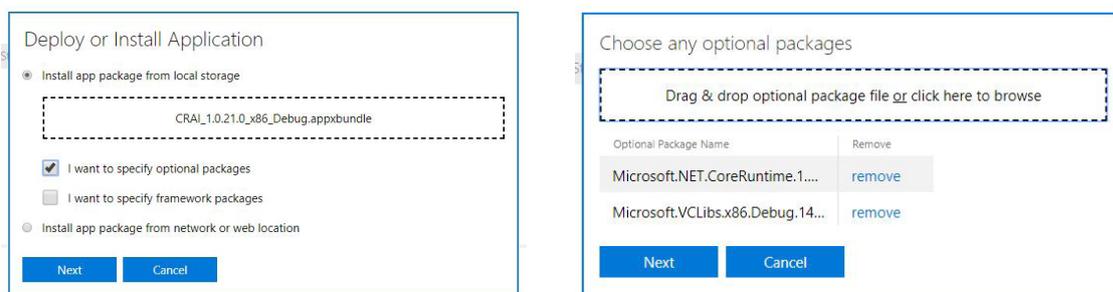


Fig.22 Différentes instructions pour installer l'application, Captures d'écran personnelles

16/03/2018]

55 ODOM Jason, How to Install & Set Up Vuforia in Unity [en ligne]. Disponible sur <<https://hololens.reality.news/how-to/hololens-dev-102-install-set-up-vuforia-unity-0175898/>> [consulté le 16/03/2018]

## La mise à jour de la bibliothèque

Afin d'importer de nouveaux éléments pour l'application, il faut les ajouter sur Unity puis recompiler l'application et l'installer sur l'HoloLens. Pour que l'application fonctionne, il faut appliquer quelques règles au niveau de la hiérarchie des objets importés. Dans un premier temps, tout nouvel objet doit être mis dans l'objet "ObjetSupprime"

Créez un objet3D/cube et nommez le. Son nom sera visible lorsque nous importerons l'objet dans l'application. Si un objet a déjà une hiérarchie similaire, il n'est pas nécessaire d'ajouter d'éléments ou de les remplacer. Néanmoins il faut ajouter des *collider* et des *renderer* sur l'ensemble des objets importés.

Lorsque nous glissons les objets dans le cube, chacun représentera une couche. Il faut savoir que le première objet sera la couche la plus basse, et le dernier la couche la plus haute. Il est possible de désactiver des *collider* sur les objets dont nous ne voulons pas voir le nom dans le mode visite. Pour Verdun, le sous fichier "ilot 41" doit avoir un *collider* désactivé par défaut. Nous pouvons ajouter des points de positionnement initiaux.

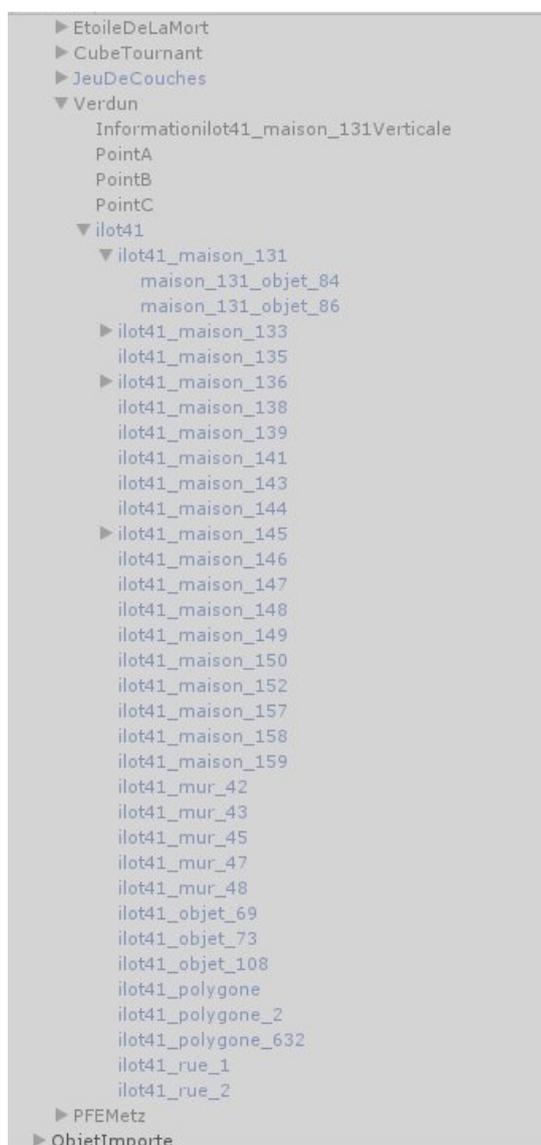


Fig.23 Hiérarchie de l'objet Verdun

# FONCTIONNALITÉS

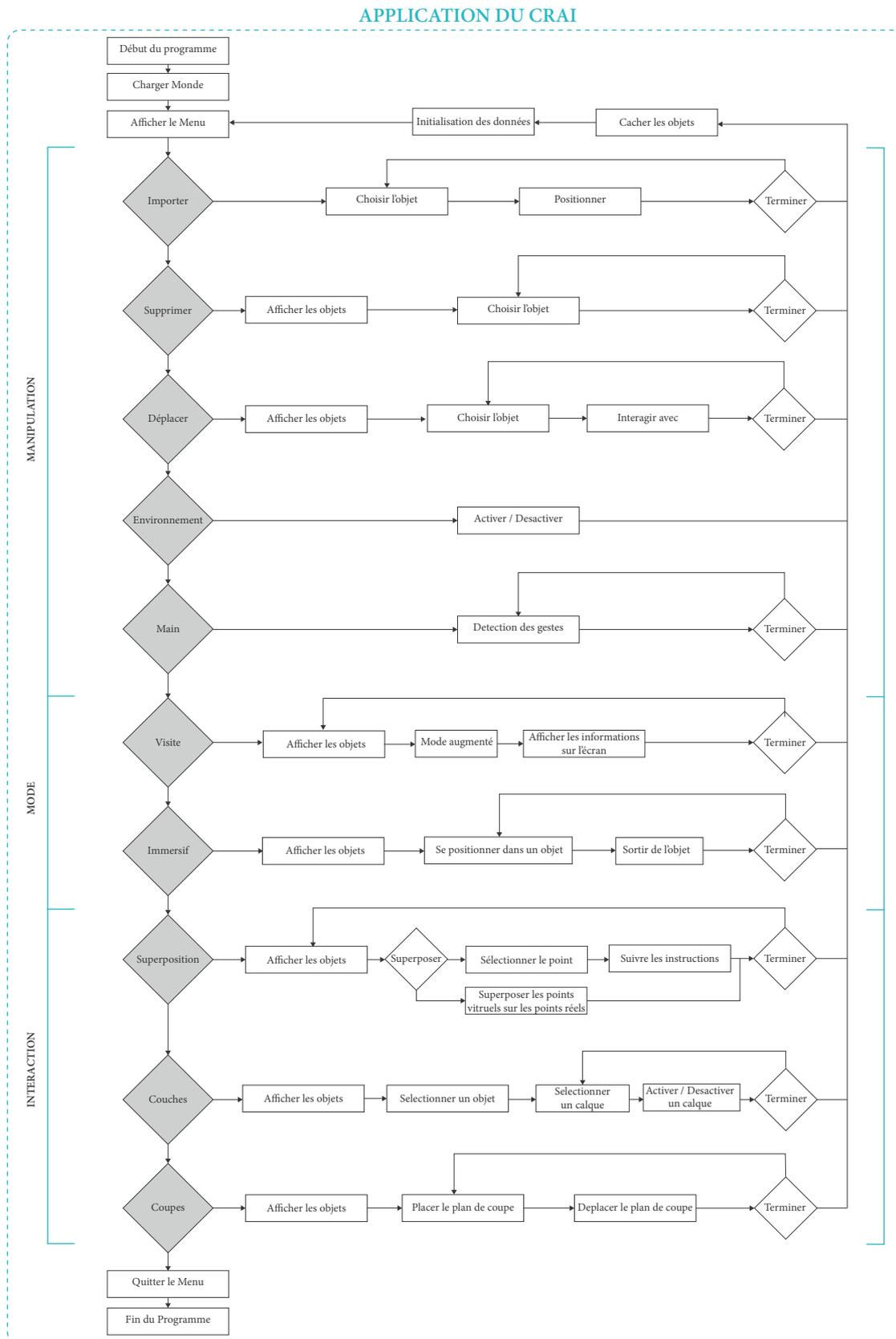


Fig.24 Logigramme du fonctionnement de l'application. Schéma personnel

## La connaissance des besoins

Afin de déterminer le menu, il faut dans un premier temps connaître les besoins de l'application. Nous les avons déterminés dans l'introduction. Nous voulons pouvoir importer un objet dans un volume existant, spécifiquement, le CRAI. Ces objets peuvent être internes mais cachés, et les importer dans l'environnement lorsque l'utilisateur le souhaite. Il est possible d'avoir un lien avec OneDrive<sup>[56]</sup> afin d'aller chercher tous les nouveaux objets créés au sein du laboratoire. C'est une méthode plus complexe à mettre en œuvre mais qui offre une plus grande flexibilité de l'application. Elle reste optionnelle. Nous pouvons importer les objets au préalable. Une fois cette manipulation réalisée, il sera possible de modifier leur position, leur taille ou leur rotation. L'intérêt de travailler avec l'HoloLens est de pouvoir interagir avec les éléments présents mais aussi avec l'environnement. Nous pourrions poser un objet que l'on a importé sur une table réelle. Ce qui est plus intéressant avec le casque de Réalité Mixte, c'est de pouvoir augmenter la réalité. Il est possible d'avoir des informations soit directement sur l'environnement du CRAI, soit sur les éléments que l'utilisateur a posés auparavant. Les informations peuvent s'afficher sur une partie de l'écran, ou nous rediriger vers une page web. Une dernière partie permettrait d'offrir une démonstration de Réalité Mixte et d'interaction avec la table tactile. Un projet présent sur cette table pourrait envoyer des informations au casque pour qu'il retransmette le même projet en 3D.

## QUELQUES DIFFICULTÉS

---

Maintenant que nous avons nos besoins, il faut les mettre en œuvre. Quelques difficultés surviennent sur certaines fonctionnalités de l'application.

### L'optimisation de l'application

Un des problèmes qui est venu rapidement lors de la création de l'application est la latence. Lorsque nous lançons l'application sur l'HoloLens, en affichant à la fois le menu, le modèle 3D de la ville de Verdun et l'environnement, l'application est ralentie. Afin de connaître la source de ce ralentissement, il faut fonctionner par étapes.

Dans un premier temps, nous lançons l'application avec les éléments séparés. Nous essayons avec seulement le menu, puis seulement avec le modèle 3D et enfin, seulement l'environnement. Dans les trois cas, l'application est plus fluide et permet une rapidité d'exécution raisonnable.

Une deuxième étape serait de simplifier les *scripts* afin de voir si nous constatons une amélioration. Les *scripts* fonctionnent en permanence lorsqu'ils s'exécutent et surtout pour la partie *Update*. Cette fonction s'exécute à chaque *Frame*. Cela consiste à un rafraîchissement de l'application plusieurs fois par seconde. Le fait de bloquer ces portions de *script* permet encore une meilleure fluidité.

Dans un troisième temps, il faudra simplifier l'ensemble des *Updates* pour que l'application puisse fonctionner et rester fluide. Néanmoins, l'application, avec un menu dans sa forme seulement graphique et la ville de Verdun, n'est pas optimale. Cela peut venir de plusieurs choses. Nous pouvons laisser de côté l'interface du menu car il n'y a qu'une vingtaine de composants. La source du ralentissement graphique vient forcément du modèle de la ville modélisée en 3D. Cet objet contient plus de huit cent composants et fait saturer l'HoloLens. Même dans une simplicité des volumes avec plus ou moins une dizaine de faces par composant, le nombre est beaucoup trop grand.

---

56 Chaîne Youtube Yining Liang : Hololens Load Data From OneDrive [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=ReE-PetXVh0>> [consulté le 16/03/2018]

Pour résoudre ces problèmes, la première idée serait de simplifier et de modifier les paramètres des matériaux sur ces volumes. C'est possible en allégeant les textures ou en changeant le type de matériau, ce qui n'a pas d'incidence. Dans Unity, il est possible de paramétrer les matériaux pour qu'ils soient plus adaptés aux applications mobiles. Pour le reste de l'application, nous n'utiliserons qu'un ou deux îlots de Verdun afin d'avoir une application plus fluide.

Dans cette partie du programme, il y avait Vuforia qui était installée. Le supprimer rend l'application plus fluide. Ceci est dû au fait que Vuforia doit scanner les images reçues pour reconnaître les objets.

Lors de la reprise du code complet en simplifiant les *Updates*, ainsi que les objets importés, l'application est plus fluide. Le plus grand changement dans cette nouvelle version de l'application, est d'avoir fait en sorte qu'il n'y ait qu'un *script* qui fonctionne à la fois. Un deuxième *script* fonctionnera tout le temps pour la reconnaissance des gestes. Le fait que le menu s'affiche uniquement lorsque c'est nécessaire permet d'optimiser l'application.

Aujourd'hui l'application fonctionne sans être optimale, mais en vue du nombre de *script*, l'application tourne autour des 60 FPS et reste agréable d'utilisation.

## Une superposition

Pour répondre aux contraintes relevées précédemment sur la précision du casque, il faut se poser la question : Comment pouvons-nous lier le réel au virtuel avec précision ?

Pour cela l'expérimentation de la superposition apporte une solution. Dans un premier temps, le but est d'avoir un outil permettant de trouver un point réel ou virtuel avec précision.

Nous pourrions dire que l'HoloLens avait une solution. Il est dommage qu'il ne scanne pas avec grande précision, surtout pour des cas spécifiques. Il n'est pas possible de reconnaître une maquette d'architecture, de déterminer une forme et de la reconnaître par le calcul des faces. Mettre cette technologie accessible sur un projet d'architecture dans l'HoloLens, pourrait même donner la possibilité de concevoir un projet complet avec le casque.

A ce moment là, plusieurs possibilités s'offrent à nous. La première est d'utiliser l'extension Vuforia. Lorsqu'un élément, image ou objet, est reconnu par l'extension, il est possible de récupérer la position de cet élément. Une fois le point défini, il est possible d'en faire ce nous voulons. Cette technologie apporte deux contraintes. Si nous voulons récupérer un objet qui "flotte", c'est à dire qu'il ne touche pas de table, ni de mur, il va être difficile de récupérer un point spécifique. Dans un second temps, comme dit précédemment, Vuforia est gourmand en ressources, car il faut analyser toutes les images d'entrée de caméra pour cibler l'élément à reconnaître. De plus, avoir une application qui fonctionne sans un apport extérieur est plus pratique dans la gestion d'une modification. Avec différents essais, Vuforia peut décaler le point ciblé de quelques centimètres. Il faut essayer de trouver une autre solution.

Nous allons voir comment la superposition se fait. Il s'agit d'une procédure pour l'ensemble des interactions dans l'application pour la superposition du réel et du virtuel. L'utilisateur doit se positionner dans une direction et viser un point précis. Puis ce même point doit être de nouveau visé dans deux autres directions afin de valider la position. Il est nécessaires d'exécuter cette procédure pour trois points différents sur l'objet virtuel et réel. Ainsi les deux plans reconnus pourront se superposer. Il s'agit de la méthode la plus précise pour obtenir un relevé correct et utilisable sur n'importe quel objet.

## La mise en valeur d'un projet patrimonial

En ce qui concerne spécifiquement l'architecture, il est possible d'incruster la maquette d'un projet dans un environnement. Avoir plusieurs idées pour un projet d'architecture est courant en phase de croquis. Créer des volumes et importer différents modèles n'est pas difficile. Ainsi nous pouvons mettre toutes les possibilités du projet dans l'environnement existant et afficher les différents modèles pour voir son insertion.



# **NIVEAU 04**

**UTILISATION DE  
L'APPLICATION**



## INTRODUCTION

---

Dans ce chapitre, nous avons des fiches explicatives permettant de faire comprendre le fonctionnement de l'application. Comme nous connaissons les besoins de celle-ci, nous sommes capables de déterminer les moyens d'interactions avec les différents boutons.

Nous retrouverons des fiches explicatives pour le menu puis les options de celui-ci. A coté de la page d'utilisation de la fonction, nous avons une page regroupant la vision idéale d'utilisation, les points négatifs et la vision future des interactions possibles.

# LES FICHES DES OPTIONS

## Le Menu

### Fonctionnement de l'option

Le menu s'affiche dès que nous sommes dans l'application. A ce stade, rien d'autre n'est activé dans cette dernière. Par défaut, si nous faisons des mouvements de tête, le menu en 3D suit notre regard. Il viendra se positionner en face de nous, même si nous nous déplaçons dans l'espace. Une fois que le menu est placé, nous pouvons regarder les boutons afin de mettre le curseur à leur niveau. En cliquant, nous pouvons entrer dans une fonction. Nous allons détailler les différentes options dans la suite des fiches. Si nous quittons une de ces fonctionnalités, nous revenons au menu.

Nous avons le bouton Suivre qui est attaché au menu. Il l'impacte directement. Par défaut, il est activé. Si nous le désactivons, le menu ne nous suit plus lorsque nous nous déplaçons. Cependant il se tourne vers la caméra, afin que nous puissions, à tout moment, atteindre les boutons.

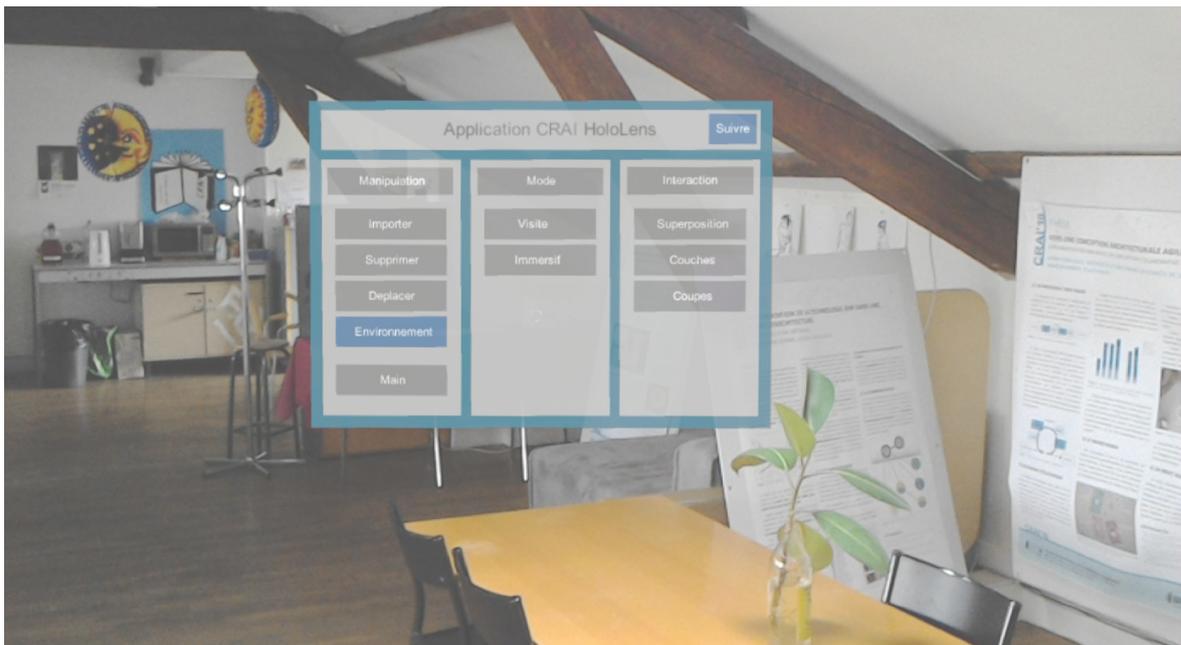


Fig.25 Le Menu dans son environnement

Interactions :

- Activer l'option Suivre ou Environnement (cliquer)
- Entrer dans une option (cliquer)

**Points positifs :**

- Déplacement du menu avec le regard.
- Fonctionnement fluide.
- Ensemble des boutons opérationnels.

**Points négatifs :**

- Non apparition des objets en permanence.
- Manque d'indication pour l'activation des boutons "suivre" et "environnement".

**Vision future de la fonction :**

- Apparition du menu en même temps que les objets.
- Commande vocale en français.
- Importer des gestes spéciaux.

## Importer

Ce bouton permet d'importer les objets de la bibliothèque de l'application. Cette bibliothèque est représentée dans Unity avec le *GameObject* : *ObjetSupprime*. Une liste d'objets apparaît à la place du menu, il est possible de faire défiler cette liste en cliquant sur les flèches du haut ou du bas. Les objets s'importent un par un. Une fois que nous avons sélectionné un élément, nous pouvons le placer dans le monde en cliquant pour valider la position.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.

## Supprimer

Une fois qu'un objet est importé, il faut pouvoir l'enlever. Pour cela, il faut cliquer sur un objet que l'on a mis dans l'environnement à l'aide d'un bouton "Supprimer". Cela permet d'alléger le fichier.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.



Fig.26 Sous menu "Importer" et exemple d'un ilot de Verdun

### Interactions Importer

- Sélectionner l'objet voulu (cliquer)
- Changement de page (cliquer)
- Placer l'objet sélectionné dans le monde (cliquer)

### Interactions Supprimer

- Sélectionner l'objet voulu (cliquer)

**Points positifs :**

- Ancrage automatique lors du positionnement de l'objet. Cela permet de retrouver l'objet quand nous retournons dans l'application.
- Possibilité de placer l'objet où l'on souhaite.
- La main gauche est utilisable.

**Points négatifs :**

- Aucune détection de la rotation de la main.
- Pas d'utilisation possible de OneDrive.
- Application plus lourde avec le fichier "ObjetSupprime".
- Boutons petits et difficile d'accès pour passer les pages du menu.

**Vision future de la fonction :**

- Importer les objets dans le monde initial de l'HoloLens.
- Pouvoir récupérer les éléments depuis OneDrive.

## Déplacer

Cette option regroupe plusieurs fonctionnalités. Elle reprend le principe de l'application HoloGrams. Nous pouvons faire la translation, la rotation, le changement d'échelle et le déplacement d'un objet. Lorsque nous entrons dans cette fonction, des cubes se dessinent autour de chaque objet présent dans l'environnement. Ces cubes représentent le squelette que nous pouvons transformer. En parcourant celui-ci, des formes apparaissent pour chaque transformation. En sélectionnant les faces, des plans s'affichent. Il est possible alors de faire une translation de l'objet en prenant la normale du plan. Lorsque nous visons une arête, un cylindre apparaît. Nous pouvons faire une rotation de l'objet sur lui même. Quand nous sommes sur un coin, des cubes apparaissent, nous pouvons alors changer l'échelle de l'objet. Enfin, si nous regardons l'objet en lui même, nous pouvons le déplacer en cliquant dessus.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.



Fig.27 Objets à déplacer dans le laboratoire

### Interactions :

- Translation d'un objet (Tenir : Gauche/Droite/Haut/Bas)
- Rotation d'un objet (Tenir : Gauche/Droite/Haut/Bas)
- Changement d'échelle d'un objet (Tenir : Avant/Arrière)
- Sélectionner un objet déplaçable (Cliquer)
- Placer l'objet sélectionné (Cliquer)
- Valider les interactions (Cliquer)

**Points positifs :**

- Ancrages qui se déplacent en même temps que les objets.

**Points négatifs :**

- Retour au menu nécessaire pour toute nouvelle interaction.

**Vision future de la fonction :**

- Pouvoir utiliser les deux mains pour interagir avec les objets.
- Utiliser la position de la main pour attraper un objet.

## Environnement

Si nous cliquons sur le bouton environnement, celui-ci permet d'afficher ou non le maillage qui nous entoure. Il faut cliquer sur le bouton pour qu'il se sélectionne ou désélectionne.



*Fig.28 Maquette virtuelle passant derrière les objets réels*

Interactions :

- Activer ou désactiver l'environnement (Cliquer)

**Points positifs :**

- Vision de l'environnement fiable.
- Interactions avec les objets que nous déplaçons.

**Points négatifs :**

- Les objets se positionnent sur l'environnement par rapport à leur centre.
- L'environnement se charge autour du point initial du lancement de l'application.

**Vision future de la fonction :**

- Faire en sorte que les objets ne passent pas à travers l'environnement.
- L'environnement se charge lors de déplacements.

## Mode Visite

Le mode "Visite" est fait pour être utilisé lorsque qu'une personne vient prendre connaissance du CRAI. Normalement, ce visiteur n'a pas d'expérience dans l'utilisation du casque. C'est pour cela que cette fonctionnalité est allégée. Il n'y a aucun geste spécifique à avoir mise à part, lever le doigt. Ce mode permet d'afficher, dans un premier temps, les objets importés dans l'environnement. Ensuite, lorsque nous visons un objet, son nom s'affiche. En ajoutant la main avec le doigt levé, nous pouvons voir des informations supplémentaires sur l'objet visé. Le curseur change de couleur si l'objet visé est attaché à une page d'informations.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.

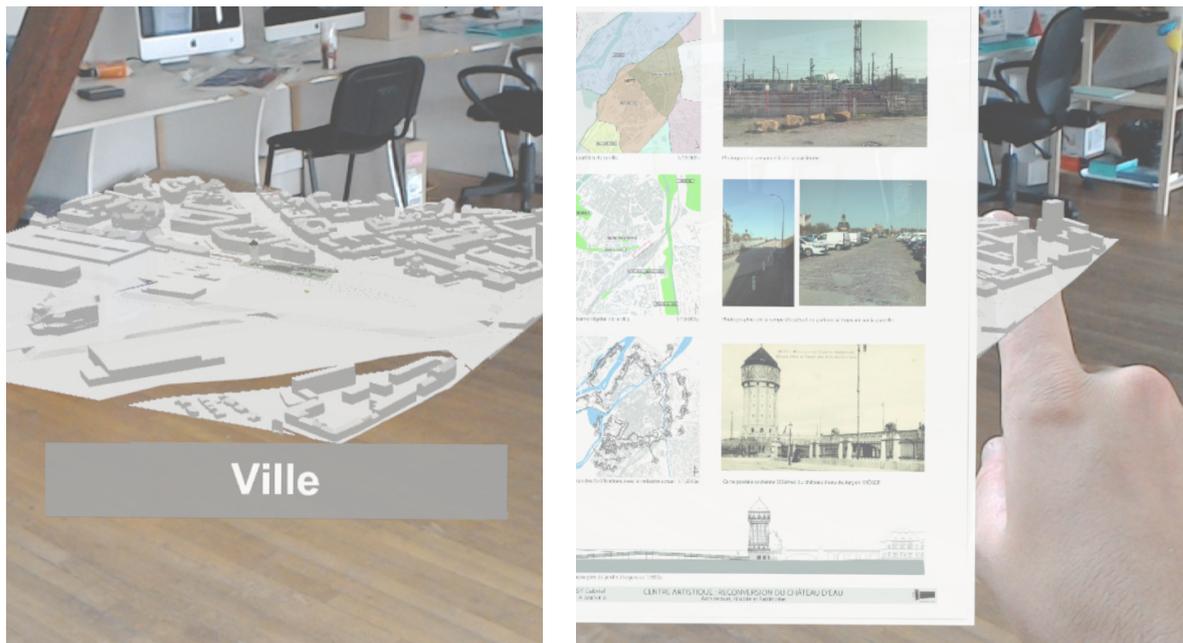


Fig.29 Annotation et panneau d'information liés à la maquette virtuelle.

Interactions :

- Afficher nom objet visé (Viser avec le curseur)
- Afficher les pages d'informations (Main avec index levé)

**Points positifs :**

- Permet de créer une exposition au CRAI.
- Simple d'utilisation pour les visiteurs.
- Affichage d'informations complémentaires.
- Changement de couleur du curseur si une page d'information est liée à l'objet visé.

**Points négatifs :**

- Aucun accès aux pages web des objets.

**Vision future de la fonction :**

- Augmenter le nombre d'interactions pour le visiteur.

## Mode Immersif

L'idée est de placer un point sur une maquette virtuelle et de projeter l'échelle 1:1 de cette maquette autour de nous par rapport au point. En vue des premières expériences avec la maquette de Verdun, il sera difficile d'importer une seconde maquette de la ville. Cela risque de ralentir considérablement l'application. De plus la maquette de Verdun n'apporte pas grand chose à grande échelle, car il n'y a pas de détails sur les façades. Ce qui est impressionnant sur cette maquette, c'est de pouvoir voir l'intégralité de la ville et de tourner autour.

Le mode immersif a également été testé avec le projet PFE de Metz. Celui-ci bénéficie de plus de détails et permet donc une immersion plus réaliste.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.



*Fig.30 Visualisation du mode immersif*

Interactions :

- Se positionner dans un objet 3D (Cliquer)
- Sortir du mode immersif (Cliquer)

**Points positifs :**

- Aucune latence car l'image est une texture appliquée sur un plan face à la caméra.
- Déplacement de la caméra en même temps que l'utilisateur.

**Points négatifs :**

- Ne prend pas en compte les dimensions des objets virtuels.
- Mode immersion non réaliste.
- Visualisation de l'environnement qui nous entoure.

**Vision future de la fonction :**

- Prise en compte de l'échelle de la maquette virtuelle.

## Superposition

Ce menu est expérimental. Il est une réponse possible à la contrainte de précision du casque.

Les trois premiers boutons permettent de positionner les points se trouvant sur des objets réels. Les trois suivants sont les points sur les objets virtuels. Il est possible de les positionner séparément et précisément à l'aide de la triangulation. Une fois ces six points placés dans l'espace, nous pouvons superposer les points virtuels sur les points réels. Deux boutons "Initialiser" permettent de remettre à zéro les coordonnées.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.

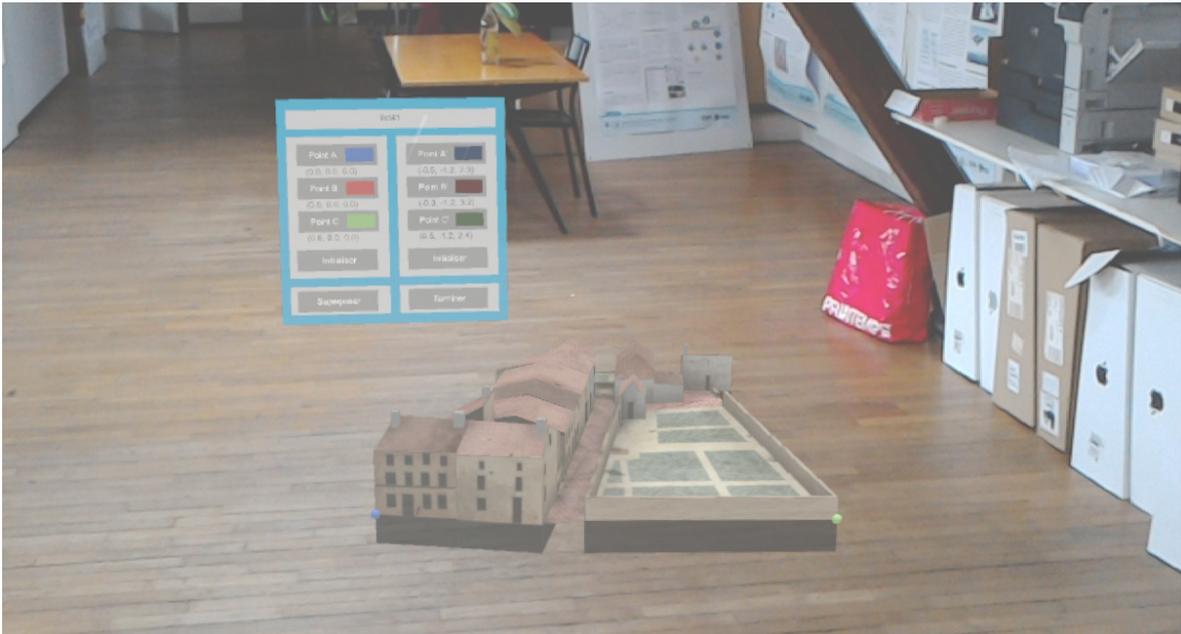


Fig.31 Superposition de la maquette de Verdun sur le sol

Interactions :

- Sélectionner un point que nous voulons placer (Cliquer)
- Initialiser les points réels ou les points virtuels (Cliquer)
- Superposer l'objet virtuel sur les marques réelles
- Quitter la fonctionnalité Superposition.

**Points positifs :**

- Possibilité de superposer un projet avec la table tactile.
- Possibilité de positionner des objets de toute sorte même si ceux-ci sont initialement mal placés.
- Résultat du positionnement des points précis.

**Points négatifs :**

- Manipulation longue.
- Superposition pas encore tout à fait au point.

**Vision future de la fonction :**

- Déplacer les marqueurs en un seul clic.
- Positionner les marqueurs rapidement.
- Améliorer la superposition.

## Couches

Cette fonctionnalité reprend l'idée des calques. Nous pouvons changer de page afin de visualiser l'ensemble des claques du projet. Il est possible de les enlever ou des les afficher en cliquant dessus avec un geste tenu de la main vers le haut. Il est possible d'éclater les différents calques afin de les mettre les uns au dessus des autres et voir la composition globale de l'objet.

Exploser le projet permet de montrer les différents couches. Il est aussi possible de voir les types d'éléments séparément et ainsi mettre en valeur ceux qui nous semblent les plus importants.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.

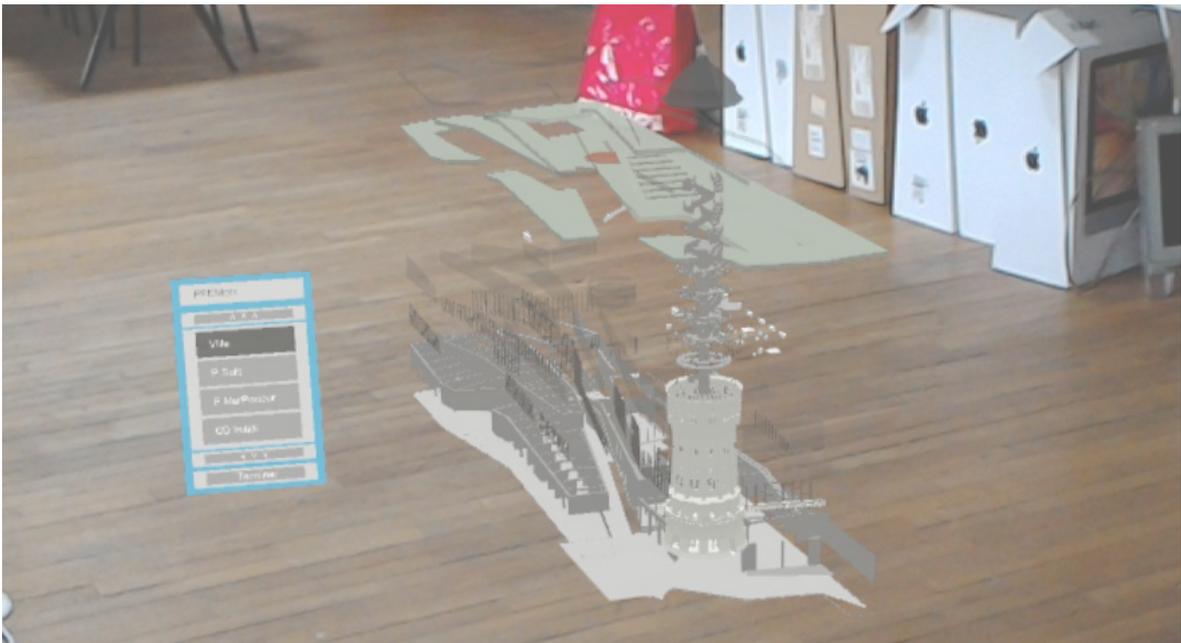


Fig.32 Maquette virtuelle éclatée

### Interactions :

- Changement de pages (Cliquer)
- Activer ou désactiver les calques (Cliquer)
- Éclater un objet en fonction de ces calques (Tenir : Haut/bas)

**Points positifs :**

- Bonne compréhension d'un projet
- Visualisation fonctionnelle

**Points négatifs :**

- Aucune remise à zéro de l'éclatement.
- Pas de remise à l'échelle de l'objet lors de la sélection des calques.
- Désélection des boutons lors du retour au menu.

**Vision future de la fonction :**

- Correction des points négatifs.

## Coupes

Dès que nous avons un objet présent dans le monde, nous pouvons en faire une coupe. Dans un premier temps, il faut cliquer sur la face d'un objet pour placer le plan de coupe et ensuite, nous pouvons le déplacer sur sa normale. Si nous cliquons sur un autre objet, le premier n'aura plus de coupe, c'est le second qui se coupera.

Il est possible de cliquer sur "Valider" pour sortir de ce mode.

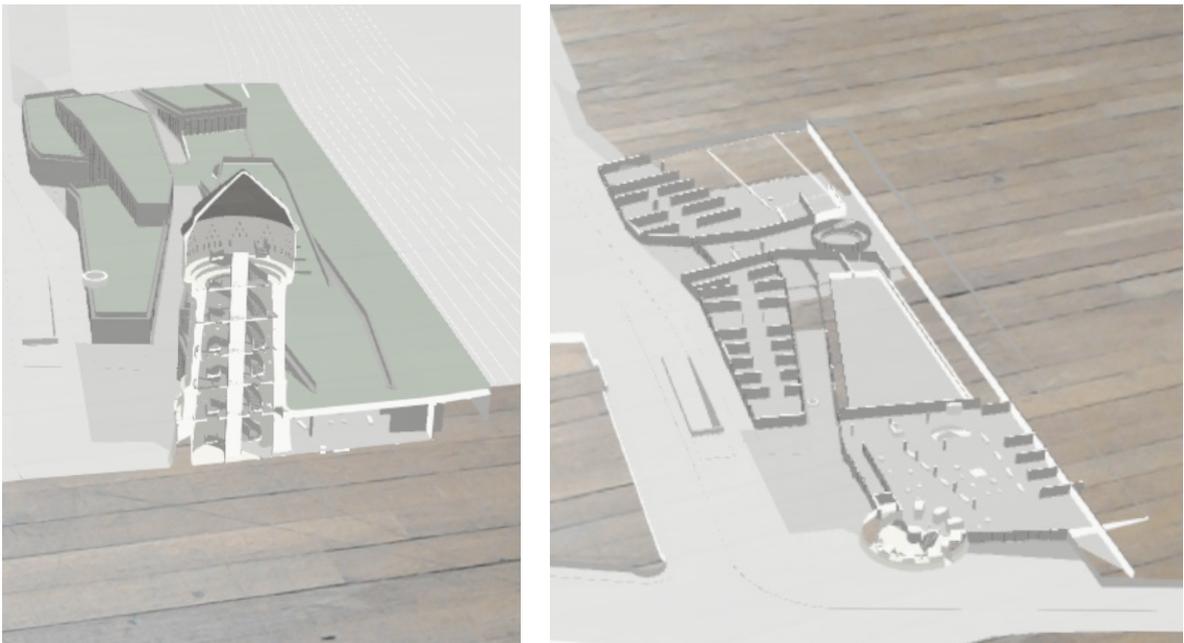


Fig.33 Coupes de la maquette virtuelle

Interactions :

- Poser le plan de coupe (Cliquer)
- Faire une translation du plan de coupe sur sa normale (Tenir : Gauche/Droite/Bas/Haut)
- Quitter la fonctionnalité (Cliquer)

**Points positifs :**

- Fonctionnement de l'application.

**Points négatifs :**

- Positionnement possible seulement sur les plans x,y,z.
- Avoir qu'un seul matériau pour que la coupe fonctionne.

**Vision future de la fonction :**

- Positionnement du plan de coupe sur toutes les surfaces.



# **NIVEAU 05**

**ENTRONS DANS LE  
CODE**



## INTRODUCTION

---

Dans cette partie, il est possible de comprendre l'intégralité de l'application. Nous retrouvons les *scripts* annotés de quatre options de l'application.

Pour commencer, il faut rappeler que je ne suis pas programmeur et j'ai déjà conscience que certaines parties peuvent être optimisées. L'application fonctionne dans l'état présent mais n'est qu'un avancement.

Les scripts présentés se trouvent dans l'annexe de ce mémoire.



## LA PROGRAMMATION DE L'APPLICATION

---

### Les Gestes

Le *script* est implémenté dans le *GameObject* au démarrage et fonctionne dans toute l'application. Il est normal que nous puissions utiliser les gestes à tout moment pour entrer ou sortir des menus. Tout ce qui concerne les gestes est implémenté dans la classe :

GestureRecognizer

Cela permet de récupérer les moments où la main apparaît dans le champs de vision des caméras. Le fonctionnement est classiquement le même pour chaque genre de gestes. Pour chaque geste il y a quatre phases, la première là où le geste commence (*start...*), le moment où le geste s'arrête (*completed...*), l'entre deux (*updates...*) et enfin le moment où le geste est annulé (*anceled...*).

Dans l'application :

Update()

Nous récupérons à tout moment les informations de la caméra, et plus précisément les objets que nous regardons. Ainsi nous pouvons appeler dans tous les *scripts*, le *GameObject* ou le *Name* de l'objet visé. Nous savons aussi si nous visons des objets ou non.

Awake()

Cette fonction n'est appelée qu'une seule fois au moment de l'initialisation du *script*, mais elle reste en veille pour que nous puissions en utiliser les fonctions. Nous remarquons qu'il y a un fonctionnement différent de ce qui est habituellement fait dans les fonctions. Lorsque nous mettons un `Debug.Log()` au début, et dans la fonction "clique", peut importe le nombre de fois que nous cliquerons, le deuxième `Debug` affichera un nombre de clics quand le premier n'en affichera qu'un seul. Cette fonction est appelée dès que tous les objets sont initialisés. Nous appelons à ce moment les fonctions que nous avons décrites précédemment : les fonctions *Manipulation* et *Navigation*. La fonction *Manipulation* est moins utilisée que la fonction *Navigation* car elle permet d'avoir l'ensemble des interactions possibles avec l'HoloLens.

La fonction *Manipulation* permet d'avoir d'autres informations. Celle-ci permet d'avoir la position de la main lorsqu'elle est dans le champs de vision du casque. C'est cette fonction qui n'a pas été facile à mettre en place, car les publications ne sont pas à jour et les parties de codes trouvables deviennent obsolètes. Toutefois, nous récupérons les coordonnées x, y et z de la main, par rapport à la position initiale de la caméra, au moment où nous avons fait entrer la main dans le cadre de vision. C'est-à-dire, au moment où la fonction est appelée :

OnManipulationStarted

## Le Menu

Le fichier menu est la partie centrale de l'application et permet de desservir l'ensemble des fonctionnalités. Premièrement, nous avons des variables en *static* qui peuvent être utilisées dans l'ensemble des *Script* du programme. Nous retrouvons : "Choix menu" qui permet de savoir quel menu est activé ou non, "initialisationVar" qui permet de dire ce qu'il faut réinitialiser dans la fonction. Ces deux variables sont en *static* pour permettre de les modifier depuis un autre script. Plus particulièrement la modifier avant de revenir dans le menu ou en quittant le menu. Ce qui permet d'avoir des éléments qui se désactivent en décalage par rapport avec l'événement d'un clique. Ensuite nous avons beaucoup de variables *GameObject* qui permettent d'avoir tous les objets dont nous aurons besoin. Nous aurions pu aller les chercher avec :

```
Find("nomObjet")
```

Cependant, les noms d'objets peuvent changer au fur et à mesure que l'application progresse. Cette méthode sera plus utile dans une utilisation future.

```
Start()
```

Nous avons cette fonction car, le *Script* est dans l'objet menu. Il fonctionne depuis le début du programme. Nous pouvons initialiser à cet instant. Nous commençons par créer un tableau qui reprend l'ensemble des boutons du menu. Cela permet d'avoir l'état d'un bouton en deuxième «colonne» et la première colonne est le nom du bouton. Nous initialisons l'ensemble des variables à cet instant.

```
Update()
```

Dans un premier temps, nous regardons les interactions avec les boutons. Nous prenons en compte le passage du curseur sur un bouton, afin qu'il se sélectionne. Puis, il se désélectionne lorsque nous ne le visons plus. C'est à cet instant que nous utilisons le tableau pour implémenter l'état de la deuxième colonne. Nous avons un *timer* afin d'avoir un menu qui paraît plus fluide. Ensuite, nous entamons le bouton "Suivre" et le bouton "Environnement". Ces boutons sont spécifiques. Ils ont seulement deux états, cliquer et décliquer.

Des difficultés m'ont poussé à sortir ce genre de boutons de la boucle. Lorsque nous cliquons, le clic ne fait pas qu'une seule *Frame*. Ceci engendre des problèmes car il est possible d'avoir un nombre pair de *Frame*, alors, le clique ne sert à rien. Un exemple aiderait à comprendre. Si nous cliquons une première fois sur le bouton, il s'active. Et, une seconde fois, il se désactivera. Par conséquent, si nous avons un nombre de *Frame* pair, cela veut dire que lorsque nous cliquons, le bouton s'enfoncé puis se désenfoncé instantanément. A ce moment, l'utilisateur ne se rend pas compte qu'il se passe quelque chose. Le fait de mettre un deuxième bouton identique et d'intervertir lorsque nous cliquons, permet de régler ce problème. Le temps est plus long entre le changement d'objets que lorsque nous faisons un *Air Tap*.

Comme dit dans les enjeux de l'application, elle doit être fluide et ne pas déranger durant son utilisation. Pour diminuer le nombre de calculs, chaque option fonctionne indépendamment. Le menu aussi. C'est pour cela que le menu n'est pas présent lorsque nous serons dans une fonctionnalité. Et lorsque nous quitterons cette fonctionnalité, son *script* sera désactivé pour laisser place à celui du menu.

Il existe trois sous menus qui peuvent être utilisés. Les deux prochains exemples montrent leur fonctionnement. Le premier qui est le sous menu "Validation" est utilisé dans "Coupe". Le second "Importe" est utilisé dans le menu "Importation". Il est expliqué dans la partie suivante et le dernier "Superposition".

## La coupe

Le sous menu "Validation" est réutilisé souvent dans les applications. Lorsque l'option est simple et n'a pas besoin d'un support visuel, il suffit de mettre un bouton pour sortir de cette fonctionnalité. A cet instant, nous utilisons le bouton "Terminer" du sous menu "Validation".

Dans la fonctionnalité "Coupe", l'utilisateur clique sur une face et déplace le trait de coupe. Tout est possible de faire sans un apport visuel supplémentaire. La problématique de cette fonctionnalité est de couper un objet. Pour cela il faut utiliser les *Shader*. Il faut créer le *Shader* du plan puis celui de l'objet. En simplifiant, le *Shader* demande de ne pas afficher la texture des objets lorsque le plan le coupe. Pour savoir où le plan se situe, il faut lui donner sa position et sa normale.

Pour en revenir au *Script*, on a l'interaction avec le bouton "terminer" qui permet de remettre les *Shader* à leur valeurs initiales. Ensuite, si on clique sur un objet, et qu'il appartient aux objets importés, nous pouvons changer le *Shader Standart* par le *Shader* spécifique qui permet de couper les objets par rapport au plan.

La variable "*allChildren*" permet d'avoir en mémoire tous les enfants de l'objet sélectionné pour pouvoir changer ses *Shader*. Puis une fois que tout est en place, nous pouvons appliquer le plan de coupe et le déplacer en fonction de la main.

Lorsque nous faisons le geste de la main tenue, et que nous bougeons de gauche à droite, nous translatons le plan sur sa normale avec une force. Dans la classe de "Gestes", nous récupérons une variable qui va de -1 à 1. Plus nous nous éloignons du point initial du geste, plus vite nous pourrons déplacer le plan.

## L'importation

### Initialisation()

Il faut connaître le nombre d'objets que nous pouvons importer à chaque fois que nous entrons dans cette fonctionnalité. Nous allons chercher tous les éléments qui ont été supprimés, c'est-à-dire ceux qui ne sont pas importés. Une fois que nous connaissons ce nombre d'objets, il faut que l'utilisateur puisse les voir. Pour cela le menu "Importation" s'affiche et calcule par sections de quatre objets. Il faut les afficher sur les quatre boutons du sous menu. Nous avons, ensuite, quelques exceptions : si le fichier "ObjetSupprime" ne contient pas un nombre d'objets correspondant à un multiple de quatre, alors il y aura des cases vides. Ensuite, il faut pouvoir aller dans les pages suivantes si le fichier "ObjetSupprime" contient plus de quatre objets. Il faut connaître le nombre de pages que nous pouvons tourner et savoir à quelle page nous sommes. Nous changeons de page quand nous cliquons sur "ObjetImportePrec" ou "ObjetImportSuiv".

Nous retrouvons ensuite, les quatre choix, lorsque nous cliquons sur un des boutons. Nous recherchons l'objet qui y est attaché et nous pouvons l'avoir dans la main pour le placer dans l'environnement. Une fois que nous cliquons avec un objet dans la main, il se place. Ensuite, nous pouvons en prendre un second et ainsi de suite jusqu'à ce que nous cliquons sur "Terminer".



# **CONCLUSION**



Pour conclure, la technologie est fonctionnelle pour un premier prototype. Il s'agit tout de même, d'une grande avancée. L'HoloLens est ouvert pour son développement d'applications. Il n'y a pas besoin de compétences poussées dans la programmation. Il reste accessible financièrement pour les entreprises car il peut devenir rentable pour celles-ci. Il est facile à prendre en mains après un temps de formation relativement court.

Cependant, j'ai constaté qu'il y avait quelques points à améliorer, tel le champs de vision, la reconnaissance des gestes, ainsi que la puissance de calculs.

En parallèle de ce mémoire, j'ai développé une application au CRAI. Celle-ci répond aux exigences du laboratoire et s'inscrit dans le développement de la thématique : "La visualisation et interaction dans les espaces virtuels". Il serait possible de reprendre ces recherches pour travailler sur les autres thèmes du CRAI, tel que le travail en collaboration. Dans ce cas précis, mon objectif était de faire un lien avec mon projet de fin d'étude, en valorisant un projet patrimonial.

Pour développer cette application, il a été nécessaire d'approfondir mes connaissances en programmation. Malgré les difficultés rencontrées, celle-ci m'a permis de montrer un panel d'options utilisables pour l'architecture: comme la possibilité de créer des coupes, changer d'échelle ou gérer différents calques. L'élaboration du programme m'a créé quelques soucis. Parmi ceux-ci, les ancrages ont été les plus complexes à gérer. L'application n'est pas, à ce jour, totalement fonctionnelle. Elle est malgré tout utilisable et les concepts clefs sont opérationnels. Le mode visite permet d'exposer les objets virtuels du CRAI. L'application s'ouvre donc, pour les visiteurs du laboratoire.

Afin d'améliorer l'application de manière générale, il serait nécessaire d'intégrer une plus grande variété d'objets dans la bibliothèque. N'ayant pas eu l'occasion d'approfondir l'importation, les éléments ne peuvent être téléchargés d'un fichier extérieur à l'application, comme OneDrive.

Il serait également intéressant de pouvoir se projeter dans l'utilisation du futur casque. Nous pourrions certainement travailler sur l'application, en temps réel, grâce à Blender ou Grasshopper. Il est certain que la technologie va évoluer. La popularisation de la technologie a commencé. Lorsque nous regardons les films, nous avons déjà une vision futuriste de l'HoloLens.

# LA BIBLIOGRAPHIE

## Ouvrages :

- CHATTERJEE Ayan, Building Apps for the Universal Windows Platform, Apress, Royaume-Unis, 2017
- KRESS Bernard C. and Cummings William J. , Towards the Ultimate Mixed Reality Experience : HoloLens Display Architecture Choices. Microsoft Corporation, Redmond, WA. USA, publié le 02 Juin 2017.
- MILGRAM Paul, Augmented Reality : A class of displays on the reality-virtuality continuum, Japon, 1995
- TAYLOR Allen, Develop Microsoft HoloLens Apps Now, Apress, Royaume-Uni, 2016

## Conférences et interviews :

- COPPENS Adrien, Parametric Modelling in Immersive Environments, conférence du 12 juillet 2018, ENCCA
- BERGOUNHOX Julien, [Interview] Comment Ford utilise la réalité virtuelle pour mieux concevoir ses voitures [en ligne]. Disponible sur <<https://www.usine-digitale.fr/article/interview-comment-ford-utilise-la-realite-virtuelle-pour-mieux-concevoir-ses-voitures.N562253>> [Consulté le 01/03/2018]
- KIPMAN Alex, A Futuristic Vision of the Age of Holograms [en ligne]. Disponible sur <[https://www.ted.com/talks/alex\\_kipman\\_the\\_dawn\\_of\\_the\\_age\\_of\\_holograms/discussion?la](https://www.ted.com/talks/alex_kipman_the_dawn_of_the_age_of_holograms/discussion?la)> [consulté le 20/02/2018]

## Publications en ligne :

- BHUSHAN JAGWANI Valendu, informaticien : What is the difference between virtual reality, augmented reality and mixed reality? [en ligne]. Disponible sur <<https://www.quora.com/What-is-the-difference-between-virtual-reality-augmented-reality-and-mixed-reality>> [consulté le 06/07/2018]
- CHALON René, Thèse sur la "Réalité mixte et travail collaboratif : IRVO, un modèle de l'interaction homme-machine" [en ligne]. Disponible sur <<https://tel.archives-ouvertes.fr/tel-00152230/document>> [consulté le 19/07/2018]
- HILAIRE Nicolas, Apprenez à développer en C# [en ligne]. Disponible sur <<https://openclassrooms.com/fr/courses/1526901-apprenez-a-developper-en-c>>. [consulté le 02/03/2018]
- LABBE Pierrick, Article du Magazine de la Réalité Virtuelle [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/realite-mixte-definition-exemples-0106>> [consulté le 20/02/2018]
- LABBE Pierrick, Article du Magazine de la Réalité Virtuelle [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/hologrammes-magic-leap-realite>> [consulté le 22/02/2018]
- LAMELOT Matthieu, Article d'un test HoloLens [en ligne]. Disponible sur <<https://www.tomshardware.fr/articles/test-microsoft-hololens,2-2602.html>> [consulté le 20/02/2018]
- L.BASTIEN : Microsoft HoloLens : solutions aux principaux bugs et problèmes [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/microsoft-hololens-problemes-bugs>> [consulté le 01/03/2018]
- ODOM Jason, How to Install & Set Up Vuforia in Unity [en ligne]. Disponible sur <<https://hololens.reality.news/how-to/hololens-dev-102-install-set-up-vuforia-unity-0175898/>> [consulté le 16/03/2018]

- ODOM Jason, 3D Live Tiles, Two-Handed Manipulation & More Finally Coming to the HoloLens 03/03/2018 [en ligne]. Disponible sur < <https://hololens.reality.news/news/dev-report-3d-live-tiles-two-handed-manipulation-more-are-finally-coming-hololens-0183241/>> [consulté le 08/03/2018]
- PECUCHET Bertrand, Rapport sur la kinect, Lille, 2013 [en ligne]. Disponible sur <<http://iut-gmp.univ-lille1.fr/fichiers/LPVI/RapportFinalkinect.pdf>> [consulté le 03/02/2018]
- PIROTTE Jeremy, Article : [Test] Microsoft Hololens : le casque de réalité augmentée de Microsoft [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/test-microsoft-hololens>> [consulté le 02/03/2018]
- RONFAUT Lucie, Article : J'ai testé l'HoloLens de Microsoft et le futur n'est pas pour tout de suite [en ligne]. Disponible sur <<http://www.lefigaro.fr/secteur/high-tech/2016/10/17/32001-20161017ART-FIG00001-j-ai-teste-l-hololens-de-microsoft-et-le-futur-n-est-pas-pour-tout-de-suite.php>> [consulté le 20/02/2018]
- RUBINO Daniel, Article : Here are the full processor, storage and RAM specs [en ligne]. Disponible sur <<https://www.windowscentral.com/microsoft-hololens-processor-storage-and-ram>> [consulté le 20/02/2018]
- SOREL François, 100% Windows : à quoi sert Hololens ? [en ligne]. Disponible sur <<https://www.01net.com/mediaplayer/video/100percent-windows-a-quoi-sert-hololens-950139.html>> [consulté le 01/03/2018]
- THOMSON Iain, Article : Hola HoloLens: Reg man gets face time with Microsoft's holographic headset [en ligne]. Disponible sur <[https://www.theregister.co.uk/2015/01/23/hands\\_on\\_with\\_the\\_hololens\\_or\\_how\\_facebook\\_wasted\\_2bn\\_on\\_oculus/](https://www.theregister.co.uk/2015/01/23/hands_on_with_the_hololens_or_how_facebook_wasted_2bn_on_oculus/)> [consulté le 22/02/2018]
- TURQUIER Amandine, Article du Magazine de la Réalité Virtuelle [en ligne]. Disponible sur <<https://www.realite-virtuelle.com/lhistoire-realite-virtuelle>> [consulté le 22/02/2018]

### **Sites et pages internet :**

- Site internet du Laboratoire de Recherche : CRAI [en ligne]. Disponible sur <<http://www.map.cnrs.fr/>> [consulté le 02/03/2018]
- Site internet du master AME [en ligne]. Disponible sur <<http://www.crai.archi.fr/wp-ame/>> [consulté le 15/09/2017]
- Site 8Ninths de l'Université de Valence, travaux sur la réalité augmentée [en ligne]. Disponible sur <<http://8ninth.com/>> [consulté le 22/02/2018]

### **Site internet de Microsoft :**

- MICROSOFT, Présentation de l'holoLens [en ligne]. Disponible sur <<https://www.microsoft.com/fr-fr/hololens>> [consulté le 20/02/2018]
- MICROSOFT, MR Spatial 230: Spatial mapping [en ligne]. Disponible sur < <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-230> > [consulté le 22/03/2018]
- MICROSOFT, Article : Building the Terminator Vision HUD in HoloLens [en ligne]. Disponible sur <<https://blogs.windows.com/buildingapps/2017/03/06/building-terminator-vision-hud-hololens/#HpM7CKKhT6khyrZH.97>> [consulté le 20/02/2018]
- MICROSOFT, Preview new mixed reality apps for HoloLens [en ligne]. Disponible sur <<https://docs.microsoft.com/en-us/HoloLens/hololens-public-preview-apps>> [consulté le 19/07/2018]
- MICROSOFT, MR Input 211: Gesture [en ligne]. Disponible sur < <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-211> > [consulté le 22/03/2018]- MICROSOFT, MR Input 210: Gaze [en ligne]. Disponible sur < <https://docs.microsoft.com/en-us/windows/mixed-reality/holograms-210> > [consulté le 22/03/2018]
- MICROSOFT, Spatial mapping [en ligne]. Disponible sur <<https://docs.microsoft.com/fr-fr/win->

dows/mixed-reality/spatial-mapping> [consulté le 19/07/2018]

- MICROSOFT, Page utilisation de la 3D de Microsoft [en ligne]. Disponible sur <<https://support.microsoft.com/fr-fr/help/13766/hololens-using-3d-viewer-on-hololens>> [consulté le 03/02/2018]

- MICROSOFT, Page explicative sur la caméra de Microsoft [en ligne]. Disponible sur <<https://docs.microsoft.com/fr-fr/windows/mixed-reality/locatable-camera>> [consulté le 03/02/2018]

### Site internet de Unity :

- API de Unity [en ligne]. Disponible sur <<https://docs.unity3d.com/ScriptReference/>> [consulté le 12/04/2018]

- API de Unity : Vuforia [en ligne]. Disponible sur <<https://docs.unity3d.com/Manual/vuforia-sdk-overview.html>> [consulté le 16/03/2018]

### Vidéos :

- AP-HP : Opération de l'épaule réalisée avec un casque de réalité mixte [en ligne]. Disponible sur <[https://www.youtube.com/watch?time\\_continue=4321&v=xUVMeib0qek](https://www.youtube.com/watch?time_continue=4321&v=xUVMeib0qek)> [consulté le 22/02/2018]

- EPOFILMS : Alex Kipman [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=Unj0kHT2XWk>> [consulté le 20/02/2018]

- I3D Past Projects : holoportation: virtual 3D teleportation in real-time (Microsoft Research) [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=7d59O6cfaM0>> [consulté 01/03/2018]

- KOSTAS PAP : Vuforia AR Controller [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=020Utr57uP8>> [consulté le 10/05/2018]

- KOTABU : Minecraft Hololens demo at E3 2015 [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=xgakdcEzVwg>> [consulté le 20/02/2018]

- LIANG Yining, Tutoriel pour importer un fichier depuis One Drive [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=ReE-PetXVh0&t=1s>> [consulté le 10/03/2018]

- MARK SINNATHAMBY : Hololens Development using Unity and Visual Studio 2015 [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=yDm9tryDiIY>> [consulté le 20/02/2018]

- MICROSOFT HOLOLENS : Microsoft HoloLens: Skype [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=4QiGYtd3qNI>> [consulté le 20/02/2018]

- NDC Conferences : Developing Object Recognition for HoloLens - Stephen Carter [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=LxEl7DN5uRg>> [consulté le 12/04/2018]

- SKETCHUP : SketchUp Viewer for Hololens Demonstration [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=dmpoCjz0Yc0>> [consulté le 10/03/2018]

- SYNCHRO SOFTWARE : Introduction to HoloSITE, utilisation application HoloLens pour un projet BIM [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=Pdl0mLKTmy4>> [consulté le 20/02/2018]

- THOUGHT EXPERIMENT : #Hololens - Tutorial #1 - Getting Started [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=OYx4qIqi0oI>> [consulté le 12/04/2018]

- U9VATA : Blender meets Hololens 2 - upgrades to HoloRenderer [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=b5wfjLPtigY>> [consulté 17/08/2018]

- VISUALLIFE : Revit to HoloLens [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=Wxfhceti0-4>> [consulté le 10/03/2018]

- WINDOWS : Envisioning the Future with Windows Mixed Reality [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=2MqGrF6JaOM>> [consulté le 20/02/2018]

- YINING LIANG : Hololens Load Data From OneDrive [en ligne]. Disponible sur <<https://www.youtube.com/watch?v=ReE-PetXVh0>> [consulté le 16/03/2018]





## **LES ANNEXES**

## LES GESTES

```
using UnityEngine;
using UnityEngine.XR.WSA.Input;

public class Gestes : MonoBehaviour
{
    //Declaration des variables
    public GameObject FocusedObject { get; private set; }
    public GestureRecognizer Recognizer { get; private set; }
    public Camera cameraPrincipaleM;
    public GameObject initial;
    public GameObject sphere;
    public Vector3 handPosition,cameraPosition;
    public Vector3 handPositionInit;

    static public Gestes Instance { get; private set; }
    static public GameObject objetClique,objetTenu, objetVise;
    static public string nomVise, nomClique;
    static public bool clique,tenu,mainVisible, objetVide;
    static public Vector3 valeurDeplacement;
    static public Vector3 positionMain;

    void Start()
    {
        objetClique = initial;
        objetTenu = initial;
    }

    private void OnNavigationStartedEvent(InteractionSourceKind source, Vector3 relativePosition, Ray ray)
    {
        valeurDeplacement = relativePosition;
        tenu = true;
    }
    private void OnNavigationUpdatedEvent(InteractionSourceKind source,Vector3 relativePosition,Ray ray)
    {
        valeurDeplacement = relativePosition;
    }
    private void OnNavigationCompletedEvent(InteractionSourceKind source, Vector3 relativePosition, Ray ray)
    {
        valeurDeplacement = relativePosition;
        tenu = false;
    }

    private void OnInteractionSourceDetected(InteractionSourceDetectedEventArgs args)
    {
        mainVisible = true;
        cameraPosition = cameraPrincipaleM.transform.position;
    }
    private void OnInteractionSourceUpdated(InteractionSourceUpdatedEventArgs args)
    {
        if (args.state.source.kind == InteractionSourceKind.Hand)
        {
            args.state.properties.sourcePose.TryGetPosition(out handPosition);
        }
    }
    private void OnInteractionSourceLost(InteractionSourceLostEventArgs Args)
    {
        mainVisible = false;
    }

    // Detecter les gestes reconnus
    void Awake()
    {
        Instance = this;
        Recognizer = new GestureRecognizer();
        Recognizer.Tapped += (argumentTap) =>
        {
            clique = true;
        }
    }
}
```

```

    if (FocusedObject != null)
    {
        objetClique = FocusedObject;
        nomClique = FocusedObject.name;
    }
};

Recognizer.NavigationStartedEvent += OnNavigationStartedEvent;
Recognizer.NavigationUpdatedEvent += OnNavigationUpdatedEvent;
Recognizer.NavigationCompletedEvent += OnNavigationCompletedEvent;

InteractionManager.InteractionSourceDetected += OnInteractionSourceDetected;
InteractionManager.InteractionSourceUpdated += OnInteractionSourceUpdated;
InteractionManager.InteractionSourceLost += OnInteractionSourceLost;

Recognizer.StartCapturingGestures();
}

void Update()
{
    positionMain = cameraPosition + handPosition -new Vector3(0,0,0.1f);

    objetVide = false;
    GameObject oldFocusObject = FocusedObject;

    var headPosition = cameraPrincipaleM.transform.position;
    var gazeDirection = cameraPrincipaleM.transform.forward;

    if (mainVisible==false)
    {
        tenu = false;
    }

    RaycastHit hitInfo;
    if (Physics.Raycast(headPosition, gazeDirection, out hitInfo))
    {
        FocusedObject = hitInfo.collider.gameObject;
        if (FocusedObject!=null)
        {
            objetVide = true;
            objetVise = FocusedObject;
            nomVise = FocusedObject.name;
        }
    }
}
}
}

```

## LE MENU

```

using UnityEngine;
using System.Collections;
using UnityEngine.XR.WSA.Input;

public class Menu : MonoBehaviour
{
    // Variables utilisables par toutes les classes
    static public bool suivre;
    static public string choixMenu;
    static public bool initialisationVar;

    // Variables
    private Vector3 nouvelleDir;

    public Material initial;
    public Material selection;
    public Material passage;

    public GameObject boutonSuivre, boutonNePasSuivre;
    public GameObject boutonEnvironnement, boutonEnvironnementActive;
    public GameObject importation, menuImportation, modeVisite, suppression, section, superposition, menuSSuper, immersion, couches, me-

```

```

nuMain;
public GameObject menu,validation, casque;
public GameObject deplacement,environnement;
public GameObject objetImporte;

private string texte;

private string[,] reconnaissance = new string[9, 2];
private string memoire;

private int i, j,iMem;
private int timer;

private float distance;
private bool mouvementMenu, environnementActive;

void Start()
{
    //Creation du tableau du menu
    reconnaissance = new string[,] { { "Importer", 0.ToString() }, { "Supprimer", 0.ToString() }, { "Deplacer", 0.ToString() }, { "Main", 0.ToString() }
    , { "Visite", 0.ToString() },{ "Immersif", 0.ToString() }
    , { "Superposition", 0.ToString() }, { "Couches", 0.ToString() }, { "Coupe", 0.ToString() } };

    // Initialisation des valeurs
    suivre = true;
    boutonSuivre.SetActive(suivre);
    boutonNePasSuivre.SetActive(!suivre);
    choixMenu = null;
    transform.position = casque.transform.position + casque.transform.forward;
}

void Update()
{
    // Ajout d'un timer pour un menu plus fluide visuellement
    // laisse la couleur de passage sur les boutons un peu plus
    timer++;
    if (timer > 5) { timer = 0; }

    for (i = 0; i < 9; i++)
    {
        if (Gestes.nomVise == reconnaissance[i, 0])
        {
            if (Gestes.nomVise == Gestes.nomClique)
            {
                if (memoire != Gestes.nomVise)
                {
                    for (j = 0; j < 9; j++)
                    {
                        if (reconnaissance[j, 1] == 1.ToString())
                        {
                            reconnaissance[j, 1] = 0.ToString();
                        }
                    }
                    iMem = i;
                    choixMenu = Gestes.nomClique;
                    reconnaissance[i, 1] = 1.ToString();
                    memoire = reconnaissance[i, 0];
                }
            }
            else
            {
                GameObject.Find(reconnaissance[i, 0]).GetComponent<Renderer>().material = passage;
            }
        }
        else
        {
            if (timer == 0 && reconnaissance[i, 1] != 1.ToString())
            {
                GameObject.Find(reconnaissance[i, 0]).GetComponent<Renderer>().material = initial;
            }
        }
    }
}

```

```

    }
}

// Bouton suivre
if (Gestes.nomVise=="LeSuivre" || Gestes.nomVise=="NePasSuivre")
{
    boutonSuivre.GetComponent<Renderer>().material = passage;
    boutonNePasSuivre.GetComponent<Renderer>().material = passage;
    if (Gestes.clique == true)
    {
        Gestes.clique = false;
        suivre = !suivre;
        boutonSuivre.SetActive(suivre);
        boutonNePasSuivre.SetActive(!suivre);
    }
}
else
{
    boutonSuivre.GetComponent<Renderer>().material = initial;
    boutonNePasSuivre.GetComponent<Renderer>().material = selection;
}

// Bouton Environnement
if (Gestes.nomVise == "EnvironnementBouton" || Gestes.nomVise == "EnvironnementBoutonActive")
{
    boutonEnvironnement.GetComponent<Renderer>().material = passage;
    boutonEnvironnementActive.GetComponent<Renderer>().material = passage;
    if (Gestes.clique == true)
    {
        Gestes.clique = false;
        environnementActive = !environnementActive;

        boutonEnvironnement.SetActive(!environnementActive);
        boutonEnvironnementActive.SetActive(environnementActive);
    }
}
else
{
    boutonEnvironnement.GetComponent<Renderer>().material = initial;
    boutonEnvironnementActive.GetComponent<Renderer>().material = selection;
}

// Deplacer le Menu en fonction de la distance de visee
// Le menu doit rester visible lorsque l'on s'eloigne
// Plus on s'eloigne, plus le menu se deplace rapidement

nouvelleDir = casque.transform.position - transform.position;
transform.rotation = Quaternion.LookRotation(nouvelleDir, Vector3.up);
distance = (menu.transform.position - (casque.transform.position + casque.transform.forward)).magnitude;

// Si le bouton Suivre est active, il faut que le menu nous suive
if (suivre)
{
    if (distance > 0.3f) { mouvementMenu = true; } // distance longue à partir du moment ou le menu bouge
    else if (distance < 0.1f) { mouvementMenu = false; } // distance courte à partir du moment ou le menu s'arrete

    if (mouvementMenu) { transform.position = Vector3.MoveTowards(transform.position, casque.transform.position + casque.
transform.forward, distance * Time.deltaTime); }
}

//Si l'environnement est active, il faut l'afficher
if (environnementActive)
{
    environnement.SetActive(true);
}
else
{
    environnement.SetActive(false);
}

validation.transform.position = menu.transform.position;

```

```

// Onglet Manipulation

// A chaque fois, afficher le sous menu, activer la fonctionnalite
// initialiser, puis desactiver le menu principal
// Cette partie peut être optimisee

if (choixMenu=="Importer")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);
    menuImportation.SetActive(true);
    importation.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}
else if (choixMenu=="Supprimer")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);
    validation.SetActive(true);
    suppression.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}
else if (choixMenu == "Deplacer")
{
    initialisationVar = true;
    Deplacement.ini = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);
    validation.SetActive(true);
    deplacement.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}
else if (choixMenu == "Main")
{
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);
    validation.SetActive(true);
    menuMain.SetActive(true);
    Gestes.clique = false;
    suivre = true;
}

// Onglet Mode
else if (choixMenu == "Visite")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;

```

```

    menu.SetActive(false);
    modeVisite.SetActive(true);
    validation.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}
else if (choixMenu == "Immersif")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    immersion.SetActive(true);
    menu.SetActive(false);
    validation.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}

// Onglet Interactif
else if (choixMenu == "Superposition")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);
    menuSSuper.SetActive(true);
    menuSSuper.transform.position = menu.transform.position;
    superposition.SetActive(true);
    suivre = true;
    Gestes.clique = false;

    objetImporte.SetActive(true);
}
else if (choixMenu == "Couches")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);

    menuImportation.SetActive(true);
    menuImportation.transform.position = menu.transform.position;
    couches.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}
else if (choixMenu == "Coupe")
{
    initialisationVar = true;
    memoire = null;
    reconnaissance[iMem, 1] = 0.ToString();
    iMem = 0;
    GameObject.Find(choixMenu).GetComponent<Renderer>().material = initial;
    menu.SetActive(false);
    validation.SetActive(true);
    section.SetActive(true);
    Gestes.clique = false;
    suivre = true;
    objetImporte.SetActive(true);
}
else {}
}
}
}

```

## LA COUPE

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Coupe : MonoBehaviour {

    // Declaration des Variables
    public GameObject cameraPrincipale;
    public GameObject validation, menu;
    public GameObject plan, objetCoupe;
    public float valeur;
    public Shader shaderObjetCoupe;
    public Vector3 vecteurDirecteur;
    public Vector3 normal, position;
    public Renderer rend;
    public bool coupeActive;
    public float moyenneDeplacement;
    public GameObject objetMemoire, objetImporte;
    public Material initial, passage;

    public Transform[] allChildren, allChildrenMem;

    // Initialiser
    void Initialisation()
    {
        for (int i = 1; i < objetImporte.transform.childCount; i++)
        {
            objetImporte.transform.GetChild(i).GetComponent<BoxCollider>().enabled = true;
        }
    }

    void Start ()
    {
        objetMemoire = GameObject.Find("InitialImp");
        Initialisation();
    }

    void Update ()
    {
        // Le menu validation regarde toujours l'utilisateur
        validation.transform.rotation = Quaternion.LookRotation(cameraPrincipale.transform.position - validation.transform.position,
        Vector3.up);

        moyenneDeplacement = (Gestes.valeurDeplacement.x + Gestes.valeurDeplacement.y) / 2;
        // Nous cliquons
        if (Gestes.clique==true)
        {
            Gestes.clique = false;

            if (Gestes.nomVise == "Terminer")
            {
                Gestes.clique = false;
                validation.SetActive(false);
                menu.SetActive(true);

                for (int k = 0; k < allChildrenMem.Length; k++)
                {
                    allChildrenMem[k].GetComponent<Renderer>().material.shader.name = "Standard";
                }
                for (int i = 1; i < objetImporte.transform.childCount; i++)
                {
                    objetImporte.transform.GetChild(i).GetComponent<BoxCollider>().enabled = false;
                }

                Menu.choixMenu = null;
                this.gameObject.SetActive(false);
            }
            else { }
        }
    }
}
```

```

// Si l'objet appartient à la liste des objets importés
if (Gestes.objetClique.transform.parent.root.name == "ObjetImporte")
{
    // Positionner le plan sur la surface
    plan.transform.position = WorldCursor.positionCurseur;
    plan.transform.rotation = Quaternion.FromToRotation(Vector3.up, WorldCursor.NormalCurseur);
    coupeActive = true;

    // Chercher tous les enfants de l'objet sélectionné
    allChildren = Gestes.objetClique.GetComponentInChildren<Transform>();
    allChildrenMem = objetMemoire.GetComponentInChildren<Transform>();
    // Mettre la texture à tous les enfants de l'objet
    // Si on clique sur un autre objet

    if (objetMemoire != Gestes.objetClique)
    {
        for (int k = 0; k < allChildren.Length; k++)
        {
            for (int m = 0; m < allChildren[k].GetComponent<Renderer>().materials.Length; m++)
            {
                allChildren[k].GetComponent<Renderer>().materials[m].shader = shaderObjetCoupe;
            }
        }

        for (int l = 0; l < allChildrenMem.Length; l++)
        {
            for (int m = 0; m < allChildrenMem[l].GetComponent<Renderer>().materials.Length; m++)
            {
                allChildrenMem[l].GetComponent<Renderer>().material.shader = Shader.Find("Standard");
            }
        }
    }

    // Si on clique sur le même objet
    else
    {
        for (int k = 0; k < allChildren.Length; k++)
        {
            for (int m = 0; m < allChildren[k].GetComponent<Renderer>().materials.Length; m++)
            {
                allChildren[k].GetComponent<Renderer>().materials[m].shader = shaderObjetCoupe;
            }
        }
        objetMemoire = Gestes.objetClique;
    }
}
else {}

}
else {}

// Déplacement sur l'axe normal
// Il faut ajouter le geste de la main
if (coupeActive == true)
{
    if (Gestes.mainVisible == true && Gestes.tenu == true)
    {
        plan.transform.Translate(Vector3.up * moyenneDeplacement * Time.deltaTime, Space.Self);
    }

    // Donner les coordonnées du plan sur le matériau de chaque objet
    normal = plan.transform.TransformVector(new Vector3(0,1,0));
    position = plan.transform.position;

    for (int k = 0; k < allChildren.Length; k++)
    {
        for (int m = 0; m < allChildren[k].GetComponent<Renderer>().materials.Length; m++)
        {
            if (allChildren[k].GetComponent<Renderer>().materials[m].shader.name == "CrossSection/OnePlaneBSP")
            {
                allChildren[k].GetComponent<Renderer>().materials[m].SetVector("_PlaneNormal", normal);
                allChildren[k].GetComponent<Renderer>().materials[m].SetVector("_PlanePosition", position);
            }
        }
    }
}

```

```

    }
    }
}
else { }

//Quitter
if (Gestes.nomVise == "Terminer")
{
    Gestes.objetVise.GetComponent<Renderer>().material = passage;
}
else { GameObject.Find("Terminer").GetComponent<Renderer>().material = initial; }

//Securisation
if (Gestes.clique == true) { Gestes.clique = false; }
}
}

```

## L'IMPORTATION

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.WSA;

public class Importation : MonoBehaviour {

    // General
    public Camera cameraPrincipale;
    public Material passage, initial;
    public GameObject menuImportation, menu;
    public GameObject objetSupprime, objetImporte;

    private int i;

    // Sous menu
    public string[,] listeObjets;
    public string nom;
    public GameObject bouton1, bouton2, bouton3, bouton4, objetSelectionne;

    private int nombreObjets, page;
    private string fichier1, fichier2, fichier3, fichier4;
    private bool deplacement;
    private Vector3 decalage;

    // Initialisation du tableau contenant les objetsSupprime
    public void Initialisation()
    {
        page = 0;
        i = 0;
        nombreObjets = objetSupprime.transform.childCount;
        listeObjets = new string[nombreObjets, 2];
        do
        {
            nom = objetSupprime.transform.GetChild(i).name;
            listeObjets[i, 0] = nom;
            if (objetSupprime.transform.GetChild(i).childCount == 1)
            {
                listeObjets[i, 1] = objetSupprime.transform.GetChild(i).GetChild(0).name;
            }
            i++;
        } while (i < nombreObjets);

        //Test permettant de voir ce qu'il y a dans le tableau
        /*for (i=0;i<nombreObjets;i++)
        {
            Debug.Log(listeObjets[i, 0]);
        }*/
    }
}

```

```

// placer le menuImportation au même endroit que le menu
menuImportation.transform.position = menu.transform.position;
}

// Connaitre la liste des objets supprimer
// Créer les pages en fonction du nombre d'elements

private void Update()
{
    menuImportation.transform.rotation = Quaternion.LookRotation(cameraPrincipale.transform.position - menuImportation.trans-
form.position, Vector3.up);
    // Initialiser le tableau d'objetSupprime quand on entre dans le sous menu
    if (Menu.initialisationVar)
    {
        Initialisation();
        Menu.initialisationVar = false;
    }
    else {}

    // Calcul des objets à afficher dans le sous menu
    // Nous affichons 4 objets par pages au maximum

    if (nombreObjets > page * 4+1)
    { fichier1 = listeObjets[page*4 +1, 0]; }
    else
    { fichier1 = ""; }
    if (nombreObjets > page * 4+2)
    { fichier2 = listeObjets[page*4+2, 0]; }
    else
    { fichier2 = ""; }
    if (nombreObjets > page * 4+3)
    { fichier3 = listeObjets[page*4+3, 0]; }
    else
    { fichier3 = ""; }
    if (nombreObjets > page * 4+4)
    { fichier4 = listeObjets[page*4+4, 0]; }
    else
    { fichier4 = ""; }

    //Changer le texte des boutons objets du sous menu

    bouton1.GetComponent<TextMesh>().text = fichier1;
    bouton2.GetComponent<TextMesh>().text = fichier2;
    bouton3.GetComponent<TextMesh>().text = fichier3;
    bouton4.GetComponent<TextMesh>().text = fichier4;

    // Affichage : remettre les boutons dans la couleur initial
    // Pour seulement afficher celui selectionne quand on passe dessus
    if (menuImportation.activeSelf==true)
    {
        GameObject.Find("ObjetImporteSuiv").GetComponent<Renderer>().material = initial;
        GameObject.Find("ObjetImportePrec").GetComponent<Renderer>().material = initial;
        GameObject.Find("ImportationChoix1").GetComponent<Renderer>().material = initial;
        GameObject.Find("ImportationChoix2").GetComponent<Renderer>().material = initial;
        GameObject.Find("ImportationChoix3").GetComponent<Renderer>().material = initial;
        GameObject.Find("ImportationChoix4").GetComponent<Renderer>().material = initial;
        GameObject.Find("Terminer").GetComponent<Renderer>().material = initial;
    }

    // Deplacer l'objet quand nous l'avons dans la main
    if (deplacement == true)
    {
        // Deplacer l'objet par rapport au regard
        if (Gestes.clique == false)
        {
            if (objetSelectionne.GetComponent<WorldAnchor>()!=null)
            {
                Destroy(objetSelectionne.GetComponent<WorldAnchor>());
                Ancrage.NouvelleAncre = objetSelectionne;
                Ancrage.etat = "supprimerAncrage";
            }
        }
    }
}

```

```

menuImportation.SetActive(false);
if (objetSelectionne.GetComponent<Collider>()!=null)
{
    decalage = objetSelectionne.GetComponent<Collider>().bounds.center - objetSelectionne.transform.position;
}
objetSelectionne.transform.localPosition = cameraPrincipale.transform.position + cameraPrincipale.transform.forward * 3f -
decalage;
}
else
{
    // Placer l'ancrage
    Ancrage.NouvelleAncre = objetSelectionne;
    Ancrage.etat = "ajouterAncrage";

    Objets.mettreAJourObjets = true;

    Gestes.clique = false;
    deplacement = false;
    menuImportation.SetActive(true);

    objetSelectionne = null;
}
}
// Quand nous voulons changer de page, precedent et suivant
else
{
    if (Gestes.nomVise == "ObjetImportePrec")
    {
        Gestes.objetVise.GetComponent<Renderer>().material = passage;
        if (Gestes.clique == true)
        {
            Gestes.clique = false;
            if (page >= 0)
            {
                page--;
            }
            else {}
        }
        else {}
    }
    else if (Gestes.nomVise == "ObjetImporteSuiv")
    {
        Gestes.objetVise.GetComponent<Renderer>().material = passage;
        if (Gestes.clique == true)
        {
            Gestes.clique = false;

            if (page < (nombreObjets / 4))
            {
                page++;
            }
            else {}
        }
        else {}
    }

    // Si nous cliquons sur un bouton pour selectionner un objet
    // nous activons l'objet, nous le mettons dans le dossier objetImporte, puis dans la main avec la commande deplacement

    else if (Gestes.nomVise == "ImportationChoix1")
    {
        Gestes.objetVise.GetComponent<Renderer>().material = passage;
        if (Gestes.clique == true)
        {
            Gestes.clique = false;
            objetSupprime.SetActive(true);
            if (fichier1 != "")
            {
                objetSelectionne = GameObject.Find(fichier1);
                objetSelectionne.SetActive(true);
                objetSelectionne.transform.parent = objetImporte.transform;
                deplacement = true;
            }
        }
    }
}

```

```

    }
    Initialisation();
    objetSupprime.SetActive(false);

}
else { }
}
else if (Gestes.nomVise == "ImportationChoix2")
{
    Gestes.objetVise.GetComponent<Renderer>().material = passage;
    if (Gestes.clique == true)
    {
        Gestes.clique = false;
        objetSupprime.SetActive(true);
        if (fichier2!="")
        {
            objetSelectionne = GameObject.Find(fichier2);
            objetSelectionne.SetActive(true);
            objetSelectionne.transform.parent = objetImporte.transform;
            deplacement = true;
        }
        Initialisation();
        objetSupprime.SetActive(false);
    }
    else { }
}
else if (Gestes.nomVise == "ImportationChoix3")
{
    Gestes.objetVise.GetComponent<Renderer>().material = passage;
    if (Gestes.clique == true)
    {
        Gestes.clique = false;
        objetSupprime.SetActive(true);
        if (fichier3!="")
        {
            objetSelectionne = GameObject.Find(fichier3);
            objetSelectionne.SetActive(true);
            objetSelectionne.transform.parent = objetImporte.transform;
            deplacement = true;
        }
        Initialisation();
        objetSupprime.SetActive(false);
    }
    else { }
}
else if (Gestes.nomVise == "ImportationChoix4")
{
    Gestes.objetVise.GetComponent<Renderer>().material = passage;
    if (Gestes.clique == true)
    {
        Gestes.clique = false;
        objetSupprime.SetActive(true);
        if (fichier4 != "")
        {
            objetSelectionne = GameObject.Find(fichier4);
            objetSelectionne.SetActive(true);
            objetSelectionne.transform.parent = objetImporte.transform;
            deplacement = true;
        }
        Initialisation();
        objetSupprime.SetActive(false);
    }
    else { }
}

// Si on vise "Terminer", on quitte en fermant tout
else if (Gestes.nomVise == "Terminer")
{
    Gestes.objetVise.GetComponent<Renderer>().material = passage;
    if (Gestes.clique == true)
    {

```

```
Gestes.clique = false;
objetImporte.SetActive(false);
menuImportation.SetActive(false);
menu.SetActive(true);
Menu.choixMenu = null;
this.gameObject.SetActive(false);
}
else {}
}
else {}
}

//Securisation
if (Gestes.clique == true) { Gestes.clique = false; }
}
}
```

