

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement 2017-2018

Hololens, Réalité Augmentée et Epaule

Tuteurs académiques

Jordan NICOT

Mohamed SLIMANE

Julien BERHOUE

Christian PROUST

Étudiant

Alexis BARON (DI5)



Liste des intervenants

Nom	Email	Qualité
Alexis BARON	alexis.baron@etu.univ-tours.fr	Étudiant DI5
Jordan NICOT	jordan.nicot@outlook.com	Tuteur académique, Département Informatique
Mohamed SLIMANE	mohamed.slimane@univ-tours.fr	Tuteur académique, Département Informatique
Julien BERHOUET	julien.berhouet@gmail.com	Tuteur académique, Département Informatique
Christian PROUST	christian.proust@univ-tours.fr	Tuteur académique, Département Informatique



Avertissement

This document is written by Alexis BARON above-named the author.

L'Ecole Polytechnique de l'Université François Rabelais de Tours is represented by Jordan NICOT, Mohamed SLIMANE, Julien BERHOUE et Christian PROUST above-named the academic tutors.

By the utilisation of the document, all the project's intervenants are accepting the conditions defined below

The author recognizes assume full responsibility about the content of the document as well as any legal consequences that may result from the violation of laws or copyright.

The author attests that the words of the document are sincere and assume full responsibility for the veracity of the statements.

The author attests to do not appropriate the work of others and that the document contains no plagiarism.

The author attests that the document does not contain any defamatory or condemnable statement before the law.

The author acknowledges that he disseminates this document in whole or in part in any form without prior consent Of the academic tutors and of the company.

The author authorizes Polytechnic University of the François Rabelais University of Tours to disseminate all or part of this document, in any form whatsoever, including after transformation, citing the source. Such dissemination shall be free of charge and shall be accompanied by this warning.



Pour citer ce document

Alexis BARON, *Hololens, Réalité Augmentée et Epaule*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2017-2018.

```
@mastersthesis{
  author={BARON, Alexis},
  title={Hololens, Réalité Augmentée et Epaule},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2017-2018}
}
```

Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	v
Remerciements	1
I Introduction	2
1 Introduction	3
1 Contexte de la réalisation	3
2 Objectifs	5
2.1 Objectif 1 : Importation	5
2.2 Objectif 2 : Matching	6
2.3 Objectif 3 : Poursuite de l'objet	6
3 Hypothèses	6
4 Base méthodologiques	7
4.1 Gestion	7
4.2 Démarche scientifique	7
2 Description générale	9
1 Environnement du projet	9

2	Caractéristiques des utilisateurs	10
3	Fonctionnalités du système	11
4	Structure générale du système	11
4.1	Structure générale matérielle et logicielle	11
4.2	Structure générale de l'application	12
II	États de l'art	13
3	La réalité augmentée	14
4	Les HoloLens	16
1	Introduction	16
2	Caractéristiques	17
3	Marché et domaine d'application.....	18
3.1	Le marché de la réalité augmentée.....	18
3.2	Domaine d'application	19
4	Interactions	21
4.1	Vision	22
4.2	Gestuelle	23
4.3	Voix	24
5	Le Mapping	25
1	Principe du mapping.....	25
2	Erreurs de mapping.....	26
2.1	Les différentes erreurs	26
2.2	Quelques facteurs produisant ces erreurs	26
3	Conclusion	27
III	Analyse et conception	29
6	Sujet approfondi	30
7	Solutions proposées	31
1	Solution : Importation	31
2	Solution : Matching	32
3	Solution : Poursuite de l'objet.....	32
4	Architecture	32
4.1	Architecture des objets dans la scène.....	32
4.2	Architecture du code C#.....	33

8 Solutions finales du projet	35
1 Solution finale : Objectif Importation.....	35
2 Solution finale : Objectif Matching	35
3 Solution finale : Objectif Poursuite de l'objet	36
 IV Bilan	 37
9 Bilan PRD1	38
1 Difficultés rencontrées	38
2 Bilan personnel.....	38
3 Conclusion	39
 10 Bilan PRD2	 40
1 Difficultés rencontrées	40
2 Bilan personnel.....	40
3 Perspective du projet	41
4 Conclusion	41
 Annexes	 42
A Spécifications	43
1 Interfaces Homme/Machine.....	43
2 Spécifications fonctionnelles.....	43
3 Spécifications non fonctionnelles.....	46
3.1 Contraintes de développement et conception.....	46
3.1.1 Matériels	46
3.1.2 Langages de programmation	46
3.1.3 Logiciels et bibliothèques	46
3.2 Contraintes de fonctionnement et d'exploitation	47
3.2.1 Performances	47
3.2.2 Maintenance et évolution du système	47
 B Gestion de projet	 48
1 Méthode de gestion de Projet	48
2 Planning.....	48
 C Mise en place de l'environnement	 51
1 Installation.....	51
2 Configuration	51
2.1 Unity	52
2.2 Visual Studio.....	55

D Développement	57
1 Apprentissage et manipulation des objets dans la scène.....	57
2 Affichage de l'omoplate en surimpression.....	57
3 Première application avec ImageTarget et marqueur 2D	58
3.1 ImageTarget : configuration	59
3.2 ImageTarget : utilisation	61
3.3 ImageTarget : 1er résultat	61
3.4 ImageTarget : Stabilité de l'image	62
3.4.1 ImageTarget : Changement du comportement pour le Tracking ..	62
3.4.2 ImageTarget : désactivation du comportement Vuforia.....	63
3.5 ImageTarget : Activation du comportement Vuforia	64
3.5.1 Installation d'HoloToolkit/MixedRealityToolkit.....	64
3.5.2 Création d'un Manager.....	64
3.5.3 Création d'un nouveau script : "ClickLaunchTrackable".....	65
3.6 ImageTarget : 2ème résultat	66
3.7 ImageTarget : Clipping Plane	67
3.8 ImageTarget : 3ème résultat	69
4 Deuxième application avec ModelTarget et marqueur 3D.....	70
4.1 Model Target Generator.....	70
4.2 Model Target : Comment l'utiliser	71
4.3 Model Target : GuideView.....	72
4.4 Model Target : Résultat et hypothèses.....	72
E Démonstration	73
Comptes rendus hebdomadaires	74
Webographie	83
Bibliographie	85

Table des figures

1 Introduction

1	Épaule saine [WWW3]	4
2	Épaule abîmée par l'arthrose [WWW2]	4
3	Epaule après la pose de prothèse (http ://www.chirurgie-orthopedique-paris.com/prothese-de-lepaule/)	4
4	T. Gregory, Hôpital Européen Georges Pompidou – Paris	5
5	Pré-requis.....	5
6	Trello	7
7	Base méthodologique	8

2 Description générale

1	Environnement	10
2	Diagramme de cas d'utilisation	11

3 La réalité augmentée

1	Courbe du "Hype" des technologies	15
---	---	----

4 Les HoloLens

1	Caractéristiques techniques des Hololens [WWW4]	17
2	Caractéristiques techniques des différentes lunettes d'AR [WWW13]	18
3	Chirurgie du dos assistée par HoloLens [WWW6] [WWW5]	20
4	Utilisation automobile design [WWW14]	20
5	Logo Dassault Aviation [WWW8]	21
6	Kheops en réalité augmentée [WWW15]	21

7	Type d'action.....	22
8	Curseur HoloLens.....	22
9	Sélectionner (Air tap)	23
10	Ouverture menu et fermeture application (Bloom).....	23
5	Le Mapping	
1	Mapping d'une pièce	25
7	Solutions proposées	
1	Ajouter un masque	31
2	Architecture des objets dans la scène.....	33
3	Architecture du code C#	34
B	Gestion de projet	
1	Cycle en V	48
2	Diagramme de Gantt Prd1	49
3	Diagramme de Gantt Prd2	49
4	Diagramme de Gantt Prd2 Fin de projet	50
C	Mise en place de l'environnement	
1	Écran installation VS2015	52
2	Préférences.....	52
3	QualitySettings	53
4	BuildSettings	53
5	OtherSettings.....	54
6	Capabilities	54
7	XRSettings	55
8	Configuration Digital Eyewear	55
9	Configuration Visual	56
10	Pour avoir le code PIN sur les HoloLens.....	56
D	Développement	
1	Projet origami	57
2	Scène Unity pour l'affichage de l'omoplate	58
3	License Manager	59
4	Création de la base de données	59
5	Ajout d'un marqueur	60
6	Vuforia configuration	60

7	ImageTarget dans la scène	61
8	Méthode sur le changement d'état du Tracking	63
9	Stopper Vuforia.....	63
10	Composant de MyManager	65
11	Méthode Update().....	66
12	Gestion du tap avec Audio	66
13	Explication clipping plane	67
14	Maquette finale avec marqueur	70
15	Model Target Generator	71
16	ModelTarget dans Unity	72
 E Démonstration		
1	Démonstration du 11 Octobre (1er groupe)	73



Remerciements

En premier lieu, je souhaiterais remercier toutes les personnes m'ayant assisté et encadré durant ce projet afin de réaliser correctement mon projet de recherche et développement.

Je remercie mes encadrants : Mr NICOT Jordan, Mr SLIMANE Mohand, Mr BERHOUE Julien et MR PROUST Christian.

Ces personnes m'ont parfaitement encadré, s'impliquent et aiment vraiment ce qu'ils font. Cela motive et aide grandement à la réalisation d'un tel projet.

Je remercie également Mr CARDOT Hubert de m'avoir aidé à trouver une piste sur un problème technique de placement dans un espace 3D.

Je voudrais remercier Mr RAMEL Jean-Yves et Mr BOCQUILLON Ronan pour m'avoir montré qu'il était possible de faire des diagrammes d'architecture de nos objets même si nous travaillons dans un environnement de manipulation 3D.

Enfin, je remercie toutes les personnes ayant pu contribuer par n'importe quel moyen à l'avancée du projet.

Première partie

Introduction

1

Introduction

Ce document présente la seconde partie du projet Recherche et Développement « Lunettes 3d et prothèses d'épaule » et fait donc suite à la première partie réalisée par Jordan Nicot. Il définit les premières idées de développement avec une approche globale. Le client est le LI et le CHRU Trousseau (Dr Julien Berhouet). Le maître d'œuvre est Alexis BARON. Il est également l'auteur de ce document et le maître d'ouvrage, celui qui s'occupera de la réalisation du projet.

L'objectif du "projet Hololens, Réalité Augmentée et Épaule" est de faire le lien entre les domaines de la chirurgie orthopédique et de l'informatique.

Nous souhaitons pouvoir afficher, en temps réel, pendant l'intervention chirurgicale, sur des lunettes connectées portées par le chirurgien, une image de la scapula pathologique et/ou de la glène reconstituée virtuellement.

La superposition avec la cible (ce que voit réellement le chirurgien) doit être très précise : non seulement entre l'image de la glène et la glène elle-même - visible - mais aussi entre l'image de la scapula et la scapula réelle - elle-même non visible -.

La précision de ces superpositions devra être quantifiée à terme. Cet affichage veut dire évidemment déformation des images en adéquation avec les déplacements de son regard.

1 Contexte de la réalisation

Voici un résumé :

a) Problème médical : usure de la glène à cause de l'arthrose

Comme on peut le voir sur la figure 2, le cartilage présent sur la tête de l'humérus et la glène est détérioré.

Ce qui nécessite une intervention chirurgicale.

b) Remède chirurgical : arthroplastie (pose d'une prothèse anatomique) :

Afin de soigner l'épaule du patient il va falloir enlever les parties détériorées et les remplacer par des prothèses. Nous allons nous intéresser principalement au cas de la prothèse de la glène et la principale question est de savoir comment bien la positionner.

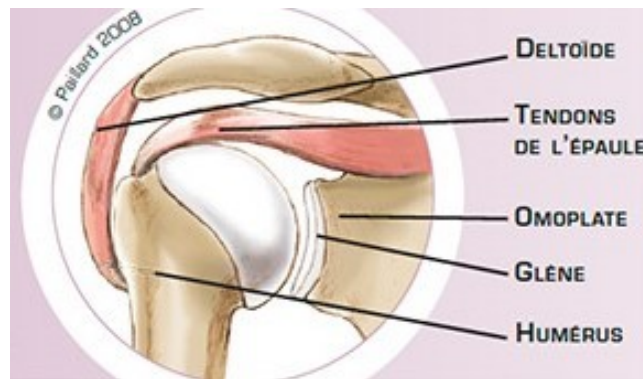


Figure 1 – Épaule saine [WWW3]



Figure 2 – Épaule abîmée par l'arthrose [WWW2]



Figure 3 – Epaule après la pose de prothèse (
<http://www.chirurgie-orthopedique-paris.com/prothese-de-lepaule/>)

c) On dispose de :

La partie verte est une image 3D réelle provenant d'un scanner que le patient a subi 15 jours avant l'intervention.

La partie rouge est une image 3D virtuelle de la partie de glène disparue. Cette partie a été mathématiquement reconstituée en probabilité.

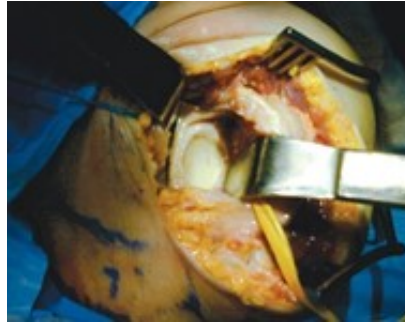


Figure 4 – T. Gregory, Hôpital Européen Georges Pompidou – Paris

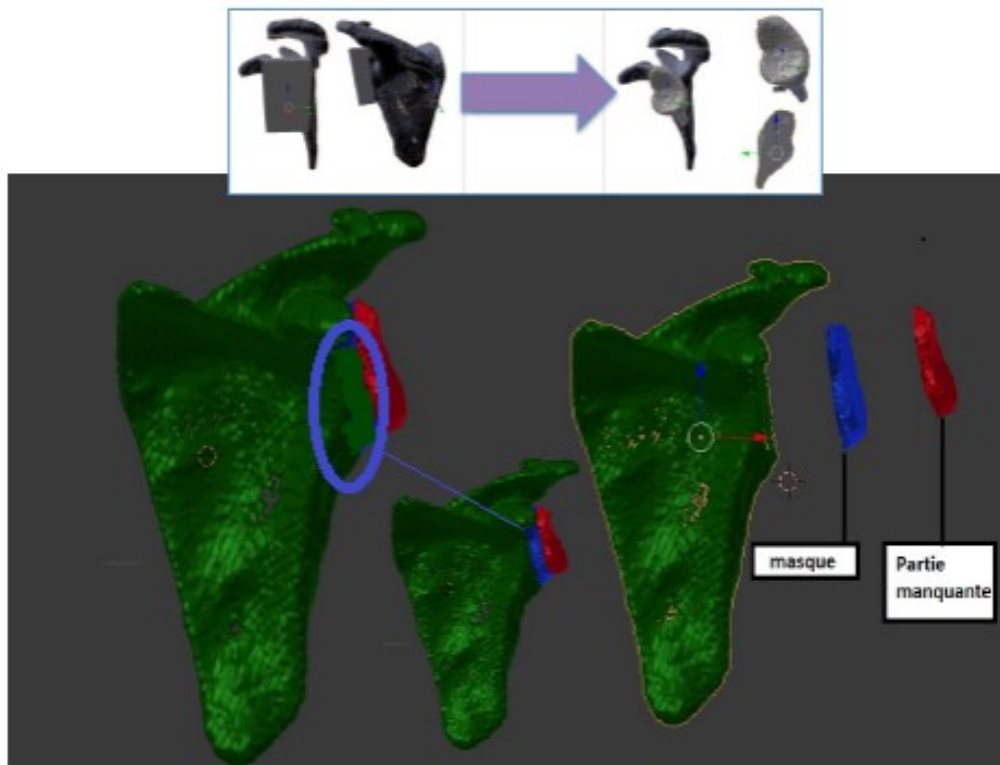


Figure 5 – Pré-requis

L'image en bleu est un extrait (avec le logiciel Blender) de l'image en vert, c'est le Masque qu'il faudra superposer parfaitement, à l'initialisation, avec la partie de glène visible par le chirurgien lors de son intervention.

2 Objectifs

Dans ce projet 3 objectifs principaux m'ont été confiés, objectifs sur lesquels il faudra que je me renseigne et acquiers des connaissances afin de pouvoir les accomplir.

2.1 Objectif 1 : Importation

L'importation d'un modèle 3D de Masque était un des problèmes majeurs l'année précédente et je vais devoir tenter de le résoudre dans ce projet.

La compagnie Vuforia a présenté en Juin 2017 un outil permettant d'importer ses propres objets au format .obj en tant que "Target" que nous appellerons Masque et ceci à la place de l'actuel moyen mis en place qui est un modèle imprimé sur une feuille.

Ce "Model Target Generator" devrait être intégré dans la version 7 de Vuforia ... à une date indéterminée.

Cependant la disponibilité de ce logiciel étant pour le moins aléatoire, nous allons partir sur un autre moyen plus accessible. Une pastille posée sur le patient avec des motifs et des couleurs très variées afin de faciliter la reconnaissance pour les HoloLens. Cette pastille une fois reconnue sur le patient et à partir des précédentes radios de celui-ci, nous pourrions calculer la position de l'omoplate réelle et donc placer l'objet virtuel.

2.2 Objectif 2 : Matching

Le matching dans notre cas va être très important car c'est le coeur même de l'application ; il faudrait importer comme cité précédemment un modèle 3D comme Masque(bleu) mais il faudrait également le modèle 3D de l'omoplate pathologique réelle actuelle (vert) qu'il faudra afficher en surimpression sur les lunettes (la scène).

Le but du matching serait donc de faire une superposition. Mais comme on vient de le dire ci-dessus, nous utiliserons, dans un premier temps, une pastille pour aider au matching initial, à la place du masque.

2.3 Objectif 3 : Poursuite de l'objet

Pour finir il faudra ensuite qu'on puisse voir l'omoplate virtuelle et que celle-ci soit fixe c'est à dire que si l'on tourne autour du patient l'omoplate virtuelle restera en place et l'image ne bougera pas.

L'année dernière des problèmes de saut d'image avaient été constatés ; il faudra que ce problème soit réglé.

3 Hypothèses

Nous utiliserons les HoloLens, ces lunettes sont récentes et sont sur le marché des développeurs depuis 1 an. De plus, la réalité augmentée étant une technologie récente les logiciels et bibliothèques sont voués à être modifiés dans le futur.

Il faudra donc se baser sur les technologies actuelles et utiliser des bibliothèques existantes auxquelles il faudra rajouter notre propre code afin de compléter nos objectifs.

De plus lors de l'Augmented World Exposition à Santa Clara aux Etats-Unis, qui s'est tenue du 31/06/2017 au 02/07/2017, un logiciel/add-on a été exposé par l'entreprise Vuforia. C'est un add-on qui permettrait de pouvoir répondre à certaines attentes du projet afin d'importer un modèle 3D du masque ce qui serait plus pratique que l'utilisation d'une image 2D comme dans le précédent PRD. Or pour avoir accès à ce logiciel en Beta, il faut passer par une discussion par mail avec Vuforia, dont Christian Proust s'occupe. Cependant en vue de la possible non disponibilité de cette add-on, il faudra une solution. Pour cela on aurait donc à placer une pastille sur le patient comme expliqué précédemment dans l'objectif 1.

4 Base méthodologiques

4.1 Gestion

En ce qui concerne la gestion de ce projet, les projets de recherche et développement étant des projets individuels je n'utiliserai donc pas de technologie de partage de fichier tel que git,svn, etc...

Cependant des comptes rendus ainsi que l'avancée du rapport de chaque semaine seront à remettre à mes encadrants afin de pouvoir évaluer l'avancement du projet.

Pour l'organisation sur l'année du projet j'utiliserai MindView afin de pouvoir générer un diagramme de Gantt sur lequel il faudra se baser pour éviter les retards sur le projet.

Un Trello¹ sera aussi mis en place afin de pouvoir gérer les tâches que je devrai effectuer tout au long de l'année, celui-ci sera utilisé majoritairement lors de la partie modélisation et développement de l'application.

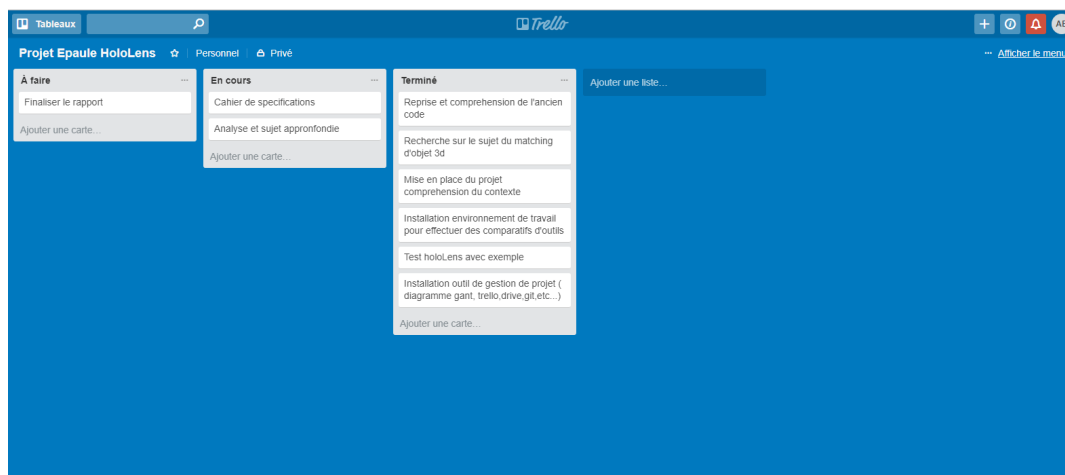


Figure 6 – Trello

4.2 Démarche scientifique

Une première ébauche de la démarche scientifique avait été faite lors du précédent PRD sur ce projet. Hors le problème était que l'image sautait ; nous en avons donc parlé et ceci est dû au fait que dans l'algorithme principal du déroulement de l'application, il y avait un positionnement de la scène (omoplate en surimpression) dans la boucle principale après l'initialisation.

C'est pourquoi au moindre mouvement de l'utilisateur l'image sautait, il est donc préférable de positionner et de fixer la scène dès le début lors de l'initialisation et de ne plus y toucher. Les lunettes se souviendront de la position de l'objet 3D à observer en surimpression.

Cependant ce genre de procédé enlève la possibilité que l'objet 3D virtuel en surimpression puisse bouger, mais dans ce type de chirurgie, c'est ce qu'il nous faut, car le patient ne bouge pas. Nous n'avons donc pas besoin de suivre un objet 3D, qui aurait un masque qui bouge, en réinitialisant sa position mais bien de le fixer et de ne plus y toucher.

Nous avons donc 3 tâches principales :

1. cf définition Wikipédia : <https://fr.wikipedia.org/wiki/Trello>

1. Détection de la cible et positionnement de la scène
2. Reconnaissance de la nouvelle orientation de l'image 3D de la cible, en fonction du regard du chirurgien
3. Rafraîchissement de l'image sur les HoloLens

Voici le schéma du fonctionnement principal de l'application correspondant à ces tâches :

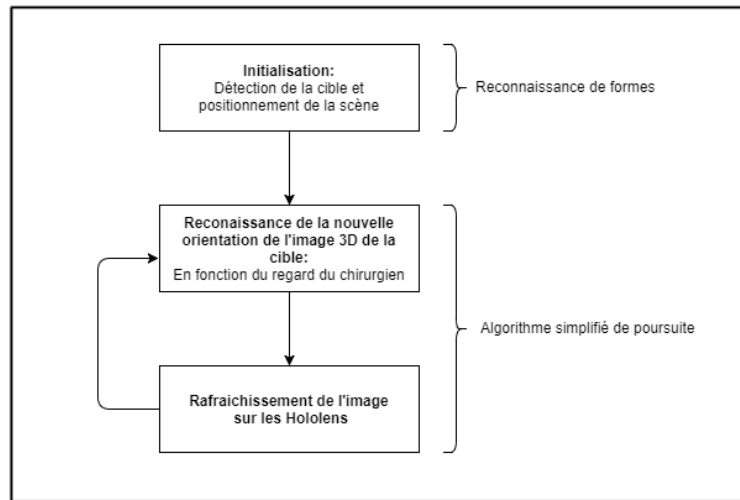


Figure 7 – Base méthodologique

2

Description générale

1 Environnement du projet

Ce projet est la suite de plusieurs anciens projets depuis 2015, depuis l'année dernière ce projet est axé sur la partie réalité augmentée. Il existe déjà un prototype, testé l'année dernière, mais qui comporte plusieurs problématiques énoncées plus tôt dans ce rapport.

Les lunettes utilisées dans ce projet sont les mêmes que celles de l'année dernière ; ce sont des HoloLens.

Il existe également une maquette, d'épaule ouverte, mise à disposition pour effectuer les tests. Pour ce qui est de l'environnement au sens technique nous utiliserons les mêmes logiciels/SDK que dans le précédent PRD c'est à dire :

1. Visual studio 2017

Faire attention, certaines versions de Visual studio 2017 ne fonctionnent pas avec Vuforia et Unity, si possible prendre Visual studio 2015 Update 3 pour avoir une version stable. Visual Studio va donc nous permettre de compiler notre code C#¹ afin d'avoir l'application sur nos lunettes. C'est un logiciel de développement Windows

2. Vuforia

Vuforia est un SDK de réalité augmentée qui peut être utilisé pour plusieurs plates-formes comme Android, IOS, UWP et Unity, c'est une grosse plateforme pour la réalité augmentée.²

3. Unity 2017 2.0b2

Finalement cette version choisie en début de PRD s'est avérée buguée lors de la phase de mise en place de l'environnement.

Il faudra donc prendre la version Unity 2017.3.0b11 qui elle, est fonctionnelle et sans bugs. Unity est un moteur multiplateforme permettant entre autre dans notre cas, de gérer toute la partie 3D de notre application, le langage engendré derrière est du C#.

Pour faire le lien entre tout nos composants, il faudra utiliser UWP qui est une plateforme Windows.

1. Langage de programmation

2. 375 000 + développeurs enregistrés / 45000+ applications [WWW19]

On pourra donc compiler une solution C# dans Unity pour UWP et ensuite utiliser cette solution dans Visual Studio 2015.

Celle-ci pourra donc être compilée et enfin être exportée sur les lunettes HoloLens.

Cette plateforme est donc vraiment utile car, sans celle-ci, il serait impossible d'avoir les liens entre les différents composants de notre environnement.

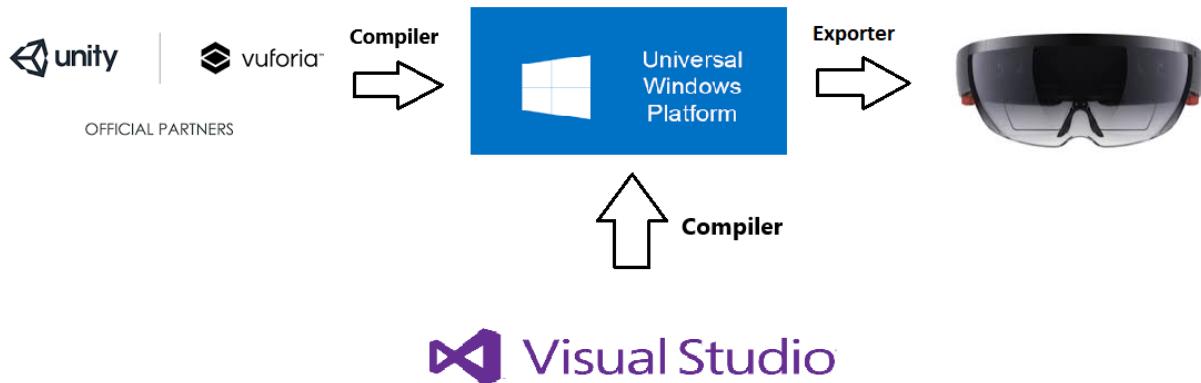


Figure 1 – Environnement

2 Caractéristiques des utilisateurs

Cette application sera utilisée ou modifiée par 2 types d'utilisateurs :

1. Le chirurgien

Le chirurgien devra suivre un manuel utilisateur pour comprendre le fonctionnement de l'appareil mais également le fonctionnement de l'application elle-même.

Pour un meilleur confort les futures versions pourraient assister le chirurgien dans son utilisation de l'application mais ceci pourrait être une future fonctionnalité sur la version finale qui ne fait pas partie des objectifs actuels.

Afin de pouvoir suivre en direct ce que voit le chirurgien, par exemple pour les assistants de celui-ci, un moniteur type PC pourrait être mis en place afin de diffuser le flux vidéo de ce que voit le chirurgien avec les HoloLens, un logiciel à ce propos est sortie en 2017. [WWW11]

2. Le technicien

Le technicien n'interviendra que très rarement. Celui-ci devra avoir les connaissances nécessaires afin de mettre à jour l'application en rajoutant des modules ou des fonctionnalités supplémentaires sur l'application finale.

3 Fonctionnalités du système

Voici notre diagramme de cas d'utilisation :

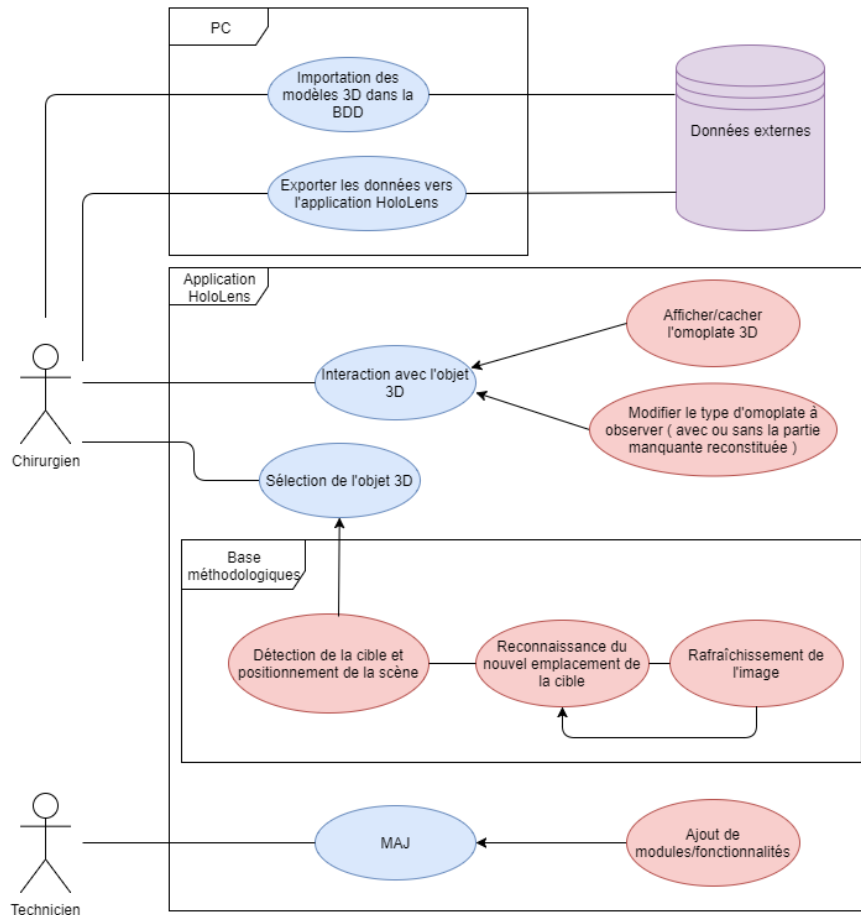


Figure 2 – Diagramme de cas d'utilisation

4 Structure générale du système

4.1 Structure générale matérielle et logicielle

Ici nous allons rester sur le même environnement matériel que l'année précédente lors du PRD de Jordan Nicot (voir rapport de Jordan Nicot).

Cet environnement se compose de Vuforia , Virtual studio (nous utiliserons la version 2015 finalement car la version 2017 a des problèmes de compatibilité), Unity 3D (version 5.5.2 pour des raisons de compatibilité également) et bien sûr les lunettes HoloLens de Microsoft. Nous aurons donc un environnement logiciel Windows.

Des liens seront à effectuer entre Vuforia et Unity et entre Unity et Visual Studio via un tutoriel qui explique comment mettre en place un environnement de développement pour la réalité augmentée.[WWW20] [WWW10]

4.2 Structure générale de l'application

L'application aura une structure simple, il faudra en premier lieu importer, dans la BDD des modèles, notre modèle 3D de l'omoplate que nous voudrions utiliser ; nous cherchons principalement à importer dans un premier temps l'omoplate verte + bleue (omoplate malade). À long terme il faudra importer l'omoplate verte + la partie bleue + la partie rouge.

Deuxième partie

États de l'art

3

La réalité augmentée

Nous entendons de plus en plus parler de réalité augmentée et virtuelle mais de quoi s'agit-il exactement ?

Cette partie sera en complément de celle présentée par Jordan Nicot dans son rapport de PRD l'année dernière.

Réalité augmentée :¹ La réalité augmentée est la superposition de la réalité et d'éléments (sons, images 2D, 3D, vidéos, etc.) calculés par un système informatique en temps réel.

Réalité virtuelle :² L'expression « réalité virtuelle » (ou multimédia immersif ou réalité simulée par ordinateur) renvoie typiquement à une technologie informatique qui simule la présence physique d'un utilisateur dans un environnement artificiellement généré par des logiciels, environnement avec lequel l'utilisateur peut interagir.

La réalité augmentée est à ne surtout pas confondre avec la réalité virtuelle.[1]

Effectivement, il existe plusieurs différences notables entre l'AR et la VR.

Après bien sûr la plus grande différence se fait au niveau de la perception ; avec la réalité augmentée vous accédez toujours au réel, dans la réalité virtuelle vous vous immergez complètement dans le virtuel et vous vous coupez du réel.

Ces deux domaines ont donc quelques points communs mais sont complètement différentes.[WWW21]

De nos jours la réalité augmentée est un secteur en pleine effervescence. C'est un secteur qu'on peut qualifier de secteur émergent et cela grâce aux multitudes de technologies permettant de faire de la réalité augmentée.

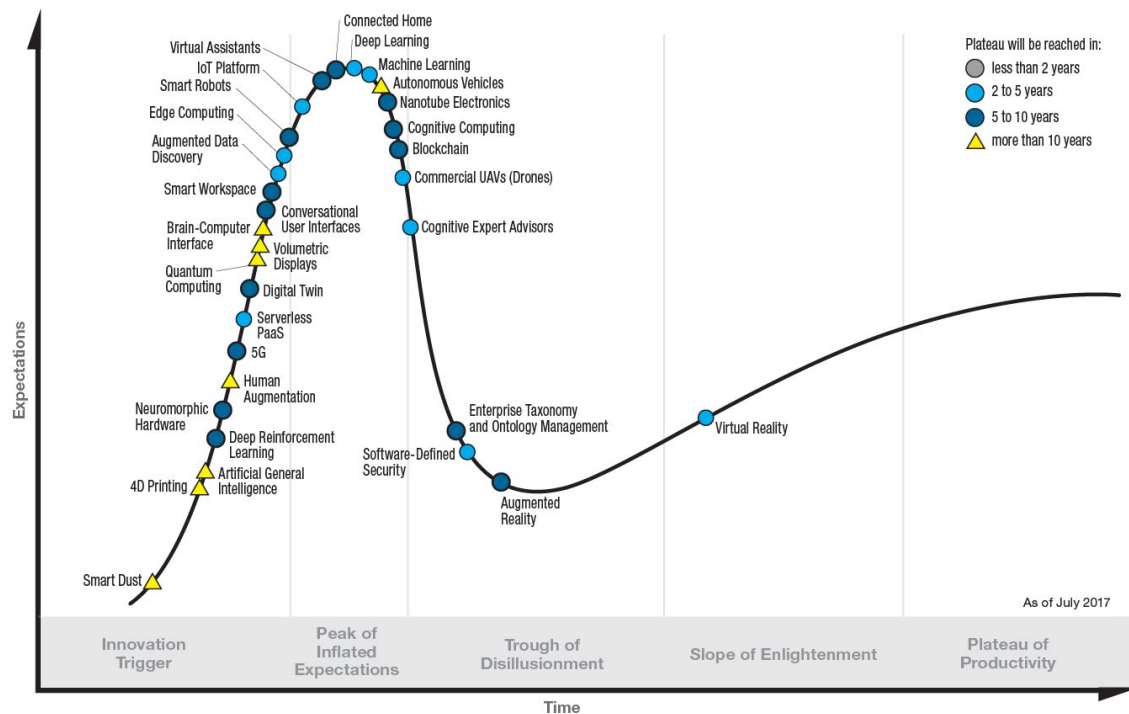
Dans une courbe du cycle de "hype" des technologies émergentes, on se trouverait donc dans le creux qui suit le pic du "Hype". Ce pic du "Hype" est la phase où l'on parle beaucoup du domaine, où l'on pense à toutes ses possibles applications mais où la technologie n'est pas encore sortie, c'est une phase d'émerveillement.

Quant au creux où nous nous trouvons, c'est une phase de désillusions car la technologie n'est pas aussi facile à prendre en main qu'on peut le penser et certaines applications que l'on pensait pouvoir réaliser lors du pic de "hype", ne sont finalement pas possibles ou difficilement réalisables. [3]

1. cf définition Wikipédia

2. cf définition Wikipédia

Gartner **Hype Cycle** for Emerging Technologies, 2017



gartner.com/SmarterWithGartner

Source: Gartner (July 2017)
© 2017 Gartner, Inc. and/or its affiliates. All rights reserved.

Gartner

Figure 1 – Courbe du "Hype" des technologies

4

Les HoloLens



1 Introduction

Dans ce projet nous utiliserons les lunettes HoloLens de Microsoft. Ces lunettes de grande qualité ont grandement poussé la réalité augmentée à passer dans ce que l'on peut appeler la phase de déploiement et qui dit phase de déploiement dit applications naissantes grâce à cette technologie.

En appliquant les lunettes HoloLens de Microsoft sur la courbe de Hype on obtient une phase de "Hype" qui a commencé le 21 Janvier 2015 date à laquelle Microsoft a dévoilé son prototype des lunettes lors de la conférence "Windows 10 : The Next Chapter". Cette période s'est finie le 29 Février 2016 date de début de pré-commande des lunettes pour les Etats-Unis et le Canada. C'est seulement 8 mois après, le 12 Octobre 2016, que les lunettes ont été ouvertes aux pré-commandes pour la France. C'est à partir de ce moment là que l'équipe, travaillant précédemment sur ce projet, a pu commander les lunettes après l'établissement d'un état de l'art approfondi sur les différentes lunettes du marché et les besoins du projet.

2 Caractéristiques

Alors en premier lieu nous allons parler des caractéristiques techniques des lunettes HoloLens de Microsoft.

Nous pouvons prendre en compte qu'actuellement ces lunettes sont les plus performantes du marché, même si Microsoft pense déjà à une version V3 de ses HoloLens pour 2019, la version V2 étant abandonnée.


Marque	MICROSOFT
Modèle	HoloLens
Photo	
Mise à jour	19/10/2017
Prix	3299 € TTC Dev Edition
Disponibilité	Uniquement pour les développeur(Dev Edition) et professionnels (Community Edition)
Type d'affichage	2d/3d
Plateforme - OS	Windows 10
SDKs/Logiciels	HoloLens Emulator, UWP, Vuforia, Unity3D,
Applications possibles	Industriel, Educatif, Médical, etc...
Communauté	http://forums.hololens.com/
Résolution écran (en pixels)	720p
Fréquence	60hz
Camera(s)	4 caméras de compréhension de l'environnement, 1 caméra de profondeur, 1 caméra HD 2MP pour photo vidéo
Champ de vision	30° Horizontal, 17,5° vertical
Processeur	Intel Atom x5-Z8100 - 1,04 GHz
RAM	64GB Flash, 2Go RAM
Mémoire interne	64Go
Poids	579g
Accéléromètre	Oui
Gyroscope	Oui
Magnétomètre	Oui
Connectivité	Wifi, Bluetooth
Liens des caractéristiques techniques	https://developer.microsoft.com

Figure 1 – Caractéristiques techniques des HoloLens [WWW4]

En voyant les caractéristiques techniques on peut comprendre très vite pourquoi ces lunettes font partie des meilleurs lunettes de réalité augmentée du moment, celles-ci ont un grand potentiel.

Cependant malgré de très bonnes performances, il existe un énorme défaut à ces lunettes, qui est le champ de vision. Ce champ de vision est d'environ 36° plus précisément, ce qui est très peu en comparant à notre réel champ de vision.

Cela implique un problème qui est le fait de devoir bouger notre tête pour observer un objet qui est trop grand pour le champ des lunettes, ce problème est surtout présent quand on se rapproche d'un objet fixé pour voir les détails de celui-ci.

3 Marché et domaine d'application

3.1 Le marché de la réalité augmentée

En tout premier lieu, comme le secteur de la réalité augmentée est en pleine expansion, une multitude de domaines professionnels voulant se "technologiser" s'intéressent à cette voie qui semble prometteuse.

Les lunettes de Microsoft sont pour le moment présentes sur le marché, elles étaient, il y a peu, seulement ouvertes au monde des développeurs avec un kit de développement coûtant 3299€ ; mais on peut actuellement voir sur le site web de Microsoft qu'une version "Community" à 5489€ est présente, il est donc possible en étant un professionnel d'acheter les HoloLens depuis début Octobre. La version grand public n'étant toujours pas annoncée.

La différence majeure entre les 2 packs est que l'un est seulement ouvert aux développeurs (kit de développement) et l'autre est ouvert aux développeurs et aux entreprises (version commerciale).

Pour ce qui est du marché de la réalité augmentée celui ci est très disputé entre les diverses paires de lunettes existantes ainsi que les smartphones qui peuvent utiliser la réalité augmentée tel que le nouveau smartphone de chez Apple, l'iPhone X.







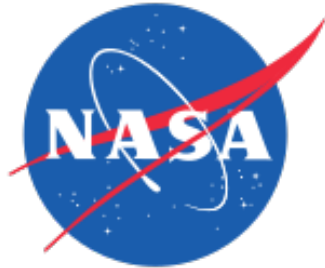
Marque	MICROSOFT	EPSON	EPSON	EPSON	ODG	METAVISION
Modèle	Hololens	MOVERIO BT-2000	MOVERIO BT-300	MOVERIO BT-200	R-7	Meta 2
Photo						
Mise à jour	19/10/2017	19/10/2017	19/10/2017	19/10/2017	19/10/2017	19/10/2017
Prix	3299 € TTC Dev Edition	3.120,00 € TTC	849 € TTC	699,00 € TTC	\$ 2.750.00	\$1495
Disponibilité	Uniquement pour les développeurs (Dev Edition) et professionnels (Community Edition)	Disponible	Disponible	Disponible	Disponible	Disponible
Type d'affichage	2d/3d	2d/3d (Side by side)	2d/3d (Side by side)		2d/3d	2d/3d
Plateforme - OS	Windows 10	Android	Android	Android	ReticleOS (Andoird)	Application Windows
SDKs/Logiciels	Hololens Emulator, UWP, Vuforia, Unity3D,	SDK Android, SDK by Espon, Vuforia, Wikitude	SDK Android, SDK by Espon, Vuforia, Wikitude	SDK Android, SDK by Espon, Vuforia, Wikitude	SDK propriétaire, Wikitude	SDK propriétaire
Applications possibles	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...
Communauté	http://forums.hololens.com/	https://moverio.epson.com/?_ga=1.2010	https://moverio.epson.com/?_ga=1.2010	Nombreux forums, mais non dédiés	-	https://www.metavision.com/faq
Résolution écran (en pixels)	720p	960x540	1280 x 720	960x540	720p	2.5K
Fréquence	60hz	60Hz	30Hz	60Hz		60Hz
Camera(s)	4 caméras de compréhension de l'environnement, 1 caméra de profondeur, 1 caméra HD 2MP pour photo vidéo	2 caméras, 15M, 1 0.3M pour la profondeur	5M	VGA	1080p 60fps et 720p 120fps	720p + capteur infrarouge 240p
Champ de vision	30° Horizontal, 17,5° vertical	23°	23°	23°	20°	90°
Processeur	Intel Atom x5-Z8100 - 1,04 GHz	Double noyau de 1,2 GHz TI OMAP 4460	Intel® Atom™ x5 1,44 GHz Quad Core	Processeur à double noyau de 1,2 GHz TI OMAP 4460	Qualcomm Snapdragon™ 805 2.7GHz quad-core Processor	Intel i5 1,5GHz
RAM	64GB Flash, 2Go RAM	1Go	2Go	1Go	3Go	4Go
Mémoire interne	64Go	8Go	16Go	8Go	64Go	128Go SSD
Poids	579g	290g (lunettes) - 265g (télécommande)	69g	88g (lunettes) - 124g (télécommande)	136g	
Accéléromètre	Oui	Oui	Oui	Oui	Oui	
Gyroscope	Oui	Oui	Oui	Oui	Oui	
Magnétomètre	Oui	Oui	Oui	Oui	Oui	
Connectivité	Wifi, Bluetooth	Wifi, Bluetooth, Micro USB	Wifi, Bluetooth	Wifi, Bluetooth	Wifi, Bluetooth, Micro USB	Wifi, Bluetooth, USB
Liens des caractéristiques techniques	https://developer.microsoft.com	https://www.epson.fr/product	http://www.epson.de/en/prod	https://www.epson.fr/product	http://www.osterhoutgroup.co	

Figure 2 – Caractéristiques techniques des différentes lunettes d'AR [WWW13]

Il existe donc, comme on peut le voir dans ce tableau, une multitude de lunettes déjà disponibles et la gamme de prix peut varier de plusieurs centaines d'euros, à plusieurs milliers, c'est

pourquoi il faut bien cibler son besoin afin de choisir au mieux ses lunettes.

Malgré une sortie tardive vis à vis des concurrents sur le marché, les HoloLens de Microsoft s'en sortent à merveille avec plusieurs partenariats tel que la NASA (Aérospatial), Lowe's (Equipeement construction Immobiliere), Stryker(Equipement medical).



(a) NASA



Improving Home Improvement™

(b) Lowe's

stryker®

(c) Stryker

3.2 Domaine d'application

1. Médical

La réalité virtuelle et la réalité augmentée sont en pleine expansion dans le domaine médical. On a pu voir plusieurs types d'application de la réalité virtuelle dans ce milieu surtout pour l'apprentissage car cela s'y prête parfaitement.

Maintenant avec l'apparition de la réalité augmentée nous pouvons importer des objets virtuels dans la réalité; cela ouvre une multitude de possibilités, telles que l'assistance chirurgicale.

Nous pouvons prendre l'exemple de Scopis qui est une firme Allemande qui travaille dans le domaine de la technologie médicale. On peut voir que depuis peu ils travaillent sur une application d'assistance chirurgicale pour les chirurgiens orthopédiques spécialisés dans les opérations de la colonne vertébrale.

Cette application nommée "Holographic Navigation Platform" permet l'optimisation des interventions chirurgicales; ce qui induit une baisse du temps d'intervention ainsi qu'une qualité améliorée de celle-ci.

Il existe bien d'autres applications médicales telles que celle créée par l'entreprise Touch Surgery qui est à l'origine de 200 programmes d'entraînement de procédures chirurgicales.

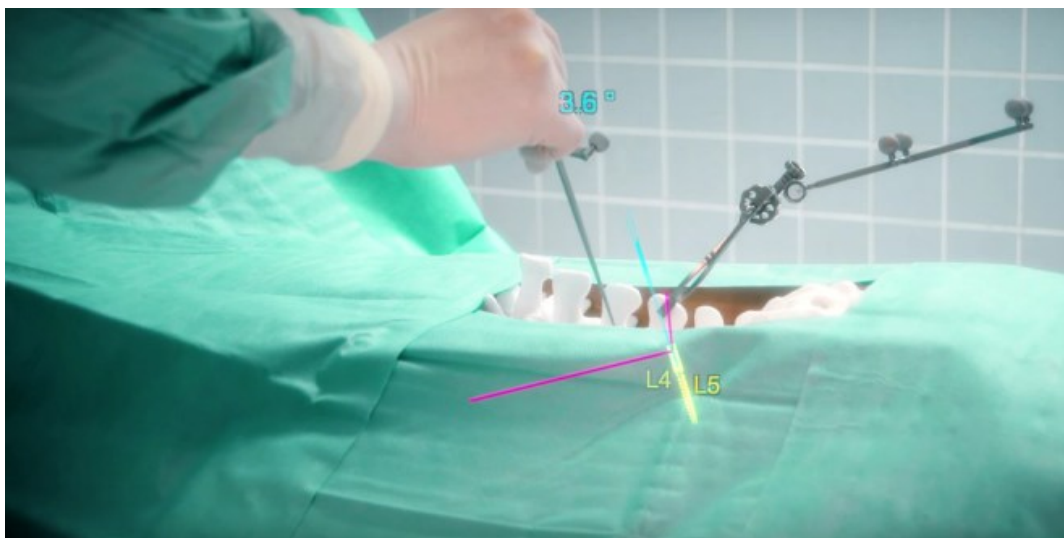


Figure 3 – Chirurgie du dos assistée par HoloLens [WWW6] [WWW5]

2. Automobile

Utilisée principalement pour le design des nouvelles voitures, la réalité augmentée est déjà présente dans de grandes firmes automobiles tel que Ford.

Depuis plus d'un an, Ford utilise les HoloLens afin de gagner en rapidité sur la conception des nouvelles voitures ; cela leur permet de coupler la réalité augmentée (utilisée pour des changements mineurs) avec une ancienne méthode qu'est l'utilisation de miniatures en argile.

Cela leur a permis de gagner en précision mais aussi d'obtenir un gain de temps non négligeable.

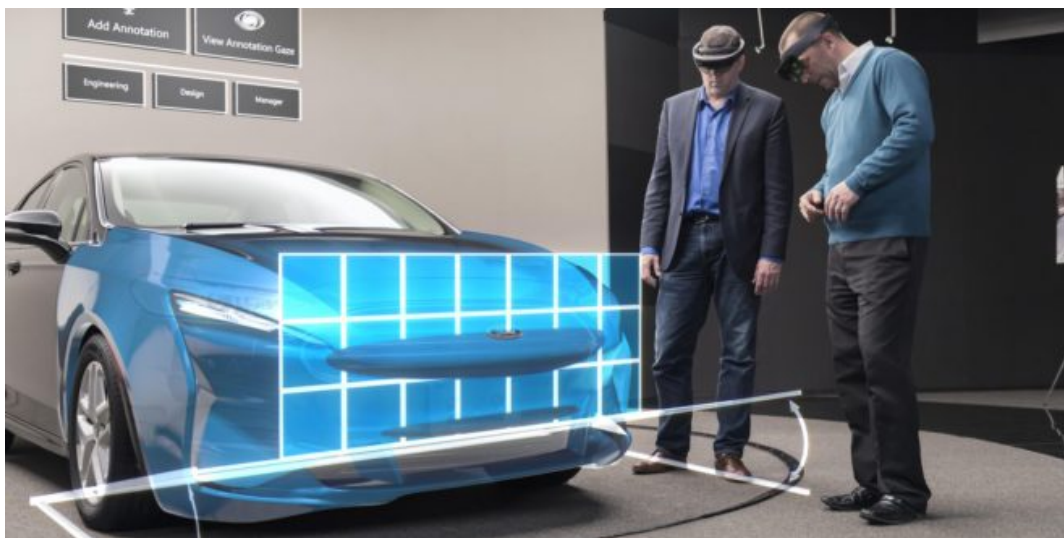


Figure 4 – Utilisation automobile design [WWW14]

3. Aéronautique

Dans le milieu de l'aéronautique la réalité augmentée commence à se démocratiser et il n'y a pas besoin d'aller à l'autre bout du monde pour en voir les conséquences.

Ici en France Microsoft a effectué un partenariat avec Dassault Aviation, les HoloLens serviront pour accompagner les équipes dans leur quotidien.

Utiliser principalement lors de la maintenance des avions de chasse tel que le Rafale , les lunettes de chez Microsoft augmenteront l'efficacité des employés mais les HoloLens permettront également de gagner beaucoup de temps sur ces maintenances.

Il reste encore beaucoup d'applications possibles au secteur de l'aéronautique tel que l'accompagnement lors de la conception ou la fabrication d'un avion.



Figure 5 – Logo Dassault Aviation [WWW8]

4. Archéologie

Nous avons pu voir dans des publications à partir du 2 Novembre 2017 qu'une nouvelle cavité dans la pyramide de Kheops a été découverte.

Il y a également eu récemment un reportage sur France 5 le mardi 28 Novembre 2017 où l'on peut voir très clairement l'utilisation des HoloLens afin de manipuler la pyramide en surimpression et voir cette nouvelle cavité. (Voir 2ème teaser du reportage¹)



Figure 6 – Kheops en réalité augmentée [WWW15]

Il existe bien d'autres secteurs d'activités dans lesquels la réalité augmentée et les HoloLens peuvent s'impliquer comme l'immobilier, le divertissement (Pokemon Go depuis 2016) et même la vie de tous les jours avec des informations supplémentaires à apporter à notre quotidien. Cependant cette technologie est encore une technologie naissante, il faut donc lui laisser le temps de mûrir et de s'implanter dans ces secteurs d'activités.

4 Interactions

Afin d'interagir avec les lunettes HoloLens, il existe une multitude de moyens afin de laisser libre cours à notre imagination et notre créativité. Nous allons donc voir comment interagir avec ces lunettes et les possibles applications qui peuvent leur être attribuées.

1. <http://www.francetvpro.fr/france-5/communiqués-de-presse/kheops-mysterieuses-decouvertes-8898148>



Figure 7 – Type d'action

4.1 Vision

Tout d'abord parlons de la vision/regard ; lors de l'utilisation des HoloLens on peut très bien voir ce qu'on pourrait définir comme curseur au centre de notre champ de vision. Il faut faire attention car le curseur est "fixe" et suit le mouvement de notre tête et non le mouvement de nos yeux.

Ce curseur va être utilisé comme une souris, c'est ce curseur qui suit notre regard et qui va permettre de prendre pour cible tout les objets 3d que nous pouvons voir.

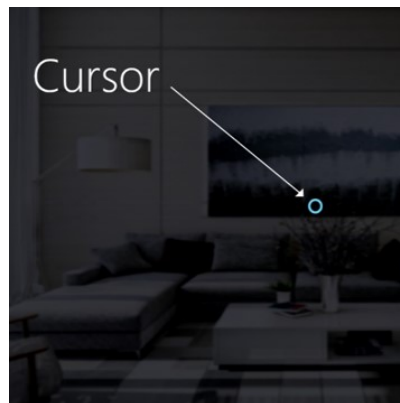


Figure 8 – Curseur HoloLens

Cependant si vous bougez votre regard en dehors de l'objet et que le champ du curseur n'est plus dessus, l'objet en question ne sera plus ciblé et aucune option d'interaction dessus ne sera possible car implicitement vous n'avez plus aucune intention d'interagir avec, si votre regard n'est plus dessus.

Cependant cette interaction visuelle se limite à quelques principes qui sont propres à la vision tels que le ciblage et non la sélection qui elle va vous permettre de modifier la place de l'objet dans l'espace.

Pour pouvoir réellement sélectionner l'objet et avoir accès à une multitude d'actions le concernant il va falloir passer par d'autres interactions tel que la gestuelle ou la voix.

De plus une fois la sélection faite par exemple pour bouger l'objet celui-ci va s'adapter à votre regard et donc vous pourrez bouger cet objet en bougeant la tête.

Les HoloLens sont capables de mettre en relation le mapping d'une pièce avec ce regard, c'est pourquoi les objets virtuels que vous bougerez pourront se heurter à des éléments réels.

En conclusion cette action permet de cibler un élément 3D en utilisant le regard afin que cet élément puisse être ensuite traité par une action à travers la gestuelle ou la voix.

4.2 Gestuelle

En premier lieu, la gestuelle fait partie des actions dites "interface" qui vont permettre de sélectionner et de modifier un objet 3d dans l'espace.

Plusieurs points sont importants vis à vis de ce geste qui paraît banal.

Le "Gaze and commit" est le principe fondamental des actions liées à la gestuelle. Ce principe consiste en une suite d'actions que sont le regard afin de cibler notre objet et ensuite un mouvement de sélection avec nos doigts afin d'interagir avec celui-ci.

Le deuxième point important, même s'il paraît implicite et normal pour des utilisateurs, est la reconnaissance des mains. Les HoloLens sont capables de reconnaître le mouvement des mains en suivant leurs positions.

Cependant les HoloLens ne reconnaissent les mains que dans une position spécifique (voir figure 9), la reconnaissance se fait seulement en voyant le dos de la main ; tout autre action sera ignorée par les HoloLens pour la sélection.

Cette action peut être difficile à prendre en main au début car il faut bien positionner sa main en face de soi et ne pas garder le bras le long du corps car il faut que la caméra puisse voir votre main afin d'analyser l'action qui est à effectuer.

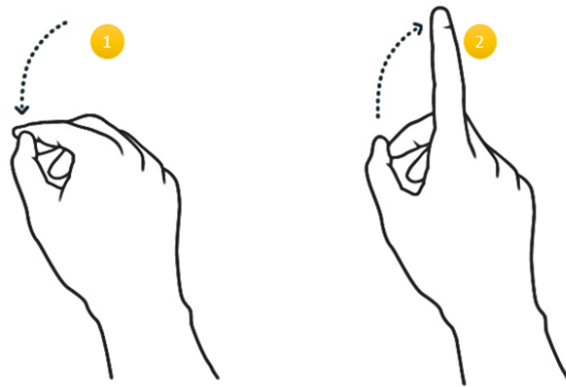


Figure 9 – Sélectionner (Air tap)

Cependant il existe une multitude de gestuelles enregistrées dans les HoloLens que nous pouvons également utiliser comme l'ouverture de menu et fermeture d'applications courantes, etc... et dans ce cas précis l'application va devoir reconnaître autre chose que le dos de votre main.

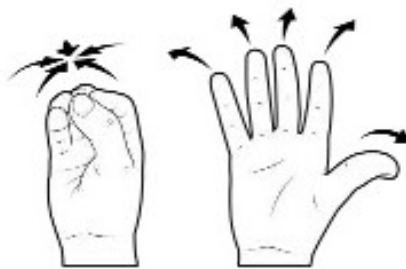


Figure 10 – Ouverture menu et fermeture application (Bloom)

Bien entendu il existe des interactions beaucoup plus complexes comme la manipulation d'un

objet, dans le mouvement de sélection on pince puis relâche mais on peut également juste pincer et ne pas relâcher afin de pouvoir bouger certains objets qui peuvent être bougés.

4.3 Voix

Dans les HoloLens il existe déjà une multitude de commandes vocales nous permettant d'interagir avec les objets 3d et pouvant ainsi remplacer la gestuelle.

De plus comme les lunettes sont produites par Microsoft nous avons accès à Cortana qui peut nous suivre et nous guider si on le lui demande.

Pour le moment la reconnaissance de la voix est seulement pour la langue anglaise ; on ne peut donc pas parler français et être reconnu par les HoloLens.

Toutes ces actions peuvent donc être une piste possible afin d'améliorer le confort du chirurgien. Prenons donc l'exemple d'une intervention, le chirurgien se doit d'avoir les mains libres, donc des actions utilisant les mains sont le plus souvent à éviter car un moment critique peut précipitamment survenir . Il est donc préférable d'utiliser un mode d'interaction par voix pour faire apparaître et disparaître par exemple l'objet 3d tout en gardant sa position dans l'espace. C'est une problématique en dehors de notre sujet, pour l'instant.

Pour plus d'informations, voir le site Microsoft dans la partie basics :

<https://developer.microsoft.com/en-us/windows/mixed-reality/development>

5

Le Mapping

1 Principe du mapping

Le mapping fournit une représentation détaillée des surfaces du monde réel dans l'environnement HoloLens, permettant aux développeurs de créer une expérience de réalité augmentée convaincante. En fusionnant le monde réel avec le monde virtuel, une application peut donner l'impression que les hologrammes sont réels.[2]

Pour obtenir ce résultat les HoloLens vont scanner la pièce grâce à une caméra qui va récupérer des données par infrarouge [WWW17]; ces infrarouges se refléteront sur les différentes surfaces à l'image d'un sonar sauf qu'ici on utilise la lumière et non le son; et en recoupant les données obtenues par la caméra on obtiendras après cela un maillage 3D formé avec des petits triangles (cf figure 1).

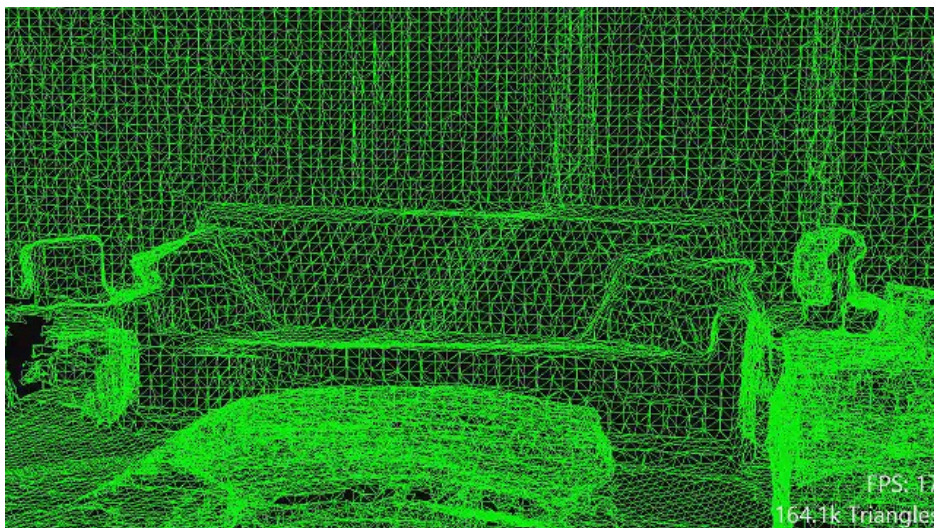


Figure 1 – Mapping d'une pièce

Ceci va donc permettre de reconnaître les différents volumes dans une pièce tels qu'une plateforme, un mur, le sol, etc.

Les éléments virtuels doivent se comporter comme un élément réel afin d'améliorer l'immersion

au sein de la réalité augmentée.

Ce mapping sert donc à rendre plus réels les éléments virtuels, par exemple en les occultant lorsqu'ils sont derrière un mur ou en évitant, quand on les place, que ceux-ci traversent un volume de la pièce comme une table, des chaises ou autres.

Cependant le mapping n'est pas très précis dans certains cas et peut laisser des trous ou même inventer des surfaces qui n'existent pas, nous allons donc voir les erreurs que nous pouvons rencontrer ainsi que les facteurs produisant ces erreurs afin d'éviter de les rencontrer.

2 Erreurs de mapping

2.1 Les différentes erreurs

Voici les différentes erreurs que l'on peut rencontrer dans le mapping d'une pièce. [WWW12] [WWW1] :

1. Les trous

Certains maillages du mapping n'ont pas été correctement effectués et cela peut contribuer à la formation de trous. Le problème majeur est que le mapping n'ayant pas de maillage à cet endroit, les opérations de type occultations si c'est un mur, ou même collision ne vont pas fonctionner car techniquement ce sera un trou qui sera à cette place.

De ce fait un objet pourra être vu alors qu'il y a un mur, ou même celui-ci pourrait traverser une table, etc. Ce genre de problème nuit gravement à l'immersion de l'utilisateur qui peut ne pas comprendre la place d'un objet dans l'espace qui l'entoure.

2. Les hallucinations

Ici c'est l'inverse des trous. Nous allons avoir un maillage d'une partie de la pièce où rien n'est présent. Ce qui implique que les objets que nous allons poser à ces endroits auront des problèmes car ils pourront subir des collisions en l'air par exemple ; vous pourriez même voir un objet disparaître par occultations si vous passez derrière cette hallucination qui n'existe pas.

3. Les décalages

Les décalages sont un peu moins fréquents que les hallucinations et les trous mais ils sont existants. Cela impliquera implicitement des hallucinations et des trous car les surfaces réelles auront subi un décalage virtuellement ; nous aurons donc un maillage décalé sur certaines parties ce qui entraînera une possibilité par exemple d'avoir un morceau du maillage d'une table dans le vide à côté de cette même table et donc d'avoir à l'opposé de la table un trou.

2.2 Quelques facteurs produisant ces erreurs

Les erreurs que nous avons vues précédemment sont dues à des facteurs bien précis dont nous allons énumérer les principaux.

1. Les mouvements de l'utilisateur

La caméra utilisée lors du scan est une caméra très importante c'est pourquoi il faut faire attention à ses caractéristiques ; elle gère les données dans un cône de 70° à un minimum

de 0.8 mètres et à un maximum de 3.1 mètre.

Les surfaces à scanner ne seront donc analysées que dans cet intervalle précis.

Il faut donc bien tourner la tête afin d'avoir le meilleur scan et donc le meilleur mapping de la pièce possible. Par exemple si vous ne levez jamais la tête vous n'aurez jamais le plafond comme surface ou si un objet est trop proche ou trop loin, il ne sera jamais interprété.

C'est pourquoi les mouvements de l'utilisateur sont vraiment très importants pour avoir un mappage de qualité.

2. Les mouvements dans l'environnement

Parlons maintenant des mouvements dans l'environnement. Les HoloLens s'adaptent assez vite aux changements dans l'environnement si ces changements sont effectifs dans le champ des caméras et donc clairement visibles.

Effectivement si des changements sont effectués hors champs lorsque nous allons tourner la tête pour revenir sur un endroit où il y a eu du changement, les HoloLens peuvent avoir un retard sur le scan et qui dit retard dit qu'il pourra éventuellement y avoir des trous voir des bosses/hallucinations lors du mapping.

3. La lumière

La lumière est un facteur très gênant lors du mapping d'une pièce car le mapping se base sur un principe de sonar par lumière comme expliqué précédemment, si la différence de lumière est vraiment importante cela peut entraîner des erreurs de mapping.

Nous pouvons prendre l'exemple par exemple de rayon lumineux provenant d'une fenêtre, si la pièce possède des coins n'ayant pas une grande luminosité nous pouvons avoir des déformations dues à cette grande différence de luminosité.

De plus cela ne se limite pas qu'aux rayons lumineux ; les surfaces réfléchissantes et brillantes peuvent, elles aussi, interférer lors du scan de la pièce en interprétant mal cette réflexion lumineuse et en créant des surfaces invisibles n'existant pas réellement.

4. Les types de surface

La lumière et les types de surface sont étroitement liés car ici nous allons parler des types de surface et comment ils influent sur l'analyse d'un objet selon sa surface.

Certaines surfaces ont une matière qui reflète mieux la lumière infrarouge.

Il est donc possible que quelquefois une surface plutôt sombre ne reflète quasiment pas la lumière ; il faudra donc se rapprocher pour que cet élément puisse être correctement scanné.

Il existe donc un objet précis qu'il faut absolument éviter dans ce genre de manipulation : le miroir ; cela peut générer des trous et des hallucinations dus à la réflexion d'un environnement.

3 Conclusion

En conclusion, le mapping proposé sur les HoloLens est un principe intéressant. Effectivement grâce à cette technique le monde de la réalité augmentée a pu être perfectionné et être amené à un tout autre niveau, les applications envisageables apportées par un tel système sont immenses. Cependant c'est une technique très compliquée, dû au fait que cela nécessite un environnement stable, avec des conditions de scan quasi parfaites.

Néanmoins le mapping proposé par les HoloLens est convaincant même si parfois quelques bugs peuvent s'immiscer à cause de l'environnement.

Dans notre cas, le milieu de la chirurgie est le plus souvent bien éclairé surtout autour du patient. Il ne devrait y avoir aucun soucis dans un environnement de ce type lors du scan. Or des erreurs peuvent quand même s'immiscer, des tests seront donc à prévoir afin d'examiner si ces conditions sont parfaites et ainsi calculer une marge d'erreur qui doit absolument être proche du 0 .

Troisième partie

Analyse et conception

6

Sujet approfondi

Suite à notre état de l'art il est plus facile de cerner le sujet, de comprendre les tâches à accomplir et comment les accomplir. Pour notre projet les HoloLens que nous utiliserons devront donc être couplé à un environnement logiciel stable permettant la configuration et le développement de notre application.

Nous avons donc 3 objectifs principaux décrits dans la partie objectifs. Le premier est surtout lié à la configuration des lunettes et à l'importation de nos modèles 3D.

Le second objectif est lié au matching; pour cela il faut bien comprendre que pour afficher en surimpression un objet à une place précise les lunettes HoloLens ont besoin d'un point de repère appelé "masque" comme décrit dans notre état de l'art; il faudra donc trouvé une solution à ce matching.

Le dernier objectif est l'un des plus importants; il faudra que l'omoplate en surimpression (vert) soit bien positionné suite au matching et qu'elle ne bouge plus. Ici nous devons bien comprendre que l'objet en question a une place fixe dans l'espace, cet objectif est vraiment important.

Pour résoudre les problématiques, il fallait proposer des solutions viables et précises.

7

Solutions proposées


Nous allons donc voir dans cette partie les différentes solutions proposées pour chaque objectif.


1 Solution : Importation


Pour le premier objectif qui consiste en l'importation de notre modèle 3D. Il existera donc des modèles 3D(vert) créés sous Blender et exportés en .obj. Ces modèles seront donc à importer dans Unity. [WWW18] Pour ce qui est de l'importation du modèle pour le masque(bleu) celui ci ne sera pas 3D. Nous verrons dans la partie matching les causes de cette importation 2D. Pour importer ce modèle 2D il faudra donc utiliser Target Manager un outil de Vuforia permettant de gérer tout les masques que nous pouvons avoir.


Add Target

Type:


Single Image


Cuboid


Cylinder


3D Object

File:

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Figure 1 – Ajouter un masque

2 Solution : Matching

Cette partie matching est une partie importante du projet. Comme cité précédemment nous allons avoir un masque 2D comme point de repère et non 3D. Notre masque 2D sera un sticker que nous devons importer comme une image à l'aide du Target Manager de Vuforia.

Mais comme on peut le voir dans la capture d'écran précédente, il existe un onglet 3D Object donc pourquoi ne pas l'utiliser ?

Tout simplement, ce type 3D Object n'accepte qu'un format bien particulier (.OD) qui est la sortie de l'application mobile de Vuforia (Vuforia Object Scanner).

Il n'y a aucun moyen à l'heure actuelle de transformer un .OBJ en .OD. Dans notre cas l'application mobile est inutilisable, car il faudrait imprimer en 3D le .OBJ correspondant à notre omoplate choisie et ensuite utiliser l'application Vuforia pour scanner cette omoplate imprimée, afin d'avoir un fichier .OD. Nous avons donc oublié cette idée car beaucoup trop fastidieuse pour obtenir un .OD.

Vuforia a également présenté en 2017 un outil très intéressant (Vuforia Model Target) qui n'est pas encore disponible et qui permettrait d'importer cette fois-ci nos .OBJ. Il n'y aurait plus besoin de passer par un .OD.

Malheureusement nous avons essayé de prendre contact et n'avons pas pu obtenir l'accès à ce logiciel.¹

C'est pourquoi nous utiliserons un masque 2D dans cette version du projet. A l'avenir l'outil Vuforia Model Target sera sûrement intégré dans les prochaines versions, tels que Vuforia 7.

Ceci est une possible amélioration que pourra subir l'application pour améliorer le matching.

3 Solution : Poursuite de l'objet

Cette section se rapportera plus à une partie sur le code. Ici il faudra aller voir dans les classes de Vuforia telle que la classe Tracker pour pouvoir modifier le comportement de Vuforia pour qu'il corresponde à notre base méthodologique et ainsi avoir une poursuite de l'objet sans que l'objet en surimpression disparaisse ou ne soit pas stable.

Cette objet sera fixe et ne devra donc pas bouger pour pouvoir obtenir une précision chirurgicale à 1mm.

4 Architecture

Ce projet n'ayant pas de code avec des classes bien précises, j'ai donc fait 2 diagrammes permettant de comprendre l'organisation des objets dans ma scène Unity et un autre diagramme permettant de comprendre l'architecture de l'application C# générée par Unity pour UWP. Nous verrons donc ces 2 diagrammes dans les parties suivantes.

4.1 Architecture des objets dans la scène

Voici l'architecture des objets qui ont été mis en place dans notre scène sur Unity.

1. Exemple avec Vuforia Model Target : [WWW16]

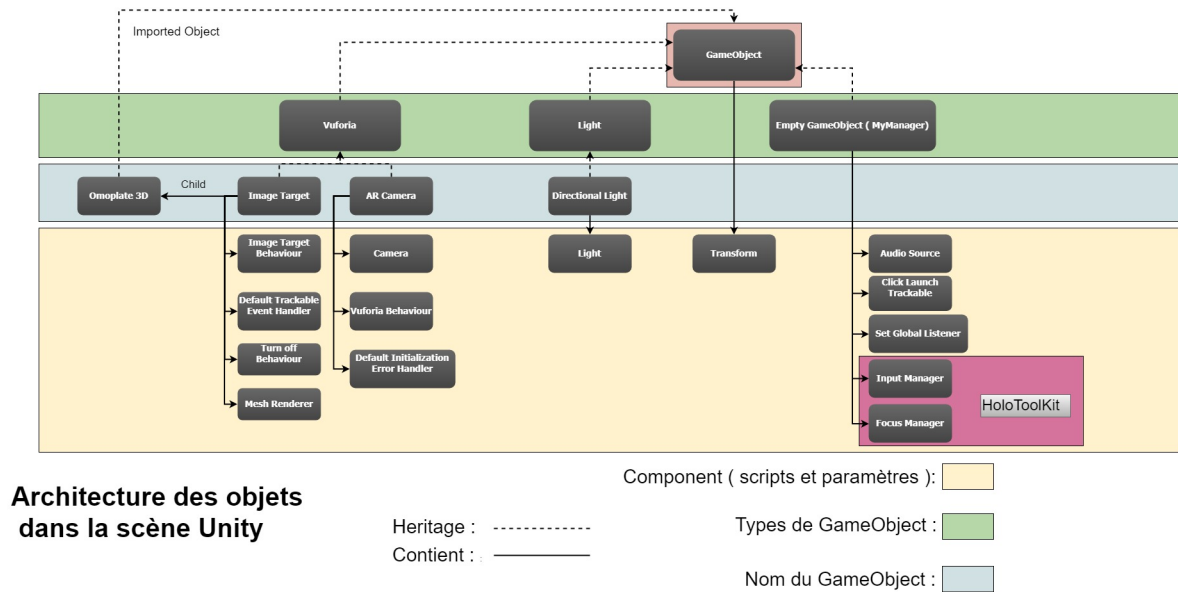


Figure 2 – Architecture des objets dans la scène

4.2 Architecture du code C#

Comme on peut le voir dans le schéma ci-dessous, le code C généré depuis Unity est en fait découpé en 2 parties.

La deuxième partie sert pour UWP et comprend beaucoup de librairie mais surtout le point d'entrée de l'application.

C'est-à-dire le script App.cs qui va lancer toute l'application.

Il y a également plusieurs dossiers contenant une multitude de fichier.

Dans les assets nous allons retrouver les ressources images de Microsoft.

Dans unprocessed, c'est là que nous aurons un lien avec Unity car c'est ici que tous les fichiers .dll et .pdb, relatif à Unity , se trouvent.

Il y a également d'autres dossiers et fichiers mais ceux listés ci-dessus sont les principaux.

[WWW9]

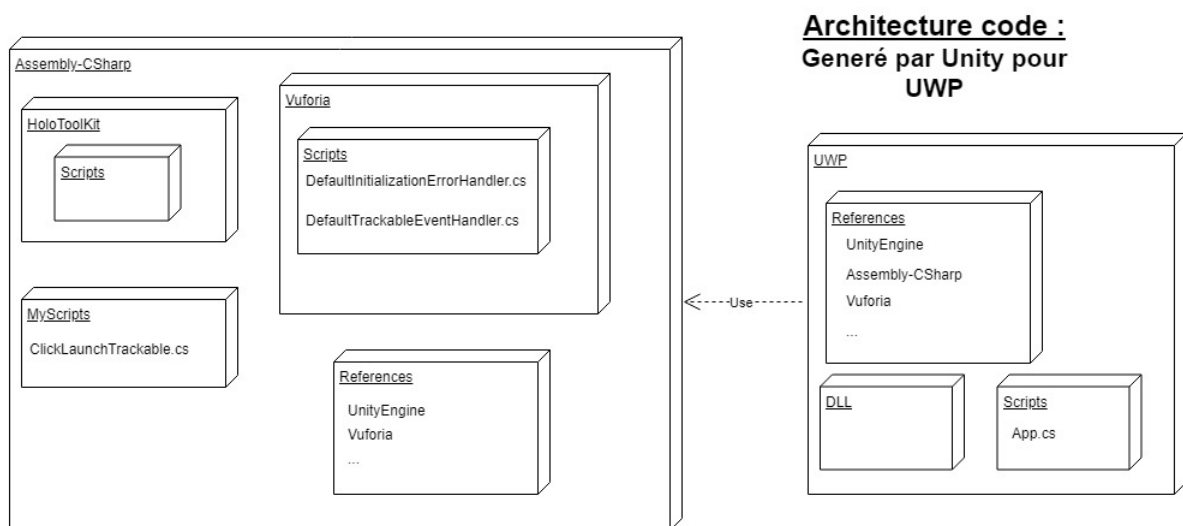


Figure 3 – Architecture du code C#

8

Solutions finales du projet

Nous allons donc voir dans cette partie les différentes solutions proposées à l'issue du projet pour chaque objectif en comparaison des solutions que j'avais proposées dans la première partie du PRD.

1 Solution finale : Objectif Importation

Dans un premier temps, l'importation de l'objet en surimpression se fera comme décrite lors de la première partie.

Nous allons récupérer une représentation 3D de l'omoplate du patient depuis un scanner, puis celle-ci sera traitée dans Blender en diminuant le nombre de points(modèle 3D vert).

Ensuite, grâce à cet objet simplifié dans Blender, nous allons générer un fichier au format .obj. Ce fichier sera ensuite importé à Unity. Pour la partie technique de cette importation, il est possible de voir comment la faire dans la partie développement de ce rapport.

Pour finir, il fallait importer notre marqueur.

Nous avons donc décidé à la base d'importer via TargetManager un marqueur 2D. Pour l'application utilisant ImageTarget, nous sommes restés sur cette idée.

Cependant, pour l'application utilisant ModelTarget, il fallait utiliser le nouvel outil de Vuforia qui est sorti avec la version 7.0.

Ce nouvel outil nommé ModelTargetGenerator va donc permettre de prendre en entrée un fichier .obj comme pour notre modèle 3D(vert).

Il en ressortira un objet utilisable par Unity. (Voir partie [D.4.1] des annexes)

2 Solution finale : Objectif Matching

Pour la première partie des livrables de ce PRD, une application utilisant ImageTarget a été réalisée. Celle-ci consiste à utiliser un marqueur 2D au préalable importer avec le premier objectif et à afficher notre modèle3D(vert) en surimpression au bon endroit sur l'omoplate réelle du patient.

Une maquette a donc été réalisée avec un marqueur 2D positionné à un endroit précis comparé à notre omoplate. (voir partie [D.3] des annexes)

Donc le matching avec ce marqueur 2D ce fait via des mesures précises qui pourront très bien être récupérées durant le scanner de l'épaule du patient dans une position précise 15 avant l'opération. Celui-ci devra être remis dans la même position lors de l'opération pour garder les mesures.

Pour l'application utilisant ModelTarget le matching doit se faire à partir des guideView. (voir partie [D.4.3] des annexes)

Cependant, le résultat attendu n'étant pas fonctionnel, cette application et l'étude dont elle découle, représente une piste pour le futur du projet.

3 Solution finale : Objectif Poursuite de l'objet

Pour le 3ème et dernier objectif, je suis resté sur l'idée de base qui était de modifier un script de Vuforia.

Plus précisément, ce script est "DefaultTrackableEventHandler". Les changements effectués sur ce script, pour modifier le comportement et l'adapter à notre projet, sont précisés dans la partie [D.3.4.1] des annexes de ce rapport.

J'ai donc respecté la base méthodologique qui avait été présentée lors de la première partie du PRD dans les spécifications.

Quatrième partie

Bilan

9

Bilan PRD1

1 Difficultés rencontrées

En premier lieu ce PRD est sur un sujet innovant sur une technologie récente sortie il y a 1 an. Il y a donc en ligne très peu de documentation comparée à certaines technologies. Ce fut la première difficulté que j'ai rencontrée dans ce projet, le manque d'informations évidentes sur le sujet.

Au niveau de la gestion du projet, je pense avoir passé trop de temps sur l'état de l'art même si dans notre cas la partie analyse devait être plus faible que la partie état de l'art en raison de la nature du projet sur une technologie récente. Je reste donc dubitatif sur le choix que j'ai fait de passer plus de temps à faire des recherches sur le fonctionnement des lunettes adapté à notre projet.

Pour finir, la partie mise en place de l'environnement fut un vrai calvaire. Le problème étant qu'il y a des gros problèmes de versions entre les logiciels utilisés. Pour avoir un environnement stable qui peut fournir un logiciel qui s'exécute à la fin, il faudra s'attarder aux différents bugs possibles. J'ai donc dû lire les différents patches et bugs connus des versions de Visual et de Unity, afin d'arriver enfin à une version stable.

Je n'ai également pas pu prendre la version de Jordan de l'année dernière, car Vuforia est passé en SDK intégré dans les versions récentes de Unity. Il ne met plus à disposition leurs SDK pour cette plate-forme comme il le faisait avec les anciennes versions de Unity.

2 Bilan personnel

Dans cette partie je vais parler de ce qu'a pu m'apporter cette première partie du projet recherche et développement.

En premier lieu, il fallait que je m'organise sur comment j'allais répartir les tâches de ma partie recherche et j'étais un peu perdu au bout d'un moment mais j'ai pu apprendre comment m'organiser plus efficacement dans mon travail. De plus j'ai bien été encadré tout au long de cette première partie ce qui m'a permis de ne pas me disperser sur mon sujet.

J'ai également apprécié la partie état de l'art; le domaine de la réalité augmentée est vaste et il y a plusieurs informations sur l'aspect global de ce qu'est la réalité augmentée avec beaucoup de suppositions sur ce que l'on pourrait faire. Cette partie fut très intéressante même si j'ai eu des

difficultés sur la partie technique comme cité précédemment.

Pour la partie analyse il fallait que je puisse trouver des solutions aux problématiques posées pour mon sujet. Pour cela j'ai utilisé les ébauches faites l'année dernière et j'ai pu ainsi améliorer la solution et trouver des réponses pour le sujet de cette année. Ceci devrait donc conclure à un produit final qui respectera les problématiques posées en début de projet.

Pour la partie mise en place de l'environnement, j'ai pu constater que nos premiers choix ne sont pas toujours les bons. J'avais préconisé de prendre des versions de logiciels qui au final ne se sont pas avérées stables. J'ai donc eu un esprit critique sur mon propre travail et j'ai donc du rebondir et trouver rapidement une solution pour avoir un environnement stable avant la fin de la première partie du projet.

3 Conclusion

En conclusion, malgré plusieurs difficultés rencontrées tout au long du projet j'ai su rebondir pour finalement avoir un état de l'art qui m'a permis d'obtenir les connaissances nécessaires pour bien cerner le projet et bien l'analyser.

J'ai donc pu en ressortir des solutions pour nos objectifs.

Pour finir, l'environnement étant mis en place je vais pouvoir directement passer à la phase de développement. Je ne suis donc pas en retard sur le planning que j'avais prévu au début de mon PRD même si celui-ci a été modifié à quelques semaines près.

10

Bilan PRD2

1 Difficultés rencontrées

Dans cette partie du prd2, j'ai donc eu dans un premier temps encore des difficultés avec l'environnement, car voulant utiliser HoloToolKit, il fallait rajouter un élément dans notre environnement sensible aux versions.

Dans notre cas, il n'y avait pas de versions de l'HoloToolKit allant avec les versions de notre environnement donc j'ai dû repasser sur la dernière version stable de Visual Studio 2017 et depuis la partie du PRD1 un bug que j'avais été fixé et cette fois-ci cela fonctionnait et j'ai donc pu rajouter correctement HoloToolKit pour la gestion du tap (voir annexes du rapport). Ce genre de problème résume assez bien un des plus gros problèmes de ce type de projet utilisant des technologies qui évoluent très rapidement.

De plus, le fait que Vuforia soit un environnement très fermé n'améliore pas les choses et apporte une difficulté sur la compréhension du code qui s'effectue à l'arrière-plan de notre application. La deuxième application du projet utilise Model Target, cette fonctionnalité est sortie il y a peu et il existe encore très peu de documentation ou même d'éléments comparatifs sur lesquels se baser.

Pour moi, cela était vraiment difficile. À un moment du projet, je ne savais pas comment avancer car j'avais des bugs incompréhensibles par manque de connaissances dans ce domaine.

Celle-ci était impossible ou difficile à acquérir car impossible de naviguer dans le code des scripts de Vuforia pour comprendre comment cela fonctionne.

2 Bilan personnel

Dans cette deuxième partie du projet, j'ai donc appris à utiliser Unity qui est un environnement agréable d'utilisation. Cependant Vuforia restera un point noir sur la documentation accessible et le peu d'information qu'on peut avoir sur le fonctionnement de celui-ci.

J'ai pu également regarder comment les interactions avec les HoloLens fonctionnaient dans le code C car tous les scripts de l'HoloToolKit sont ouverts et c'est d'autant plus agréable à utiliser. Donc sur un point technique, j'ai pu vraiment étendre mes connaissances sur un domaine de l'informatique que je n'avais encore jamais pratiqué et c'était vraiment très instructif et plaisant. De plus je suis vraiment content d'avoir pu avoir la première application fonctionnelle avec un

marqueur 2D même si à terme ces marqueurs 2D devront disparaître pour laisser place à des marqueurs 3D.

J'en viens donc à la deuxième application, celle-ci me laisse un sentiment de frustration, car je n'ai pas pu aboutir à quelque chose de viable. Cependant, mon travail restera une piste pour les projets à venir.

En ce qui concerne la gestion de projet, j'ai pu m'apercevoir qu'il n'était pas si simple de réussir à respecter ce qu'on a pu mettre en place lors du début d'un projet avec un diagramme de gantt par exemple.

Dans mon cas, "ModelTargetGenerator" et "ModelTarget" sont sortis tardivement et cela a complètement changé mon diagramme de gantt. J'ai dû travailler sur 2 choses différentes à cause de changement, mais cela restait une hypothèse que nous avions émise dans la première partie du PRD.

3 Perspective du projet

Pour ce qui concerne les perspectives du projet, nous en avons parlé lors d'une réunion.

Il faudra continuer sur la piste d'un marqueur 3D et essayer au maximum de se défaire du sdk de Vuforia et de ModelTarget.

Pour cela, il faudra étudier la piste de la reconnaissance d'objets en essayant de faire des algorithmes à la main sans Vuforia et tester ceux-ci dans un environnement de réalité augmentée.

Une fois cela effectué, il faudra être capable de reconnaître la glène comme notre marqueur 3D et d'afficher un élément à partir de ce marqueur comme pour ImageTarget avec notre marqueur 2D.

4 Conclusion

En conclusion, bien que la deuxième application, avec un marqueur 3d, ne soit pas opérationnelle.

Celle avec un marqueur 2D l'est et nous avons fait une vidéo avec un autre projet de PEIP afin d'avoir une preuve scientifique du bon fonctionnement de l'application.

Ce projet fut très enrichissant dans mon parcours et m'a montré la difficulté qu'on pouvait rencontrer dans des projets de recherche sur un domaine en pleine évolution.

Dans ce projet j'ai très bien été encadré et ce fut vraiment agréable d'avoir des personnes à qui soumettre ces idées et qui peuvent soit les accepter ou non en apportant une argumentation.

Techniquement, selon moi Unity est un moteur 3D vraiment très puissant et j'ai vraiment apprécié d'utiliser cet environnement hormis le SDK Vuforia qui est très fermé.

De plus, j'ai vraiment été très impressionné et agréablement surpris par la réalité augmentée avec les HoloLens, ce genre de technologie sera celle de demain et pourra peut-être devenir le quotidien dans certains domaines comme celui de la médecine.

Pour finir, cette dimension de gros projet seul en fin d'études est un vrai plus à la formation d'ingénieur et permet d'entrevoir nos futur travaux au sein d'entreprises selon le milieu où nous travaillerons.

Annexes

A

Spécifications

1 Interfaces Homme/Machine

Dans le cadre de ce PRD et au vu de nos objectifs, la seule IHM sera les HoloLens, il n'y aura aucune interface dans l'application (par exemple pour avoir des informations supplémentaires). Ceci sortirait complètement des limites du projet défini au début de celui-ci. Effectivement une amélioration future, pour cette application dans un autre projet, serait de rajouter des informations et donc d'avoir une réelle interface.

2 Spécifications fonctionnelles

Importation des modèles 3D .obj (omoplate (verte)) dans la bdd.

ImportModel

Rôle : primaire

Spécification :

Entrée : Modèle 3D omoplate(verte) ".obj" + vignette(pour le moment ce sera une image 2D ".jpg")

Sortie : Null

Pré-conditions :

Fichier au format .obj

Post-conditions :

Null

Exportation des données vers l'application Hololens.

ExportData

Rôle : primaire

Spécification :

Entrée : Modèle 3D omoplate(verte) .obj + vignette .jpg

Sortie : Null

Pré-conditions :

Fichier au format .obj

Post-conditions :

Null

Cette partie des spécifications concerne donc la partie des données. Tout nos modèles, que ce soit l'omoplate (verte) qui sera un objet 3D ou une image de la vignette comme référence pour faire apparaître en surimpression l'omoplate verte, seront donc concernés.

A terme il faudra que la vignette disparaisse et que nous utilisions le masque(bleue) comme référence, ces spécifications seront donc à changer dans un futur projet.

Sélection de l'objet 3D.

SelectObject

Rôle : primaire

Spécification :

Entrée : Liste de modèle 3D

Sortie : Le modèle sélectionné

Pré-conditions :

Il faut un modèle 3D à afficher

Post-conditions :

Base méthodologique à suivre et qui s'exécute en fond

Cacher l'omoplate

HideOmop

Rôle : Hors limite du sujet

Spécification :

Entrée : omoplate(verte)

Sortie : Null

Pré-conditions :

Omoplate 3D déjà affichée

Post-conditions :

Le modèle doit pouvoir être ré-affiché via un bouton

Changer de type d'omoplate (vert ou rouge+vert)

SwapOmop

Rôle : tertiaire

Spécification :

Entrée : omoplate(verte) ou omoplate (verte) avec partie manquante (rouge)

Sortie : omoplate(verte) ou omoplate (verte) avec partie manquante (rouge)

Pré-conditions :

Omoplate 3D déjà affichée

Post-conditions :

Null

Interaction avec l'objet 3D.

InteractObject

Rôle : secondaire

Spécification :

Entrée : Un modèle 3D

Sortie : Null

Pré-conditions :

Objet sélectionné

Post-conditions :

Null

Matching

Matching

Rôle : primaire

Spécification :

Entrée : masque (pour le moment vignette .jpg) + omoplate (verte)

Sortie : Null

Pré-conditions :

Modèle bien sélectionné et masque dans le champ de vision

Post-conditions :

Null

Repositionnement de la scène

ReposScene

Rôle : primaire

Spécification :

Entrée : omoplate(verte)

Sortie : Null

Pré-conditions :

Masque toujours dans le champ de vision

Post-conditions :

Rafraîchissement de l'image obligatoire

Rafraîchissement de l'image

RefreshImage

Rôle : primaire

Spécification :

Entrée : omoplate(verte)

Sortie : Null

Pré-conditions :

Masque toujours dans le champ de vision + bouger par rapport au masque
Post-conditions :
 Null

Dans cette 2ème partie nous avons pu voir toutes les différentes spécifications que nous allons avoir sur l'application. Il y aura une sous-partie portée sur l'interaction avec notre objet en surimpression. Dans cette sous-partie, il sera possible dans une version future d'avoir une IHM avec des boutons permettant l'interaction avec les fonctionnalités présentées ci-dessus. Nous avons également une deuxième partie qui est primordiale c'est la base de l'application, cela comporte le principe de Tracking à partir d'un masque qui sert de point de référence pour afficher l'omoplate (verte) en surimpression.

3 Spécifications non fonctionnelles

3.1 Contraintes de développement et conception

Ici nous verrons les différentes contraintes liées aux matériels, langages et logiciels.

3.1.1 Matériels

En premier lieu nous utiliserons bien évidemment, comme lunettes de réalité augmentée, les HoloLens. Il faudra également avoir un environnement Windows récent sur son ordinateur pour pouvoir utiliser les différents logiciels requis.

3.1.2 Langages de programmation

L'utilisation d'un environnement de programmation utilisant Unity entraîne, dans le cadre de notre projet, l'utilisation du langage C#. Cependant la réalisation d'une application avec Unity reste de la programmation à haut niveau. Il faudra cependant dans Visual Studio retoucher en C# certaines classes afin d'obtenir le résultat attendu.

3.1.3 Logiciels et bibliothèques

Pour ce qui est des logiciels utilisés, j'ai évoqué la majorité d'entre eux dans le début de ce rapport.

Cependant après un essai de mise en place de l'environnement avec un projet lambda présent sur le site Microsoft j'ai pu constater qu'il y avait des problèmes avec les versions 2017 de Unity et Visual Studio. Ces versions sont récentes et les utiliser pour de la réalité augmentée reste très instable.

Je suis donc repassé à une version antérieure pour Visual Studio qui est Visual Studio 2015 Update 3. Pour Unity j'ai dû passer à une version plus récente car celle que j'avais installée en début de projet comportait un bug connu qui a été ensuite corrigé dans les versions suivantes. Ces 2 logiciels sont impératifs pour le bon fonctionnement de ce projet. Pour le SDK j'utiliserai Vuforia qui s'intègre parfaitement dans cet environnement.

3.2 Contraintes de fonctionnement et d'exploitation

3.2.1 Performances

La qualité d'image est plus que correcte, la navigation est fluide. Les HoloLens ont des performances très satisfaisantes.

Pour ce qui est de la programmation de l'application, l'utilisation de Unity et de Visual Studio en même temps nécessite un Pc assez puissant pour faire tourner les 2 logiciels en même temps car ces 2 logiciels sont assez gourmands en ressources.

3.2.2 Maintenance et évolution du système

Comme décrit dans le diagramme de cas d'utilisateur, un technicien doit pouvoir intervenir sur les lunettes pour des modifications ou des corrections de bugs.

Le code de l'application se doit donc d'être évolutif et le plus clair possible pour pouvoir être repris sans problèmes.

B

Gestion de projet

1 Méthode de gestion de Projet

Dans ce projet je suivrai une méthode de cycle en V qui semble adaptée au projet. Effectivement il y aura en premier lieu une phase d'analyse des besoins qui avait déjà été faite en partie l'année précédente par Jordan Nicot. Ensuite il faudra en ressortir un contexte et des spécifications, situés plus haut dans ce rapport. Cette phase d'analyse va ensuite être suivie d'une phase de conception où l'on va réfléchir à comment concevoir l'architecture et l'organisation de notre application (logiciel utilisé, diagramme ,etc...). La partie développement sera effectuée lors de la deuxième partie du PRD.

De plus ce choix du cycle en V est aussi justifié par le fait que le besoin du client est un besoin fixe et les objectifs sont fixes eux aussi au sein du projet. Il n'y a donc aucune utilité à passer sur une méthode Agile dans le cadre de ce projet.

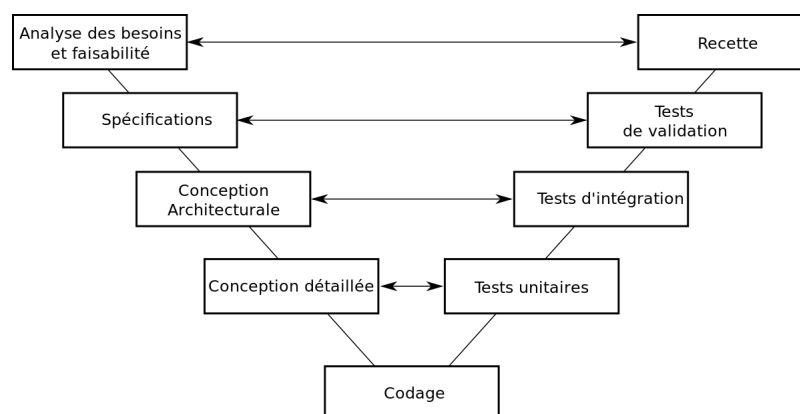


Figure 1 – Cycle en V

2 Planning

Pour la partie planification du projet, un premier diagramme de Gantt avait été fait au début du projet. Finalement celui-ci a été respecté au maximum ; seule la phase d'état de l'art a été un

peu plus longue que prévue, ce qui a débordé sur la partie spécification et conception, au final beaucoup moins fournie. Au vu de la nature du projet qui est un projet sur une technologie très récente, il m'a semblé correct de bien se renseigner sur le sujet et de fournir un travail plus conséquent sur la partie État de l'art que sur la partie Conception.

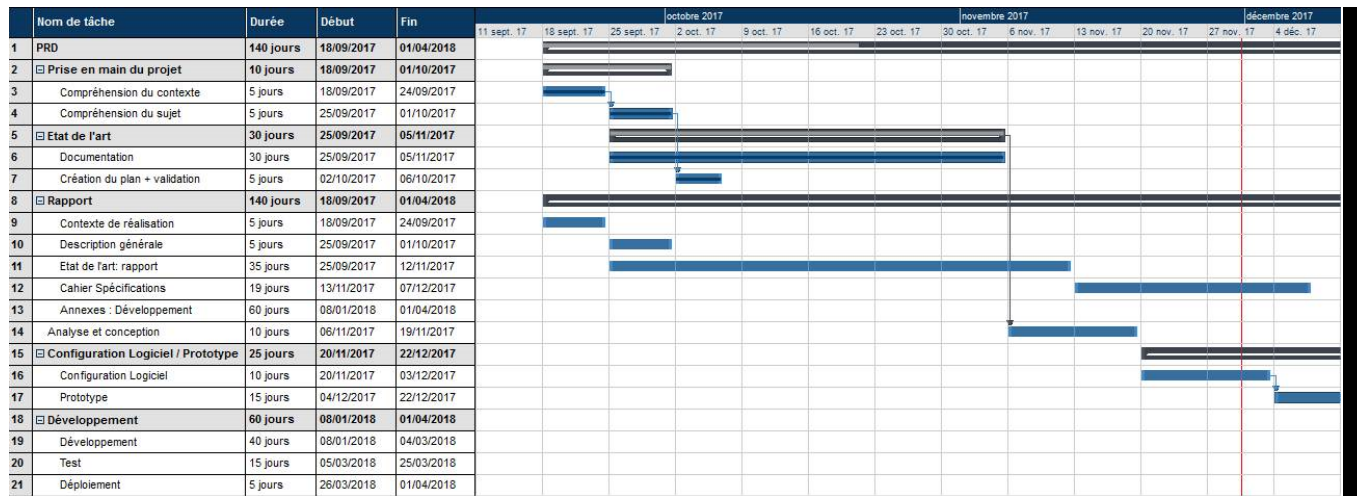


Figure 2 – Diagramme de Gantt Prd1

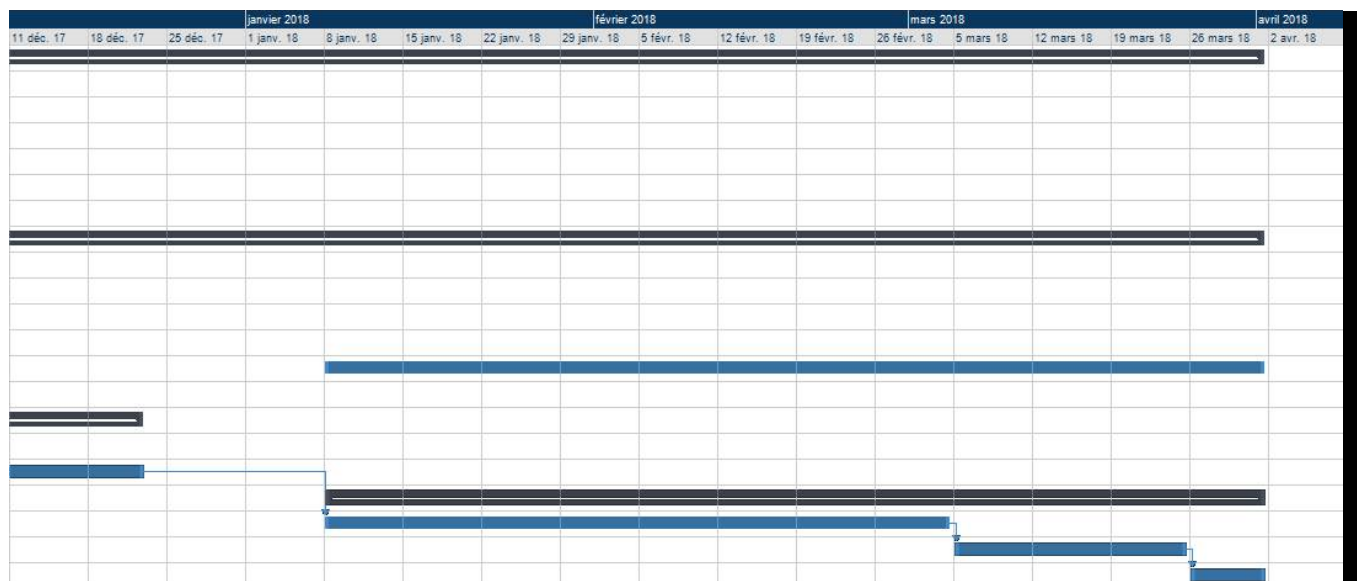


Figure 3 – Diagramme de Gantt Prd2

Pour la partie 2 de ce Gantt, il s'agira de prévisualiser comment faire notre développement tout comme pour la partie état de l'art et analyse.

Nous allons donc suivre le cycle en V que nous avons choisi pour ce projet durant la phase de "développement" de ce projet.

Il faudra donc avoir une phase de développement, une phase de test et une phase de déploiement.

Durant la phase de développement il faudra bien faire attention à rester dans les limites du projet et veiller à ne pas se disperser sur des fonctionnalités annexes.

Le but ici est d'avoir le coeur de l'application fonctionnelle et stable.

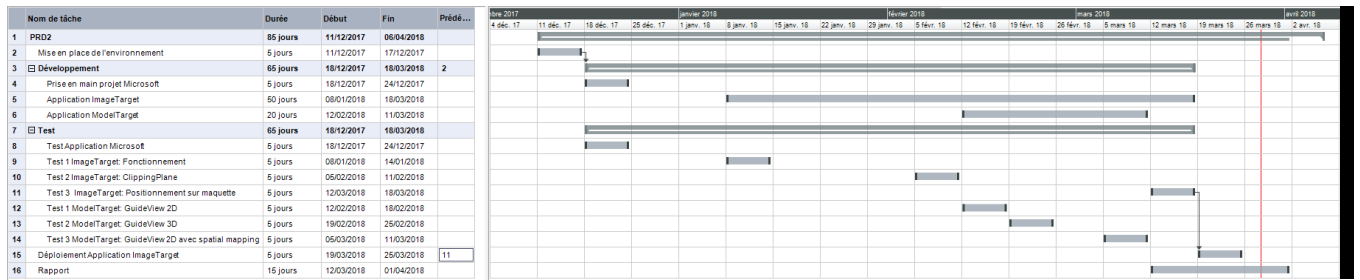


Figure 4 – Diagramme de Gantt Prd2 Fin de projet

Comme on peut le voir dans ce diagramme de Gantt de fin de projet, j'ai fait les phases de tests au fur et à mesure du développement afin d'avoir des livrables fonctionnelles et une avancée correcte.

Ma démarche était de faire des fonctionnalités nécessaires au projet. Puis de les tester immédiatement pour ne pas avancer à l'aveugle sur une mauvaise piste.

Pour cela je compilais mon application et je la testais sur les lunettes HoloLens.

Au final seul l'application ImageTarget aura fait l'objet d'un déploiement final lors d'une vidéo prise.

C

Mise en place de l'environnement

Alors en tout premier lieu, avant même de parler configuration de logiciels, il faudra dans un premier temps télécharger et installer nos logiciels.

1 Installation

Pour télécharger et installer nos logiciels, il faudra donc prendre les bonnes versions sous peine de rentrer dans un cercle vicieux à propos des versions qui ne sont pas compatibles.

Il faudra donc installer :

1. Visual Studio 2015 update 3

<https://www.visualstudio.com/fr/vs/older-downloads/>

Il faudra bien faire attention à avoir la version update 3. Lors de l'installation, il faudra bien cocher les composants dont nous avons besoin c'est à dire le SDK.

Veillez à ce que la case Tools 1.4.1 avec le SDK 10.0.14393 soit cochée, c'est très important.

2. Unity 2017.3.0b11

<https://unity3d.com/fr/unity/beta/archive>

Faire attention, sur Unity les versions beta ont tendances à vite disparaître et à se concaténer entre elles lorsqu'elles sortent en version définitive. C'est pourquoi on peut s'y perdre très rapidement.

2 Configuration

Les installations enfin effectuées, nous pouvons passer aux configurations des logiciels afin d'obtenir un projet convenable et viable qui compile et qui puisse tourner sur les lunettes. Nous commencerons par la configuration de Unity, avec en son sein Vuforia, car c'est par là que tout commence et que nous allons prendre en main la création de l'application.

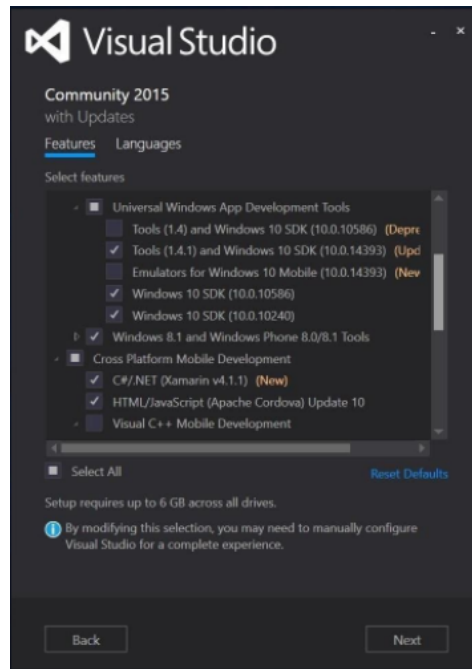


Figure 1 – Écran installation VS2015

2.1 Unity

Alors en premier lieu, afin de tester votre environnement, il est plus facile d'aller sur le site officiel de Microsoft chercher le projet nommé "Origami"¹. Il faut prendre la dernière version et importer ensuite ce projet dans le menu principal des projets Unity.

Ouvrez le et maintenant passons à la configuration du logiciel.

Pour cela en premier lieu il faudra aller dans l'onglet :

Edit\Préférences\External Tools

Une fois dans ce menu il faudra changer "External Script Editor" avec la valeur " Visual Studio 2015".

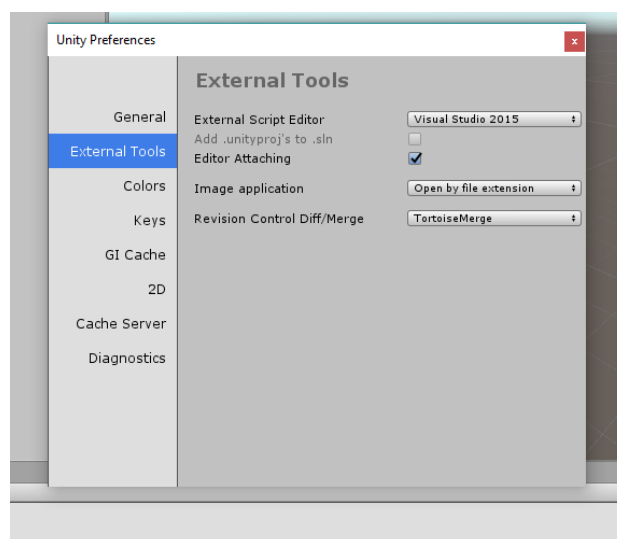


Figure 2 – Préférences

1. <https://developer.microsoft.com/en-us/windows/mixed-reality/holograms101e>

Il y aura une dernière chose à configurer avant de passer à la partie build. Il faudra aller dans : Edit\Project Settings\Quality
 Dans cet onglet, il faudra bien faire attention à ce que "Fastest" soit coché pour UWP ; normalement dans la version de Unity que nous avons installée juste avant, cette configuration devrait y être de base.



Figure 3 – QualitySettings

Ensuite pour pouvoir compiler et obtenir une solution en C# utilisable par Visual Studio, il faudra aller dans l'onglet : File\Build Settings

Ici il y a beaucoup de choses à faire pour pouvoir compiler.

Dans un premier temps, il faudra aller dans l'onglet UWP car bien sûr nous allons compiler pour cette multi-plateforme.

Une fois dans cette onglet, il faudra bien faire attention à avoir les mêmes paramètres que le screenshot suivant :

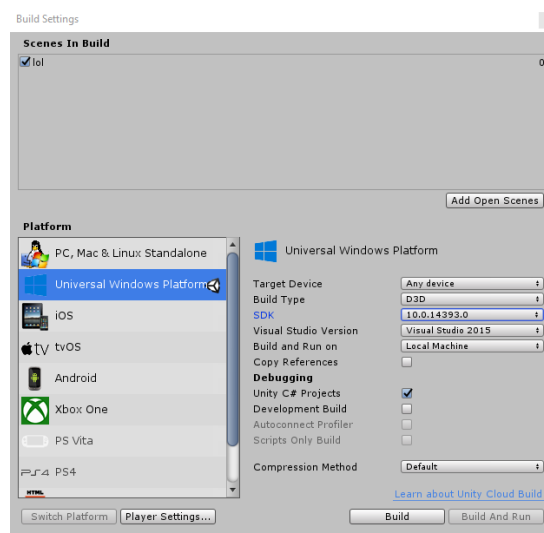


Figure 4 – BuildSettings

Mais ce n'est pas tout, maintenant il faut cliquer sur le bouton "Player Settings" dans la même fenêtre.

Lorsque vous cliquez sur "Player settings", vous pouvez voir sur votre droite un panneau contenant différents onglets. Les 3 derniers sont les plus importants et nous allons devoir y faire très attention.

Le premier que nous allons vérifier et modifier si besoin est "Other settings". Dans la sous-partie "Configuration", il faudra faire attention à avoir les bons paramètres.



Figure 5 – OtherSettings

Ensuite dans "Publish Settings", il faudra aller dans la sous-partie "Capabilities" et vérifier que "Microphone", "Webcam" et "InternetClient" soient bien cochés.

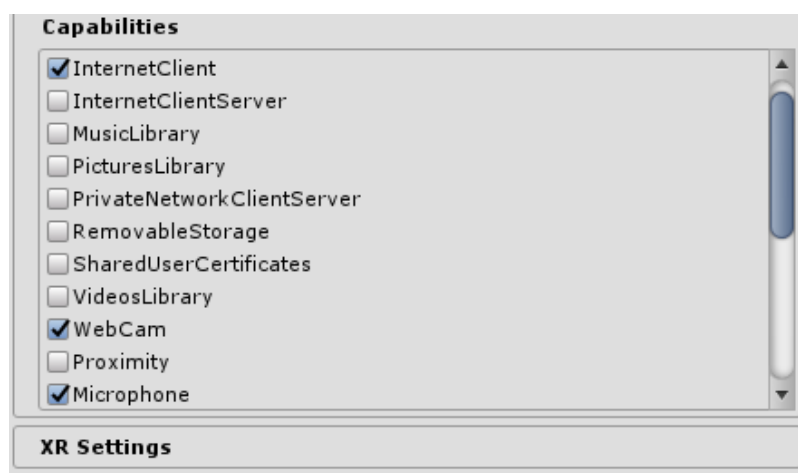


Figure 6 – Capabilities

Il nous reste l'onglet "XR Settings" qui est important car il contient tout ce qui a un rapport avec Vuforia. La majorité des erreurs que j'ai eues lors des mes diverses installations venaient de cette onglet.

Il faudra donc bien faire attention à avoir la configuration suivante :

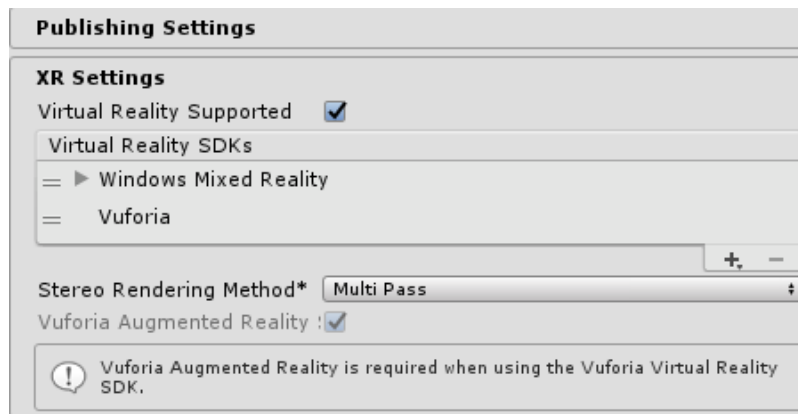


Figure 7 – XRSettings

Pour finir il reste une configuration à faire. Il ne faut surtout pas oublier la partie de configuration Vuforia.

Il faut aller dans l'onglet :

Window\Vuforia Configuration

Un panneau sur votre droite, comme pour "Player Settings", s'affichera.

Dans la section "Digital Eyewear", il faudra changer le "Device Type" actuel par "Digital Eyewear" et "Device Config" avec "HoloLens".

Cette configuration est vraiment importante, sans ça votre projet compilera sous Visual Studio mais votre programme ne fonctionnera pas sur les lunettes.

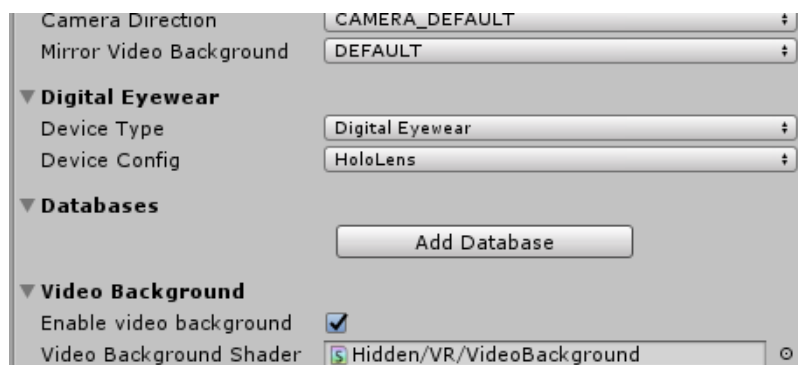


Figure 8 – Configuration Digital Eyewear

Ces derniers paramètres étant vérifiés, nous concluons la partie configuration de Unity pour la compilation d'une application de "test".

2.2 Visual Studio

Dans Visual Studio, il y a très peu de réglages à faire. Il faudra seulement bien faire attention aux paramètres de compilation.

Nous partons sur le fait que nous allons lancer l'exécution du code sur les lunettes via USB, elles devront donc être allumées. C'est donc un réglage spécifique. Pour tout autre réglage, il faudra aller voir sur le site de Microsoft.²

2. https://developer.microsoft.com/en-us/windows/mixed-reality/using_visual_studio_deploying_an_app_over_wifi_28hololens.29

Il faudra s'assurer d'avoir les réglages suivant avant de lancer l'exécution sans débogage.

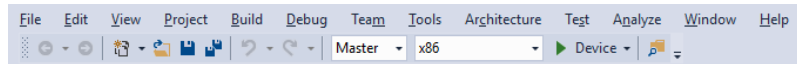


Figure 9 – Configuration Visual

Suite à l'exécution, il sera demandé dans Visual Studio un code PIN pour lier le PC aux lunettes. A ce moment précis ce code devrait apparaître sur l'interface des lunettes.

Si ce n'est pas le cas, il faudra aller dans les paramètres des lunettes, dans l'onglet "update" puis "For developers".

Une fois arrivé dans cet onglet, il faudra cliquer sur "Pair" et là un code PIN s'affichera.

Il faut rentrer ce code PIN dans Visual Studio et la compilation se terminera.

Cette dernière manipulation installera et lancera l'application sur les lunettes HoloLens.

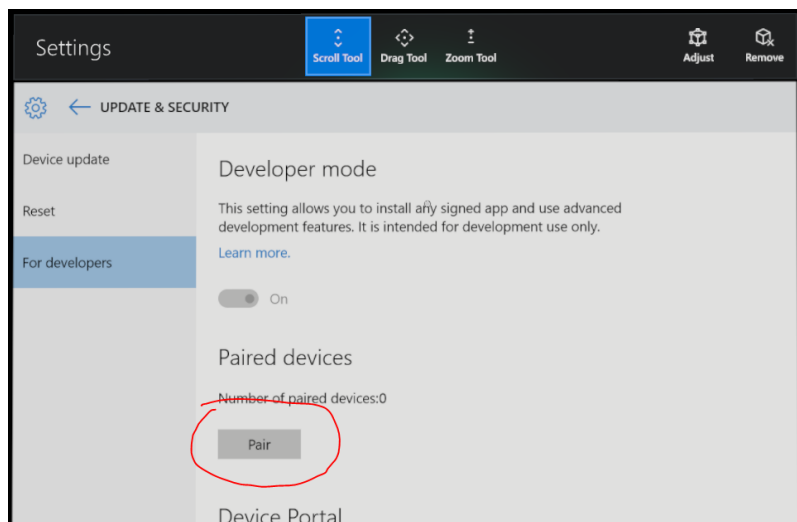


Figure 10 – Pour avoir le code PIN sur les HoloLens

Cela conclut la partie configuration de Visual Studio, mais aussi la partie configuration et déploiement de notre application sur les lunettes HoloLens.

D

Développement

1 Apprentissage et manipulation des objets dans la scène

Dans un premier temps, il fallait que je regarde les différents objets nécessaires au projet. J'ai donc dû m'exercer sur un exemple de projet de réalité augmentée Microsoft pour comprendre le fonctionnement d'une scène augmentée dans Unity.

De plus, j'ai dû lire une partie de la documentation de Vuforia pour analyser les classes qui allaient être utilisées.

Bien entendu, Vuforia étant un organisme privé, nous avons juste une définition et l'architecture des classes, mais pas le code. Nous ne pouvons donc pas analyser celui-ci en profondeur.

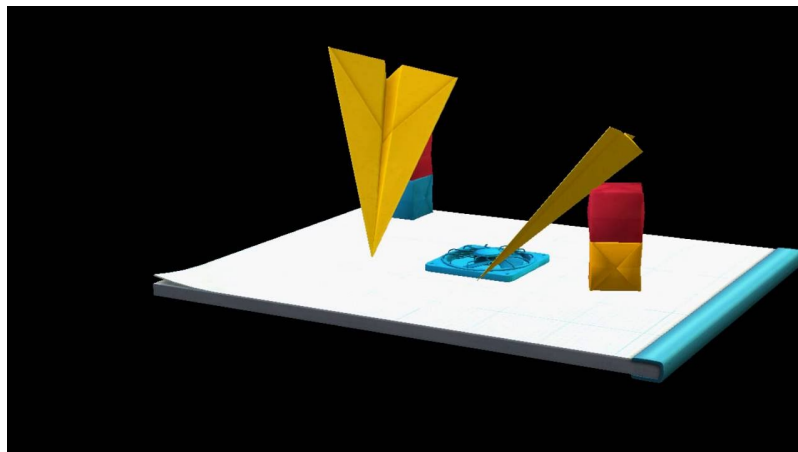


Figure 1 – *Projet origami*

2 Affichage de l'omoplate en surimpression

J'ai ensuite fait un projet similaire à celui-ci, en remplaçant les objets présents dans la scène par notre objet à nous.

Dans notre cas cet objet est une omoplate.

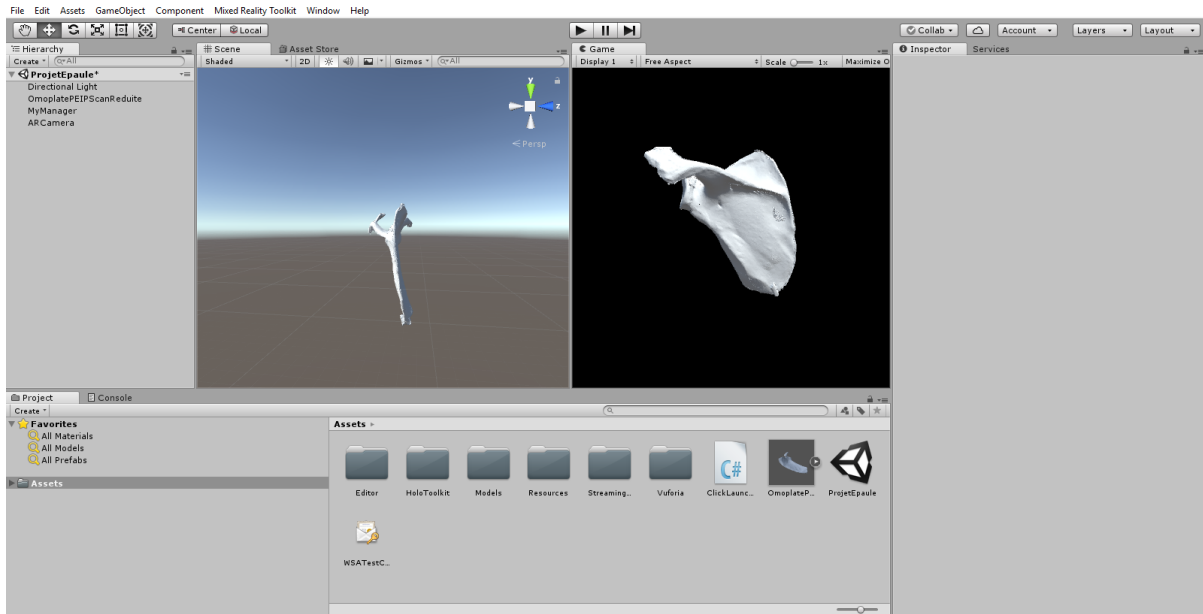


Figure 2 – Scène Unity pour l’affichage de l’omoplate

Pour arriver à ce résultat, il faut 3 composants principaux qui sont :

1. **CameraAR**

Disponible avec Vuforia, c’est une caméra qui a un comportement adapté à la réalité augmentée.

2. **Light**

Une lumière, le type de lumière est à définir selon la nature du projet. Dans notre cas pour les tests, nous avons choisi une lumière dirigée.

3. **Omoplate**

Le dernier objet est l’omoplate 3D obtenue grâce à une modélisation Blender. Cette modélisation est obtenue depuis un scanner effectué sur un patient 15 jours avant l’opération. Ce modèle a un nombre de points réduits pour pouvoir être plus facilement traité par les lunettes.

A ce point-ci du projet, il est donc possible d’afficher l’omoplate 3D, en surimpression sur les Hololens. Cependant, la notion de marqueur n’avait pas encore été mise en avant.

Une partie de l’objectif consistant à importer notre modèle et à l’afficher est donc, à ce moment du projet, finalisé et une première application a pu être testée sur les lunettes.

Jusque-là, il s’agissait donc d’un simple affichage en réalité augmentée de notre objet qui suivait notre mouvement de tête à un point fixe au centre des lunettes.

3 Première application avec ImageTarget et marqueur 2D

Après avoir affiché en surimpression l’omoplate 3D au milieu des lunettes, le but était de pouvoir afficher cette omoplate par rapport à un marqueur 2D.

Pour cela il faut utiliser l’objet nommé "ImageTarget".

ImageTarget est un objet délivré grâce à Vuforia dans Unity et qui permet d’avoir un objet de référence en 2D pour pouvoir afficher un objet3D quand celui-ci apparaît dans les lunettes.

3.1 ImageTarget : configuration

Pour ce faire il faudra dans un premier temps aller sur le site de Vuforia et se rendre dans l'onglet License Manager pour créer une clé licence.

Ensuite il faudra appuyer sur Get Development Key pour pouvoir la créer.

Cette clé servira dans Unity pour pouvoir correctement faire fonctionner l'application et les bases de données de marqueurs.

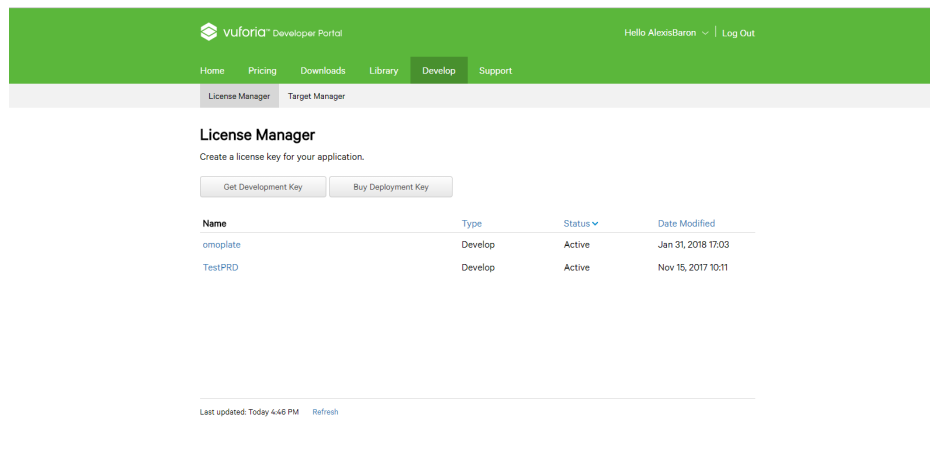


Figure 3 – License Manager

Suite à cela nous pouvons aller dans l'onglet Target Manager.

En premier lieu, il faudra créer une base de données en lui précisant le type "device".

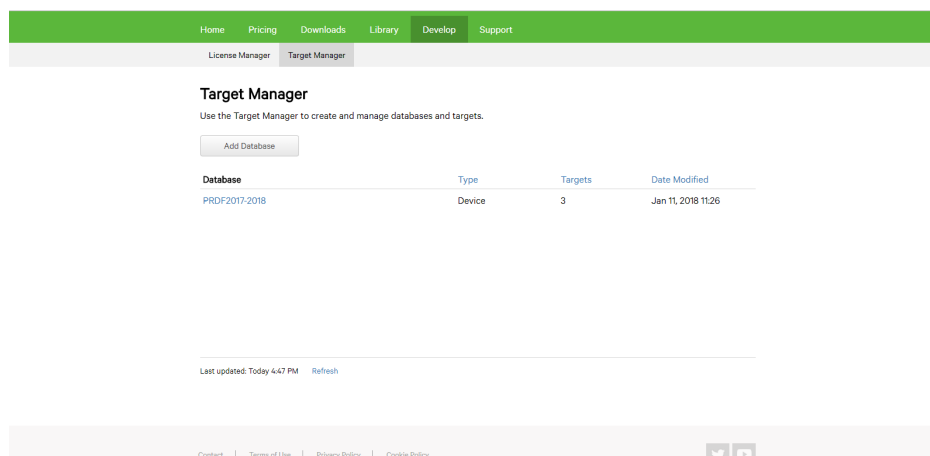


Figure 4 – Création de la base de données

Une fois cette base créée, il faudra la remplir avec nos marqueurs (si possible marqueur 5 étoiles trouvable en ligne) *Figure5*.

Ensuite, on pourra tout simplement télécharger la base de données.

Ces étapes seront les seuls à faire sur le portail développeur Vuforia.

Ensuite il faudra retourner dans Unity et incorporer le package à votre projet.


Pour cela il faudra aller dans l'onglet :

Assets/Import package/CustomPackage


et bien laisser cocher tous les éléments à importer.

Add Target


Type:




Single Image



Cuboid



Cylinder



3D Object

File:

Choose File Browse...

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Cancel
Add

Figure 5 – Ajout d'un marqueur

Nous pouvons dès à présent configurer Vuforia pour l'utilisation d'une ImageTarget. Il faut se rendre dans l'onglet :

Window/VuforiaConfiguration

ou utiliser le raccourci : ctrl + Shift + v

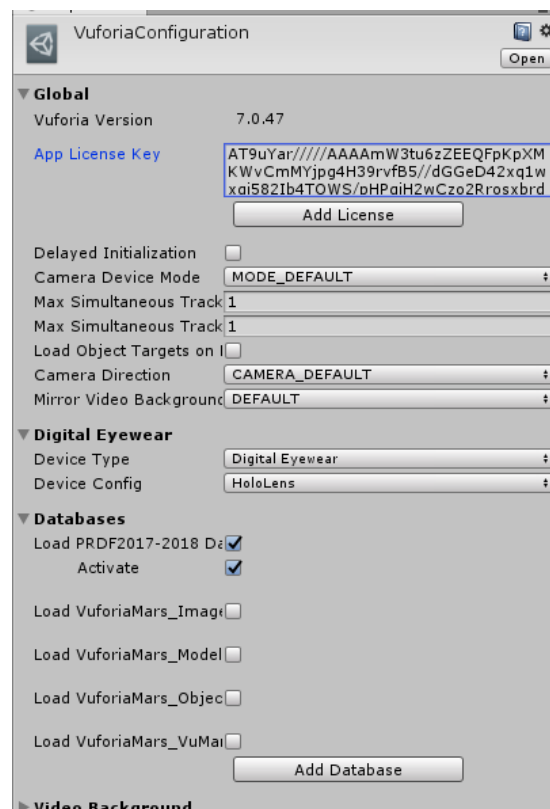


Figure 6 – Vuforia configuration

Dans cette fenêtre, il faudra inscrire sa clé licence dans le champ " App License Key" et il faudra bien cocher la base de données ainsi que le champ "active".
Ces étapes étant faites, l'utilisation de l'imageTarget pour notre usage est effective.

3.2 ImageTarget : utilisation

Maintenant nous allons créer dans notre architecture une imageTarget est adapté les paramètres de notre objet à nos besoins.

Pour ce faire nous allons modifier le composant " Image Target Behaviour", on va donc lui préciser notre base de données mais aussi le marqueur à utiliser.

Il faudra bien faire attention dans advanced à avoir la bonne taille pour le marqueur normalement vous devriez les avoir renseignés à la création de celui-ci dans le portail développeur vuforia.

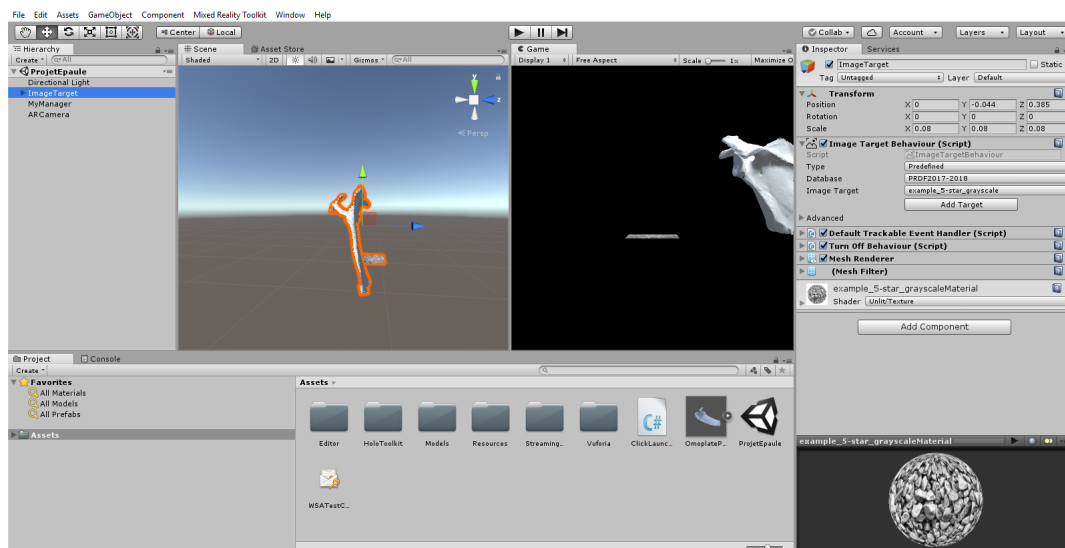


Figure 7 – ImageTarget dans la scène

Ensuite pour avoir une liaison entre notre omoplate 3D et ce marqueur, il faudra mettre notre objet omoplate 3D (qu'on aura importée dans assets au préalable) en tant qu'enfant d'ImageTarget dans l'architecture.

Comme ça l'omoplate dépendra de l'ImageTarget. On peut choisir seulement l'omoplate pour pouvoir la déplacer par rapport au marqueur.

Dans notre cas pour les premiers tests, j'ai placé l'omoplate au-dessus du marqueur afin de faciliter la visualisation de celle-ci.

3.3 ImageTarget : 1er résultat

La première version d'une application avec marqueur étant faite, il était possible de l'utiliser sur les Hololens pour voir le résultat.

En visualisant le résultat, j'ai pu constater un problème majeur.

L'omoplate 3D en surimpression tremblait beaucoup.

Ceci était dû au script par défaut de tracking de Vuforia nommé "DefaultTrackableEventHandler".

C'était un des objectifs du projet qui consistait à avoir une poursuite de l'objet même si le marqueur disparaît avec une stabilité de cet objet.

3.4 ImageTarget : Stabilité de l'image

Dans un premier temps, j'ai fait plusieurs recherches afin d'améliorer la stabilité de l'image. Je suis donc tombé sur un paramètre que l'on pouvait cocher dans ImageTarget, dans le composant "Image Target Behaviour".

Ce paramètre est "Extended Tracking". Celui-ci modifie le comportement par défaut lors du tracking Vuforia. En conséquence notre omoplate aurait dû être stable avec ce paramètre mais aurait dû également pouvoir être continuellement visible même après perte du marqueur. Cependant après quelques tests sur cette nouvelle application et mon marqueur, j'ai pu constater que cette option ne correspondait pas à notre projet car l'image tremblait toujours. Je suis donc parti sur la modification du script "DefaultTrackableEventHandler" pour changer le comportement par défaut de Vuforia lors du Tracking. Ce tracking doit se comporter comme la démarche présentée dans la première partie du PRD.

3.4.1 ImageTarget : Changement du comportement pour le Tracking

Pour cela "DefaultTrackableEventHandler" comporte 2 méthodes nommées "OnTrackingFound()" et "OnTrackingLost()".

Dans le comportement par défaut de l'application, "OnTrackingLost()" est appelée une première fois au tout début pour cacher les objets augmentés qui sont présents dans la scène. Ensuite une fois que notre marqueur est visible dans le champ des lunettes on passe dans la fonction "OnTrackingFound()" qui affiche les objets augmentés. Si on perd de vue le marqueur, on passe de nouveau dans "OnTrackingLost()" pour cacher les objets augmentés comme pour la toute première itération du programme. Or dans le comportement voulu, on ne veut pas passer dans "OnTrackingLost()" lorsque l'on perd de vue notre marqueur afin de garder en surimpression notre omoplate 3D.

Pour cela il suffit de ne pas appeler la méthode "OnTrackingLost()" lorsque l'on perd le marqueur de vue :

Comme on peut le voir dans cette partie du code qui nous intéresse, si le statut précédent était un Tracking et que le statut actuel ne présente aucun Tracking, alors nous nous trouvons dans la situation où l'on vient de perdre le marqueur de vue.

Dans ce cas là, on n'appelle pas la fonction "OnTrackingLost()".

Cette partie du code modifie donc le comportement par défaut du Tracking du SDK Vuforia. Cependant ces changements ne seront pas suffisants pour avoir une image stable, cela permettra juste de pouvoir continuer à voir l'objet en surimpression après la disparition du marqueur de notre champ de vision.

```

/// <summary>
///     Implementation of the ITrackableEventHandler function called when the
///     tracking state changes.
/// </summary>
public void OnTrackableStateChanged(
    TrackableBehaviour.Status previousStatus,
    TrackableBehaviour.Status newStatus)
{
    if (newStatus == TrackableBehaviour.Status.DETECTED ||
        newStatus == TrackableBehaviour.Status.TRACKED ||
        newStatus == TrackableBehaviour.Status.EXTENDED_TRACKED)
    {
        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " found");
        OnTrackingFound();
    }
    else if (previousStatus == TrackableBehaviour.Status.TRACKED &&
        newStatus == TrackableBehaviour.Status.NOT_FOUND)
    {
        Debug.Log("Trackable " + mTrackableBehaviour.TrackableName + " lost");
    }
    else
    {
        //First iteration here
        OnTrackingLost();
    }
}

```

Figure 8 – Méthode sur le changement d'état du Tracking

3.4.2 ImageTarget : désactivation du comportement Vuforia

Lors d'une réunion nous avons fait le choix de désactiver le comportement de Vuforia après détection du marqueur. Donc dans un premier temps, il fallait trouver comment désactiver le comportement de Vuforia sur le Tracking.

```

TrackerManager.Instance.GetTracker<ObjectTracker>().Stop();

```

Figure 9 – Stopper Vuforia

Ce code sera donc à rajouter à la fin de la méthode "OnTrackingFound()" car une fois le Tracking effectué et l'omoplate en surimpression affichée, on veut que Vuforia se stoppe.

Grâce à cette modification nous avons une image qui ne tremble quasiment plus sauf lors de mouvements brusques ou lorsque les lunettes sont bougées sur notre tête.

Cependant, le comportement de Vuforia étant désactivé, si on bouge le marqueur, l'omoplate restera à la même place. C'est le résultat escompté mais le problème est qu'il faudrait pouvoir relancer ce comportement via une entrée de notre choix.

3.5 ImageTarget : Activation du comportement Vuforia

Pour la réactivation du comportement de Vuforia, j'ai donc choisi comme entrée un "Tap". Comme indiqué dans la partie interaction du PRD, un "Tap" est un click avec les doigts dans l'environnement des Hololens.

Pour pouvoir interagir avec les différents points d'entrées des Hololens, il faut au préalable installer un kit, nommé anciennement HoloToolKit et maintenant MixedRealityToolkit, qui permettra de récupérer et de modifier les scripts de management des entrées tels qu'un tap, le regard, la voix.

3.5.1 Installation d'HoloToolKit/MixedRealityToolkit

Dans un premier temps, il faudra télécharger le package MixedRealityToolkit à l'adresse suivante :

<https://github.com/Microsoft/MixedRealityToolkit-Unity/releases>

Une fois le package télécharger, il faudra l'importer dans le projet. Pour cela il faudra aller dans l'onglet :

Assets/Import package/CustomPackage

et bien laisser cocher tous les éléments à importer. Pour éviter des problèmes de dépendances entre les classes.

Si l'intégration à Unity c'est bien passée, un onglet Mixed Reality Toolkit apparaîtra en haut de la fenêtre.

3.5.2 Création d'un Manager

Pour réactiver le tracking de Vuforia, il faudra créer un script personnalisé. Cependant avant de créer ce script il va lui falloir un conteneur, comme pour les autres objets qui contiennent des composants de type script.

On va donc créer un "Empty GameObject" qui doit être rajouté à notre scène. Cette objet sera donc le conteneur pour notre script. On pourra le nommer "MyManager".

On peut voir dans la capture d'écran ci-dessous que l'on a rajouté plusieurs composants à notre Manager :

1. ClickLaunchTrackable

Ce composant est notre nouveau script qui va gérer le tap qui réactivera Vuforia Tracking.

2. AddGlobalListener

Script d'HoloToolkit permettant d'ajouter à notre conteneur un Listener global, notre objet sera donc sensible aux différents appels possibles venant des entrées telles que le tap.

3. InputManager

Script d'HoloToolkit, à mettre pour gérer les différentes entrées possibles.

4. FocusManager

Script d'HoloToolkit, à mettre pour éviter des erreurs liées aux dépendances entre scripts.

5. AudioSource

Comme indiqué plus tôt, notre script va donc, grâce à un tap, réactiver Vuforia Tracking. Lors de la Journée Portes Ouvertes de Polytech Tours, un problème a été soulevé par le Dr Julien Berhouet.

Le problème était qu'aucune information visuelle ou sonore n'était disponible dans la première version présentée.

Plusieurs autres personnes m'ont dit qu'il serait bien qu'il y ait un avertissement visuel ou sonore lors de l'activation du Tracking avec le tap.

C'est pourquoi, j'ai mis ce composant AudioSource.

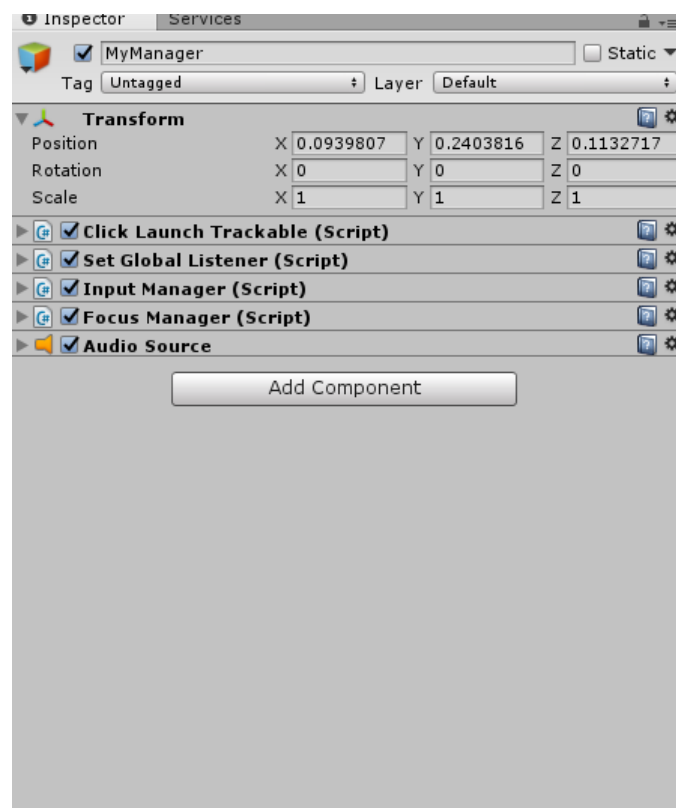


Figure 10 – Composant de MyManager

3.5.3 Création d'un nouveau script : "ClickLaunchTrackable"

Tout d'abord avant de tester avec un tap je voulais voir si ma fonction pour réactiver Vuforia Tracking fonctionnait.

J'ai donc, dans la méthode update() de mon script, fais en sorte qu'on reconnaisse les clics de notre souris sur l'ordinateur afin de pouvoir, par l'intermédiaire de l'environnement Unity, voir à travers la caméra de l'ordinateur le changement effectué.

Après avoir vu que cela fonctionnait, j'ai donc réfléchi pour effectuer la même manipulation mais avec un tap pour l'application Hololens.

```

//Test with mouse clic on computer
void Update()
{
    Debug.Log("Update done");
    if (Input.GetMouseButtonDown(0))
        TrackerManager.Instance.GetTracker<ObjectTracker>().Start();

    if (Input.GetMouseButtonDown(1))
        TrackerManager.Instance.GetTracker<ObjectTracker>().Start();

    if (Input.GetMouseButtonDown(2))
        TrackerManager.Instance.GetTracker<ObjectTracker>().Start();
}

```

Figure 11 – Méthode Update()

```

GestureRecognizer gr = null;

private void Start()
{
    // be careful, this method will be depracted in a futur version

    gr = new GestureRecognizer();
    gr.TappedEvent += Tap;
    gr.StartCapturingGestures();

    //InputManager.Instance.PushFallbackInputHandler(gameObject);
}

// Methode Tap() which launch Vuforia tracking and a noise to indicate that you did a tap
private void Tap(InteractionSourceKind source, int tapCount, Ray headRay)
{
    Debug.Log("Tap done");
    TrackerManager.Instance.GetTracker<ObjectTracker>().Start();
    launchMusic();
}

// Get AudioSource attached and start it
private void launchMusic()
{
    GetComponent<AudioSource>().Play();
}

```

Figure 12 – Gestion du tap avec Audio

Comme on peut l'observer dans cette capture d'écran, il y a 3 méthodes.

La première "start()" s'effectue lors de l'initialisation et met en place la reconnaissance d'un tap.

La méthode "tap()" est celle qui sera lancée quand notre tap sera effectué.

On peut voir qu'on relance le Tracking de Vuforia puis on appelle la 3ème et dernière méthode qui est "launchMusic()".

Comme son nom l'indique cette méthode va tout simplement récupérer, sur l'objet courant (MyManager), le composant AudioSource et va le lancer.

Donc à chaque tap nous aurons un bruit de clic.

3.6 ImageTarget : 2ème résultat

Grâce aux étapes précédentes, nous avons donc une application fonctionnelle. Quand nous regardons notre marqueur, une omoplate en surimpression s'affiche à l'endroit voulu.

Nous obtenons un résultat satisfaisant cependant, si on se rapprochait trop près on pouvait

rapidement voir l'objet disparaître.

J'ai donc dû quantifier par des tests et regarder pourquoi l'objet commençait à disparaître en faisant varier certains paramètres que nous verrons.

3.7 ImageTarget : Clipping Plane

Après plusieurs recherches sur des forums, j'ai pu observer que plusieurs personnes obtenaient le même problème.

Le paramètre "clipping plane" attaché à la caméra était responsable des problèmes rencontrés et pouvait donc faire disparaître des objets alors qu'on n'était pas vraiment proche et inversement. Comme on peut le voir sur le schéma suivant, le clipping plane possède 2 plans un qui est proche (near) et un qui est éloigné (far). Grâce à cela on peut agir sur la distance, depuis la caméra, à partir de laquelle on va pouvoir commencer à voir des objets mais aussi la distance, depuis la caméra, jusqu'où on va pouvoir voir des objets.

Les objets sont donc visibles seulement entre ces 2 plans.

De plus on peut gérer l'ouverture du focus de la caméra avec le paramètre "Field of View".

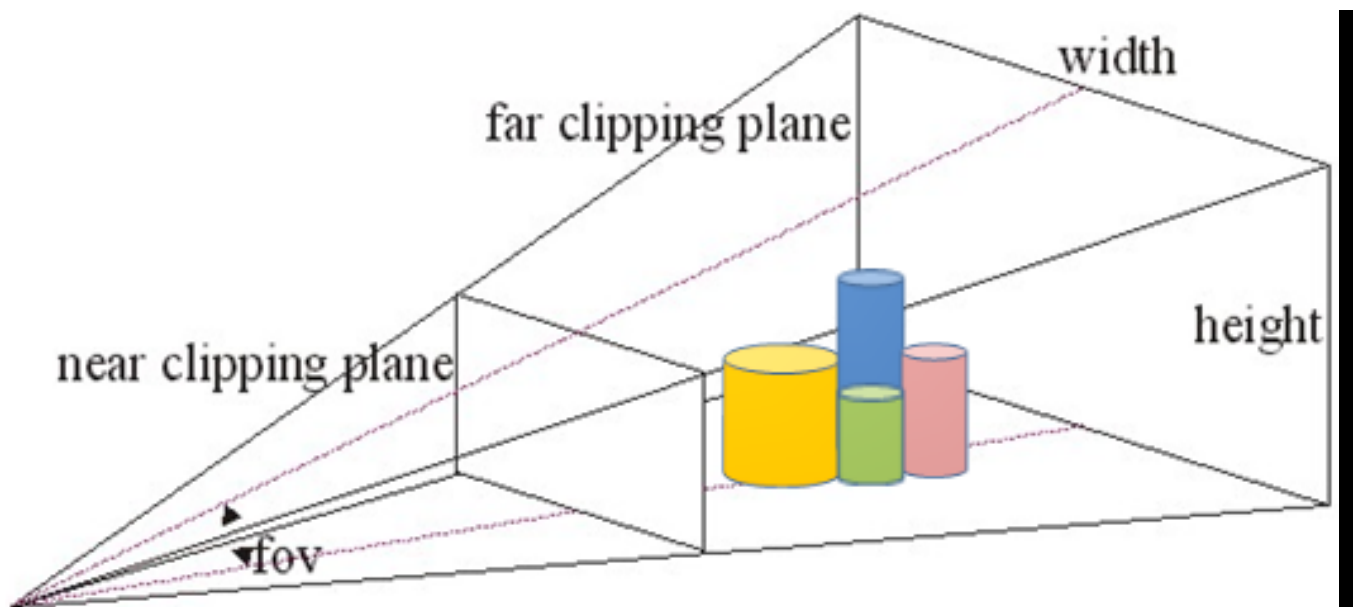


Figure 13 – Explication clipping plane

Voici la note de test envoyé par mail le 11 février 2018 :

Extended Tracking activ   : probl  me

Extended Tracking non activ   : pas de probl  me

Sans la partie du script qui stop vuforia tracking des qu'on voie le marqueur

Test 1 :

Camera clipping plane :

near : 0.05

far : 5

field of view : 100

diff  rence z entre cam  ra et ImageTarget : 0.582

diff  rence y entre cam  ra et ImageTarget : -0.089

dist = distance cam  ra-marqueur

dist \geq 80cm \longrightarrow perte du marqueur

70 \leq dist < 80cm \longrightarrow l'omoplate tend    bouger et    se replacer avec de gros mouvements.

25 \leq dist < 70cm \longrightarrow tout ce passe parfaitement

dist < 25cm \longrightarrow trop pr  s de l'omoplate nous avons donc un morceau qui dispara  t

Test 2 :

Camera clipping plane :

near : 0.01

far : 5

field of view : 100

diff  rence z entre cam  ra et ImageTarget : 0.582

diff  rence y entre cam  ra et ImageTarget : -0.089

dist = distance cam  ra-marqueur

dist \geq 80cm \longrightarrow perte du marqueur

70 \leq dist < 80cm \longrightarrow l'omoplate tend    bouger et    se replacer avec de gros mouvements.

15 \leq dist < 70cm \longrightarrow tout ce passe parfaitement

dist < 15 cm \longrightarrow trop pr  s de l'omoplate nous avons donc un morceau qui dispara  t

Test 3 :

Cam  ra clipping plane :

near : 0.01

far : 5

field of view : 50

diff  rence z entre cam  ra et ImageTarget : 0.582

diff  rence y entre cam  ra et ImageTarget : -0.089

dist = distance cam  ra-marqueur

dist \geq 80cm \longrightarrow perte du marqueur

70 \leq dist < 80cm \longrightarrow l'omoplate tend    bouger et    se replacer avec de gros mouvements.

15 \leq dist < 70cm \longrightarrow tout ce passe parfaitement

dist < 15 cm \longrightarrow trop pr  s de l'omoplate nous avons donc un morceau qui dispara  t

Conclusion :

—> Après recherche Vuforia adapte le field of View à la Camera donc ce paramètre sert juste aux tests sur ordinateur.

<https://developer.vuforia.com/forum/unity-extension-technical-discussion/ar-camera-field-view>

Far est bien trop grand 5 unités = 5m dans Unity, de plus near and far doivent être le plus proche possible pour éviter des scénarios instables.

Avec la partie du script qui stop vuforia tracking dès qu'on trouve

Test 1 :

Camera clipping plane :

near : 0.05

far : 5

field of view : 100

difference z entre camera et ImageTarget : 0.582

difference y entre camera et ImageTarget : -0.089

A partir du moment où l'on voit le marqueur, arrêt du tracking de Vuforia pour s'en séparer.

Avec clic souris de l'ordinateur, on relance le tracking et la Vuforia track si il y'a modification du marqueur et se ré-éteint.

Clicker Hololens non fonctionnel pour le moment

J'ai donc adapté ce "clipping plane" aux besoins du projet, en réduisant les valeurs du clipping plane qui par défaut sont très élevés.

La distance maximale de 80cm est dû à la taille du marqueur. Pour 1cm sur la taille max du marqueur cela correspond à une distance de 10cm. La taille max de notre marqueur étant de 8cm pour la longueur, on obtient donc les 80cm.

3.8 ImageTarget : 3ème résultat

Pour finir, nous avons mis en place, avec la maquette de l'année dernière, un environnement de démonstration grâce à l'application finale utilisant ImageTarget.

Cette application remplit donc 2 objectifs(1 et 3) que sont la poursuite de l'objet et l'importation de notre objet 3D.

Pour ce qui est du matching dans le cas de notre environnement de démonstration, j'ai placé le marqueur face à l'épaule comportant l'omoplate réelle.

J'ai donc mesuré avec une règle la distance entre le coin supérieur gauche de l'image et le centre de l'omoplate3D définie dans Blender et visible dans Unity.

En rentrant ensuite ces distances entre le marqueur et l'omoplate dans Unity, avec la bonne échelle, j'ai pu avoir une superposition avec l'omoplate réelle. Dans notre cas, il faut faire attention car l'omoplate3D et l'omoplate réelle sont différentes donc la superposition n'est pas précise mais donne une bonne indication sur ce qui est réalisable.

Nous pouvons donc dire qu'avec une omoplate et son fichier .obj correspondant nous pour-



Figure 14 – Maquette finale avec marqueur

rions, avec des distances précises obtenues lors de scanners, avoir un matching parfait et donc compléter le 2ème objectif dans son intégralité et avec un test exact.

4 Deuxième application avec ModelTarget et marqueur 3D

Comme indiqué dans la première partie du PRD, nous avons essayé en vain d'avoir l'outil Model Target Generator.

Cependant avec la mise à jour vers Vuforia 7, Model Target Generator est sorti et j'ai donc pu le télécharger et l'essayer via l'url :

<https://developer.vuforia.com/downloads/tool>

4.1 Model Target Generator

Donc Model Target Generator va nous permettre, de prendre en entrée un objet 3d qui nous servira du marqueur.

C'est à dire à partir d'un fichier .obj nous allons pouvoir récupérer un marqueur 3D utilisable par Unity/Vuforia.

Cependant en tant que compte non VIP, nous n'avons accès qu'à 10 générations maximum de package utilisable.

Une fois l'outil téléchargé et installé, il faudra se connecter avec les mêmes identifiants que ceux utiliser sur le portail Vuforia.

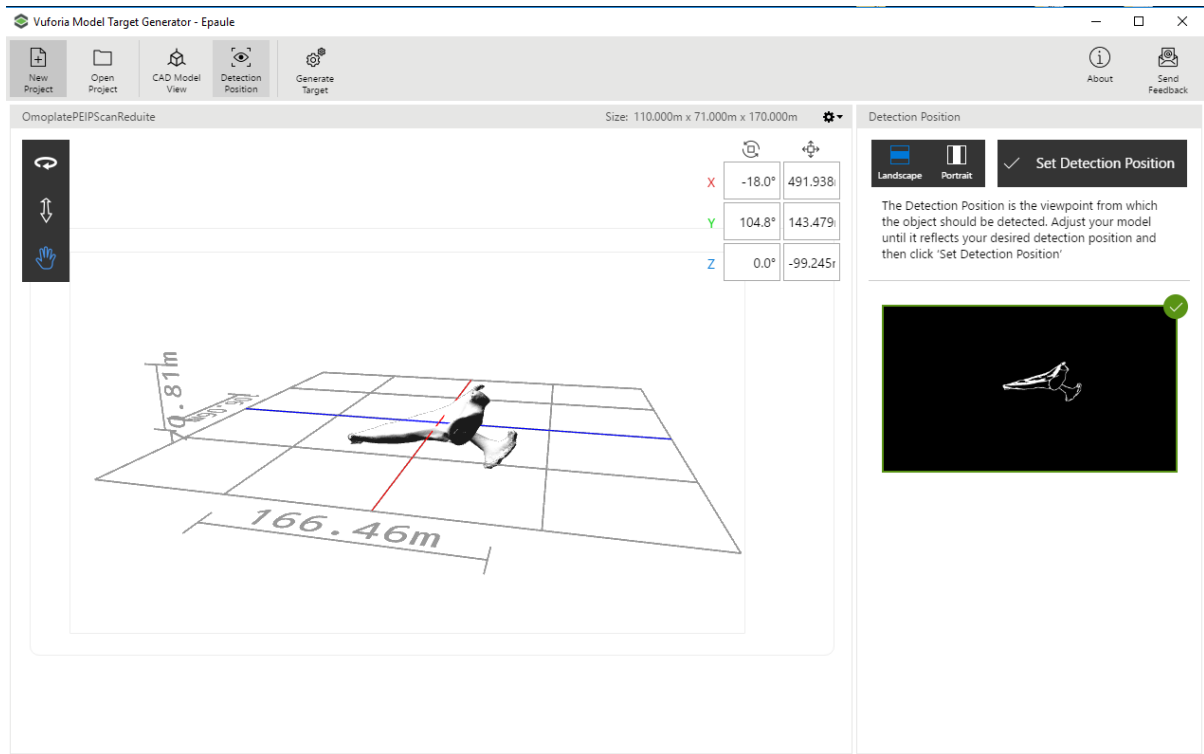


Figure 15 – Model Target Generator

Voici l'écran principal de ce nouvel outil.

On peut y voir une grande fenêtre comportant ici notre modèle 3d générer à partir d'un fichier .obj

Ensuite, il faudra, comme l'indique la capture d'écran, cliquer sur l'onglet "Detection Position" puis sur "setDetection Position". Cela va permettre de créer un guide nommé GuideView.(Nous expliquerons plus tard ce qu'est un GuideView) Pour finir il faudra générer notre "marqueur" 3D avec generateTarget.

Après la génération, on retrouvera dans un dossier un package à incorporer à Unity. Pour trouver ce package il suffit d'aller dans le dossier créer puis de suivre ce chemin :

resources\app.asar.unpacked\sampladata\NomDuProjet\dataset

Après, il faudra importer ce package comme pour les packages d'Holotoolkit cités avec Image-Target.

4.2 Model Target : Comment l'utiliser

Une fois importé, nous pouvons entrer dans le vif du sujet à l'intérieur de Unity.

On va donc créer, comme pour ImageTarget, un GameObject nommé ModelTarget cette fois-ci. Nous nous retrouverons avec l'arborescence présentée dans la capture d'écran ci-dessous.

Ensuite il faudra paramétrer ce ModelTarget. Pour cela il faudra aller dans le composant Model-TargetBehaviour comme sur la capture d'écran.

Il faudra remplir la base de données utilisées ainsi que le ModelTarget , ce sont ceux que vous avez importés avec le package unity généré depuis ModelTargetGenerator.

Il faudra bien faire attention aux valeurs sur la taille physique de l'objet, car la conversion entre ModelTargetGenerator et Unity peut mal se faire et au lieu d'avoir des valeurs en centimètres vous aurez des valeurs en mètres.

Le dernier et plus important paramètre est le guideView généré lui aussi grâce à ModelTarget-Generator.

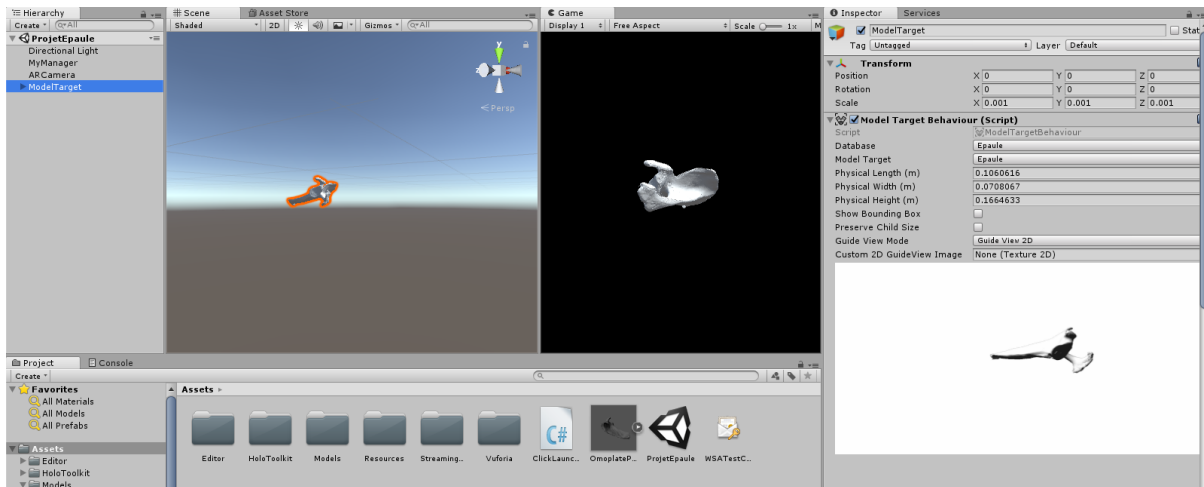


Figure 16 – ModelTarget dans Unity

4.3 Model Target : GuideView

Le guideView est un guide qui peut être 2D ou 3D.

Pour le GuideView 2D, il représente une coupe en 2 dimensions de l'objet qu'on a passé en tant que marqueur 3D.

C'est pour cela qui faut bien positionner notre objet sur la grille dans Model Target Generator. Pour le GuideView 3D, il représente plusieurs coupes 2D.

Ayant très peu d'informations sur Model Target nous avons supposé que ces guideView effectuaient du contouring sur l'objet et si il y a matching de contour alors il y aura reconnaissance de l'objet réel et donc affichage en surimpression d'un objet qu'on pourra définir.

Cependant pour arriver à ce résultat il faut bien entendu que le .obj qu'on a désigné comme marqueur 3D soit la copie de l'objet réel.

4.4 Model Target : Résultat et hypothèses

Pour arriver à des résultats s'approchant de ce que l'on veut, j'ai utilisé le guide 2D pour me baser sur une seule coupe de mon objet réel.

Malgré le travail accompli sur cette technologie, je n'ai pas réussi à avoir une application reconnaissant notre objet réel.

Bien faire attention avec un GuideView 3D, il est possible qu'il y ait une reconnaissance.

Cependant après des tests effectués, j'ai pu voir que ce guide view reconnaissant mon bras et non notre omoplate réelle.

Il est possible qu'il y ait un seuil de tolérance.

E

Démonstration

11/09/2017 : Démonstration Lycéen

J'ai du présenter le projet à 60 lycéens divisés en 3 groupes de 20. A la suite de chaque présentation, il y a eu une démonstration ainsi qu'un essai des HoloLens sur l'application Galaxy Explorer ce qui a captivé la majorité d'entre eux.



Figure 1 – Démonstration du 11 Octobre (1er groupe)

17/02/2018 : Journée Portes Ouvertes Polytech Tours

Durant la journée Portes Ouvertes de Polytech Tours, j'ai pu présenter mon projet aux familles venant découvrir l'école.

J'ai également pu présenter ce projet au directeur de l'Université de Tours.



Comptes rendus hebdomadaires

Compte rendu n°1 du 18/09/2017 au 22/09/2017

Présentation du projet et de son contexte. Ces 2 premiers jours m'ont permis en premier lieu de comprendre le contexte du projet grâce aux diverses explications données ainsi qu'aux rapport de l'année dernière.

Le projet a donc 3 objectifs qui sont définis dans le Power Point donnés dans le cr0bis :

- * Transmettre la scène
- * Superposition parfaite entre le masque et la cible
- * Poursuite en temps réel parfaite

Ensuite j'ai pu effectuer quelques recherches sur la réalité augmentée, le matériel (lunettes HoloLens), environnement de travail(Visual studio, Vuforia, Unity).

Une première installation de ces logiciels m'ont permis de comprendre qu'ils allaient tous être connectés les uns aux autres car des partenariats existent entre eux pour faciliter le développement.

De plus j'ai voulu commencer à regarder comment le matching pouvait s'effectuer entre le masque et la cible et si l'import d'objet 3d pouvait s'effectuer.

J'ai donc trouvé que depuis début de cette été 2017 une pre-release pour l'import d'objet 3d existant a été mise en place via vuforia avec un logiciel nommée "Vuforia Model Targets", cette annonce a été faite lors de l'Augmented World Expo 2017 fin mai 2017 aux Etats Unis , des informations complémentaires sur cette convention peuvent être retrouvées ici : <http://info.vuforia.com/modeltargets> <http://www.augmentedworldexpo.com/>

Tout ceci m'a permis d'avoir une première vision du projet et de comment j'allais devoir appréhender les tâches qui m'ont été confié.

Pour finir le vendredi 22/09/17 j'ai rencontré Jordan afin qu'il m'explique plus en détails le projet et qu'il me face une démonstration du fonctionnement des HoloLens.

Je joins également le début de mon rapport à ce cr1.

Pour les horaires le PRD s'effectuera le mercredi et jeudi de 8h15 à 12h30 et de 14h à 18h15.

Compte rendu n°2 du 25/09/2017 au 29/09/2017

Essai plus en profondeur des lunettes avec les différents logiciels déjà présent + essai sur la maquette au fond de la bibliothèque.

Continuité de la recherche sur le matching pour finir sur le même problème que l'année dernière c'est à dire la conversion d'un fichier .obj en un fichier .od.

Vuforia autorisant seulement l'import d'objet 3D .obj en tant que Target Model. Pour le moment une image imprimée sert de "Target Model" (solution trouvée pour effectuer des tests l'année dernière)

A terme il faudra un matching et donc un model 3d allant se positionner sur un "Target Model" également en 3D et non un motif imprimée sur feuille (Le "Target Model" n'est pas une chose visible c'est l'objet de référence pour placer notre modèle 3d ici l'omoplate).

Solutions trouvées :

* Passé par une impression 3d est utilisé l'application Vuforia Android pour scanner l'omoplate imprimée ce qui ferait un objet .od car à l'heure actuelle le seul moyen de pouvoir importer un objet 3d pour le tracking est l'import de fichier .od est le seul moyen d'en avoir est par l'application Vuforia car cette extension de fichier est propriété privé de Vuforia pour le moment.

Cependant solution trop longue et fastidieuse donc a oublié complètement (vu avec Christian).

* Cette dernière solution est la plus convaincante , comme cité précédemment dans le cr1 il existe un outil en pré-release nommée "Vuforia Model Targets" permettant l'import de fichier .obj,etc... je vous laisse dans ce mail un lien vers un article de blog d'un artiste 3d concernant cette outil ainsi qu'un lien vers une vidéo promotionnelle faite par Vuforia via un telephone mais pouvant être utilisé via les lunettes, le support importe peu dans ce contexte.

[http ://www.severin3d.com/blog/2017/6/2/new-in-ar-vuforias-object-tracking-form-3d-models-and-live-annotations-in-video-chats](http://www.severin3d.com/blog/2017/6/2/new-in-ar-vuforias-object-tracking-form-3d-models-and-live-annotations-in-video-chats)

[https ://www.youtube.com/watch?v=y70yStPCBHA](https://www.youtube.com/watch?v=y70yStPCBHA)

Ce nouvel outil remplacerait donc l'application Android Vuforia Scan permettant de scanner un objet, vu que maintenant on pourrait directement importer des objets 3d au format ".obj".

J'ai également fait une recherche sur les différentes interactions possible avec les lunettes afin de faire l'état de l'art, il existe donc 4 principal axes d'interactions :

- * Gestuelle
- * Voix
- * Regard
- * Son de l'environnement

Ceci pourrait être d'éventuel piste pour améliorer le confort de l'utilisation de l'application par le chirurgien.

Je joins à ce mail l'avancée de mon rapport, une ébauche de plan pour la partie Etat de l'Art a été rédigé ainsi que la rédaction de la première partie introductive sur les lunettes HoloLens.

Compte rendu n°3 du 02/10/2017 au 06/10/2017

Cette semaine j'ai principalement travaillé sur la partie état de l'art du projet. J'ai pu ainsi constater que la version commerciale des lunettes était sortie et était déjà disponible à un prix de 5489€

<https://www.microsoft.com/fr-fr/store/d/microsoft-hololens-commercial-suite/944xgcf64z5b>

De plus Microsoft pense déjà à l'évolution de ses lunettes avec une deuxième version des lunettes qui aurait du être les V2 mais qui sera donc nommée V3 prévue pour 2019.

<http://www.zdnet.fr/actualites/hololens-microsoft-annulerait-la-v2-pour-une-v3-lancee-en-2019-39848732.htm> <https://www.realite-virtuelle.com/hololens-microsoft-version-3-2002>

Pour ce qui est de l'état de l'art sur le secteur d'activité et l'état du marché des lunettes et de la réalité augmentée. J'ai pu constater que les lunettes HoloLens étaient utilisées dans de multiple secteurs d'activités tel que le milieu médical, automobile, aéronautique et bien d'autres encore. Les HoloLens se démocratisent petit à petit et de plus en plus d'entreprise voulant se mettre en avant technologiquement utilisent ces lunettes et font des partenariats avec Microsoft. Pour ce qui est des interactions avec les HoloLens j'ai regardé la documentation présente sur le site officiel de Microsoft et j'ai pu constater qu'il y'avait une multitude d'informations qu'il faudra que je traite la semaine prochaine en attendant de voir ce qu'il en est du nouveau logiciel de Vuforia. Ce pourrait être une bonne idée d'avoir une interaction par la voix au lieu des gestes surtout pendant une opération.

Compte rendu n°4 du 09/10/2017 au 13/10/2017

En premier lieu cette semaine était encore un peu floue sur comment faire mes états de l'art sur quoi ils devaient être faits ainsi que les limites sur lesquelles ils devaient s'appliquer. J'ai donc travaillé et fini la partie sur les interactions avec les HoloLens au vue de la discussion avec Mr Proust comme quoi il ne fallait pas trop s'étaler sur ce genre de fonctionnalité qui sont pour le moment secondaire. Le mercredi après-midi j'ai donc fais la présentation du projet ainsi qu'une démonstration des HoloLens à 60 lycéens, divisés en 3 groupes de 20. C'était une bonne expérience et cela m'a permis de pouvoir expliquer le projet et de voir qu'est ce qui pourrait être approfondis quand je le présente. Pour finir, jeudi soir nous avons eu une réunion avec Mr Proust et Mr Nicot, où en premier lieu nous avons parlé de cette problématique à propos de mes états de l'art des limites que je devais fixé afin de me concentrer sur le sujet. Nous avons également discuté de la démarche à adopter sur comment allez fonctionner le programme et sur la démarche scientifique ainsi qu'un probable plan pour mon rapport. Ensuite nous avons commencer à parler plus dans le technique car le logiciel de chez Vuforia n'étant pas vraiment disponible il nous faut une solution première qui elle l'est. Nous avons donc discuté, à propos d'une pastille reconnaissable avec une multitude de couleur afin que les lunettes puisse reconnaître cette pastille comme marqueur (marqueur 5 étoiles sur Vuforia obligatoire pour plus de précision). Cette pastille serait donc collé sur le patient et avec des calculs possible grâce aux précédentes radio on pourrait placer l'objet virtuel à la bonne position.

Quelques unes des remarques faites sur le rapport par Jordan Nicot et Christian Proust, tel que bien justifier et non faire des affirmations ainsi que bien présenter les sources, seront prise en compte dans le prochain rapport. Il me manquait quelques informations à propos du format latex sur la bibliographie c'est pourquoi des liens étaient encore présent en brut dans le texte.

Compte rendu n°5 du 16/10/2017 au 20/10/2017

Cette semaine je me suis renseigné sur la partie mapping des HoloLens comment elles fonctionnent, quelles sont les erreurs possibles que nous pouvons rencontrer lors du scan et

pourquoi cela arrive. Cette partie était vraiment très intéressante car c'est un peu le cœur même des HoloLens. Cependant cette partie était très compliqué à comprendre car il y a plein de subtilité à comprendre et les informations sont bien cachées. J'ai également fini la première partie sur la réalité augmentée ainsi que réorganisait certaines parties du rapport en fonction de ce que nous avons parlé lors de la dernière réunion. Grâce aux diverses informations sur Latex données par Jordan, j'ai pu enfin mettre une bibliographie et une webographie au sein du rapport. J'ai également pris un peu de temps pour passer sur une version en ligne pour écrire mon rapport afin de faciliter la correction par Christian. Nous avons également vu avec Christian une vidéo sur un faux cœur où le logiciel Model Target de chez Vuforia est utilisé, ce nouveau logiciel/add-on est vraiment très impressionnant et nous avons pu constater dans les commentaires de la vidéo, que la personne a réussie suite au contact par mail à obtenir un lien de téléchargement pour ce logiciel, en attendant une possible obtention de ce logiciel nous resterons sur les idées évoquées dans le précédent CR.

Lien de la vidéo :

<https://www.youtube.com/watch?v=4SQdHVgmEmY>

Compte rendu n°6 du 23/10/2017 au 27/10/2017

En premier lieu cette semaine j'ai commencé par rédiger et réfléchir à la conclusion possible sur le mapping en quoi les erreurs pouvaient être gênante dans notre cas et si il y en aurait que devons nous faire. Entre autre il faudra tester tout ces paramètres dans une salle en condition réelle d'éclairage pour ainsi pouvoir calculer la marge d'erreur celle-ci devant être le plus proche de 0 car nous nous permettons pas de décalage ou de trou dans le mapping sur un thème tel que la chirurgie.

Dans un second temps j' ai passé la majeure partie de cette semaine sur le thème des spécifications de ce projet, des utilisateurs,etc.. C'est une partie qu'il faut effectué dans le rapport et qui est plutôt longue, j'ai donc refait le schéma de l'algorithme de fonctionnement de l'application comme nous en avons parlé la dernière fois mais j'ai également émis les hypothèses,etc... qui sont nécessaires au projet De plus j'ai réorganisé et complété le rapport sur les premières parties afin que cela colle avec ce qui c'est dit lors d'une réunion sur la forme du rapport avec Mr Ragot dans son bureau.

Compte rendu n°7 du 06/11 au 10/11/2017

Pour cette semaine j'ai finalisé la partie spécifications du rapport ainsi que les petites parties du rapport manquants. Le diagramme de cas d'utilisation a été fait, à vérifier ensemble si celui-ci est correcte. Il reste une petite partie analyse + compréhension des classes de unity à faire c'est pourquoi j'ai commencé à lancer unity et à également regarder les différentes classes Unity/Vuforia ,avec lesquels nous pouvons interagir dans le cadre de la réalité augmentée, via la documentation que nous pouvons trouvé en ligne :

<https://library.vuforia.com/content/vuforia-library/en/reference/unity/index.html>

J'ai donc remarqué plusieurs classes intéressante comme la classe de Tracker il est possible que cette classe soit à modifier, nous avons vu avec Jordan lors de la première réunion que certaines classes seront à modifier pour assurer la stabilité de l'objet en surimpression. Pour la semaine qui arrive il faudrait que la compréhension des classes intéressantes à utiliser de unity soit effectués et que cette compréhension soit mise au propre sur le rapport dans la partie analyse. Il faudra également que je manipule un peu plus le logiciel Unity pour afficher quelque chose ne serait ce que sur le logiciel via des models 3D (pour ça suivre tuto ou rapport de Jordan) .

Compte rendu n°8 du 13/11/2017 au 17/11/2017

Cette semaine le jeudi était consacré au forum des entreprises je n'ai donc pas pu travailler toute la journée sur le projet. Pour ce qui est des tâches accomplies durant cette semaine. J'ai donc regarder d'autres classes et leurs liens dans l'application et j'ai donc suivi également le tutoriel de Jordan pour configurer un projet de test avec l'application Unity.

En premier lieu j'ai eu du mal à lancer la compilation car certains paramètre n'était pas de la même version que celle de Jordan j'ai donc du chercher comment pouvoir compiler et configurer mon projet. A la toute fin j'ai réussi a compiler le projet mais des erreurs sont survenues j'ai du faire une erreur dans la configuration, j'ai donc retoucher aux paramètres mais sans succès cela ne compile toujours pas et les erreurs sont incompréhensibles, à voir avec Jordan si il n'y a pas eu un problème de configuration de Unity et non du projet.

Pour la réunion technique, mercredi soir me semble parfait je serais disponible.

Compte rendu n°9 du 18/11/2017 au 24/11/2017

Cette semaine j'ai fait les annexes à fournir avec le rapport c'est à dire les interfaces possibles , diagramme de gantt ,etc... De plus après la réunion du mercredi 22/11 au soir, nous avons pu voir ensemble certains problèmes que j'avais rencontrés lors de la configuration de l'environnement de travail Visual Studio/Unity etc... J'ai donc pu le jeudi travailler sur cette partie en téléchargeant des versions plus anciennes pour éviter le problème. Grâce à cela, j'ai pu avancer dans mon travail malgré une nouvelle erreur persistante que j'ai essayer de corriger ce week-end et qu'il faudra que je réussisse à corriger cette semaine absolument à propos de Vuforia cette fois ci. Le rapport quant à lui pour cette première partie du PRD est bientôt terminé il reste quelque parties à compléter et à faire vérifier.

Compte rendu n°10 du 27/11/2017 au 01/12/2017

Cette semaine, j'ai eu 2 tâches importantes a effectuer : .Faire fonctionner l'environnement de développement, pour pouvoir commencer la partie développement. .Finir le rapport qui doit être rendu le 10/12 maximum Pour la première étape, j'ai réussi à faire fonctionner l'environnement de travail après plusieurs téléchargement. Il est maintenant possible de compiler le projet sur les lunettes avec Visual Studio. Les versions des logiciels utilisés sont : .Visual Studio 2015 update 3 .Unity 2017.3.0b11 Pour le rapport celui-ci est presque terminé il me reste quelque corrections vu avec Christian comme l'explication succinctes des logiciels utilisés au lieu de juste les énumérer. Il faudra aussi que j'écrive les spécifications fonctionnelles et le rapport sera terminé. La date limite de rendu de rapport étant le 10/12 il faudra absolument que jeudi le rapport soit terminé.

Compte rendu n°11 du 04/12/2017 au 08/12/2017

Cette semaine j'ai finis le rapport qui doit être rendu pour dimanche. J'ai donc pris en compte les remarques du dernier mail de Mr Proust et j'ai donc mis toute la partie technique, que j'ai faite, dans le rapport. Certaines remarques avaient déjà était corrigées dans la version 11 que j'avais faite cette semaine, comme l'explication des logiciels utilisés. Il est vrai qu'il manquait un aspect technique à ce rapport et je m'en excuse de ne pas avoir penser à mettre cette partie que j'avais effectué, je pensais qu'il fallait la garder pour la partie développement du rapport. Je l'ai donc ajouter aujourd'hui dans son intégralité en ce qui concerne le téléchargement, l'installation et la configuration des logiciels, pour la mise en place de l'environnement de travail. Certaines affirmations que j'avais écrites sans justification ont été justifié avec des notes en bas de page (soit un lien vers des sites où j'ai recherché, soit une brève explication) et j'ai corrigé les fautes du cr10. Pour ce qui est de parler mercredi matin de la soutenance de jeudi , c'est pour moi une très bonne idée. Je serais comme d'habitude à la bibliothèque.

Compte rendu n°12 du 11/12/2017 au 15/12/2017

Cette semaine il y'a eu les soutenances. Il n'y a donc pas eu de changements majeurs dans le projet. Toutefois après la soutenance, j'ai repris plusieurs point du rapport sur lesquels nous avons discutés. J'envoie donc dans le cr de cette semaine avec les transparents de la soutenance ainsi que le rapport modifié.

Compte rendu n°13 du 08/01/2018 au 12/01/2018

Je viens de me rendre compte que le compte-rendu pour la semaine précédant les vacances n'avait pas été envoyé. Je me permets donc d'ajouter à ce compte rendu, le travail effectué avant les vacances.

Ce compte-rendu marque le début de la deuxième partie du PRD, c'est-à-dire la partie développement et réalisation. Pour le moment aucune modification n'a été apporté au rapport.

1) Semaine du 18-12 au 22-12 :

J' avais donc commencé à prendre Unity en main avec le modèle de l'omoplate mis à ma disposition. J'avais effacé l'exemple de test que j'avais précédemment et j'avais commencé par essayer d'afficher l'omoplate correctement. Finalement avant les vacances j'avais pu afficher sur les lunettes l'omoplate sans marqueur. Cependant l'omoplate affichée dans Unity était énorme et avait des proportions démesurées.

Semaine du 08-01 au 12-01 :

Cette semaine je suis passé à l'utilisation d'un marqueur afin d'afficher l'omoplate grâce à celui-ci. J'ai donc pris un marqueur 5 étoiles pour que celui-ci soit plus facilement reconnu. Ensuite je me suis penché sur la taille que devait faire ce marqueur j'ai donc fait 2 échantillons un de 2.5 cm de largeur et un de 6 cm. Le marqueur de 2.5cm n'arrivant pas être reconnu car sûrement trop petit aux vues des diverses réponses que j'ai pu voir sur des forums. J'ai donc pris un marqueur de 6cm de largeur. Ensuite après intégration dans Unity j'ai eu pas mal de soucis car au niveau de mon ordinateur avec sa caméra cela fonctionnait. J'avais donc bien l'omoplate qui s'affichait lorsque j'approchais de ma camera le marqueur imprimé sur une feuille. Le problème étant que lorsque je compilais et lançais l'application sur les lunettes le marqueur n'était jamais reconnu. J'ai donc essayé plusieurs solutions et lors du dernier essai de Jeudi, l'omoplate c'est bien affichée sur les lunettes et sans tremblements, grâce à une option nommée "Extended Tracking" qui permet de garder la position fixe de l'objet dans l'espace après qu'il soit reconnu grâce au marqueur. Il faudra donc que je vérifie avec exactitude d'où venait le problème pour éviter que cela se reproduise.

Compte rendu n°14 du 15/01/2018 au 19/01/2018

Cette semaine j'ai donc vérifié avec exactitude d'où venait le problème que j'avais eu la semaine dernière. Finalement les marqueurs de couleur sont mal reconnus par les lunettes HoloLens et il faut mieux travailler avec des versions en nuances de gris. Il m'est aussi venu plusieurs questions et surtout par rapport au placement du marqueur que l'on voudrait utiliser. Du fait qu'on ne connaît pas spécialement la position exacte de l'omoplate à l'intérieur du patient à la conception de l'application, il va y avoir un problème. Le marqueur étant un référentiel immobile et l'épaule un référentiel mobile entre la conception de l'application et le patient qui s'installe il va donc y avoir un problème sur les distances entre le marqueur et notre objet en surimpression et il sera donc quasi impossible de faire un Matching entre l'omoplate réelle et virtuelle j'en ai donc parlé avec Mr Proust. Après notre discussion, nous en avons ressorti deux problématiques sur lesquelles je peux me pencher :

- Peut-on utiliser plusieurs marqueurs en même temps. - Une photo de l'épaule ouverte avec la glène visible au centre pourrait elle faire office de marqueur.

La semaine prochaine, je vais donc travailler sur ces 2 problématiques qui peuvent peut-être résoudre le deuxième objectif : le Matching. De plus les premiers tests pour afficher en surimpression étant concluant et validant l'objectif 1, il faudra maintenant que je travaille avec un éclairage constant afin d'avoir une rigueur scientifique sur mes manipulations et mes essais.

Compte rendu n°15 du 22/01/2018 au 26/01/2018

Cette semaine j'ai donc regardé et testé l'hypothèse des multi-marqueur émis la semaine dernière. J'ai donc cherché un moyen efficace pour nous d'utiliser plusieurs marqueurs. Dans notre cas, je pense qu'il serait judicieux d'utiliser 3 marqueurs. Chaque centre de marqueur devra être à une distance d'un point précis de l'omoplate. Chaque distance entre le marqueur et son point sur l'omoplate devra être précisée auparavant. Bien évidemment les 3 marqueurs ne seront pas côte à côte. Je pense qu'il est judicieux d'avoir une forme de triangle autour de l'omoplate qui recouvre correctement celle-ci en 3 dimensions afin d'avoir un positionnement précis sur un vecteur $v(x, y, z)$. De plus le jeudi en fin d'après-midi, j'ai vu que depuis fin décembre/début Janvier, le logiciel que nous voulions en début de projet (Vuforia Model Target) est disponible au téléchargement et utilisable avec la dernière version d'Unity qui contient Vuforia 7. J'ai donc testé un peu ce week-end le logiciel et il prend effectivement nos ".obj", on peut ensuite avoir un marqueur 3D représentant notre omoplate dans Unity. Cette représentation sera dans un élément nommée ModelTarget au lieu d'un Image Target pour un marqueur 2D. Il faudrait qu'on voie ensemble si je pars sur cette solution ou si l'on reste sur les marqueurs, mais je pense qu'il est possible de gagner beaucoup de temps avec ce nouvel outil. Je pourrais peut-être essayer en dehors des horaires de PRD et continuer à travailler sur la solution des marqueurs pendant les heures de PRD. Il faudrait que nous ayons une réunion pour voir tout ça ensemble et parler de quelques points techniques concernant les multi-marqueurs dans Unity.

Compte rendu n°16 du 29/01/2018 au 02/02/2018

Cette semaine nous avons donc eu la réunion qui a porté sur 3 points :

.Extended tracking : fonctionnalité qui répond aux besoins ou non? .Multiple marqueurs
.Vuforia Model Target

Pour le premier point nous en avons donc convenu que la fonctionnalité Extended tracking n'était pas le mieux adaptée pour notre situation et que nous voulions nous détacher aux plus des fonctionnalités gourmandes de Vuforia. Pour la fin de semaine prochaine, il faut donc que je modifie le script d'événement sur le tracking afin qu'on puisse toujours voir l'objet en surimpression même quand le marqueur disparaît de notre champ de vision. Il faudra également arrêter le tracking de Vuforia pour s'en séparer au plus. Cependant il faudra pouvoir relancer la fonctionnalité de Vuforia et donc pour ça, il faudra que j'ajoute un script qui permettra, par un clic ou autres, de relancer le tracking de Vuforia.

Pour les multiples marqueurs, cette idée a été abandonnée car il y a peu d'intérêt finalement à les utiliser. Cependant je devrais quand même en parler succinctement dans mon rapport aux vues des recherches et du travail fourni dessus.

Pour Vuforia Model Target, il faudra que j'imprime un patron papier correspondant au contour de notre omoplate car la représentation sur Unity étant un contour de notre objet, nous pensons que la reconnaissance se fait grâce au contour de celui-ci. Il faudra donc faire des tests pour valider cette hypothèse et commencer à utiliser ce logiciel en parallèle.

J'ai donc réussi cette semaine à modifier le script d'événement sur le tracking Pour ce qui est de l'arrêt du tracking de Vuforia il est fonctionnel et il est possible via un clic de souris de relancer cette fonctionnalité. Il faudra donc faire des tests pour valider les 2 scripts.

Compte rendu n°17 du 05/02/2018 au 09/02/2018

Cette semaine j'ai donc travaillé sur ce que nous avons vu la semaine dernière ensemble, je me suis donc penché sur le premier point.

J'ai modifié le script d'événement sur le tracking afin qu'on puisse toujours voir l'objet en surimpression même quand le marqueur disparaît de notre champ de vision. Cette partie la est correct, j'ai également fait des tests en modifiant certains paramètres dans Unity afin d'avoir une meilleur précision. Suite à ces tests, j'ai pu en conclure que le "clipping plane" jouait un rôle quant à la distance à laquelle l'objet disparaissait. ce paramètre correspond à quand la camera commence à faire apparaître des objets (parametre "near") et à quand les objets disparaissent (paramètre "far"). Seul les objets entre les 2 plans sont donc visibles , le minimum possible pour le "near clipping plane" est 0,01 . Je joins les notes de mes tests à ce compte rendu.

Dans le script, la méthode OnTrackingFound() est appelée je stop le tracking de Vuforia comme nous en avions convenu (celui-ci doit être ré-activable grâce au "Clicker" ou à un air tap).

Pour ce faire, il a donc fallu que je télécharge HoloToolkit et que je l'intègre à mon environnement de travail. C'est à ce point que des difficultés ont commencé, ces difficultés étaient encore liées aux versions. Les anciennes versions d'HoloToolkit étant "Deprecated/Obsolete", j'ai donc du installer la dernière version. Malheureusement, la dernière version avait besoin d'un Sdk récent pour compiler avec Unity sinon on obtenait cette erreur :

"The type or namespace name 'Haptics' does not exist in the namespace 'Windows.Devices'"

La dernière version de Sdk sur Unity était la seule compatible et compilable. Le problème étant que la solution générée ne pouvait être ouverte qu'avec Visual Studio 2017 et non le 2015. J'ai donc dû faire une mise à jour de mon Visual Studio 2017 celui qui ne marchait pas en début de projet. Finalement avec la dernière version stable de Visual Studio 2017 cela fonctionne.(ils ont dû fixer les anciens bugs liés aux clés qu'on avait eu.) Je n'ai donc plus eu d'erreur logicielle et j'ai donc pu créer un script pour récupérer un événement de click sur les HoloLens. Malheureusement avec le retard pris sur les version logicielles et le téléchargement, ce click n'est pas encore fonctionnel. Le click n'est pas reconnu et il faut que je cherche pourquoi celui-ci ne l'est pas. (je ne passe même pas dans la fonction qui devrait être reconnue).

Compte rendu n°18 du 05/03/2018 au 09/03/2018

Suite à la réunion de la semaine dernière j'ai donc travaillé cette semaine sur 3 points importants. Le premier étant de placer correctement l'objet en surimpression par rapport au marqueur. J'ai donc pu constater dans Unity que c'était seulement un décalage. Ce décalage sera quantifié la semaine qui arrive en adaptant la surimpression 3D avec notre maquette grâce au marqueur. On pourra donc voir exactement de combien est le décalage. Mis à part cela l'application est fonctionnelle.

Le deuxième point était de faire plusieurs applications en changeant les paramètres afin de ne pas avoir qu'une seule application quand on doit faire une présentation. Pour ce point il suffisait juste d'aller dans les "player settings" et faire les modifications nécessaires.

Le dernier point était de tester les différentes fonctionnalités de Model Target pour comprendre son fonctionnement en faisant varier les paramètres. Pour cela j'ai donc modifié et fait différentes applications, chacune avec un et un seul paramètre qui change. Le paramètre le plus important est le " guide view" celui-ci peut être 2d ou 3d. Le problème, il y a 2 semaines,était que ce guide view ne s'affichait pas sur les Hololens alors qu'il était bien présent sur le rendu camera du pc. J'ai donc cherché en premier lieu à afficher ce "guide view" et cette partie la est fonctionnelle.

Après plusieurs tests variés en superposant le guide avec notre objet réel, j'ai pu constater grâce aux logs en continue qu'on ne reconnaissait jamais l'objet. (pas de passage dans "onTrackingFound()") J'ai donc cru au début à un problème de mapping, j'ai donc ajouté à ma scène plusieurs éléments pour que le mapping soit visible. Le mapping de la pièce étant relativement correct ce n'était pas la source du problème. J'ai donc testé différents sens pour le guide view de l'omoplate afin de voir si ce n'était pas un problème de reconnaissance de forme et au final ça ne venait pas de la non plus. Pour l'instant le problème n'est pas résolu.

Après discussion jeudi soir avec Jordan, la semaine prochaine je vais me concentrer sur 2 tâches :

.La première est de faire une maquette fonctionnelle pour avoir une base solide et fonctionnelle pour l'application avec le marqueur.

.Pour la deuxième tâche celle-ci ira jusqu'à la fin du prd et consistera en une étude détaillée de model Target et de l'outil Model Target Generator. Il faudra donc que je décrive un workflow de comment l'utiliser, comment le mettre en place, mais aussi voir avec les classes et les paramètres comment model Target se décompose et fonctionne. Cela sera effectué dans la limite des possibilités offertes par la documentation et par les tests possibles , Vuforia étant un environnement très fermé on ne peut pas rentrer dans le code de certains scripts.

Webographie

- [WWW1] BLOGS.INFINITESQUARE. *HoloLens : Retour d'expérience sur le développement de Room-Manager*. URL : <http://blogs.infinitesquare.com/posts/windows/hololens-retour-d-experience-sur-le-developpement-de-roommanager>.
- [WWW2] CHIRURGIE-ORTHOPEDIQUE-PARIS. *Epaule Arthrose*. URL : <http://www.chirurgie-orthopedique-paris.com/prothese-de-lepaule/>.
- [WWW3] CHIRURGIE-ORTHOPEDIQUE-PARIS. *Epaule saine*. URL : <http://www.chirurgie-orthopedique-paris.com/prothese-de-lepaule/>.
- [WWW4] ETR.FR. *Caractéristiques des HoloLens*. URL : <http://www.etr.fr/actualite/1936-le-champ-de-vision-des-microsoft-hololens-30-17-5.html>.
- [WWW5] GOGGLASSES.FR. *les applications en réalité virtuelle et augmentée qui révolutionnent la médecine*. URL : <https://www.gogglasses.fr/realite-virtuelle/sante-medecin-traitement-formation-vr-ar>.
- [WWW6] GOGGLASSES.FR. *Scopis lance une application en réalité mixte pour les chirurgies du dos*. URL : <https://www.gogglasses.fr/realite-augmentee/application-realite-mixte-chirurgie-hololens-scopis>.
- [WWW7] LEPARISIEN. *Quand le chirurgien place une prothèse d'épaule grâce... à des hologrammes*. URL : <http://www.leparisien.fr/societe/quand-le-chirurgien-place-une-prothese-d-epaule-grace-a-des-hologrammes-05-12-2017-7435155.php>.
- [WWW8] MICROSOFT. *Dassault Aviation passe à la réalité mixte avec HoloLens de Microsoft*. URL : <https://news.microsoft.com/fr-fr/2017/06/19/dassault-aviation-passe-a-realite-mixte-hololens-de-microsoft/>.
- [WWW9] MICROSOFT. <https://docs.microsoft.com/en-us/windows/uwp/xbox-apps/development-lanes-unity-versioning>.
- [WWW10] MICROSOFT. *Install the tools*. URL : https://developer.microsoft.com/en-us/windows/mixed-reality/install_the_tools.
- [WWW11] MICROSOFT. *Mixed reality capture for developers*. URL : https://developer.microsoft.com/en-us/windows/mixed-reality/mixed_reality_capture_for_developers
- [WWW12] MICROSOFT. *Spatial mapping design*. URL : https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_mapping_design.

- [WWW13] REALITE-VIRTUELLE.COM. *Caractéristiques des lunettes d'AR*. URL : <https://www.realite-virtuelle.com/guide-comparatif-lunettes-realite-augmentee#1450459894625-255711f8-92d61531-81963f9c-b225>.
- [WWW14] REALITE-VIRTUELLE.COM. *Ford utilise le Microsoft Hololens pour créer de nouvelles voitures*. URL : <https://www.realite-virtuelle.com/ford-hololens-2109>.
- [WWW15] SCIENSETAVENIR. https://www.sciencesetavenir.fr/archeo-paleo/archeologie/historique-une-cavite-de-30-metres-de-long-decouverte-dans-la-pyramide-de-kheops_17964.
- [WWW16] The Advanced Perioperative Imaging Lab TGH. *Augmented Reality Overlay on 3D Printed Heart*. URL : <https://www.youtube.com/watch?v=4SQdHVgmEmY&t=96s>.
- [WWW17] ULAVAL. *Real-time High Resolution 3D Data on the HoloLens*. URL : http://vision.gel.ulaval.ca/~jflalonde/pubs/papers/garon_ismarposter_16.pdf.
- [WWW18] UNITY. *How do I import models from my 3D app?* URL : <https://docs.unity3d.com/Manual/HOWTO-importObject.html>.
- [WWW19] VUFORIA. <https://www.vuforia.com>.
- [WWW20] VUFORIA. *Vuforia App for HoloLens*. URL : <https://library.vuforia.com/articles/Training/Developing-Vuforia-Apps-for-HoloLens>.
- [WWW21] ZDNET. *AR vs VR*. URL : <http://www.zdnet.fr/actualites/realite-augmentee-vs-realite-virtuelle-le-match-39841616.htm>.



Bibliographie

- [1] Dhiraj AMIN et Sharvari GOVILKAR. « Comparative Study of Augmented Reality Sdk's ». In : 5 (fév. 2015), p. 11–26.
- [2] Adrien COPPENS. « Merging real and virtual worlds : An analysis of the state of the art and practical evaluation of Microsoft Hololens ». In : *CoRR* abs/1706.08096 (2017). URL : <http://arxiv.org/abs/1706.08096>.
- [3] Brian Lystgaard DUE. « The future of smart glasses : : An essay about challenges and possibilities with smart glasses ». In : *Working papers on interaction and communication* 1.2 (nov. 2014), p. 1–21. ISSN : 93300.

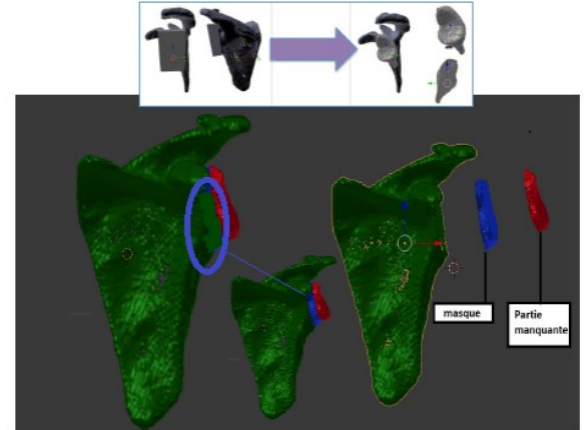
Alexis BARON

Encadrement : Jordan NICOT, Mohamed SLIMANE, Julien BERHOUE et Christian PROUST

Objectifs

3 axes principaux:

- Importation des modèles 3D
- Matching masque/cible
- Poursuite et tracking de l'objet .

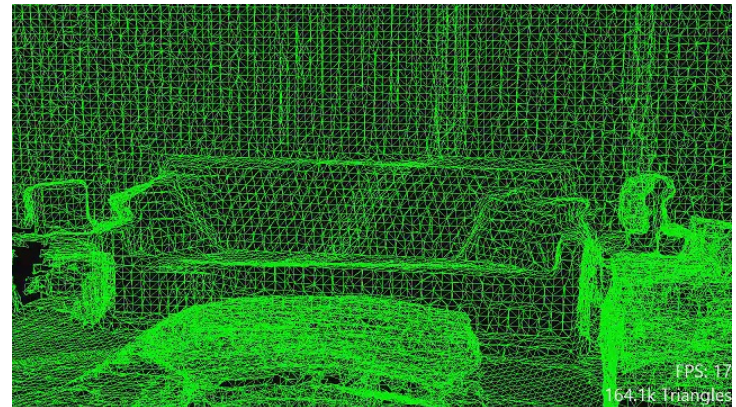


Représentation de l'omoplate et de ses différentes parties dans l'application

Mise en œuvre

Études pour bien délimiter et comprendre le sujet:

- Études de la réalité augmentée
- Études des HoloLens
- Études du mapping
- Études du tracking



Représentation du mapping d'une pièce

Résultats attendus

- Placement et surimpression d'un modèle 3D (Précision à 1mm)
- Modification du Tracking pour éviter les sauts d'images.(Version stable)
- Tester dans environnement lumineux (type chirurgicale) pour éviter les problèmes de mapping.



Hôpital Avicenne, Dr Gregory utilisation des Hololens durant l'opération.[WWW7]

Hololens, Réalité Augmentée et Epaule

Alexis BARON

Encadrement : Jordan NICOT, Mohamed SLIMANE, Julien BERHOUET et Christian PROUST

Objectifs

3 axes principaux:

- Importation des modèles 3D
- Matching masque/cible
- Poursuite et tracking de l'objet .

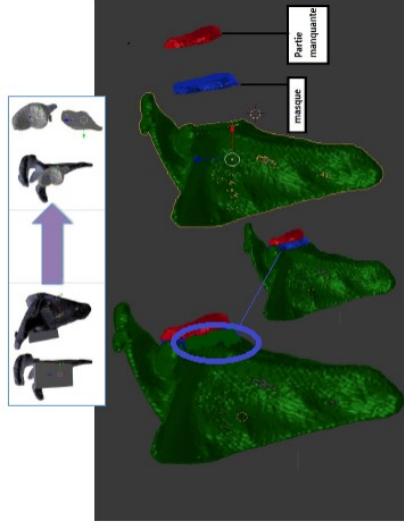
Mise en œuvre

Études pour bien délimiter et comprendre le sujet:

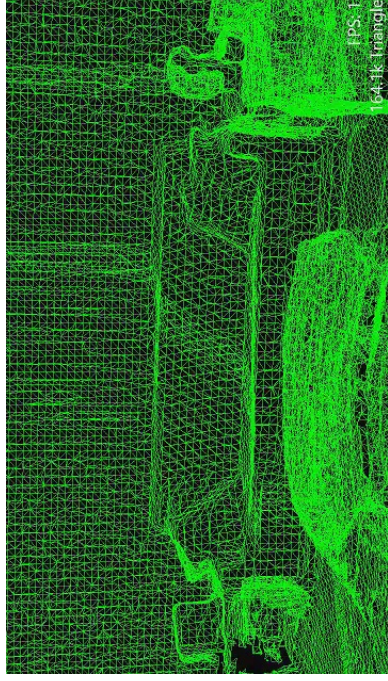
- Études de la réalité augmentée
- Études des HoloLens
- Études du mapping
- Études du tracking

Résultats attendus

- Placement et surimpression d'un modèle 3D (Précision à 1mm)
- Modification du Tracking pour éviter les sauts d'images.(Version stable)
- Tester dans environnement lumineux (type chirurgicale) pour éviter les problèmes de mapping.



Représentation de l'omoplate et de ces différentes parties dans l'application



Représentation du mapping d'une pièce



Hôpital Avicenne, Dr Gregory utilisation des HoloLens durant l'opération.[[www.w7](http://www.w7.fr)]

Hololens, Réalité Augmentée et Epaule

Résumé

Le projet "Hololens, Réalité Augmentée et Epaule" est la suite de plusieurs projets ayant pour but d'assister un chirurgien, en réalité augmentée, lors de l'opération d'une épaule. Cela permettra donc de faire un lien entre le domaine médical et le domaine de l'informatique. Pour cela nous utiliserons les lunettes de réalité augmentée : les Hololens. Il faudra donc afficher en surimpression sur les lunettes l'omoplate malade mais aussi la partie de la glène reconstituée. Il faudra donc une précision chirurgicale en résultat de la superposition de cette surimpression 3D sur les lunettes avec l'omoplate réelle.

Mots-clés

Réalité augmentée, HoloLens, Visual Studio, Unity, Domaine medical, Epaule

Abstract

The project "Hololens, Augmented Reality and Shoulder" is the result of some projects aiming to assist a surgeon in augmented reality during the operation of a shoulder. This will make a link between the medical field and the field of computer science. In order to achieve our purpose, we will use augmented reality glasses: the Hololens. It will be necessary to display a 3D model of the shoulder scapula on the glasses but also the reconstituted glenoid. It will require surgical precision as a result of the superposition of this 3D overprint on the glasses with the actual scapula.

Keywords

Augmented Reality, HoloLens, Visual Studio, Unity, Medical field, Shoulder

Tuteurs académiques

Jordan NICOT
Mohamed SLIMANE
Julien BERHOUE
Christian PROUST

Étudiant

Alexis BARON (DI5)