

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

www.polytech.univ-tours.fr

Projet Recherche & Développement 2016-2017

Projet Epaule et Lunettes Connectées

Tuteurs académiques

Julien BERHOUE

Mohamed SLIMANE

Christian PROUST

Étudiants

Jordan NICOT (DI5)

Liste des intervenants

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

Avertissement

Ce document a été rédigé par Jordan NICOT susnommé l'auteur.

L'école polytechnique de l'université François Rabelais de Tours est représentée par Julien BERHOUET, Mohamed SLIMANE et Christian PROUST susnommés les tuteurs académiques.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable des tuteurs académiques.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.

Pour citer ce document :

Jordan NICOT, *Projet Epaule et Lunettes Connectées*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2016-2017.

```
@mastersthesis{
  author={NICOT, Jordan},
  title={Projet Epaule et Lunettes Connectées: },
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2016-2017}
}
```

Table des matières

I	Introduction	1
1	Introduction	2
1	Contexte de la réalisation	5
2	Objectifs	5
2.1	Objectif 1	5
2.2	Objectif 2	6
3	Objectif 3	6
3.1	Objectif 4	6
4	Hypothèses	7
5	Bases méthodologiques	7
5.1	Gestion de projet	7
5.2	Démarche scientifique	7
2	Description générale	9
1	Environnement du projet	9
2	Caractéristiques des utilisateurs	9
3	Fonctionnalités du système	9
4	Structure générale du système	10
4.1	Structure générale matérielle	10
4.2	Environnement matériel	10
4.3	Structure générale de l'application	13
II	Etats de l'art	14
3	Réalité augmentée	15
1	Histoire	15

2	Techniques	16
2.1	Les éléments de mesure spatial	16
2.2	Inertial measurement unit (IMU).....	17
3	Le techniques de reconnaissance	17
3.1	Marker based AR	17
3.2	Markerless AR	18
4	Application	18
5	Avantages/Limites.....	20
6	Dénominations associées au projet	20
4	Les logiciels adéquats	22
1	Les applications (CMS) pour la réalité augmentée	22
2	Les SDKs pour la réalité augmentée	23
2.1	Vuforia	23
2.2	Dioata.....	23
2.3	ARToolkit.....	24
2.4	Wikitude	24
2.5	ARMedia SDK.....	24
3	Problématiques.....	24
5	Les lunettes 3D	26
1	Tableaux comparatifs	27
2	Les lunettes 3D Hololens par Microsoft.....	28
2.1	Conditions d'achat	29
2.2	Caractéristiques techniques.....	29
2.3	Système d'exploitation et plateforme de développement.....	29
III	Analyse et conception	31
6	Analyse	32
1	Conception des objectifs 1 et 3.....	32
2	Conception des objectifs 2 et 3.....	33
IV	Mise en œuvre	36
3	Bilan sur la mise en œuvre qualité.....	37
	Conclusion	38

Annexes	40
A Annexe A	41
1 Interface Homme-Machine	41
2 Spécifications fonctionnelles.....	42
3 Spécifications non fonctionnelles.....	43
3.1 Contraintes de développement et conception.....	43
3.1.1 Matériels	43
3.1.2 Langages de programmation.....	43
3.1.3 Logiciels et bibliothèques	44
3.2 Contraintes de fonctionnement et d'exploitation	44
3.2.1 Performances	44
3.2.2 Capacités.....	44
3.2.3 Contrôlabilité.....	44
3.2.4 Sécurité	44
3.2.5 Maintenance et évolution du système	44
B Gestion de projet	45
C Comparatif SDK	47
D De la configuration au déploiement	49
1 Le TargetManager de Vuforia.....	49
1.1 Création d'un compte et d'une licence Vuforia	49
1.2 Vuforia AR Starter Kit	50
2 Configuration dans Unity	50
2.1 Configuration de Vuforia dans Unity.....	50
2.2 Création de cible et configuration de la caméra Vuforia	51
2.2.1 Lier la cible Vuforia avec la scène	52
3 Configuration de la caméra	53
3.1 Export du projet Unity C# vers Visual Studio	54
4 Déploiement du projet vers les lunettes	55
4.1 Exportation du projet vers les lunettes Hololens.....	55
5 Guide d'utilisation.....	56
5.1 Démarrage de l'application	56
6 Utilisation de l'application	57
E Dossier de tests	58
1 Plan d'expérience.....	58
2 Résultats des tests.....	59
2.1 Type 1 – Single Image	59
2.1.1 Test 1 (19/01/2017).....	59
2.1.2 Test 2 (19/01/2017)	59
2.1.3 Test 3 (20/01/2017)	60
2.1.4 Test 4 (2/03/2017)	60

2.2	Type 2 – Cuboid	61
2.2.1	Test 1 (12/02/2017)	62
2.2.2	Test 2 (19/02/2017)	62
2.3	Type 3 – Association de marqueurs avec une maquette d'épaule	63
2.3.1	Test 1 (8/03/2017)	63
2.3.2	Test 2 (15/03/2017)	63
2.3.3	Test 3 (22/03/2017)	64
3	Conclusion	64

Table des figures

1 Introduction

1	Avant.....	2
2	Maintenant, une partie de la glène a disparu	3
3	Mais comment être sûr de « bien » positionner la prothèse de la glène?	3
4	T. Gregory, Hôpital Européen Georges Pompidou – Paris	3
5	Prérequis.....	4
6	Résultat souhaité - 1	4
7	Résultatsouhaité - 2	4
8	Résultat souhaité - 3.....	4
9	Résultat souhaité - 4.....	4
10	Résultat souhaité - 5.....	5
11	Objectif 1	6
12	Objectif 2	6
13	Objectif 4	6
14	Base méthodologique	8

2 Description générale

1	Diagramme des cas d'utilisation.....	10
2	Diagramme de déploiement.....	10
3	Caractéristiques techniques des Hololens	12

3 Réalité augmentée

1	Première machine de réalité augmentée.....	15
2	Premier marqueur 2D	16
3	Les six degrés de libertés.....	17
4	QR-Code.....	17

5	Marqueur sur un objet 3D.....	18
6	Application du secteur médical	19
7	Pokémon Go.....	19
8	Application du secteur de la mode	20
9	Le réel et la scène.....	20
10	La cible et le masque	21
4	Les logiciels adéquats	
1	Logo de Vuforia	23
2	Logo de Diota.....	23
3	Logo de ARToolKit	24
4	Logo de Wikitude	24
5	Logo de ARMedia	24
5	Les lunettes 3D	
1	Comparatif lunettes 3D : 1ère partie	27
2	Comparatif lunettes 3D : 2ème partie.....	28
3	Les lunettes 3D Hololens.....	29
4	Plateforme de développement de Windows	29
6	Analyse	
1	TargetManager sous Vuforia.....	33
2	Exemple avec OpenCV	33
3	Spatial understanding	34
4	Spatial understanding 2	35
5	Maquette côté droit	37
6	Maquette de face.....	37
	Conclusion	
7	Présentation du projet à Marisol Touraine	39
A	Annexe A	
1	IHM des applications PC et Lunettes.....	41
B	Gestion de projet	
1	Diagramme de Gantt PRD1.....	45
2	Diagramme de Gantt PRD2.....	46
3	Diagramme de Gantt PRD2 effectué.....	46
C	Comparatif SDK	
1	Tableau comparatif des SDKs - 1ère partie.....	47
2	Tableau comparatif des SDKs - 2ème partie	48

D De la configuration au déploiement

1	Licence Manager	50
2	Vuforia AR Starter Kit	50
3	Vuforia Configuration	51
4	Add Database.....	51
5	Add Target	51
6	Création d'une cible	52
7	Bouton "Download Database"	52
8	Download Database.....	53
9	.unitypackage.....	53
10	Activer la base de données	53
11	ARCamera.....	53
12	Qualité de l'application	54
13	Configuration de l'application 1	54
14	Configuration de l'application 2	55
15	Versionning.....	55
16	Récupérer l'IP des lunettes.....	56
17	Connexion à distance	56
18	Appareillage	57
19	App Manager	57

E Dossier de tests

1	Test 1	59
2	Test 2.....	59
3	Test 3.....	60
4	Test 4.....	61
5	Mise en place de la cible et la scène	61
6	Test 1	62
7	Test 1	62
8	Image original.....	63
9	Image original.....	63
10	Test 1	64

Première partie

Introduction

1

Introduction

Ce document présente la première partie du projet Recherche & Développement « Lunettes 3d et prothèses d'épaule ». Il définit les premières idées de développement avec une approche globale. Le client est le LI & le CHRU Trousseau (Dr Julien Berhouet). Le maître d'œuvre est Jordan NICOT. 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 Epaule » 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.

Voici un résumé :

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



Figure 1 – Avant



Figure 2 – *Maintenant, une partie de la glène a disparu*

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



Figure 3 – *Mais comment être sûr de « bien » positionner la prothèse de la glène ?*

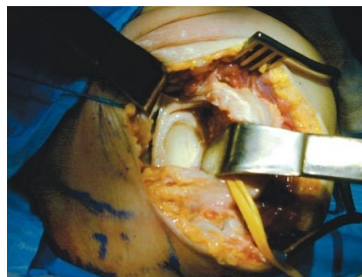


Figure 4 – *T. Gregory, Hôpital Européen Georges Pompidou – Paris*

c) On dispose de :

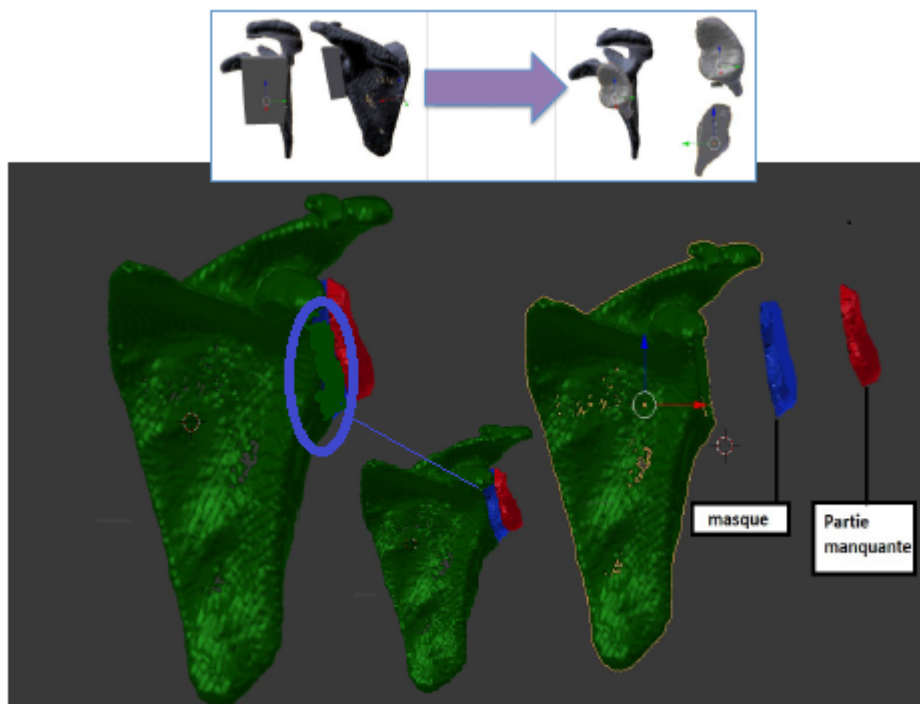


Figure 5 – Prérequis

d) On veut :

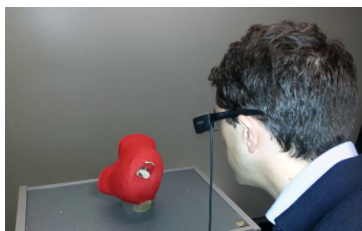


Figure 6 – Résultat souhaité - 1

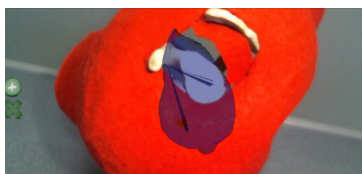


Figure 7 – Résultat souhaité - 2

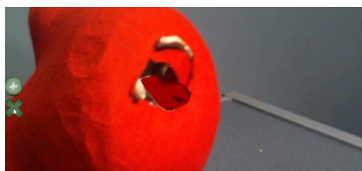


Figure 8 – Résultat souhaité - 3

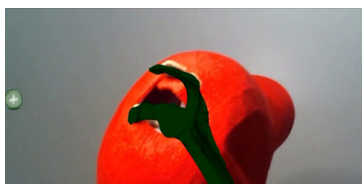


Figure 9 – Résultat souhaité - 4

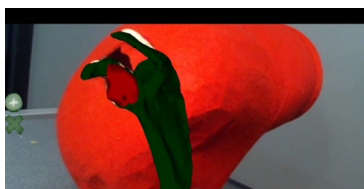


Figure 10 – Résultat souhaité - 5

Il s'agit donc de prendre en main les récentes technologies de Réalité Augmentée, notamment les « optical see-through head-mounted display »¹ et, plus précisément le fonctionnement des lunettes intelligentes connectées (« smart glasses »).

Nous reviendrons prochainement plus en détails sur cette technologie, d'un point de vue historique, technique, applicatif, etc... Nous aborderons également de manière approfondie les avantages et les limites de ce système.

1 Contexte de la réalisation

Depuis près de cinq ans, plusieurs étapes ont été abordées afin de tendre vers l'objectif final d'une assistance 3d lors d'une opération chirurgicale avec des lunettes de réalité augmentée. Dans un premier temps il y a eu la reconstitution du modèle 3D de la scapula pathologique à partir d'images de scanner. Ensuite il y a eu un travail d'analyse de données, puis un travail de reconstruction 3D à partir d'un nuage de points, afin de reconstruire, en 3D, la partie disparue de la glène de la scapula pathologique, en fonction d'autres parties intactes de la scapula.

Concernant la dernière étape avec des lunettes 3d, un premier travail exploratoire, en 2014, a dressé un état de l'art des lunettes électroniques 3D. Un premier prototype a été réalisé (en 2014-2015, 2015-2016) sur des lunettes BT200 de Epson, avec le logiciel Creator de Metaio [1] [2].

2 Objectifs

Il y a plusieurs objectifs lors d'un projet de Recherche & Développement. Il y a l'objectif général du projet comme expliqué précédemment. La réalisation de cet objectif nous amène à fournir différents travaux. Dans un premier temps nous allons fournir un état de l'art approfondi sur les lunettes 3d permettant de faire de la réalité augmentée. Ensuite nous réaliserons un second état de l'art, cette fois-ci sur les logiciels et SDKs de la réalité augmentée. Ces états de l'art vont nous permettre de sélectionner les meilleurs outils pour la réalisation du projet. Nous devons être en adéquation entre les matériels et les logiciels, car tous les logiciels ne sont pas toujours compatibles avec les différents matériels.

Dans le cadre du Projet de Recherche & Développement, nous devons également gérer toute la partie conception et analyse, ainsi que la gestion de projet. Ce document inclut, dans un découpage différent, les principaux éléments de notre cahier de spécifications.

2.1 Objectif 1

Afin de parvenir à la réalisation du projet nous allons passer par des étapes intermédiaires. On souhaite, au travers des lunettes 3d, superposer précisément un objet virtuel 3d (« une scène ») sur un objet réel de type cube (« la cible »), puis pouvoir visualiser l'ensemble sous tous les angles, selon la position de la tête et donc du regard. La cible possède un marqueur de type QR code ou une image. Nous souhaitons arriver à un résultat semblable à l'image suivante :

1. Il s'agit du dispositif où l'utilisateur perçoit le monde réel à travers un dispositif optique dans lequel sont projetés les images virtuelles destinées à l'augmentation



Figure 11 – Objectif 1

2.2 Objectif 2

Ici, la cible est un cube (un parallélépipède), mais sans image ni QR Code.

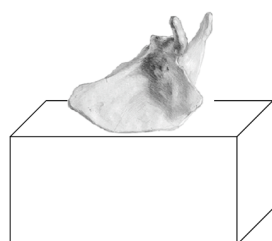


Figure 12 – Objectif 2

3 Objectif 3

La cible est un objet « à nous », c'est à dire que l'on pourra importer notre cible d'un format 3d. Un artifice de type image pourrait être utilisé, en ajoutant un QR-Code sur la glène par exemple.

3.1 Objectif 4

L'objectif final sera de répondre à la problématique initiale. La scène est la scapula pathologique reconstituée, et la cible est la glène réelle visible pendant l'intervention chirurgicale². La figure suivante illustre notre objectif.

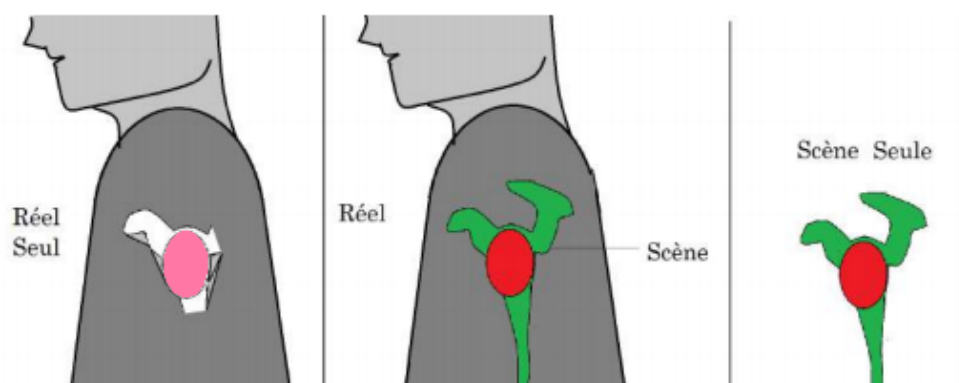


Figure 13 – Objectif 4

Lors de la réalisation de ces objectifs, nous utiliserons un modèle 3d d'une omoplate comme objet virtuel à incruster dans le réel.

2. La glène est un objet 3D bien plus complexe qu'un simple cube.

4 Hypothèses

Nous souhaitons utiliser une technologie qui est récente sur le marché. Aussi bien au niveau logiciels que matériels. L'état de l'art sur les logiciels de réalité augmentée permet d'avoir un vaste panorama des solutions; ainsi on espère avoir des solutions de replis, en cas de suppression de la plateforme par exemple (comme lors d'un précédent projet). L'univers de la réalité augmentée semble très fermé. En effet, les possibilités d'innovation sont presque infinies, les différents acteurs et développeurs restent prudents quant à la diffusion de leur travail.

Il sera peut-être envisageable d'ajouter des fonctionnalités à des logiciels existants (si cela est possible) ou alors créer nos propres fonctionnalités à travers des bibliothèques de plus bas niveau, comme OpenCV par exemple.

Il est également possible que les différents objectifs « lunettes » ne nécessitent pas les mêmes outils. Nous allons tenter de réaliser le premier objectif avec des outils permettant de réaliser le deuxième objectif, et ainsi de suite. Ainsi, on gagnera en temps d'apprentissage. Si cela n'est pas possible, nous aurons l'occasion d'utiliser plusieurs outils, et ainsi d'enrichir nos connaissances sur les différentes fonctionnalités de chaque outil.

5 Bases méthodologiques

5.1 Gestion de projet

Nous allons utiliser la méthode de cycle en V concernant la gestion de projet. Celle-ci s'adapte très bien à notre projet de recherche. Les quatre objectifs se complètent et permettent de suivre un cycle identique.

Nous n'utiliserons pas la plateforme Redmine de Polytech Tours, afin de partager notre application. En effet, je serai le seul à travailler sur le code produit, de ce fait il n'y a pas d'intérêt à le partager. Nous utilisons le logiciel MindView, qui est fourni par l'école, et qui nous permet de créer nos diagrammes de Gantt. Les tests seront définis avant de coder. Ils sont basés sur le besoin fonctionnel, la validation de chaque test est obligatoire.

Vous trouverez en annexe de ce document plus de détails sur notre gestion de projet.

5.2 Démarche scientifique

Dans cette section nous allons lister les différentes fonctionnalités de réalité augmentée nécessaires :

- Détection de la cible;
- Reconnaissance du nouvel emplacement et de la nouvelle orientation de la cible;
- Repositionnement de la scène;
- Rafraîchissement de l'affichage;

Le schéma suivant illustre le cycle mis en œuvre :

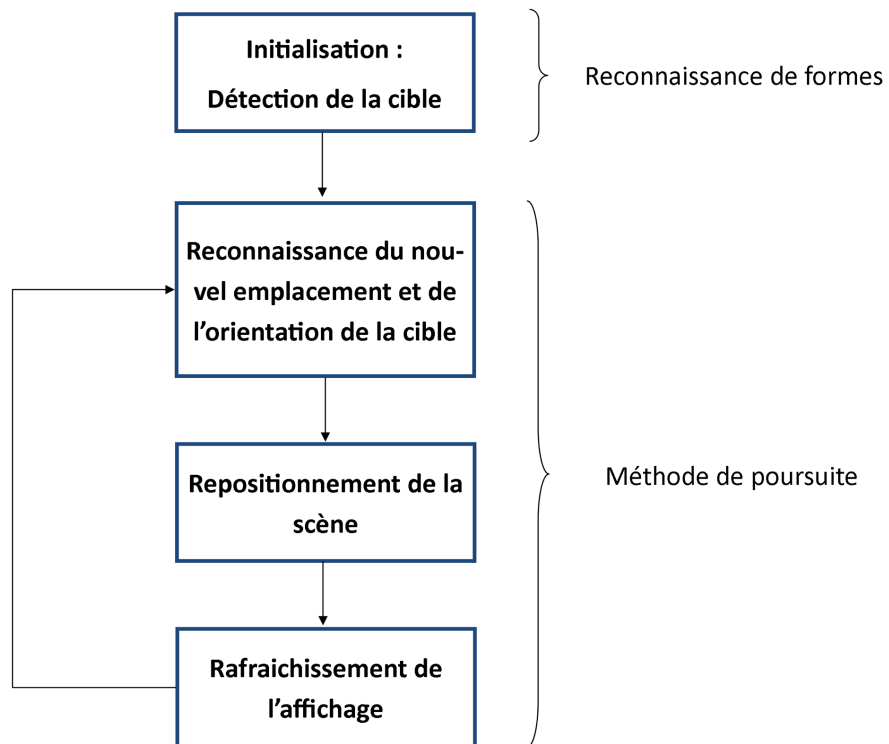


Figure 14 – Base méthodologique

2

Description générale

1 Environnement du projet

Dans cette section nous allons énoncer l'état de l'environnement préexistant. Ce travail a déjà fait l'état de plusieurs recherches. Nous avons pu nous appuyer sur un état de l'art antérieurement effectué sur les lunettes 3d [1] ainsi que sur quelques notions de la réalité augmentée. L'objectif de surimpression d'une omoplate a déjà été effectué grâce au logiciel Métio Créator (disparu du marché après avoir été racheté par Apple), sur des lunettes connectées BT200 de Epson. Mais ce logiciel fonctionne comme une boîte noire, c'est-à-dire que nous n'avons pas accès aux différents algorithmes utilisés. Dans le cas contraire, nous aurions pu utiliser la même méthodologie avec un environnement de développement différent, en utilisant des fonctionnalités semblables.

Nous travaillons avec des lunettes 3d différentes de celles utilisées lors du précédent projet, les caractéristiques matérielles sont très différentes (nombre de caméras, résolution, capteurs), de ce fait nous ne pouvons pas nous appuyer sur la méthodologie utilisée précédemment. Nous sommes donc contraints de repartir de zéro sans pouvoir utiliser un environnement logiciel préexistant.

2 Caractéristiques des utilisateurs

On distingue deux types d'utilisateurs du système :

Un utilisateur dit '*chirurgien*'. L'objectif est qu'il soit quasiment autonome avec l'application. Il devra pouvoir intégrer l'omoplate en fichier 3d dans le logiciel à partir d'un scanner sur le patient fait au préalable, faire la manipulation d'extraction de la glène et la définir comme "masque" (i.e image) de la cible. Ensuite, il pourra intégrer aux lunettes 3d, ce travail préliminaire.

Un utilisateur dit '*technicien*' qui possède des compétences informatiques suffisantes en programmation afin d'ajouter des modules/outils complémentaires. C'est un utilisateur occasionnel, qui n'interviendra que si l'on souhaite rajouter des fonctionnalités.

3 Fonctionnalités du système

On a choisi ici de représenter les fonctions utilisateurs du système au travers d'un diagramme de cas d'utilisation :

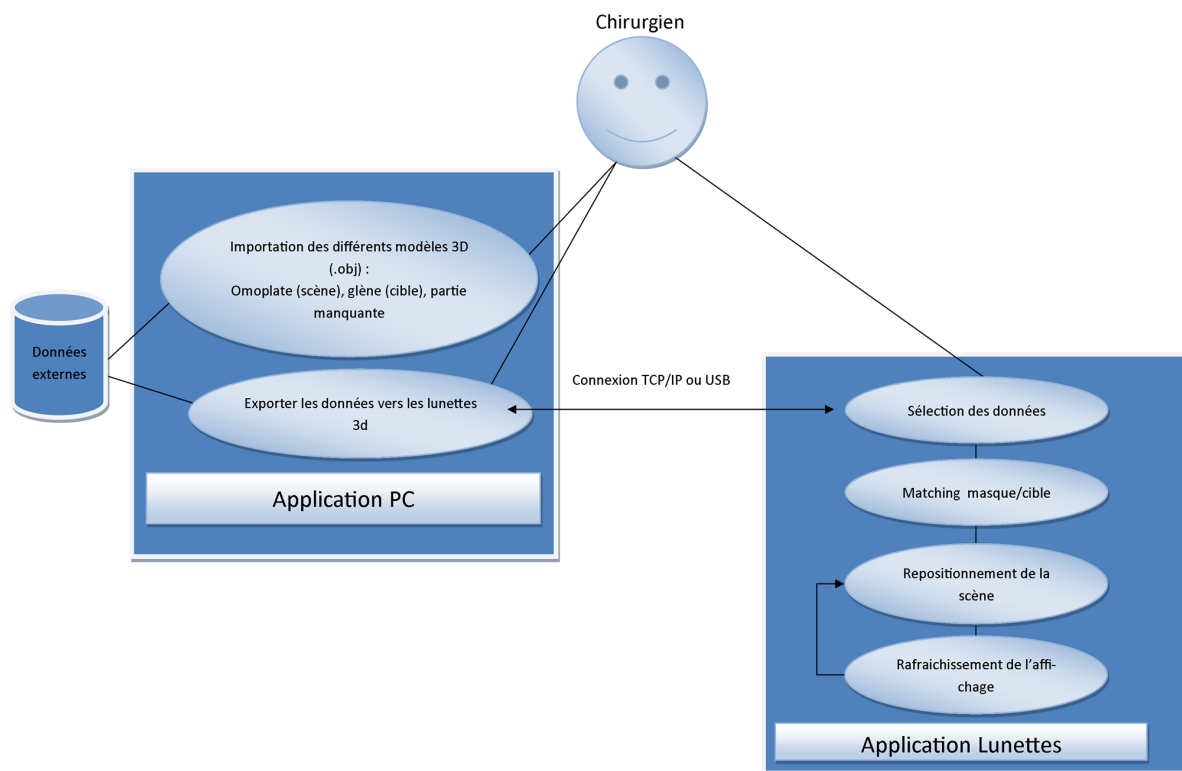


Figure 1 – Diagramme des cas d'utilisation

4 Structure générale du système

4.1 Structure générale matérielle

Après une phase de recherche bibliographique, présentée dans la suite du document, nous avons choisi d'utiliser des lunettes 3d développées par Microsoft. Nous allons donc utiliser Windows comme environnement logiciel. Le schéma suivant montre, dans une première analyse, les outils logiciels que nous allons utiliser durant le projet.

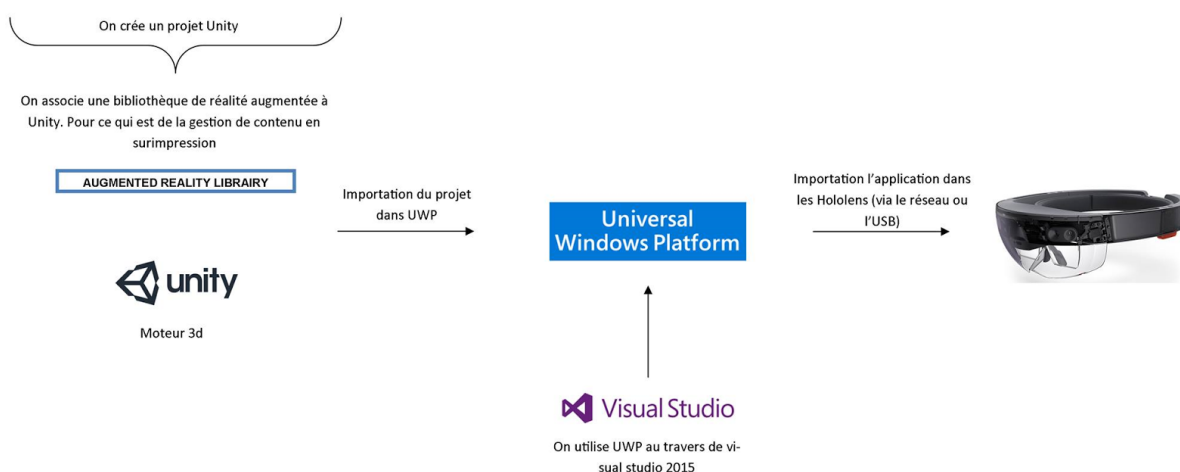


Figure 2 – Diagramme de déploiement

4.2 Environnement matériel

Nos contraintes sont bien évidemment liées aux lunettes 3d que nous utilisons. Nous nous adaptons aux caractéristiques des Hololens de Microsoft.

D'un point de vue optique, les Hololens embarquent deux écrans HD 16 :9¹. Elles embarquent aussi un nombre important de capteurs, essentiels à la réalité augmentée. On compte 5 caméras. Une de profondeur, et 4 pour la compréhension de l'environnement. On a la possibilité de prendre des photos de 2M et de faire des vidéos en haute définition. Il y a 4 microphones et capteurs de lumière. En terme de connectivité, il y a le Wifi 802.11ac, le Micro USB 2.0 et le Bluetooth 4.1 LE. L'autonomie de la batterie de 2-3 heures est convenable compte tenu de la technologie embarquée. Niveau performance, on a un processeur Intel 32 bit cadencé à 1,04 GHz, 2GB de RAM et 64GB de mémoire flash. Les Hololens possèdent aussi le Microsoft Holographic Processing Unit (HPU)[20]. Ce processeur permet d'interagir avec l'environnement, en reconnaissant les gestes, la voix et la parole.

Pour un projet de réalité augmentée, il semble que ces lunettes 3d soit un outil idéal. Les caractéristiques puissantes, notamment au niveau de la prise de vue, vont nous permettre d'avoir une vue de l'environnement optimale.

Vous trouverez ci-dessous un tableau récapitulatif des caractéristiques techniques des lunettes Hololens :

1. HD 16 :9 : La résolution est de 1280x720, avec 16 :9 comme proportion d'image.


Marque	MICROSOFT
Modèle	Hololens
Photo	
Mise à jour	29/09/2016
Prix	3000\$
Disponibilité	Uniquement pour les développeurs
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	
Camera(s)	5 caméras, dont une de profondeur et de 2MP, 1 capteur de lumière
Champ de vision	40°
Processeur	Intel Atom x5-Z8100 - 1,04 GHz et HPU
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/fr-fr/windows/holographic/hardware_details

Figure 3 – Caractéristiques techniques des Hololens

L'un des inconvénients important de ce matériel concerne les fonctionnalités déjà existantes. Est-ce que les méthodes de calibration² sont déjà existantes? Si ce n'est pas le cas, il y aura un travail supplémentaire à fournir. Nous devons envisager toutes les éventualités.

2. Calibration : la calibration consiste à réaliser des tests pour initialiser les distances, et autres mesures (couleur, luminosité, etc...)

4.3 Structure générale de l'application

L'utilisateur utilise dans un premier temps une application Pc, afin d'importer les différentes données 3d de l'omoplate pathologique (en vert), du masque (image de la glène pathologique) (en bleu) et de la partie manquante de la glène, reconstituée (en rouge). Dans une version future il sera intéressant de pouvoir faire les différentes manipulations d'extraction de la glène (i.e création du masque) et de la génération de la partie manquante directement au travers de l'application. Aujourd'hui ces étapes sont faites, antérieurement, au travers du logiciel 3d Blender. Ensuite ces données sont exportées vers les lunettes 3d soit grâce à une connexion internet de type TCP/IP soit via un câble USB.

Deuxième partie

Etats de l'art

3

Réalité augmentée

La réalité augmentée (« AR ») désigne les systèmes informatiques qui rendent possible la superposition d'un modèle virtuel 2D ou 3D à la perception que nous avons naturellement de la réalité et ceci en temps réel. Elle désigne les différentes méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans une séquence d'images. Elle s'applique aussi bien à la perception visuelle (superposition d'image virtuelle aux images réelles) qu'aux perceptions proprioceptives comme les perceptions tactiles ou auditives.¹

Il ne faut pas confondre Réalité Augmentée et Réalité Virtuelle.[21]

Les éléments virtuels peuvent être en deux ou trois dimensions. On a alors la possibilité d'ajouter du contenu à des fins principalement décisives. En effet, on peut accélérer le temps en visualisant un changement par rapport à l'environnement présent qui aurait nécessité beaucoup de temps et de ressources sinon. L'exemple de l'aménagement du mobilier est très parlant. Il devient possible "d'essayer" l'effet d'un nouveau canapé dans votre salon ou encore le résultat d'une nouvelle peinture, et cela de manière quasiment instantanée. Cette technologie est récente et reste le plus souvent au stade de recherche. Les domaines d'application sont nombreux, comme l'industrie, l'éducation ou encore la santé. Le marché mondial de la réalité virtuelle et augmentée représenterait 150 milliard de dollars d'ici 2020, selon Digi-Capital[9], dont pas moins de 120 milliards pour la réalité augmentée. On remarque qu'avec de nombreux événements à travers le pays, la France cherche à se positionner dans ce domaine qui est en train de devenir un des enjeux de demain.

1 Histoire

C'est en 1968 qu'Ivan Sutherland crée la première machine de réalité augmentée. Elle met en œuvre les six degrés de liberté avec un casque d'écran transparent. Elle n'est bien évidemment pas comparable à ce que l'on fait aujourd'hui, de part l'évolution des technologies en 40 ans. [2]



Figure 1 – Première machine de réalité augmentée

Le terme de Réalité augmentée vient plus tard, dans les années 90 par Tom Caudell et David Mizell. Il sera réellement exploité avec les travaux de Paul Milgram et Fumio Kishino en 1994.

1. cf Wikipédia

En 1996, il y a l'apparition des premiers marqueurs 2D, avec la visualisation d'objets virtuels. C'est une très grande avancée réalisée par Jun Rekimoto.

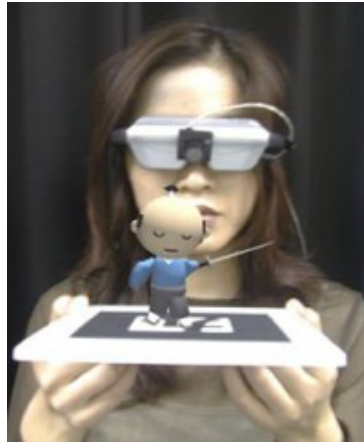


Figure 2 – Premier marqueur 2D

C'est depuis le début des années 2000, que la réalité augmentée, sous la forme que l'on connaît aujourd'hui, arrive avec les premiers téléphones. Les énormes progrès faits en traitement de l'image sont aussi très importants dans l'évolution de cette technologie. En effet, la reconnaissance de forme est une partie essentielle de la réalité augmentée, et permet de décupler le champ des possibilités.

2 Techniques

2.1 Les éléments de mesure spatial

Magnétomètre :

Ce capteur mesure le champ magnétique. Il permet de détecter les directions, comme une boussole.

Accéléromètre :

L'accéléromètre permet de se situer dans l'espace, au travers des 3 axes. Son atout principal est la connaissance du mouvement et de la translation.

Gyroscope :

Ce troisième capteur est complémentaire à l'accéléromètre. Il calcule l'orientation et peut ainsi corriger les données de l'accéléromètre.

La combinaison de ces trois capteurs permet de connaître précisément le degré de mouvement. Tous les déplacements peuvent donc être quantifiés. On l'appelle les six degrés de liberté (six-degrees-of-freedom (6DoF)).^[10]

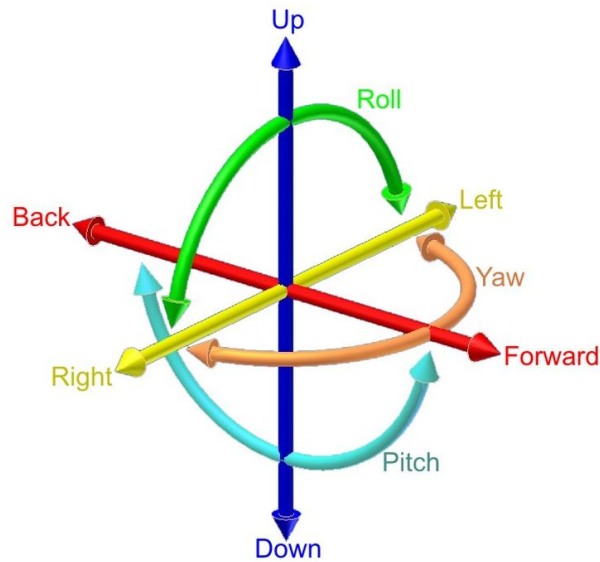


Figure 3 – Les six degrés de liberté

2.2 Inertial measurement unit (IMU)

La navigation inertielle est une technique qui permet de mesurer le mouvement. Cette technologie est aujourd'hui présente sur tous les smart phones que nous utilisons. Elle utilise pour cela des capteurs d'accélération et de mouvement. Il faut 3 capteurs d'accélération, un pour chaque axe, afin de mesurer l'accélération d'un mobile par rapport au référentiel galiléen. Il faut également coupler ces capteurs à trois gyroscope pour la mesure de la rotation. On peut ainsi récupérer les repères de position terrestre : la latitude, la longitude et l'altitude.

3 Le techniques de reconnaissance

On distingue deux catégories de reconnaissance pour la réalité augmentée : l'AR avec marqueurs (Marker bases AR) et l'AR sans marqueurs (Markerless AR). A ce jour, la première est largement la plus utilisée, car également la plus simple à mettre en place, elle comprend les QR-Code et les images sur plan 2d. La seconde catégorie est Markerless AR.[7]

3.1 Marker based AR

Les QR-Code : On considère comme un Qr-code, une figure en noir et blanc avec des contours précis de forme droite.



Figure 4 – QR-Code

Marqueurs 2d : On place dans la catégorie des marqueurs 2d, les images. Elles sont généralement imprimées.

Marqueurs 3d : On peut utiliser des marqueurs sur des objets 3d. L'objet est une forme simple, connue, sur lequel a été placé un marqueur.

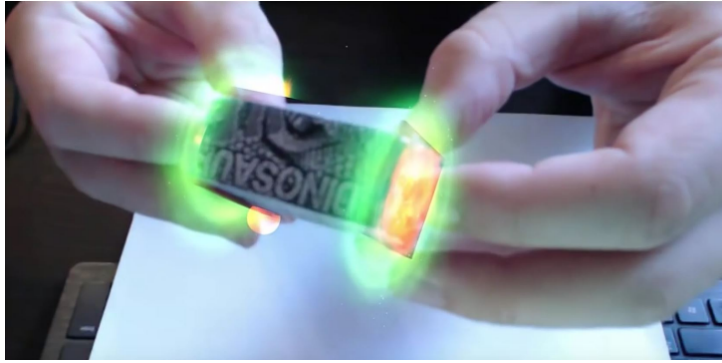


Figure 5 – Marqueur sur un objet 3D

3.2 Markerless AR

Le terme sans marqueur est sans doute un abus de langage, car il y a toujours un marqueur. Mais celui se trouve sous une autre forme. La réalité augmentée sans marqueurs consiste à détecter un objet en temps réel sans marqueurs spécifiques. La cible qui était une image par exemple devient un objet, ou le marqueur est l'objet lui-même, de par sa surface.

La réalité augmentée sans marqueurs est incontestablement plus puissante qu'avec des marqueurs. L'approche est différente et le champ des possibilités décuplé.

Notre problématique demande d'utiliser la seconde technologie. Notre marqueur est la partie de la scène. C'est un marqueur complexe, qui n'a pas une forme géométrique standard (cube, cône, etc...).

4 Application

L'univers d'application de réalité augmentée est quasiment infini. On note cependant plusieurs secteurs dans lequel les progrès seraient les plus remarquables. Les secteurs éducatifs, médicaux et industriels sont les plus prometteurs.

Prenons l'exemple d'une usine où l'on monte des moteurs. L'ouvrier pourrait alors utiliser une paire de lunettes de réalité augmentée qui le guidera en temps réel. Il aura alors accès à la procédure de montage en surimpression avec, en indiquant avec une flèche la pièce à monter, les outils à utiliser, la position où la monter sur le moteur mais surtout quelles sont les étapes du montage. On peut également prendre l'exemple d'un plombier virtuel. Imaginez que votre lavabo soit bouché, vous pourriez appeler un plombier à distance qui vous guidera sur les étapes de réparation. Il pourra alors ajouter des informations dans votre champ de vision, car il aura accès aux mêmes images que vous grâce à l'enregistrement.

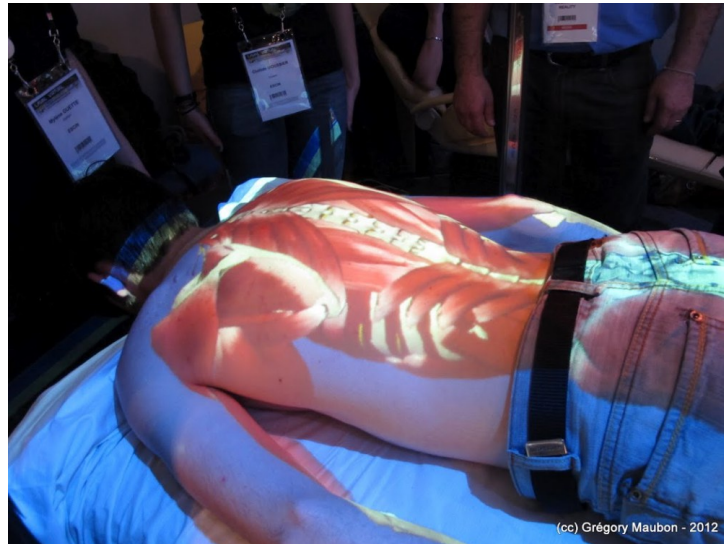


Figure 6 – *Application du secteur médical*

On peut également utiliser ces exemples pour illustrer les possibilités dans l'éducation. Les méthodes d'enseignement seront alors révolutionnées, avec une interaction sur le réel démultipliée. Dans le secteur médical, notre projet est un très bon exemple. Une intervention chirurgicale est une étape délicate, où l'on ne voit pas toujours ce que l'on cherche. Le corps humain est très complexe et composé de nombreuses couches, comme les muscles, les tendons, les veines, etc... Visualiser l'ensemble des éléments du corps humain sans avoir à l'ouvrir sera un gain de temps considérable, aussi bien pour le chirurgien que pour le patient. Les jeux vidéo s'améliorent avec la réalité augmentée. Il devienne plus interactif. Le jeu Pokémon Go est le meilleur exemple aujourd'hui. Il y a également le secteur Marketing, et notamment le street marketing. Il est possible de « déposer » du contenu publicitaire un peu partout dans une ville, un magasin ou encore dans une gare.

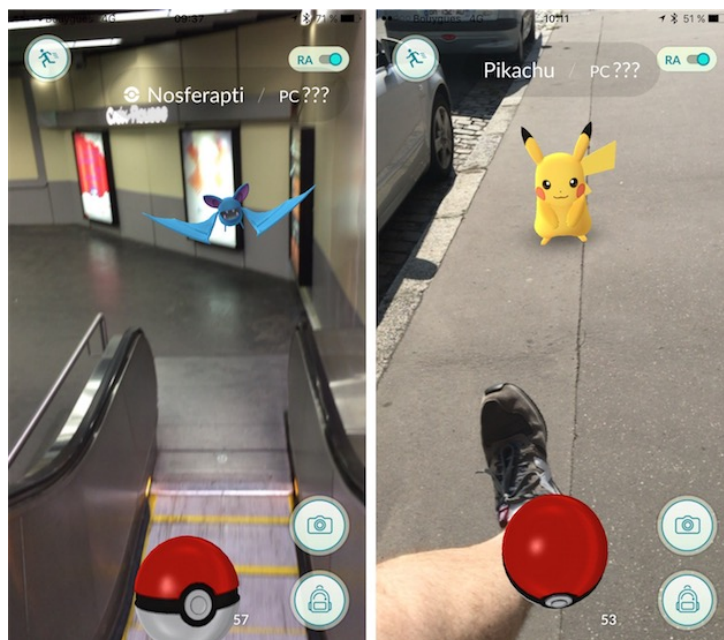


Figure 7 – *Pokémon Go*

On compte aussi la mode, le tourisme ou encore la presse qui peuvent tirer parti de cette technologie grandissante.



Figure 8 – Application du secteur de la mode

5 Avantages/Limites

Après ces premiers chapitres, il semble n'y avoir que des avantages à la réalité augmentée. Pourtant des limites se ressentent au niveau de l'interaction homme-machine. La réalité augmentée peut faire diminuer nos aptitudes de réflexion. En effet pourquoi réfléchir si on obtient toutes les informations voulues en un temps record. On devient assisté dans toutes les tâches du quotidien. Certains parlent même d'humanité diminuée.[8]

Les questions de propriétés de l'image interpellent, comment garantir le respect de la vie privée avec un appareil qui enregistre tous ce que l'on voit ?

J'ai découvert de nombreux écrits sur l'émergence des nouvelles technologies et l'évolution de l'Homme, mais ce n'est pas le sujet. Il est important de garder à l'esprit qu'elles sont les retombées sur l'éthique en manipulant ces nouvelles technologies. Elles apportent principalement des avantages, mais il ne faut pas ignorer les inconvénients qui en découlent.

6 Dénominations associées au projet

Il est important que l'on ait tous le même vocabulaire. En effet, il est relativement récent et on se trompe facilement. Nous allons également utiliser un code couleur pour chaque objet indépendant.

On appellera ce qui existe le **réel**. Le virtuel sera appelé la **scène**. La **scène** est l'ensemble des objets virtuels qui seront affichés en surimpression au réel. Sa couleur est le vert et le rouge.

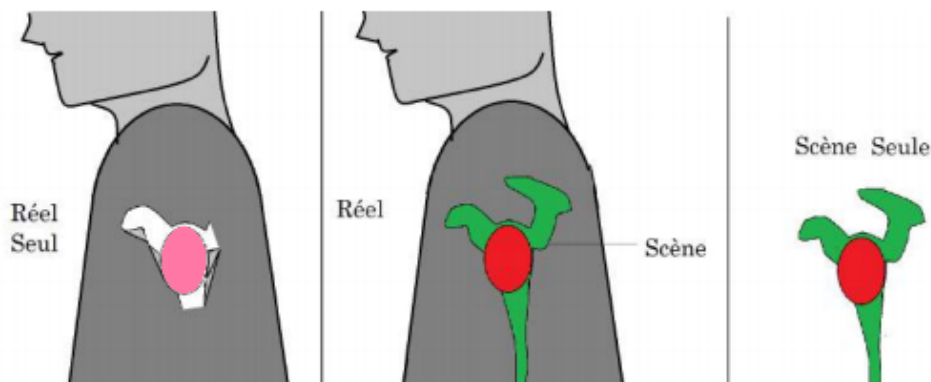


Figure 9 – Le réel et la scène

Dans la reconnaissance de forme, on appellera **cible** l'objet qu'on cherchera à reconnaître Ici, sa couleur est le rose. Le **masque** est le modèle que l'on utilise pour la reconnaissance, sa couleur est le bleu. La

partie de la glène manquante reconstituée mathématiquement est de couleur rouge. Elle fait partie de la scène.

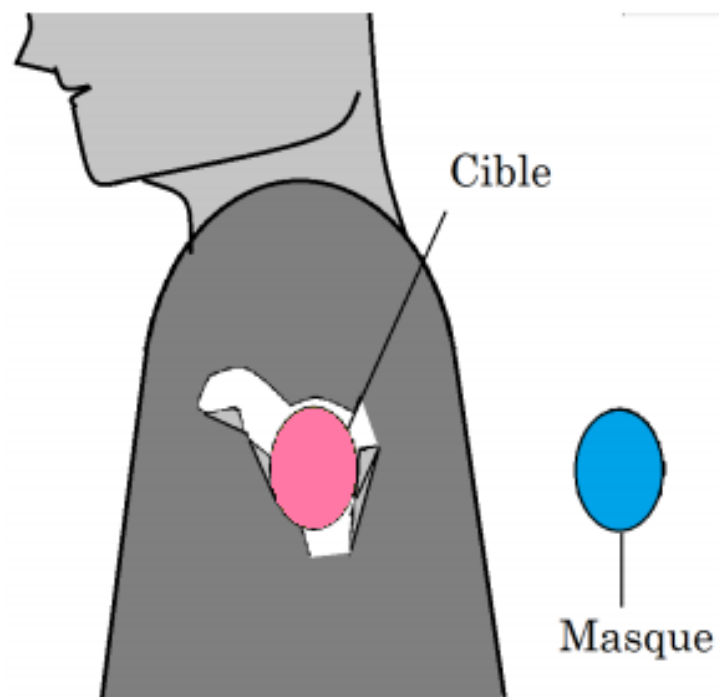


Figure 10 – La cible et le masque

4

Les logiciels adéquats

Pour faire de la réalité augmentée il “suffit” d’incruster un élément de type image, vidéo ou 3d en surimpression au réel, en général une vidéo de ce que l’on perçoit en temps réel. Il y a alors plusieurs moyens d’y parvenir.[3]

Il existe deux types de solution pour l’AR. Les CMS et les SDK. Un CMS est un logiciel ou un service en ligne qui permet de réaliser rapidement des projets de réalité augmentée. Les choix des possibilités est réduit, on doit se contenter de ce que propose le logiciel. Des connaissances de l’informatique ne sont pas nécessaires. On ne peut pas utiliser une application qui utilise ces plateformes à son nom. A l’inverse il y a les SDKs. Ce sont des plateformes de développement qui permettent de développer une application en utilisant les fonctionnalités disponibles. Il est également possible d’en modifier et d’en ajouter. Des connaissances en développement sont indispensables. Notre choix se portera davantage sur les SDKs.

1 Les applications (CMS) pour la réalité augmentée

Plusieurs applications sont disponibles sur Smartphones. Elles permettent de choisir un objet 3d dans une base de données ou d’en importer, puis de choisir le marqueur. Ensuite il faut placer la caméra de son ordinateur ou de son smartphone devant le marqueur, l’application se charge ensuite de créer la réalité augmentée avec l’objet choisi. Voici une liste non exhaustive de quelques applications :

- Auresma par Hewlett-Packard ;
- Layar Creator par Layar ;
- Augment ;
- Blippbuilder par Blippar, en développement ;
- Zooburst, création de livres animés ;
- Zapworks ;

Ces solutions ne répondent pas à notre problématique. D’une part, elles ne semblent pas disponibles sur des lunettes de réalité augmentée. De plus ces applications ne peuvent pas être modifiées pour être adaptées à nos besoins. Ces applications sont capables d’incruster des plans, et quelques objets 3d à partir de marqueurs. Il n’y a pas de reconnaissance d’objets 3d complexe.

2 Les SDKs pour la réalité augmentée

2.1 Vuforia

Vuforia est un standard de la réalité augmentée. Il est supporté par plus de 250 000 développeurs. Cela en fait donc le leader dans ce domaine. Il fournit un kit de développement de réalité augmentée et un outils en ligne (CMS) Vuforia Studio. Le SDK permet entre autre la reconnaissance et le tracking d'objet planaire et d'objet 3D. Vuforia a été développé par Qualcomm avant d'être racheté par PTC en novembre 2015. PTC est un leader mondial en matière de technologie. Ils cherchent à accompagner les industriels en leur fournissant des solutions technologiques de pointe. C'est en avril 2016 qu'il annonce vouloir prendre en charge le casque Hololens de Microsoft sur leur dernière acquisition, la plateforme de réalité augmentée Vuforia.[5]

A première vue, Vuforia semble être une solution idéale à notre problématique. Elle est disponible sur de nombreux support, comme les smartphones, tablettes et ordinateurs, les lunettes de réalité augmentée et les casques de réalité virtuelle. Le nombre d'applications utilisant Vuforia est énorme. On peut en voir un ensemble sur leur site.[13] Au niveau des systèmes d'exploitation, le SDK de Vuforia est compatible avec Android, iOS et UWP (Universal Windows Platform). Les applications peuvent être développées avec Android Studio, XCode, Visual Studio et Unity.

L'utilisation du SDK Vuforia est gratuite. Elle devient payante dès lors que l'on utilise les services de reconnaissance à partir du Cloud. Jusque-là, cette plateforme semble répondre à nos besoins. Nous allons nous intéresser plus particulièrement aux fonctionnalités offertes ainsi qu'à la liberté que l'on peut avoir à en développer d'autres plus adaptées à nos besoins. La reconnaissance d'objets 3d est un peu particulière. Les nombreuses documentations annoncent le matching d'objet 3d pas trop complexe (bon pour notre glène) seulement il faudra scanner au préalable l'objet avec les hololens, sinon ça n'est pas possible. A ce jour, on ne peut pas importer d'objets 3d d'un autre support.



Figure 1 – Logo de Vuforia

2.2 Dioata

L'avantage numéro un de Diota est qu'elle permet de faire de la réalité augmentée "sans marqueurs", comme expliqué plus haut. Il est capable d'identifier directement un objet 3d en temps réel. Les inconvénients de cette solution est qu'elle n'est compatible qu'avec Windows 10. Elle n'est développable qu'avec le logiciel DiotaStudio, compatible avec Unity. Le tout s'embarque seulement avec un tablette Microsoft Surface Pro 4. L'ensemble du kit ne vaut pas moins de 7000€.



Figure 2 – Logo de Diota

2.3 ARToolkit

ARToolkit[1] est une bibliothèque similaire à Vuforia. Elle est portable sur iOS, Android, Linux et Windows. Il est possible d'ajouter et modifier des fonctionnalités directement sur le code source de la bibliothèque. On peut ainsi l'utiliser pour une application bien spécifique.

On trouve sur leur site une documentation plutôt complète, avec de nombreux tutoriels. Une communauté existe avec un forum.



Figure 3 – Logo de ARToolKit

2.4 Wikitude

Le SDK Wikitude est disponible gratuitement en version d'essai, mais devient vite payant lorsque l'on veut ajouter des fonctionnalités. Il est disponible sous Android et iOS. Disponible sur plusieurs paires de lunettes 3D, il a l'inconvénient de n'avoir que la reconnaissance d'image, donc d'objet 2D. Une version bêta est en cours de développement afin de pouvoir travailler avec des objets 3D.



Figure 4 – Logo de Wikitude

2.5 ARMedia SDK

Ce SDK n'est pas encore disponible pour windows, ce qui est malheureux car ce SDK embarque la détection d'objet 3d complexe.



Figure 5 – Logo de ARMedia

3 Problématiques

Pourquoi utiliser ces librairies avec Unity? Unity semble incontournable lorsque l'on veut faire de la réalité augmentée. Les points suivants l'expliquent :

La **compatibilité**, Unity est compatible avec quasiment tous les environnements de développement (iOS, Android, Windows, Linux).

Il est **gratuit**, du moment que l'entreprise qui l'utilise ne gagne pas plus de 100 000 dollars par an.

Unity est incontournable, mais surtout indispensable. Unity est un moteur 3d. Un moteur 3d[19] est un logiciel permettant de transformer des données dimensionnelles en image. Les processus utilisés sont lourds, et à la différence des API's de bas niveau, les moteurs 3d embarquent des fonctions complexes permettant de charger des formats de fichiers différents, d'animer une scène, de gérer la caméra, etc... Et cela de manière plus transparente pour le développeur. Les lunettes de réalité augmentée, n'embarquent pas de tels systèmes, c'est pourquoi il faut en inclure un si on veut obtenir de bonnes performances au niveau de la 3d.

Une autre solution consisterait à développer soit même les fonctions nécessaires à la réalité augmentée en utilisant des librairies tels que OpenCV. OpenCV est une bibliothèque open source, disponible sur de nombreuses plateformes. Elle est cependant payante si l'on veut l'utiliser avec Unity (95\$).

5

Les lunettes 3D

La connaissance des technologies de lunettes de réalité augmentée est indispensable à la réussite du projet. Il est important de connaître au mieux le marché actuel afin de se positionner sur une technologie qui réponde au mieux à nos besoins. Pour cela nous avons fait un état de l'art des lunettes disponibles ou en sortie proche. Il faut faire attention aux publicités, car étant un secteur en pleine expansion, certaines entreprises veulent aller parfois trop vite. On peut se retrouver avec des dates de sorties plusieurs fois repoussées, un matériel non terminé ou disponible sans kit de développement. L'exemple des "Google Glass" est parlant, en effet, Google a choisi de faire un pas en arrière sur leur commercialisation. Ils les ont retirés du marché car "la technologie n'était pas assez mûre pour ce produit". L'affichage 3d est un des critères les plus importants. Nous nous intéressons aussi aux plateformes de développement associées aux lunettes (SDK) et aux logiciels proposés.

Nous avons détaillé les lunettes les plus intéressantes pour la réalisation du projet. Il est indispensable de pouvoir visualiser du contenu 3d et d'avoir un environnement de développement disponible. **Vous trouverez dans la section suivante un tableau récapitulatif des produits que nous avons analysés.** Savoir si les lunettes sont compatibles avec du contenu 3d dépend principalement de l'environnement de développement (i.e. le SDK). Lors de cette recherche bibliographique, nous nous sommes donc focalisés sur les SDK disponibles qui permettent l'utilisation de la 3d.

1 Tableaux comparatifs




Marque	MICROSOFT	EPSON	EPSON
Modèle	Hololens	MOVERIO BT-2000	MOVERIO BT-300
Photo			
Mise à jour	29/09/2016	29/09/2016	29/09/2016
Prix	3000\$	3.120,00 €TTC	799 €
Disponibilité	Uniquement pour les développeur	Pas encore disponible	Octobre 2016??
Type d'affichage	2d/3d	2d/3d (Side by side)	2d/3d (Side by side)
Plateforme - OS	Windows 10	Android	Android
SDKs/Logiciels	Hololens Emulator, UWP, Vuforia, Unity3D,	SDK Android, SDK by Espon, Vuforia, Wikitude	SDK Android, SDK by Espon, Vuforia, Wikitude
Applications possibles	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...	Industriel, Educatif, Médical, etc...
Communauté	http://forums.hololens.com/	https://moverio.epson.com/?_ga=1.20102072.1219241715.1474546220	https://moverio.epson.com/?_ga=1.20102072.1219241715.1474546220
Résolution écran (en pixels)	720p	960x540	1280 x 720
Fréquence		60Hz	30Hz
Camera(s)	5 caméra, dont une de profondeur et de 2MP, 1 capteur de lumière	2 caméras, 1 5M, 1 0.3M pour la profondeur	5M
Champ de vision	40°	23°	23°
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
RAM	64GB Flash, 2Go RAM	1Go	2Go
Mémoire interne	64Go	8Go	16Go
Poids	579g	290g (lunettes) - 265g (télécommande)	69g
Accéléromètre	Oui	Oui	Oui
Gyroscope	Oui	Oui	Oui
Magnétomètre	Oui	Oui	Oui
Connectivité	Wifi, Bluetooth	Wifi, Bluetooth, Micro USB	Wifi, Bluetooth
Liens des caractéristiques techniques	https://developer.microsoft.com/fr-fr/windows/holographic/hardware_details	https://www.epson.fr/products/see-through-mobile-viewer/moverio-pro-bt-2000/Caracteristiques-techniques	http://www.epson.de/en/products/see-through-mobile-viewer/moverio-bt-300

Figure 1 – Comparatif lunettes 3D : 1ère partie




Marque	EPSON	ODG	METAVISION
Modèle	MOVERIO BT-200	R-7	Meta 2
Photo			
Mise à jour	29/09/2016	29/09/2016	29/09/2016
Prix	699,00 TTC	\$ 2,750.00	\$949
Disponibilité	Disponible	janv-17	Q3 2016, pre-order
Type d'affichage		2d/3d	2d/3d
Plateforme - OS	Android	ReticleOS (Andoird)	Application Windows
SDKs/Logiciels	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...
Communauté	Nombreux forums, mais non dédiés	-	https://www.metavision.com/faq
Résolution écran (en pixels)	960x540	720p	2560 x 1440
Fréquence	60Hz		60Hz
Camera(s)	VGA	1080p 60fps et 720p 120fps	720p + capteur infrarouge 240p
Champ de vision	23°	20°	90°
Processeur	Processeur à double noyau de 1,2 GHz TI OMAP 4460	Qualcomm SnapdragonTM 805 2.7GHz quad-core Processor	Intel i5 1,5GHz
RAM	1Go	3Go	4Go
Mémoire interne	8Go	64Go	128Go SSD
Poids	88g (lunettes) - 124g (télécommande)	136g	
Accéléromètre	Oui	Oui	
Gyroscope	Oui	Oui	
Magnétomètre	Oui	Oui	
Connectivité	Wifi, Bluetooth	Wifi, Bluetooth, Micro USB	Wifi, Bluetooth, USB
Liens des caractéristiques techniques	https://www.epson.fr/products/see-through-mobile-viewer/moverio-bt-200	http://www.osterhoutgroup.com/products-r7-glasses	

Figure 2 – Comparatif lunettes 3D : 2ème partie

2 Les lunettes 3D Hololens par Microsoft

Nous allons, cette année, travailler avec cette paire de lunettes. Elles ont été commandées, car nous n'avons pas de latitude sur notre délais de réflexion, pour des raisons administratives.



Figure 3 – Les lunettes 3D Hololens

2.1 Conditions d'achat

A ce jour, ces lunettes ne sont disponibles que pour les développeurs. De plus, elles sont seulement en vente dans les magasins US et Canadien. Il faut donc passer par un intermédiaire pour ce les procurer.

2.2 Caractéristiques techniques

Se référer à la section 1, page 27.

2.3 Système d'exploitation et plateforme de développement

Le système d'exploitation des Hololens est Windows 10. Sans parler de développement, on retrouve un environnement connue. On a accès à un navigateur internet, le Windows Store pour ajouter des applications et tout ce qui est possible sous Windows 10 (ce qui est possible avec les configurations matériels)..

Plusieurs outils sont nécessaires au développement d'application.

Windows a développé un outils commun à plusieurs plateforme pour développer des applications Windows 10, le SDK Windows 10. Il permet de créer des applications Windows universel, UWP (Universal Application Platform), que l'on peut embarquer sur tous les matériels utilisant Windows 10. On peut quand même spécifier certains facteurs pour qu'elles soient plus spécifiques au fonctionnement d'un type d'appareil.



Figure 4 – Plateforme de développement de Windows

L'environnement de développement utilisé est Visual Studio 2015 upgrade 3. Le langage de programmation conseillé est le C#. Si l'on n'a pas encore les lunettes Hololens on peut utiliser un simulateur.

Le simulateur permet de développer et tester ses applications. L'émulateur nécessite d'avoir plusieurs outils. Il faut posséder le système d'exploitation Windows 10 mais dans une version supérieur à la version familiale ou Home, comme la version Education ou Pro. Ce prérequis est dû au composant Hyper-V qui ne peut être installé que sur ces versions. L'inconvénient de l'émulateur est qu'il ne permet pas d'utiliser directement sa webcam. En effet il considère que les résultats ne peuvent être probant compte tenu des technologies mise en oeuvre sur les Hololens. On peut démarrer les applications, qui s'exécutent sur fonds noirs. On peut cependant importer des images ou vidéos pour simuler le réel.

Le développement d'applications pour Hololens nécessite d'avoir Visual Studio 2015 Upgrade 3. Cela est dû au composant UWP (Universal Windows Platform). Le SDK Windows 10 est également nécessaire.

Ces outils sont le minimum requis.

Il faut ensuite, en fonction, de ses besoins ajouter d'autres outils, comme des bibliothèques de réalité augmentée ou encore Unity pour le rendu 3d.

Le puissant moteur 3d Unity est compatible avec les Hololens (comme la plupart des lunettes d'AR de cette gamme). Comme expliqué précédemment, Unity permet l'intégration d'objet 3d.

Nous avons vu dans la partie précédente que de nombreuses bibliothèques de réalité augmentée étaient compatibles avec les Hololens. Il y a Vuforia, Wikitude, etc. Nous verrons un peu plus tard quelle bibliothèque convient le mieux à notre problématique parmi celles disponibles.

Troisième partie

Analyse et conception

6

Analyse

Notre état de l'art nous permet d'avoir une première idée des outils que nous allons utiliser, aussi bien matériels que logiciels. C'est une première idée car au niveau logiciel, une phase importante d'expérimentation doit nous permettre de nous assurer de la faisabilité du projet.

Le matériel utilisé va être les Hololens de Microsoft. C'est un produit puissant doté d'une technologie impressionnante. Nous les avons commandées avant la fin de l'état de l'art, car nous n'avons pas de latitude sur notre délai de réflexion, pour des raisons administratives.

Niveau logiciel, il nous faut approfondir nos recherches. Des traitements de détection d'objets simples sont plutôt faciles à mettre en place. Cependant pour répondre à notre problématique, nous devons nous assurer que les SDK étudiés permettent de reconnaître des objets 3d, que l'on peut importer comme cible.

1 Conception des objectifs 1 et 3

La conception et la réalisation du premier objectif est connu. Pour rappel il s'agit de déposer un objet 3d virtuel (une omoplate) sur un cube sur lequel se trouve un ou plusieurs marqueurs de type image ou QR-Code. Nous allons utiliser le SDK Vuforia. Il est compatible avec notre paire de lunettes 3D.

Nous allons coupler notre SDK avec le moteur 3D Unity. En effet il s'avère être un outil de transition indispensable. Il permet d'insérer notre objet 3D, à partir d'une importation .obj. Cela répond à notre demande, car les différents objets (omoplate, masque et glène reconstitué) sont exportés via le logiciel Blender. C'est via ce même logiciel que nous découpons les différentes parties.

Une fois notre objet importé dans Unity, il reste une étape de configuration. Il faut configurer le projet Unity afin qu'il puisse être exporté vers Visual Studio. Pour cela nous devons régler certains paramètres comme la caméra, le type de scène ou encore le type de fond. Nous pouvons également choisir ou insérer notre objet, en terme de distance par rapport à la position des lunettes. Cela correspond à un premier programme de découverte, car par la suite on souhaite positionner notre objet en fonction d'un objet réel.

Nous allons revenir vers Unity, mais d'abord il faut configurer notre cible. C'est ici que l'on rentre dans la configuration de Vuforia. Il faut passer par le TargetManager de Vuforia. C'est une plateforme qui permet de créer ses cibles. On a le choix entre des cibles 2D ou 3D. Nous sommes cependant limités au niveau des objets 3D. En effet on ne peut pas importer notre cible au format .obj, et c'est une des raisons pour lesquelles on ne pourra pas utiliser ce SDK pour la réalisation de l'objectif 2. Nous allons choisir un cube, qui est une forme géométrique disponible avec le cone ou le cylindre. Nous pouvons placer une image sur notre cube qui fait office de marqueur. Notre cible est maintenant prête et nous allons l'exporter vers Unity. Ensuite nous allons positionner notre omoplate sur le cube fraîchement importé et pouvoir lancer l'application.

Add New Target

Target Name ⓘ

RockyStones

Target Type

Single Image

Cube

Cuboid

Cylinder

Target Dimension ⓘ

Width: 540

Target Image File ⓘ

Choose File rocky-stones.jpg

Cancel

Add

Figure 1 – TargetManager sous Vuforia

2 Conception des objectifs 2 et 3

Ici, on ne peut plus utiliser Vuforia, car nous sommes restreints par l'utilisation de marqueur. Nous allons cette fois nous tourner vers la bibliothèque OpenCV, que l'on peut associer à Unity. Une fonction nous intéresse plus particulièrement : SurfaceMatching. Nous allons devoir programmer davantage via Visual Studio en utilisant notamment les fonctions de récupération du flux vidéo du SDK des Hololens. La phase d'expérimentation va être essentielle car c'est à ce moment que l'on saura si cette solution est bien la bonne. La figure ci-dessus nous montre comment un objet a été matché via cette fonction. En détectant la surface de l'objet courant, on peut confondre cet objet avec un masque importée en .obj. Si cette solution s'avère faisable, alors on pourra très bien l'utiliser pour réaliser l'objectif 4, qui est l'objectif final.

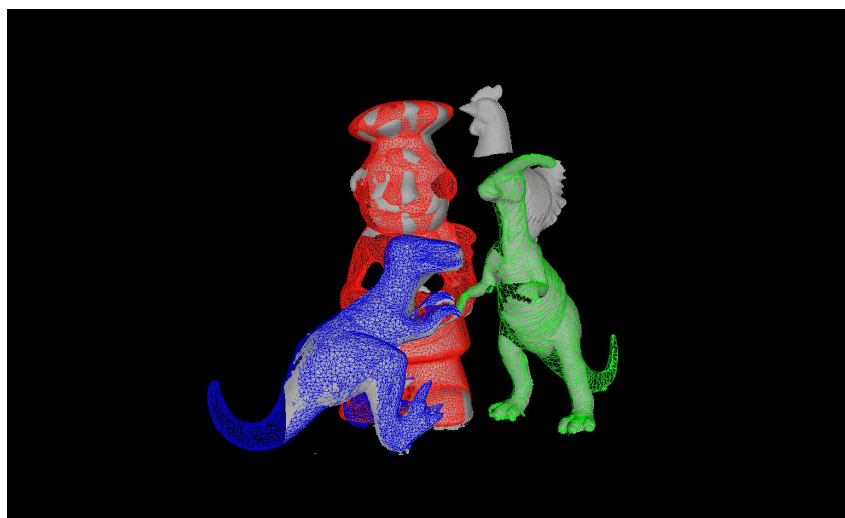


Figure 2 – Exemple avec OpenCV

Les objectifs 2 et 4, comme prévu, demandaient une phase d'apprentissage mais surtout de faisabilité importante. La connaissance de cette nouvelle technologie qu'est les Hololens est parfois imprévisible. L'élément qui nous a empêchés d'aller au bout de ces objectifs est l'absence des informations de profondeur. Vendant 5 caméras dont une de profondeur sur les lunettes, cette information est récupérable de plusieurs façons. D'abord la caméra de profondeur est censée apporter l'information de la distance opposant la caméra et les éléments de l'image, c'est son rôle. La deuxième consiste en l'utilisation d'au moins deux caméras en stéréo. Un paramétrage permet d'utiliser deux caméras pour récupérer les données de profondeur.

Ces notions n'ont pas été abordées lors de la partie Recherche du projet. L'accès a plutôt été dirigée vers la découverte des outils et matériels de réalité augmentée, et la recherche d'outils de prototypage rapide. Le problème avec le kit de développement des Hololens, ne permet que l'utilisation d'une caméra RGB. Alors il n'est pas possible de récupérer les informations de profondeur. En réalité cela est possible, mais nécessiterait un temps de formation très long. Il y a une phase d'apprentissage, puis de paramétrage. Nous avons donc pris la décision de ne pas aller vers ces sentiers. Nous avons remarqué que cette problématique était traitée par plusieurs équipes de recherches. En lisant les différents travaux, nous savons que nous n'aurons pas le temps de les réaliser (ils pourraient faire l'objet d'un PRD à eux seuls). On retrouve le projet CurvSurf[4] de Google et les articles[6] de l'université de Laval.

Ce dernier lien montre bien la capacité des Hololens à transformer l'environnement réel en un mappage 3d, avec une capacité très limitée. Leur méthode consiste à coupler les Hololens avec une caméra RealSense. Cette dernière caméra permet de récupérer les données de profondeur et de les envoyer vers les lunettes. En utilisant ces données et les données géographiques des lunettes on possède un système de mappage de l'environnement puissant. Ils ont géré le graph matching afin de confondre la figurine d'un dragon et son modèle 3d virtuellement.

La stratégie commerciale de la mise en vente des Hololens nous a fortement induits en erreur. Et nous ne sommes pas les seuls, à voir les différents retours sur les blogs ou forums.

Une des fonctionnalités du SDK Hololens, Holotoolkit, est la fonction Spatial Understanding. Cependant, elle n'est pas précise. Elle est suffisante pour la détection de surface plane, comme l'illustre la figure suivante :

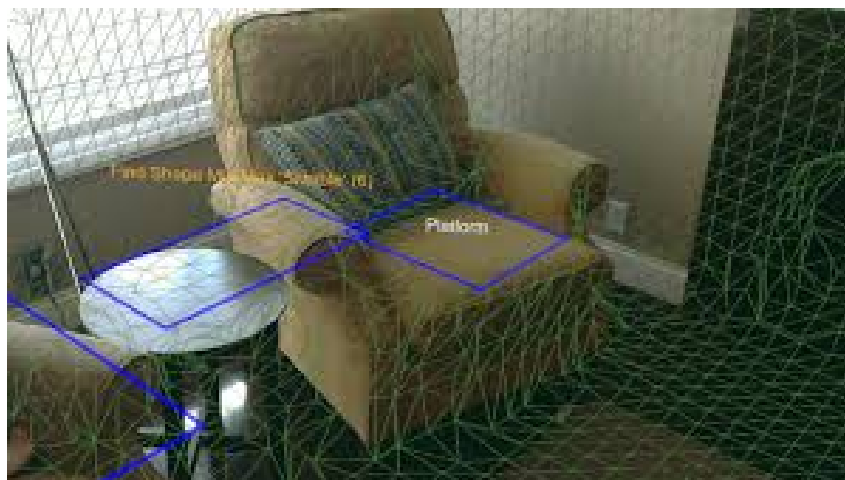


Figure 3 – *Spatial understanding*

L'illustration de l'équipe de l'université de Laval[6] est pertinente. On remarque quel est le degré de qualité du mappage de l'environnement des Hololens comparé à celui d'une caméra RealSense. On remarque qu'à ce stade, aucune analyse n'est possible avec ce type de résultat.

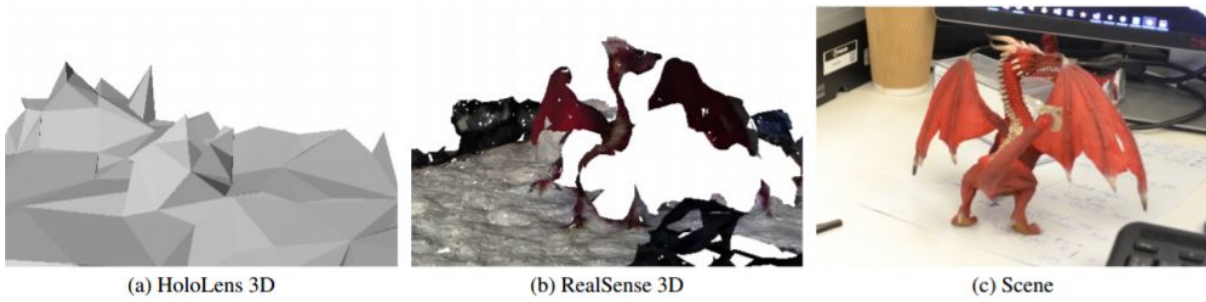


Figure 4 – *Spatial understanding 2*

De mon point de vue, le projet devrait être segmenté en plusieurs parties. D'abord, un projet sur l'étude et la mise en œuvre d'outils de prototypage rapide de réalité augmentée. D'un autre côté, un projet plus technique, sur les techniques de récupération des données de profondeur à partir des HoloLens. Il serait intéressant d'obtenir un mappage 3D et précis de l'environnement en temps réel. Ensuite un autre projet de "Graph Matching" permettrait d'utiliser ce mappage pour le confondre avec un modèle 3D préexistant.

Quatrième partie

Mise en œuvre

En connaissance des éléments de la partie précédente, nous avons revu nos objectifs. Concernant les objectifs 2 et 4, notre travail n'est pas plus dans la réalisation, mais dans l'étude de la manière de les réaliser pour un projet futur. Pour cela, nous avons poursuivi notre analyse afin qu'il existe des premiers éléments de réalisation pour la suite. Nous avons donc réduit les temps de travail sur ces deux objectifs et augmenté ceux des objectifs 1 et 3, afin d'avoir une forte compréhension des outils et une analyse des meilleures conditions de réalisation.

La mise en œuvre des objectifs 1 et 3, c'est faite comme prévu à l'aide de Vuforia.

Vous trouverez en annexe le guide de création d'un projet de réalité augmentée avec Vuforia, pour les HoloLens. Vous y trouverez les différentes étapes, ainsi que des explications sur les paramètres de configuration. Il y a également les résultats de tous les tests effectués. Ainsi, vous trouverez les modalités des paramètres qui induisent sur la qualité de la réalité augmentée.

Nous avons développé plusieurs applications, dont la dernière avec la maquette de l'épaule est la mieux réalisée. Nous avons mis en place deux marqueurs. L'un d'eux est face à la maquette et l'autre sur le côté droit de la maquette. Ainsi, on peut se déplacer autour de deux des quatre "faces" de la maquette.

La figure suivante illustre la maquette et les deux marqueurs :



Figure 5 – Maquette côté droit



Figure 6 – Maquette de face

3 Bilan sur la mise en œuvre qualité

L'intérêt d'un projet est de capitaliser son travail. Tout ce qui a été fait doit être noté, documenté et expliqué, car lors de la reprise du projet, les successeurs feront face aux mêmes difficultés. Il est important de leur donner un maximum d'outils afin qu'ils entrent le plus aisément possible dans le projet. Pour cela, la rédaction de plusieurs documents est nécessaire :

- Tutoriel de l'outil : Vuforia (création de marqueurs, configuration, paramétrage)
- Guide de déploiement
- Guide d'utilisation
- Un manuel de test et les résultats

Il est également important de mettre en œuvre le versionning des différentes applications créées.



Conclusion

Il est temps de conclure sur la 1ère partie du projet. Les états de l'art ont permis de clairement identifier les outils nécessaires à la bonne réalisation du projet. Étant un projet plus orienté recherche que développement, il y a des zones d'ombres qui vont nécessiter une phase d'expérimentation. Les procédures de spécifications et analyse du problème ont permis de se projeter vers le rendu de l'application. C'est une première approche qui permet d'anticiper certaines demandes du client. La partie de gestion de projet a également été importante, en effet il est important de savoir gérer son temps afin de remplir ses objectifs. Il est indispensable de garder à l'esprit que nous sommes dans un projet pédagogique et par conséquent il y a plusieurs facteurs à prendre en compte, et pas seulement la problématique du projet. La démarche génie logiciel permet d'éviter des surprises lors du développement. Le planning prévisionnel a été respecté. Concernant celui du S10, il est fort probable qu'il soit changé car les temps donnés pour les phases d'expérimentation sont indicatifs.

La conclusion de la seconde partie du projet est positive. Ce projet m'a permis d'obtenir un stage dans la santé associée à la réalité augmentée dans un grand groupe international. Nous avons présenté le projet de nombreuses fois. Les plus importantes ont été la journée portes ouvertes de l'école, où nous avons rencontré tout au long de la journée des personnes de tout horizon. C'était une formidable occasion de présenter les outils qui feront peut-être notre quotidien dans les prochaines années. Nous avons également présenté le projet à la Ministre de la Santé et des Affaires Sociales, Marisol Touraine. C'était dans le cadre de la journée portes ouvertes de l'Amic 37 avec lequel Polytech est en partenariat. Elle a découvert et testé les lunettes à travers notre application sur la maquette avec beaucoup d'enthousiasme.



Figure 7 – *Présentation du projet à Marisol Touraine*

Annexes

A

Annexe A

1 Interface Homme-Machine

La fenêtre du chirurgien au niveau de l'application PC permet d'importer les différents objets 3d. Voici un exemple de rendu :

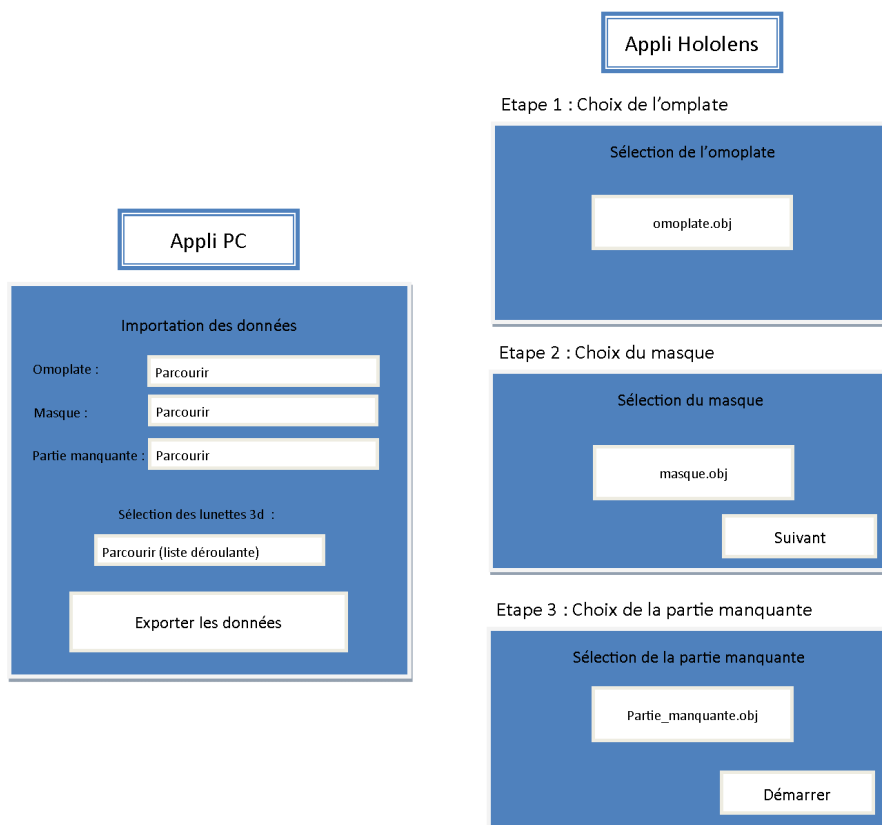


Figure 1 – IHM des applications PC et Lunettes

Nous avons défini trois étapes afin de sélectionner les différentes parties de l'omplate en objet 3d. On pourra par exemple imaginer que nous n'avons qu'un seul fichier 3d, où l'application est capable de discerner les trois parties avec un nommage stricte fait au préalable sous Blender. En effet, sous Blender, on peut séparer un objet 3d en différents sous-objets sur une même scène, le tout contenu dans un seul et

même fichier .obj. Nous avons indiqué qu'il y avait plusieurs fichiers d'omoplates. En effet, le chirurgien pourrait importer plusieurs fichiers lors d'une même séance, et faire, ainsi, plusieurs opérations sans mettre à jour ces lunettes 3d. La capacité de stockage des Hololens est de 64Go. Le nombre de fichiers serait à déterminer, mais il est certain que l'on pourrait en stocker plusieurs. Pour diverses raisons, de sécurité notamment, cette possibilité ne sera pas retenue dans notre travail. Un seul couple de fichiers sera présent dans la mémoire des lunettes.

2 Spécifications fonctionnelles

Dans cette partie nous allons exprimer les besoins fonctionnels. C'est une pré-analyse à l'évaluation de la complexité et à la planification du projet. On retrouve ci-dessous les définitions des fonctions principales ainsi qu'un diagramme détaillé des cas d'utilisation.

Importation des fichiers de données .obj (omoplate, masque, partie manquante)

ImportDataObj;

Rôle : secondaire

Spécification :

Entrée : Fichier omoplate .obj

Sortie : nulle

Pré-conditions :

Fichier au format .obj

Peut-être spécifié une taille maximale ?

Post-conditions :

Cette fonction interagit avec le dossier présent sur l'ordinateur

Export des fichiers de données vers les lunettes 3d

ExportDataToGlasses

Rôle : secondaire

Spécification :

Entrée : Fichier omoplate .obj

Sortie : nulle

Pré-conditions :

Fichier au format .obj

Peut-être spécifié une taille maximale ?

Post-conditions :

La connexion est-elle active ?

Sélection des données .obj (omoplate, masque, partie manquante) (Appli lunettes)

SelectData

Rôle : secondaire

Spécification :

Entrée : Fichier omoplate .obj

Sortie : Les données ont bien été sélectionnées

Pré-conditions :

Y-a-t-il des données disponibles ?

Post-conditions :

Nulle

Fabrication du masque

MaskRecognition

Rôle : primordiale

Spécification :

Entrée : Fichier du masque

Sortie : L'élément est reconnu

Pré-conditions :

Nulle

Post-conditions :

Nulle

Surimpression de l'omoplate/partie manquante, la cible, avec le masque.

DisplayScapula

Rôle : primordiale

Spécification :

Entrée : Fichier du masque

Sortie : Omoplate/partie manquante en surimpression

Pré-conditions :

Le masque a bien été « matché » avec la cible

Ses données de géolocalisation permettent de toujours connaître sa position

Post-conditions :

Nulle

Selon la méthode de poursuite retenue :

- reconnaissance du nouvel emplacement et de la nouvelle orientation de la cible
- repositionnement de la scène
- rafraichissement de l'affichage

3 Spécifications non fonctionnelles

3.1 Contraintes de développement et conception

Dans cette partie nous devons préciser les contraintes liées aux matériels, aux langages de programmation, aux logiciels et bibliothèques, à l'environnement nécessaires et aux protocoles de communication.

3.1.1 Matériels

Au cours du développement logiciel nous aurons besoin d'un poste de programmation, sous Windows. Nous aurons également des lunettes 3d de réalité augmentée, il s'agit des Hololens de Microsoft. Si nous utilisons l'émulateur Hololens nous devons respecter certaines conditions matérielles. L'émulateur nécessite d'avoir plusieurs outils. Il faut posséder le système d'exploitation Windows 10 mais dans une version supérieure à la version familiale ou Home, comme la version Education ou Pro. Ce prérequis est dû au composant Hyper-V qui ne peut être installé que sur ces versions.

3.1.2 Langages de programmation

Aucun langage de programmation n'est imposé par le client, cependant le matériel utilisé nous impose d'utiliser le langage Windows C#.

3.1.3 Logiciels et bibliothèques

Le développement d'applications pour Hololens nécessite d'avoir Visual Studio 2015 Upgrade 3. Cela est dû au composant UWP (Universal Windows Platform). Le SDK Windows 10 est également nécessaire. Nous n'avons pas encore décidé quel SDK de réalité augmentée nous allons utiliser.

3.2 Contraintes de fonctionnement et d'exploitation

3.2.1 Performances

Les performances des lunettes 3d Hololens sont très bonnes.

3.2.2 Capacités

L'un des enjeux importants du projet est de démontrer sa faisabilité.

3.2.3 Contrôlabilité

Le logiciel est capable de créer des fichiers d'erreurs (fichier de log). Ceux-ci sont à destination d'un utilisateur confirmé.

3.2.4 Sécurité

Il faudra mettre en place une procédure de validation préopératoire qui empêche le chirurgien de se tromper de fichiers. Cette préoccupation est hors du champ de cette étude de faisabilité.

3.2.5 Maintenance et évolution du système

Il doit pouvoir y avoir la possibilité de supprimer, modifier ou de rajouter des fonctionnalités, par des développeurs qui ne soient pas forcément les développeurs d'origine. Pour cela nous devons garantir un développement de type génie logiciel.

B

Gestion de projet

La méthode du cycle V semble la plus adaptée à notre problématique. Ici, le cycle en V est clairement identifié et il n'y pas de raison que le client intervienne en cours d'année pour changer les objectifs. Notre travail ne convient pas par exemple à une méthode agile où des lots sont à fournir à chaque sprint.

Chaque objectif nécessite une phase ascendante commune. Il y a d'abord la phase d'analyse des besoins et de faisabilité. Selon les objectifs l'analyse diffère peu et l'étude de faisabilité est un enjeu final du projet, elle est à démontrer pour l'objectif 4. S'en suivent les phases de spécifications et de conception, où une grande partie de l'application est la même quelques soient les objectifs.

Notre cycle en V est semblable à un cycle standard, à la différence qu'il y a quatre phases en bout de V correspondant à chaque objectif. Ces quatre phases se suivent, elles ne se chevauchent pas. Chaque phase comprend les phases de conception détaillées, de codage et de tests unitaires.

Au niveau du planning du projet, nous nous sommes bien tenus aux prévisions de la première partie du PRD. Chaque tâche a été identifiée préalablement, et correspond principalement à du travail de recherche sur les différents états de l'art. Nous y avons également ajouté des tâches pour la mise en œuvre du projet, comme la rédaction des différents documents, ou encore la préparation de la soutenance. Voici le diagramme de Gantt du PRD1 :

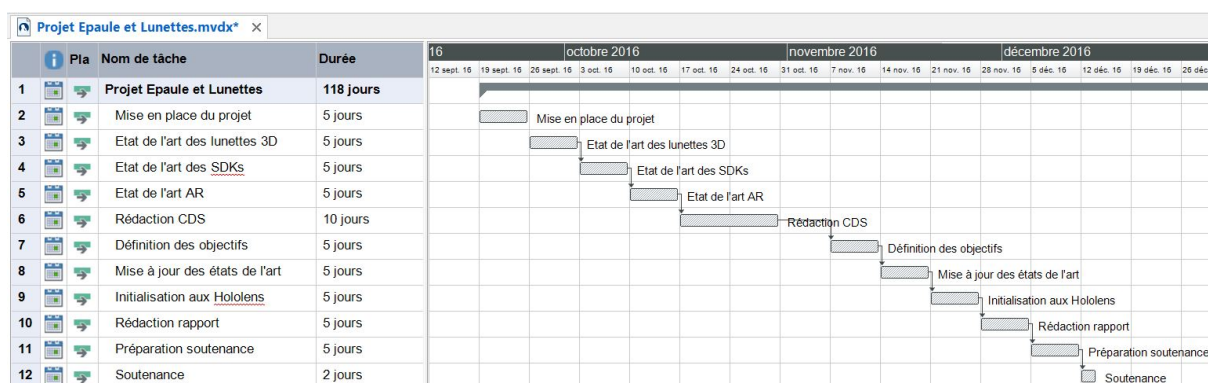


Figure 1 – Diagramme de Gantt PRD1

Lorsque nous allons entamer la seconde partie du PRD, nous avons identifié des tâches qui correspondent à des temps de formation technique. En effet, nous allons utiliser certaines technologies, comme les différents SDKs, qui nécessitent certainement un temps d'adaptation. Ces temps de formation sont

incluent dans les objectifs 1, 2, 3 et 4. Ces tâches comprennent les temps de conception et de codage.

Il y a aussi des tâches d'expérimentations techniques. Le protocole expérimental est clairement identifié pour les objectifs 1 et 3, contrairement aux objectifs 2 et 4. Il faut donc s'attendre à tester différentes technologies, pour s'assurer que l'une d'elles répondent à notre besoin.

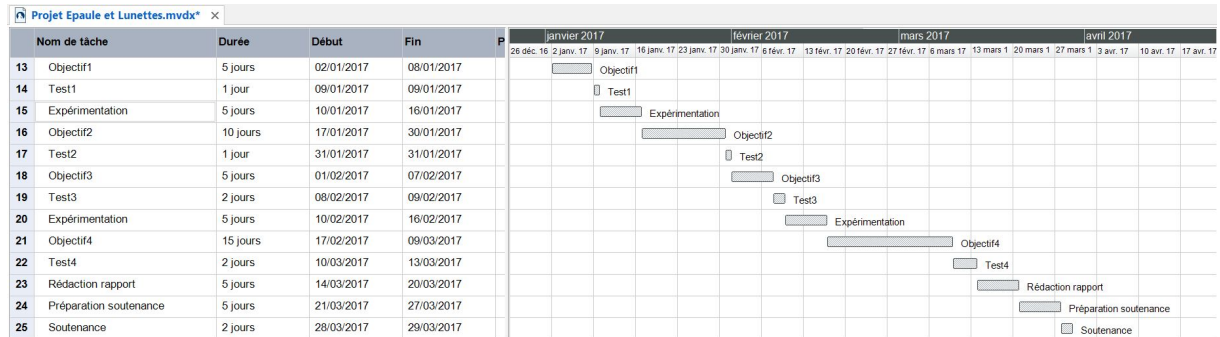


Figure 2 – Diagramme de Gantt PRD2

A noter également que ce projet est très largement susceptible d'être repris par d'autres étudiants, il faut donc tout mettre en œuvre pour que le projet puisse être réutilisé et facilement amélioré pour un futur projet.

Lors de la partie 1 du projet, nous avons mis les risques des objectifs 2 et 4 en avant, mais pas à une si grande mesure. Lorsque nous avons été sûr que nous ne pouvions pas avancer davantage dans cette direction, nous avons décidé avec le groupe de travail, de réaliser un nombre de tests plus important sur les objectifs réalisables, afin d'en conclure sur une création et configuration d'un projet de réalité augmentée optimale.

Le diagramme de Gantt suivant présente le déroulement de la seconde partie du projet. Comme annoncé, celui-ci a fortement été modifié en raison de la finalité de l'analyse des objectifs 2 et 4.

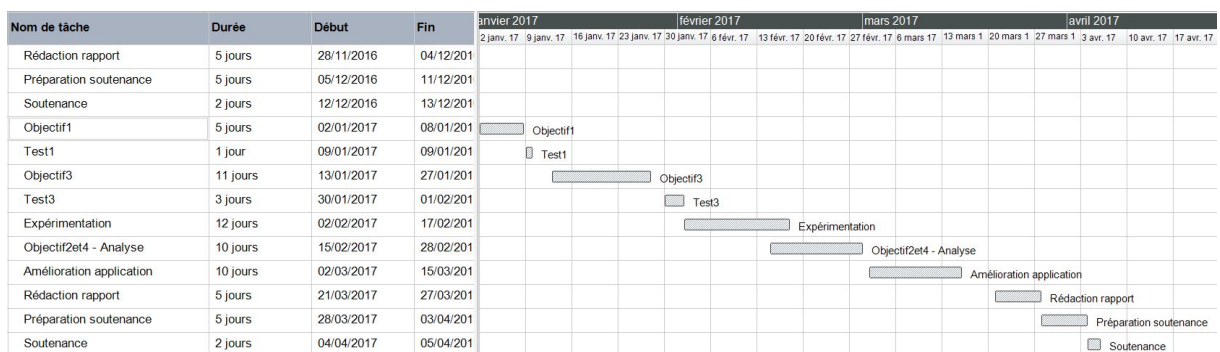


Figure 3 – Diagramme de Gantt PRD2 effectué

L'avancée du projet a été mesurée chaque semaine. Des rencontres bilans avec Mr. Proust étaient souvent organisées le jeudi en fin d'après-midi, afin que je lui présente les travaux effectués depuis le mercredi. Une démonstration permettait de valider ou non les étapes. Nous avons eu plusieurs jalons, où des présentations étaient prévues. Alors, nous avons tout intérêt de fournir une démonstration propre et fonctionnelle pour les utilisateurs.

C

Comparatif SDK





Nom	Vuforia	VuforiaStudio	DiotaPlayer	ARToolkit
Fournisseur	PTC	PTC	Diota	ARToolkit
Logo				
Type	SDK	CMS	CMS	SDK
Plateforme	Android, iOS, UWP	Web (Windows et Mac), iPhone et iPad	Windows 10	iOS, Android, Linux, Windows
Licence	Gratuit (options payantes)	??	~7000€	OpenSource LGPL 3.0
Compatibilité matériel	Mobile, Lunettes AR (Hololens)	Webcam, iPhone et iPad	Surface Pro 4	Mobile, Lunettes AR
Moteur 3d	Unity		Unity	Unity
Matching	Marqueurs 2d, Objets 3D après scan	"ThingMark" : image sur un objet physique	Objets 3d	Images et Marqueurs 2d, Objets 3D
Marker based	Oui	Oui	Oui	Oui
Markerless	Non	Non	Non	Oui
3d tracking	Oui	Non	Non	Oui
Tracking	Oui	Oui	Oui	NFT
Evolutif	Oui	Non	Non	Oui, très largement (critère de communication)
Sites web	https://www.vuforia.com/	https://www.vuforia.com/studio	http://www.diota.com/#diotaplayer	https://artoolkit.org/
Communauté	https://developer.vuforia.com/forum	https://developer.vuforia.com/forum		Forte

Figure 1 – Tableau comparatif des SDKs - 1ère partie



Nom	Wikitude SDK	Wikitude Studio Creator	Armedia
Fournisseur	Wikitude	Wikitude	Armedia
Logo			
Type	SDK	CMS	SDK
Plateforme	Android et iOS	On web plateforme	iOS, Andoird
Licence	Version d'essai, SDK LITE: 590€, SDK PRO:990€	Verssion d'essai, 49€/mois ou 490€/an	Payant (1000€/moi ou 3500€/an)
Compatibilité matériel	Lunettes AR (Epson Moverio, Vuzix M100, ODG R-7	Webcam	Mobile
Moteur 3d	Unity	Oui	Unity
Matching	Images	Images et marqueurs	Complet
Marker based	Oui	Oui	Oui
Markerless	Oui	Non	Oui
3d tracking	Oui (version beta)	Non	Oui
Tracking	Oui	Oui	Oui
Evolutif	Oui	Non	Oui
Sites web	http://www.wikitude.com/	http://www.wikitude.com/	http://dev.inglobetechnologies.com/index.php
Communauté	http://www.wikitude.com/developer-developer-forum	http://www.wikitude.com/developer-developer-forum	http://dev.inglobetechnologies.com/helpdesk/

Figure 2 – Tableau comparatif des SDKs - 2ème partie

D

De la configuration au déploiement

Lors de ce projet, Vuforia a été notre outil principal. Toutes les applications fournies ont été générées par ce SDK. Vuforia permet de faire de la réalité augmentée à partir de différents marqueurs. D'un point de vue langage on appelle la réalité sans marqueur (markerless) lorsque l'on utilise un marqueur différent d'un QRCode, c'est-à-dire que l'utilisation d'une image comme cible est considérée comme de la réalité augmentée « markerless ». Ce document permet d'aiguiller un utilisateur lors de la création d'un projet de réalité augmentée avec Vuforia, et en utilisant les lunettes 3d HoloLens de Microsoft.

1 Le TargetManager de Vuforia

Le lien du site vers le portail développeur de Vuforia[16].

Une partie importante dans la création d'un tel projet est le paramétrage des cibles via l'outil de gestion des cibles (TargetManager[17]). C'est le point d'entrée pour pouvoir utiliser leurs fonctionnalités dans notre projet, créer et générer des cibles de différentes formes et exporter la base de données de cibles vers notre projet Unity.

1.1 Création d'un compte et d'une licence Vuforia

Rendez-vous sur le lien suivant et création d'un nouveau compte[12]. Nous aurons besoin de cette clé pour une prochaine étape, lors de la configuration du SDK. Le lien suivant permet d'accéder directement sur la page[14].

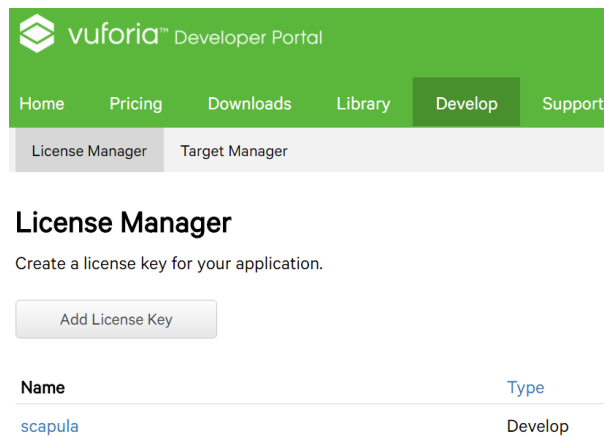


Figure 1 – Licence Manager

1.2 Vuforia AR Starter Kit

Dans Unity, vous pouvez vous rendre directement dans l'Asset Store, recherchez et téléchargez « Vuforia AR Starter Kit ». Une autre alternative consiste à télécharger le SDK Unity Vuforia sur le site web de Vuforia[18]. Il suffit de glisser/déposer le package téléchargé vers Unity.

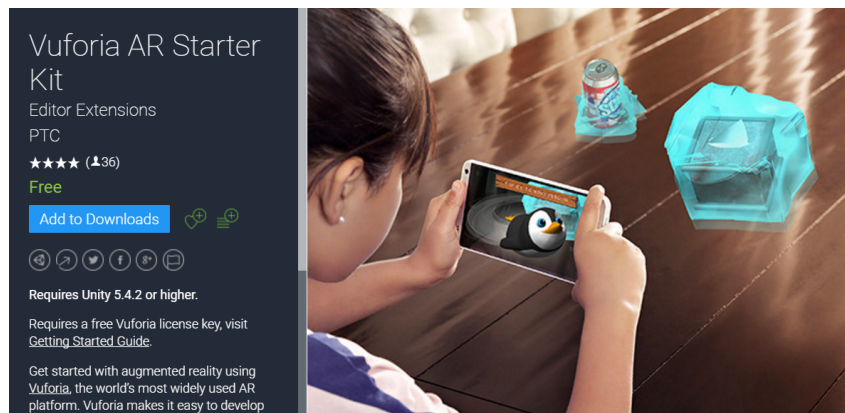


Figure 2 – Vuforia AR Starter Kit

2 Configuration dans Unity

2.1 Configuration de Vuforia dans Unity

Il faut récupérer la clé de licence précédemment créée et l'insérer dans l'onglet VuforiaConfiguration (Vuforia -> Configuration), à l'emplacement App Licence Key. Il faut également configurer la partie Digital Eyewear, et mettre Eyewear Type à Optical See-Through et See-Through Config à Hololens.

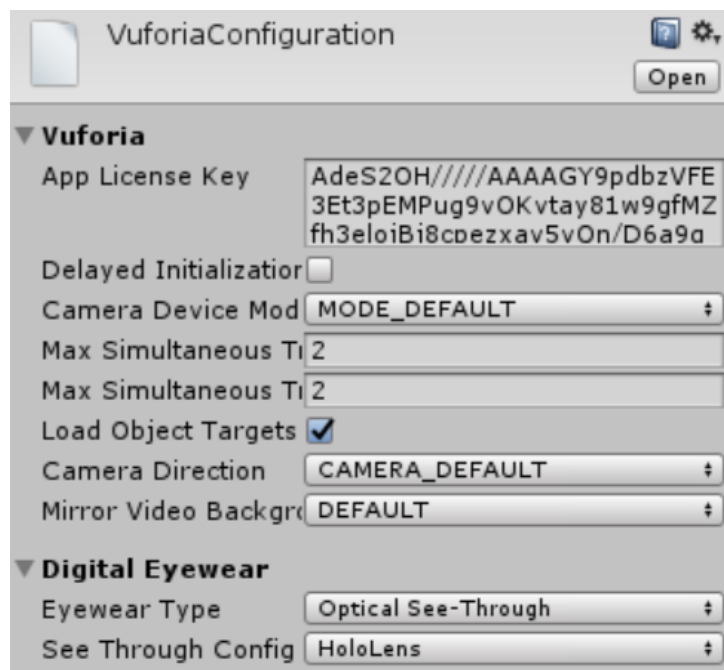


Figure 3 – Vuforia Configuration

2.2 Création de cible et configuration de la caméra Vuforia

Dans le Target Manager, créez une nouvelle base de données (DataBase), puis dans cette base de données ajoutez une cible (Add Target). Ajouter une base de données, puis ouvrez la base de données puis ajoutez une cible :

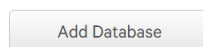


Figure 4 – Add Database

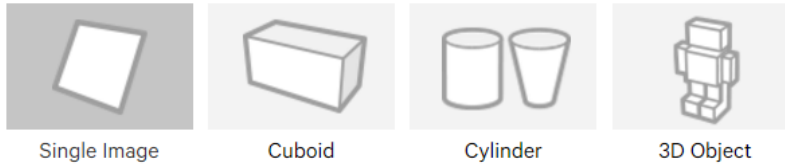


Figure 5 – Add Target

On ouvre ainsi un formulaire de création d'une cible :

Add Target

Type:



File:

marqueur_maquette2.png

Browse...

Invalid file format. Only 8 bit gray scale or 24 bit RGB of file type JPG or PNG are allowed.

.jpg or .png (max file 2mb)

Width:

0.06

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:

marqueur_maquette2

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 6 – Création d'une cible

Ici, on a le choix d'utiliser plusieurs types de cibles. « Single Image » : il s'agit d'un marqueur sur un plan i.e. une feuille, un cube : on renseigne l'image de chaque face, un cylindre et un objet 3d. Attention, l'objet 3d est généré à partir du scanner de Vuforia.

Notation de la cible : lorsque que la création d'une cible est terminée, on peut obtenir sa note, entre 0 et 5, 5 étant la meilleure note. Cette notation est faite en fonction du nombre de point caractéristique de l'image. En résumé, plus une image est nette et contrastée, et mieux elle est notée. Plus les formes ont une géométrie droite et plus y aura de points caractéristiques, contrairement aux formes arrondies. Ce lien[15] est une guide pour l'évaluation des cibles.

2.2.1 Lier la cible Vuforia avec la scène

Ici, il faut choisir la cible. Nous allons prendre l'exemple d'une image. La première étape consiste à ajouter l'image dans le TargetManager de Vuforia. Pour cela rendez-vous dans le Target Manager, téléchargez la base de données, puis glissez-déposez le fichier téléchargé dans Unity.

Download Database (All)

Figure 7 – Bouton "Download Database"

Attention : il faut faire attention à bien sélectionner "Unity Editor" afin que le format de la cible soit bien adapté.

Download Database

1 of 1 active targets will be downloaded

Name:
cereale

Select a development platform:

- ☐ Android Studio, Xcode or Visual Studio
☒ Unity Editor



Figure 8 – Download Database

On obtient un fichier de type *.unitypackage.

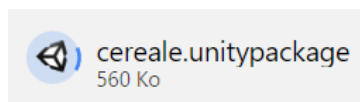


Figure 9 – .unitypackage

Pour importer ce fichier sous Unity, il suffit de le glisser-déposer dans l'Asset de Unity. De retour dans l'onglet VuforiaConfiguration il faut activer la base de données comme dans la figure suivante :

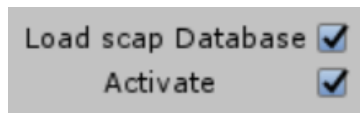


Figure 10 – Activer la base de données

3 Configuration de la caméra

A la place de la caméra standard d'Unity il faut utiliser la caméra de Vuforia. Celle-ci est configurée pour utiliser la caméra des Hololens comme référence. Dans la partie Assets, sélectionnez : Vuforia->ARCamera, et la glisser dans la fenêtre hierarchy :

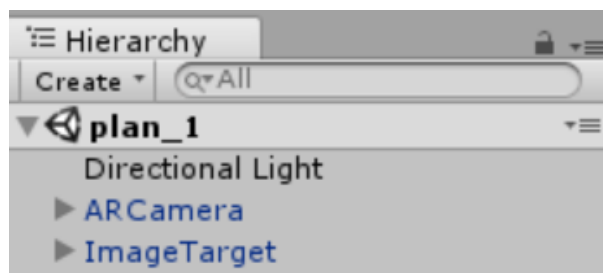


Figure 11 – ARCamera

Il faut maintenant configurer cette caméra. Dans l'inspecteur (Inspector) de la caméra mettre « CAMERA » pour le champ World Center Mode.

Et voilà, la configuration est terminée, il faut maintenant exporter le projet vers les lunettes. Il y a une étape intermédiaire qui consiste à passer par Visual Studio.

3.1 Export du projet Unity C# vers Visual Studio

Sélectionnez Edit -> Project Settings -> Quality. Ici, il faut cliquer sur la flèche qui pointe vers la bas et sélectionner "fastest".



Figure 12 – Qualité de l'application

Sélectionnez File -> Build Settings, puis Windows dans la liste. Il faut changer la valeur de SDK à Universal 10 et celle Build Type à D3D. Nous exporterons le projet en C#, il faut donc cocher « Unity C Projects ».

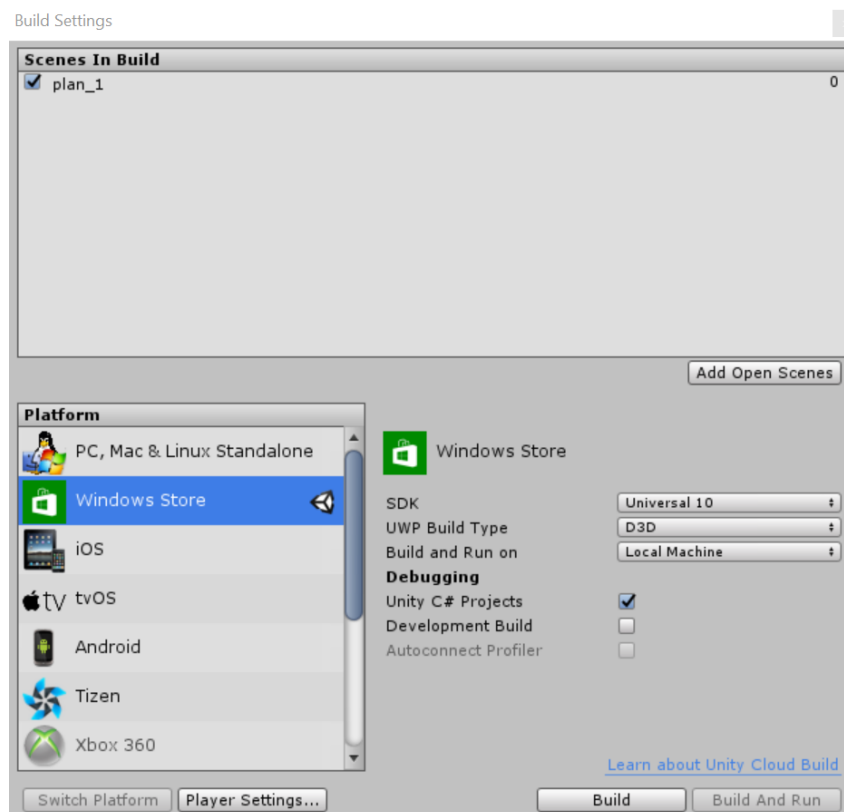


Figure 13 – Configuration de l'application 1

Toujours dans cette fenêtre ouvrez « Player Settings ». Un nouveau panneau s'ouvre sur la gauche, il faut cocher « Virtual Reality Supported et ajouter Windows Holographic dans la liste.

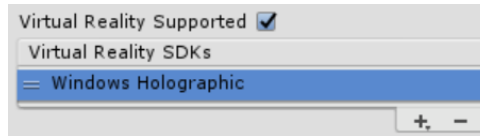


Figure 14 – Configuration de l'application 2

C'était les paramètres obligatoires, maintenant il y a une autre partie permettant de gérer le nom des applications, une description et le versionning :

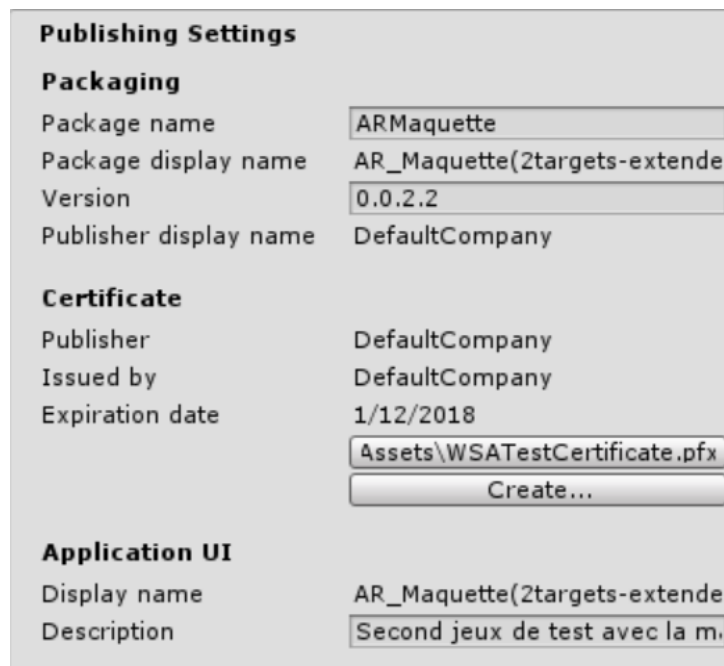


Figure 15 – Versionning

4 Déploiement du projet vers les lunettes

4.1 Exportation du projet vers les lunettes Hololens

La configuration est terminée, on peut donc exporter le projet, toujours dans l'onglet « Build Settings », on appuie sur « Build ». On crée un nouveau dossier App à l'emplacement que l'on veut, puis on sélectionne ce dossier.

Un projet Visual Studio est généré en C#. Il y a plusieurs composantes dans le projet généré. D'abord, il y a le fichier App.cs. Ce fichier est le fichier principal, qui gère l'application (initialisation, démarrage, etc...). Il y a aussi le Package.appxmanifest, celui-ci sert à la gestion du versionning et du nommage de l'application.

Il faut d'abord modifier le fichier Package.appxmanifest (Ouvrir le code source). Et à la ligne TargetDeviceFamily modifier Name="Windows.Universal" par Name="Windows.Holographic" et MaxVersionTested="10.0.10240.0" par MaxVersionTested="10.0.10586.0".

On modifie la cible de Debug vers Release et d'ARM vers X86.

Pour exécuter le projet on a plusieurs possibilités. Le premier est de l'exécuter à travers le réseau. Pour cela il faut que les lunettes et l'ordinateur soient sur le même réseau.

On récupère l'adresse IP des lunettes. Allez dans les paramètres, Network Internet puis Wifi. L'adresse IP se trouve dans les options avancées.

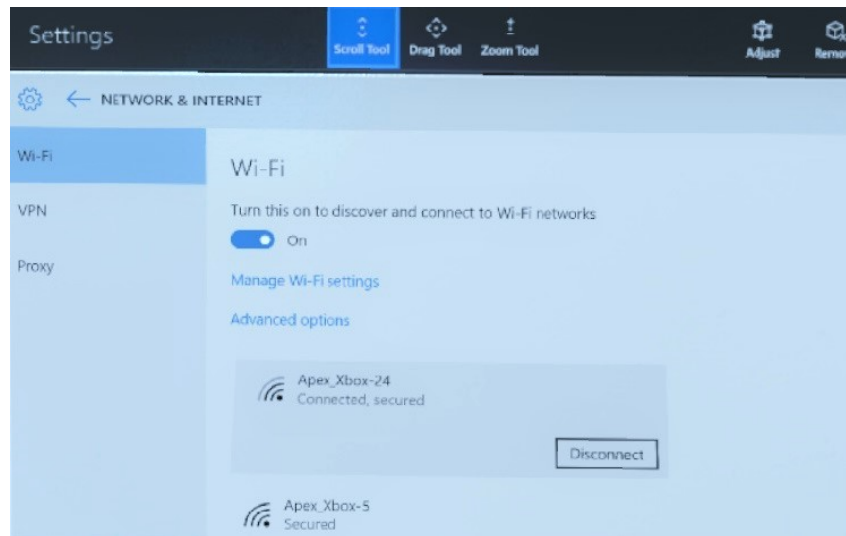


Figure 16 – Récupérer l'IP des lunettes

De retour dans l'IDE Visual, on modifie "Local Machine" par "Remote Machine", On ajoute l'adresse IP et on change le mode d'authentification à "Universel".

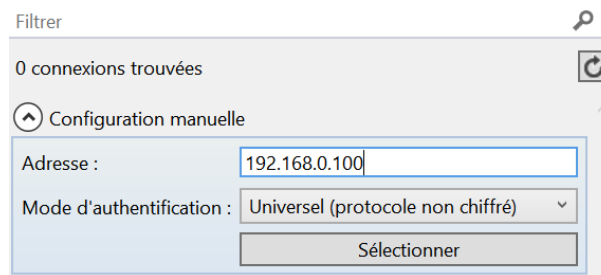


Figure 17 – Connection à distance

La dernière étape est d'exécuter sans débogage dans l'onglet Debug, attendre le que le projet soit déployé et visualiser le résultat à travers les lunettes.

La seconde solution consiste à connecter l'ordinateur et les lunettes via USB. Ici, on ne modifie pas Local Machine. On déploie sans débogage et l'application démarre.

5 Guide d'utilisation

5.1 Démarrage de l'application

Il y a plusieurs moyens de lancer l'application. Soit lors du déploiement du projet vers les lunettes, comme expliqué précédemment, soit en se rendant sur le portail Hololens. De même, il y a deux possibilités pour s'y rendre. Si l'ordinateur et les lunettes sont connectés sur le même réseau alors il faut se rendre à l'adresse suivante : <http://IP>, exemple : <http://192.168.100.120>. A ce moment on nous demande de créer un compte. On entre le nom, puis un code Pin est demandé. Vous le trouverez à travers les lunettes, en vous rendez dans les propriétés/réglages, puis Update Security et enfin dans l'onglet développeur, comme sur la figure suivante :

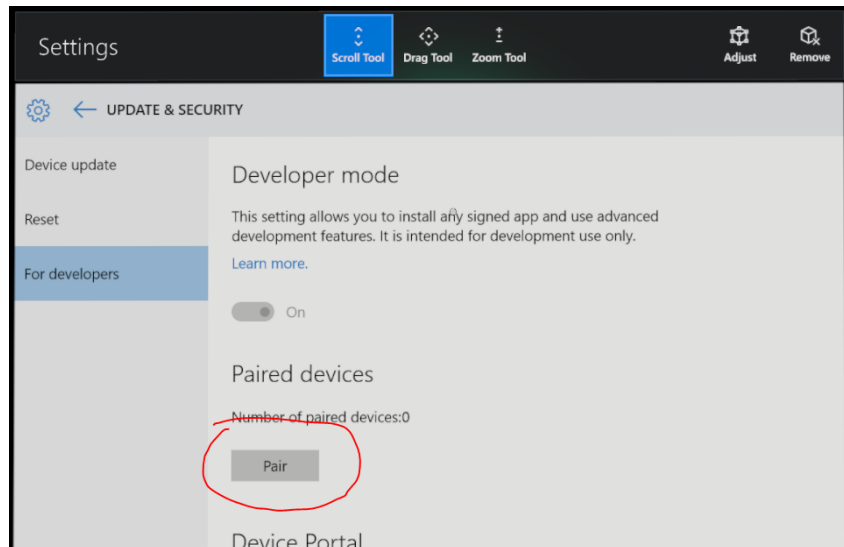


Figure 18 – Appareillage

Cliquez sur Pair et à ce moment un code s'affiche, entrez ce code, et vous pouvez maintenant créer un utilisateur. Vous avez maintenant accès au portail de votre Hololens. Pour lancer l'application, rendez dans l'App Manager (onglet sur le côté à gauche). Ouvrir la liste et lancer notre application. La figure suivante illustre ces étapes :

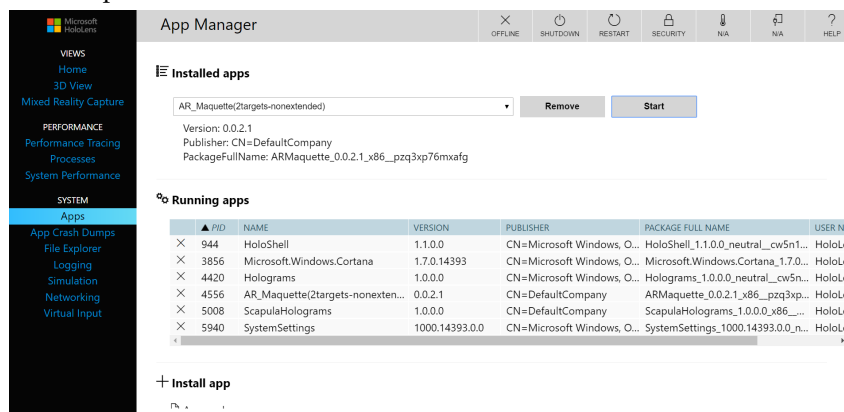


Figure 19 – App Manager

Attention, l'application est présente seulement si l'application a déjà été déployée.

Il existe une autre manière d'accéder à ce portail si vous n'avez pas de connexion internet. Il suffit de brancher les lunettes à l'ordinateur via l'USB. L'adresse à entrer dans votre navigateur est : <http://127.0.0.1:10080>. La manipulation reste la même que précédemment.

6 Utilisation de l'application

Vous constatez que l'application a bien démarré lorsque l'image "Made with Unity" apparaît. Vous n'avez plus qu'à regarder les marqueurs afin de laisser agir la réalité augmentée. Regardez le document « Plan d'expérience » afin de comprendre quelles sont les différents paramètres qui impacte sur la qualité de la réalité augmentée.

E

Dossier de tests

1 Plan d'expérience

Les plans d'expérience permettent d'obtenir un maximum de renseignement sur les expérimentations effectuées. Ils sont un outil de choix dans la compréhension et l'analyse des méthodes mises en place. Dans un plan d'expérience, on ne parle pas de variables mais de facteur. Les grandeurs mesurées sont appelées réponse. La valeur donnée à un facteur est appelée niveau. Pour étudier l'influence d'un facteur on s'intéresse à sa limite basse, niveau bas et la limite haute, niveau haut. Lorsque l'on a plusieurs facteurs, on parle de l'espace expérimental. Il y a $n+1$ dimensions pour une expérience à n facteurs, l'unité ajoutée correspond à la réponse.

Les facteurs utilisés sont :

- La taille du marqueur : qu'il s'agisse d'une single image ou d'un cuboïde, le principe reste le même, des images sont utilisées comme marqueurs. La taille de l'image est exprimée en centimètre. Il faut faire attention à avoir un ratio cohérent (exactement cohérent) entre la taille et la dimension de l'image en pixel.
- La note du marqueur : L'outil de réalité augmentée Vuforia nous permet d'obtenir une note à nos marqueurs. Celle-ci est calculée en fonction du nombre des points caractéristiques dans l'image.
- Le nombre de marqueurs
- La distance à la cible : lors de la configuration nous allons évaluer si la position du marqueur et la scène par rapport à la caméra à une importance.
- La luminosité : la luminosité est mesurée en lux.

La réponse étudiée est :

- La stabilité : en enregistrant une vidéo de l'expérience, nous prendrons un intervalle d'une dizaine de seconde où nous mesurerons le temps de bon positionnement de la scène. Nous obtiendrons un pourcentage de stabilité. Dans un premier temps nous donnerons une note de 0 à 5.
- La distance où cela fonctionne : en fonction de l'angle de vue des lunettes et de la taille de l'écran ;

2 Résultats des tests

2.1 Type 1 – Single Image

2.1.1 Test 1 (19/01/2017)



Image originale



Points caractéristiques

Figure 1 – Test 1

Taille du marqueur : A4

La note du marqueur : 2/5

Lorsque que l'on regarde le traitement fait sur l'image de droite, on constate que le nombre de points caractéristiques est faible. Lors du test on constate que la réalité augmentée ne fonctionne pas. Peu importe la distance à la cible ou la luminosité.

Résultat : Echec, aucune visualisation de la scène.

Le nombre de marqueurs : 1 ; la distance à la cible : N/A ; la luminosité : N/A.

2.1.2 Test 2 (19/01/2017)



Image originale



Points caractéristiques

Figure 2 – Test 2

Taille du marqueur : A4

La note du marqueur : 4/5

Nous avons ajouté le logo de l'école ainsi que des régions jaune au niveau des écarteurs. Ainsi on ajoute du contraste à l'image (essentiel pour extraire les points caractéristiques).

Le nombre de marqueurs : 1

La distance à la cible : à une distance de 50cm à 1m la scène est visible. En étant à 1 mètre au-dessus de la cible, on peut visualiser la scène à 360°.

La luminosité : N/A

Résultat : Plutôt bon.

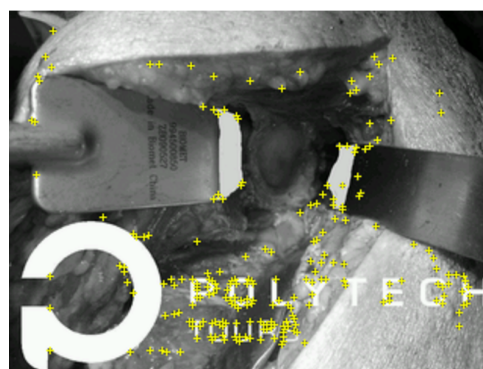
Stabilité : 4/5 ; Distance : 50cm-1m.

Ce test nous permet de mieux comprendre le système de notation d'une cible par Vuforia. 4 étoiles sont suffisant pour que la réalité augmentée fonctionne. En-dessous ça semble compliqué. La scène est dans la cible, c'est-à-dire que lorsque l'on regarde la cible, la scène apparaît, nous verrons plus tard que c'est une condition idéale. Lors du projet test nous allons tester le résultat lorsque la notation de la cible est maximale.

2.1.3 Test 3 (20/01/2017)



Image originale



Points caractéristiques

Figure 3 – Test 3

Taille du marqueur : A4

La note du marqueurs : 5/5

Toujours dans l'optique d'améliorer le tracking, nous avons changé la couleur du contraste du bleu au blanc. Avec le fond de couleur rouge cela améliore le contraste. On atteint maintenant la note maximale.

Le nombre de marqueurs : 1

La distance à la cible : à une distance de 50cm à 1m la scène est visible. En étant à 1 mètre au-dessus de la cible, on peut visualiser la scène à 360°.

La luminosité : N/A

Résultat : Bon.

Stabilité : 5/5 ; Distance : 50cm-1m.

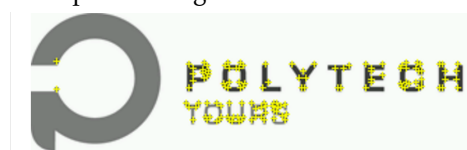
La stabilité s'est nettement améliorée. La scène ne saute plus en fonction des déplacements. La distance à la cible où cela fonctionne est grande. Les trois tests sont effectués avec une cible grande (format A4), on suppose que cela se révèle être un avantage plus on s'éloigne de la cible. En revanche on ne peut pas s'approcher à quelques centimètres car la cible n'est plus entièrement dans le champ de vision des lunettes. Pour rappel l'écran des lunettes est assez petit et l'angle de champ de vision très réduit.

2.1.4 Test 4 (2/03/2017)

Lors de ce test nous allons travailler sur une cible de taille plus réduite. En effet lorsque le chirurgien opère il a besoin de s'approcher au plus près de l'omoplate du patient. De plus, dans l'hypothèse où nous utiliserions des images, une feuille au format A4 ne serait pas envisageable.



Image originale



Points caractéristiques

Figure 4 – Test 4

Taille du marqueur : 6x12

La note du marqueur : 4/5

Le nombre de marqueurs : 1

La distance à la cible : 20cm-1m, ici la cible est face à l'utilisateur lorsqu'il est assis sur une chaise. La scène est au-dessus de la cible et à environ 5cm en arrière. L'image suivante illustre la couple cible/scène :



Figure 5 – Mise en place de la cible et la scène

La luminosité : N/A

Résultat : Moyen.

Stabilité : 3/5 ; Distance : 20cm-80cm.

La stabilité est un peu particulière. Déjà l'utilisateur a besoin d'être en face de la cible pour que la réalité augmentée fonctionne. C'est-à-dire que comme l'image est de face, lorsqu'il est bien au-dessus de la cible, elle n'est pas reconnue. L'avantage est que l'on peut s'approcher plus près que lors des tests précédents. Il nous faut trouver une solution pour améliorer l'application lorsque la cible et la scène ne se confondent pas.

2.2 Type 2 – Cuboid

Ici, nous testons un second type de forme. C'est en réalité une multi Target où chaque face est une cible de type Single Image. Nous avons été confrontés à des problèmes de configuration. Il y a aucun écart possible entre la taille du cube, et le ratio longueur/largeur de l'image en pixel. Sachant qu'un pixel n'est pas divisible, il faut prévoir au début les tailles de nos cubes. Pour le premier test, nous avons créé la forme à partir de notre patron, donc on gère les tailles. En revanche, pour le second test nous n'avions pas la main pour les tailles, le travail de mise en échelle a été compliqué.

2.2.1 Test 1 (12/02/2017)

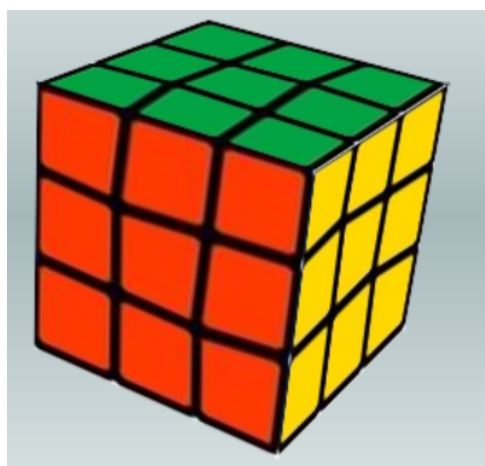
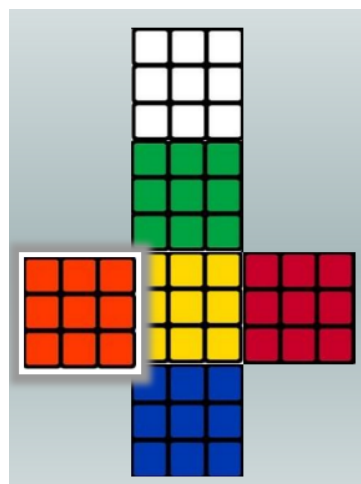


Image 3d



Patron

Figure 6 – Test 1

Taille du marqueur : 6x6x6

La note du marqueur : 0/5 pour chaque face

A ce stade des tests, nous n'avions pas conscience de la présence de cette notation. C'est d'ailleurs à ce moment que nous l'avons découverte.

Le nombre de marqueurs : 6 ; La distance à la cible : N/A ; La luminosité : N/A

Résultat : Echec. Au moins, nous avons pu mettre en avant les différentes contraintes de configuration.

2.2.2 Test 2 (19/02/2017)



Image 3d



Points patron

Figure 7 – Test 1

Taille du marqueur : 19*27,2*5

La note du marqueur : face et back : 5 ; top et bottom : 4 ; left : 3 ; right : 1 ;

Le nombre de marqueurs : 6
 La distance à la cible : 50cm-1m
 La luminosité : N/A

Résultat : Plutôt bon.

Stabilité : 4/5 ; Distance : 50cm-1,3m.

Ici, l'affichage de la scène dépend de la face (ou des faces) visible(s). On constate une très bonne reconnaissance sur les faces avant, derrière, droite et au-dessus. Ce qui est en corrélation avec les notes attribuées à chaque face. On constate que la reconnaissance sur plusieurs marqueurs visibles simultanément fonctionne.

2.3 Type 3 – Association de marqueurs avec une maquette d'épaule

L'enjeu dans cette partie est d'associer notre cible avec une maquette d'épaule, alors il faut que la scène se superpose au mieux avec la maquette.

2.3.1 Test 1 (8/03/2017)

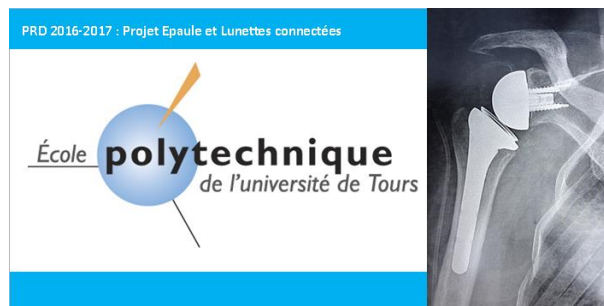


Figure 8 – Image original

Taille du marqueur : 15*6
 La note du marqueur : 5
 Le nombre de marqueurs : 1
 La distance à la cible : 20cm-1m
 La luminosité : N/A

Résultat : Plutôt moyen.

Stabilité : 3/5 ; Distance : 20cm-1m.

Ici, on reprend un des test précédent, où la cible est verticale et non posée face à une table. On a amélioré le rendu grâce à un cible de meilleur qualité.

2.3.2 Test 2 (15/03/2017)



Figure 9 – Image original

Taille du marqueur : 15*6
 La note du marqueur : 5
 Le nombre de marqueurs : 1
 La distance à la cible : 20cm-1m
 La luminosité : N/A

Résultat : Plutôt bon.

Stabilité : 4/5 ; Distance : 20cm-1,3m.

Contrairement au test précédent, la cible est placée au-dessus de la scène. C'est une condition vraiment meilleure. En effet, la scène n'est pas confondue dans la cible, et le fait que la cible soit au-dessus de la scène est bien meilleur, on respecte la position de l'écran des lunettes qui est en dessous de la caméra.

2.3.3 Test 3 (22/03/2017)

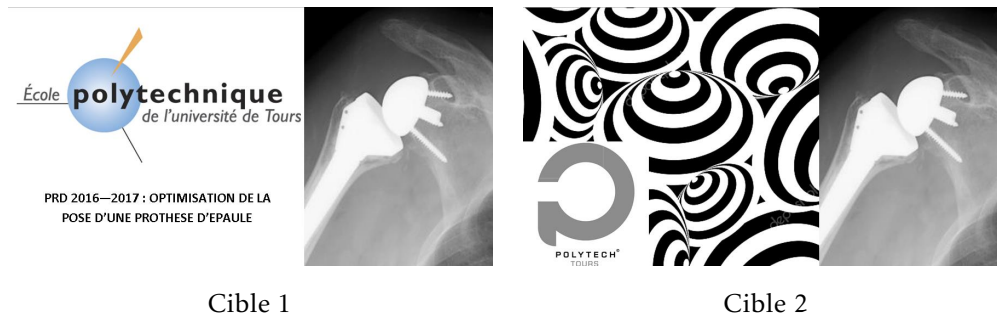


Figure 10 – Test 1

Taille du marqueur : 15*6

La note du marqueur : 5 pour les deux ;

Le nombre de marqueurs : 2

La distance à la cible : 10cm-1m

Le marqueur de droite est confondu avec la scène. Le marqueur de gauche est au-dessous de la scène. La luminosité : testée dans plusieurs conditions, nulle, moyenne et bonne.

Résultat : Très bon.

Stabilité : 5/5 ; Distance : 10cm-1,3m.

On constate une amélioration par rapport aux tests précédents. Avec la scène plus basse que la cible on gagne en stabilité par rapport à lorsque qu'elle est en-dessous. De plus, on agrandit les angles de vue, et on améliore le confort de l'utilisateur. Il n'a plus à se baisser et à regarder vers le haut. C'est plus facile de regarder vers le bas, que vers le haut. La luminosité a eu un impact sur les performances, cela est dû au fait que les premiers tests ont été faits dans un environnement très sombre. La qualité du papier et de l'impression ont été un facteur de réussite. On peut également se référer à l'écran que l'on perçoit correspondant à la taille de l'écran de visualisation ou à la vue caméra.

Paramètre « Extended tracking »

Ici, l'étape d'initialisation est meilleure lorsque la cible est au centre de cet écran, mais le tracking fonctionne toujours lorsque seule la moitié de la cible est visible dans le haut de cet écran.

"If you disable extended tracking, Vuforia runs all the time and the tracking is smooth but the teapot disappears as soon as the marker disappears. In extended mode the teapot persists even after the marker is removed, but if the marker is visible i get the aforementioned jumps."[11]

3 Conclusion

De face, l'omoplate prend de la place verticalement mais pas horizontalement. Pour que le champ de vision soit entièrement utilisé, mettre la cible sur le côté est pertinent. Lorsque que l'on regarde l'omoplate de côté, elle prend de la place sur tout le champ de vision : il est alors pertinent de confondre la cible et la scène.

Après avoir fait tous ces tests, on peut conclure que notre problème a été d'optimiser la position de la cible par rapport à la scène.



Comptes rendus hebdomadaires

Compte rendu n°1 du 25/09/2016

Après un rappel des consignes du PRD. Les 2 premiers jours de travail ont permis de se remettre dans le contexte du projet. Des premières recherches sur la réalité augmentée ont été faites ainsi que la recherche des lunettes 3D pouvant répondre à la problématique. J'oriente principalement ma recherche vers les outils de développement associés aux lunettes (logiciels/SDKs). Un panorama sera disponible la semaine prochaine. Un drive a été mis en place afin que tous les documents nécessaires soient toujours disponibles. J'ai également créé un projet sur le site de gestion de projet Trello. Il permettra de suivre l'avancée du travail. Je le partagerai cette semaine.

Compte rendu n°2 du 02/10/2016

Cette semaine j'ai travaillé sur le panorama des lunettes de réalité augmentée. Il a été présenté à Christian Proust jeudi dernier. Il est disponible sur le drive que je viens de vous partager (vous avez du recevoir un mail). J'ai choisi d'orienter ma recherche en fonction des SDKs associés. En effet, en recherchant les logiciels/SDKs pour faire de la réalité augmentée, on trouve presque à chaque fois les matériels compatibles, dont les lunettes de réalité augmentée. Vous retrouverez les différents logiciels et SDKs recherchés sur le drive, au travers d'un tableau.

Les 4 prochains jours de travail vont être dédiés aux HoloLens de Microsoft. On va chercher à les caractériser au mieux et déterminer les outils à utiliser pour développer des applications compatibles. L'objectif est de s'assurer que notre projet est réalisable avec ce matériel.

Il y a sur le drive mon rapport de stage enrichi après les deux derniers jours de travail. Christian Proust m'a fourni des lectures sur les SDKs de réalité augmentée.

Compte rendu n°3 du 09/10/2016

Cette semaine j'ai approfondi mes recherches sur les HoloLens de Microsoft, j'ai installé l'émulateur HoloLens. Le principal défaut est qu'il n'autorise pas l'utilisation d'un flux vidéo en temps réel. Ensuite je me suis rendu compte qu'il était important de terminer correctement les états de l'art. J'ai donc réfléchi sur des questions importantes liées aux différentes technologies (voir document : Comparatif des lunettes de réalité augmentée 2). Christian Proust a corrigé les premiers éléments du rapport, en repensant le plan notamment. Les mises à jour ont été faites sur le drive.

Les deux prochains jours de travail seront consacrés à finaliser les tableaux de comparatifs des lunettes et des SDKs (il manque quelques informations), ainsi que la partie approfondie sur les HoloLens.

Compte rendu n°4 du 16/10/2016

Cette semaine j'ai terminé les tableaux comparatifs des lunettes et des SDKs. J'ai complété l'état de l'art sur la réalité augmentée. L'état de l'art complet sur les Hololens a bien avancé. Je suis actuellement en train de schématiser les outils nécessaires à son développement. Les prochaines séances vont permettre de terminer le travail de recherche sur les Hololens. Il faut également commencer à réfléchir et argumenter le choix du SDK de réalité augmentée que nous allons utiliser.

L'objectif est de terminer les états de l'art avant l'arrivée des lunettes.

Vous pouvez consulter l'état du rapport et les différents tableaux sur le drive. Les tableaux terminés sont disponibles au format .xls, les autres sont des brouillons.

Compte rendu n°5 du 23/10/2016

Cette semaine j'ai travaillé sur les Hololens. Je me suis également intéressé plus spécifiquement aux SDKs et logiciels de réalité augmentée. A première vue, je n'en ai pas trouvé pouvant reconnaître un objet à partir de son modèle 3d, provenant de Blender par exemple. La plupart des SDKs limitent le nombre de cibles à travers des target manager (plateforme où l'on charge ces cibles). Je vais continuer les recherches, j'ai quelques pistes à explorer. On pourra peut-être développer nous-même nos méthodes et fonctions de réalité augmentée.

Compte rendu n°6 du 06/11/2016

Cette semaine j'ai travaillé sur le cahier de spécifications (il est disponible sur le drive). Il permet de proposer une première approche du projet, avec ses méthodes, ses interfaces, ses limites, une première réflexion des problèmes que l'on pourrait rencontrer. Nous l'avons corrigé une première fois avec Christian Proust, puis j'ai rencontré Nicolas Ragot qui m'en a fait plusieurs retours. Les premières heures des créneaux de cette semaine sont dédiées à la correction de ce document. S'en suivront la poursuite du travail à la recherche des logiciels de réalité augmentée, ainsi que la rédaction d'un document expliquant les méthodes de gestion de projet mises en place.

Compte rendu n°7 du 13/11/2016

Cette semaine j'ai travaillé à la rédaction du cahier de spécifications, et des méthodes de gestion de projet. Les documents sont presque terminés. J'ai également poursuivi mes recherches sur le choix du SDK. Je vais continuer mes recherches vers des solutions où le développement est plus important (peut-être qu'il faudra implémenter nos propres fonctions de reconnaissance par exemple), afin de voir quelles sont les possibilités.

Compte rendu n°8 du 20/11/2016

La rédaction du cahier de spécifications est terminée (il est en PJ). Il est amené à être modifié et enrichi. Les différentes parties de ce document vont être incluses dans le rapport de la première partie du PRD. J'ai discuté avec Christian Proust mercredi dernier. Nous avons défini quatre objectifs de réalité augmentée avec les lunettes (voir CDS). Nous avons reçu les lunettes ! Mercredi matin, je vais procéder à l'installation avec Mickael Rousseau. Ensuite je commencerai des premiers tests pour la réalisation du premier objectif (voir CDS). Je pense utiliser le SDK Vuforia.

Compte rendu n°9 du 27/11/2016

C'était la première semaine de travail avec les Hololens. C'est vraiment une technologie impressionnante ! Nous avons testé les différentes applications disponibles pour afficher et "déposer" des objets virtuels sur les tables (réelles). Il est vrai que le champ de vision est restreint mais je pense qu'il est suffisant pour le projet, en effet la taille de l'omoplate n'est pas trop grande. J'ai également créé une petite application permettant d'afficher une omoplate dans le champ de vision des lunettes. Celle-ci est importée au préalable dans l'application au format .obj (j'ai utilisé Unity pour créer l'application). Cette semaine je vais essayer d'afficher cette même omoplate sur un cube, sur lequel il y a des marqueurs, après avoir reconnu ce cube (Objectif1). J'utiliserai Vuforia en plus des autres outils de développement déjà utilisés. Je vais également travailler sur le rapport que je dois remettre le 5 décembre (une semaine avant les soutenances).

Je vous enverrai des vidéos des différents résultats que j'obtiens.

J'ai également fait des recherches sur les fonctionnalités de développement disponibles pour les lunettes (SDK de Windows). J'ai remarqué que l'on peut utiliser les fonctions de "mapping" de l'environnement, mais il ne semble pas possible d'utiliser ces données pour les confondre avec des objets 3d importés, mais je dois davantage explorer cette piste afin de m'en assurer. Le travail sur le choix du SDK "final" se poursuit.

Compte rendu n°10 du 04/12/2016

Cette semaine je me suis encore familiarisé avec les Hololens. J'ai également travaillé sous Unity, afin de régler la taille et la position de l'hologramme à afficher avec les lunettes (comme sur les vidéos). J'ai rencontré Mr Martineau mercredi afin qu'il réponde à quelques questions sur la gestion de projet. La majeure partie de mon temps de travail a été consacrée à la rédaction du rapport du PRD1. Ce sera également le cas pour les deux prochains jours, mon rapport doit être terminé pour la fin de semaine. Je vous le ferai suivre au plus tard vendredi. Il faudra également que je prépare ma soutenance, qui semble être prévu lundi 12.

Compte rendu n°11 du 08/01/2017

Cette semaine j'ai créé une application de réalité augmentée où la cible est une couverture de magazine (i.e un plan) (une vidéo est disponible sur le drive mais elle est de mauvaise qualité). Cela a été possible grâce au SDK Vuforia. J'ai également préparé l'application où la cible est un cube avec des marqueurs (les premiers tests n'ont pas fonctionnés, j'ai quelques réglages à améliorer, notamment la précision des marqueurs), elle sera terminée mercredi. Il s'agit de l'objectif 1. J'ai également activement travaillé sur le duo Unity/OpenCV afin de réaliser l'objectif2. Il me faut encore du temps de recherche afin de m'assurer de la faisabilité. Une rencontre est prévue jeudi après-midi avec Christian Proust afin de faire le bilan du travail. Le diaporama de la soutenance faite au mois de décembre est disponible sur le drive. J'ai oublié la version corrigée du rapport sur mon poste à l'école, je l'ajouterai sur le drive dès que possible.

Compte rendu n°12 du 17/01/2017

Cette semaine, j'ai complété mes recherches sur OpenCV avec Unity, et plus précisément les différents algorithmes de détection d'objets 3D. J'ai continué mon travail sur l'objectif1. J'ai fait face à des problèmes de configuration notamment dû à la précision des images du cube. Je vais continuer ce mercredi afin de réaliser l'objectif1. Christian et Julien, j'ai bien reçu vos images, il faudra en imprimer en grand format pour samedi. De mon côté je vais configurer les applications avec ces images (cibles).

Compte rendu n°13 du 30/01/2017

J'ai réussi à faire de la réalité augmentée avec un cube. Le cube est une boîte de céréales et les images incrustées dessus servent de cible. J'ai également travaillé les scripts Unity afin de créer nos propres méthodes de détection. Il serait bien d'utiliser OpenCV avec Unity. Cette semaine je vais donc continuer mon travail de script. Je vais également tester le scan de Vuforia, pour tester les performances.

Compte rendu n°14 du 05/02/2017

Cette semaine j'ai commencé le travail avec les scripts Unity. La première étape consiste à récupérer le flux vidéo des lunettes. Cette étape fonctionne, on récupère le flux sous forme d'image à une fréquence définie. La seconde étape est le traitement de cette image, pour cela on va utiliser OpenCV. L'étape suivante sera d'afficher les résultats du traitement sous forme d'hologramme en temps réel.

Compte rendu n°15 du 13/03/2017

La semaine dernière j'ai démarré la rédaction de différents documents. Le premier est le plan de test/expérience. Il permet d'exprimer la qualité d'un test, selon des facteurs et des réponses. Le second document est le tutoriel complet de la configuration de Vuforia (SDK de réalité augmentée). En effet, la configuration est plutôt complexe et nécessite plusieurs étapes. Cette semaine je vais continuer la rédaction de ces documents, et faire de nouveau test avec Vuforia en utilisant plusieurs marqueurs simultanément.

Webographie

- [1] ARTToolKIT. *ARToolKit*. URL : <https://artoolkit.org/>.
- [2] AUGMENTEDMEDIA.NET. *Une brève histoire de la réalité augmentée*. URL : <https://augmentedmedia.net/2011/09/18/une-breve-histoire-de-la-realite-augmentee/>.
- [3] AUGMENTED-REALITY.FR. *Comment faire de la réalité augmentée*. URL : <http://www.augmented-reality.fr/cest-quoi-la-realite-augmentee/comment-faire-de-la-realite-augmentee/>.
- [4] CURVSURF. *Projet CurvSurf*. URL : <https://plus.google.com/+CurvSurf>.
- [5] PTC FRANCE. *PTC associe la technologie Vuforia au casque de réalité augmentée Microsoft HoloLens*. URL : <http://www.ptcfrance.com/presse/2016/ptc-associe-la-technologie-vuforia-au-casque-de-realite-augmentee-microsoft-holoLens>.
- [6] Université de LAVAL. *Projet HoloLens*. URL : <http://vision.gel.ulaval.ca/~jflalonde/projects/hololens3d/>.
- [7] RESEARCHGUIDES.DARTMOUTH.ED. *Les techniques de reconnaissance*. URL : <http://researchguides.dartmouth.edu/c.php?g=59732&p=382860>.
- [8] STRABIC. *Réalité augmentée humanité*. URL : <http://strabic.fr/Realite-augmentee-humanite>.
- [9] TECHCRUNCH. *Augmented and Virtual Reality to hit 150 billion by 2020*. URL : <https://techcrunch.com/2015/04/06/augmented-and-virtual-reality-to-hit-150-billion-by-2020/>.
- [10] VRS.ORG.UK. *Capteurs*. URL : <http://www.vrs.org.uk/virtual-reality-gear/motion-tracking/sensors.html>.
- [11] VUFORIA. *Extended Tracking*. URL : <https://library.vuforia.com/articles/Training/Extended-Tracking>.
- [12] VUFORIA. *Inscription*. URL : <https://developer.vuforia.com/targetmanager/>.
- [13] VUFORIA. *Les applications Vuforia*. URL : <https://www.vuforia.com/Apps>.
- [14] VUFORIA. *Licence Manager*. URL : <https://developer.vuforia.com/targetmanager/licenseManager/licenseListing>.
- [15] VUFORIA. *Natural Features and Image Ratings*. URL : <https://library.vuforia.com/articles/Solution/Natural-Features-and-Ratings>.

- [16] VUFORIA. *Portail développeur*. URL : <https://developer.vuforia.com/>.
- [17] VUFORIA. *Target Manager*. URL : <https://developer.vuforia.com/targetmanager/>.
- [18] VUFORIA. *Téléchargement du SDK*. URL : <https://developer.vuforia.com/downloads/sdk>.
- [19] WIKIPÉDIA. *Moteur 3D*. URL : https://fr.wikipedia.org/wiki/Moteur_3D.
- [20] WIKIPÉDIA. *Windows Holographic*. URL : https://en.wikipedia.org/wiki/Windows_Holographic.
- [21] ZDNET. *Réalité augmentée vs Réalité virtuelle, le match*. URL : <http://www.zdnet.fr/actualites/realite-augmentee-vs-realite-virtuelle-le-match-39841616.htm>.



Bibliographie

- [1] PFE de MAXIME FACOMPRESZ. « Réalisation d'une application sur les lunettes à réalité augmentée visant à assister le chirurgien au cours d'une opération d'omoplate ». In : (2015).
- [2] Minghui ZANG. « Utilisation des lunettes 3D pour une opération chirurgicale d'une omoplate ». In : (2016).

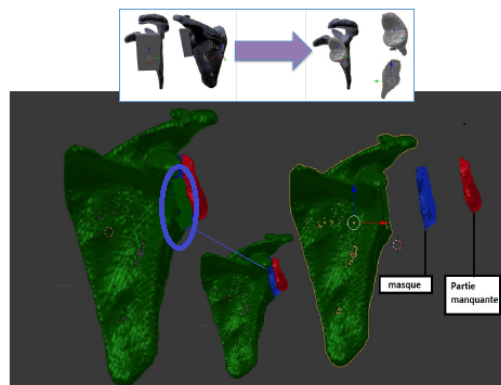
Projet Epaule et Lunettes Connectées :

Jordan NICOT

Encadrement : Julien BERHOUE, Mohamed SLIMANE et Christian PROUST

Objectifs

- Se familiariser avec les technologies de la réalité augmentée
- Réaliser une application ayant pour but d'assister les chirurgiens lors d'une opération chirurgicale à l'aide de lunettes 3D



omoplate pathologique et glène reconstituée virtuellement

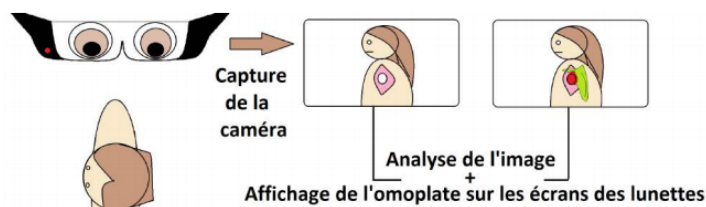
Mise en œuvre

- Étude des matériels et logiciels existants
- Reconnaissance de formes
- Poursuite d'une cible, et affichage, en temps réel (au millimètre près)



Hololens de Microsoft

Illustration de la démarche



Projet Epaule et Lunettes Connectées

Résumé

L'objectif du « projet Epaule » 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.

Mots-clés

Epaule, Réalité augmentée, Omoplate, Lunettes 3D, Hololens

Abstract

The purpose of the "Shoulder Project" is to link the fields of orthopedic surgery and computer science. We wish to be able to display, in real time, during the surgical operation, on the connected glasses carried by the surgeon, an image of the pathological scapula and / or the virtually reconstituted glenoid.

The superposition with the target (what the surgeon really sees) must be very precise: not only between the image of the glenoid and the glenoid itself - visible - but also between the image of the scapula and the actual scapula - Itself not visible. The accuracy of these overlays will have to be quantified over time. This display obviously means deformation of the images in accordance with the displacements of his gaze.

Keywords

Shoulder, Augmented reality, Scapula, 3D Glasses, Hololens

Tuteurs académiques

Julien BERHOUET
Mohamed SLIMANE
Christian PROUST

Étudiants

Jordan NICOT (DI5)