

CONSERVATOIRE NATIONAL DES ARTS ET METIERS  
CENTRE REGIONAL PACA

## **MEMOIRE**

Présenté en vue d'obtenir

Le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Informatique et Multimédia

Par

**Thomas CIANFARANI**

---

Refonte du système d'avatar virtuel

**Soutenu le 25 août 2021**

---

### **JURY**

**PRESIDENT : Annick RAZET**

**MEMBRES : Arnaud ELOI**

**Olivia PAPINI**

**Laurent-Stéphane DIDIER**

**Pierre PARADINAS**

Maître d'apprentissage

Tutrice académique

3<sup>ème</sup> membre

4<sup>ème</sup> membre



# Remerciements

---

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes auxquelles je voudrais témoigner toute ma gratitude.

Je tiens tout d'abord à exprimer ma reconnaissance à l'école d'ingénieurs du Cnam pour l'opportunité qui m'a été offerte au sein de cette formation. J'adresse mes sincères remerciements à tous les professeurs, intervenants, responsables de formation et toutes les personnes ayant participé au bon déroulement de mon cursus d'une manière ou d'une autre.

Je remercie ma tutrice académique, Olivia PAPINI, pour son accompagnement et ses conseils durant ces trois années d'apprentissage. Nos échanges et le partage de sa vision me guideront, en toute certitude, lors de mes expériences futures.

Je souhaite remercier C2Care, l'entreprise dans laquelle j'ai effectué mon apprentissage, d'avoir cru en moi le premier jour, ainsi que tout au long du parcours. L'expérience que j'ai pu y acquérir sur le plan technique, mais aussi simplement sur le monde du travail me sera indispensable pour mes prochaines aventures professionnelles.

Je voudrais particulièrement remercier mon maître d'apprentissage Jason CURE, pour sa patience, ses enseignements et la bienveillance dont il a fait preuve lors de mon temps à C2Care. Je remercie également Arnaud ELOI d'avoir accepté de prendre le relais pour le dernier semestre du cursus.

Enfin je remercie toutes les autres personnes ayant impacté positivement ma période de formation. Ma famille, Martine, Pascal et Vittorio. Je remercie également mes amis et collègues.

# Glossaire

---

Terme	Définition
<i>RV/VR</i>	Réalité virtuelle.
<i>TERV</i>	Thérapies par exposition à la réalité virtuelle.
<i>UI</i>	Interface utilisateur.
<i>Débogage</i>	Action d'identification et de suppression de bogues (ou « bugs ») dans le code d'un logiciel.
<i>2D</i>	Deux dimensions.
<i>3D</i>	Trois dimensions.
<i>IA</i>	Intelligence artificielle.
<i>Framework</i>	Ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations de tout ou d'une partie d'un logiciel.
<i>Sentiment de présence</i>	Sentiment authentique d'exister dans un monde autre que le monde physique où le corps se trouve.
<i>« Motion sickness » ou cinétose</i>	Ensemble des troubles induits par l'utilisation de la réalité virtuelle : désorientation, vertiges et nausées.
<i>SCRUM</i>	Méthode de gestion de projet agile.
<i>Sprint</i>	En agilité : plage temporelle de durée fixe, allant de 2 à 4 semaines, au cours de laquelle une partie du travail global sera effectuée.
<i>ROAM board</i>	Tableau permettant d'effectuer la gestion des risques d'un projet. ROAM : Resolved Owned Accepted Mitigated.
<i>Carte heuristique</i>	Schéma permettant de représenter visuellement des idées liées entre elles en reflétant le fonctionnement de la pensée.
<i>Moteur de jeux</i>	Logiciel principalement conçu pour le développement de jeux vidéo. Il regroupe tout ce qui a trait au fonctionnement global du jeu.
<i>Unity3D</i>	Moteur de jeux développé par Unity Technologies.
<i>Microsoft .Net</i>	Plateforme de Microsoft permettant de créer des projets pour diverses plateformes.

<i>C#</i>	Langage de programmation orientée objet commercialisé par Microsoft et destiné à développer sur la plateforme Microsoft .Net.
<i>Assets</i>	Ressources diverses centralisées dans un projet Unity.
<i>Modèle 3D</i>	Fichier contenant une représentation géométrique informatique d'un objet en trois dimensions.
<i>PNJ</i>	Personnage Non-Joueur.
<i>C2Packages</i>	Projet Unity de C2Care centralisant les éléments communs à toutes les applications de réalité virtuelle.
<i>C2Base</i>	Dossier commun à tous les projets de réalité virtuelle de C2Care, contenant notamment l'avatar virtuel.
<i>Prefab</i>	Objet dans Unity sauvegardé pour être réutilisé dans plusieurs environnements.
<i>Scène Unity</i>	Espace 3D dans lequel sont ajoutés des modèles, scripts, lumière et autres, afin de constituer un environnement virtuel dynamique.
<i>Blend Tree</i>	Composant d'Unity permettant d'effectuer un mélange entre plusieurs animations à partir de paramètres donnés.
<i>Character controller</i>	Composant d'Unity permettant de facilement déplacer et gérer les collisions d'un personnage virtuel.
<i>RigidBody</i>	Composant d'Unity permettant d'appliquer de la physique à un objet.
<i>Unity package</i>	Archive exportée depuis Unity, contenant une multitude d'assets pouvant être facilement importés dans un autre projet Unity.
<i>Cinématique inverse</i>	Ensemble des méthodes de calcul des positions et rotations d'un modèle articulaire afin d'obtenir une pose désirée.
<i>Frame</i>	Terme anglais désignant une image rendue par un moteur de jeux.
<i>Git</i>	Logiciel de gestion de versions décentralisé, permettant une collaboration avancée sur du code et des fichiers.
<i>GitHub</i>	Service web d'hébergement de projets Git.
<i>Commit</i>	Sauvegarde de l'environnement de travail à un instant t.
<i>Branche</i>	Pointeur vers un commit.
<i>Git-Flow</i>	Méthode d'utilisation de Git où des branches « feature » sont créées à partir d'un flux de développement commun.

# Table des matières

---

I.	Introduction .....	8
1.	L'entreprise C2Care.....	8
2.	Thérapies par Exposition à la Réalité Virtuelle (TERV) .....	10
3.	Présentation de mon service .....	11
4.	Notion d'avatar virtuel.....	12
II.	Problématique du PFE .....	14
III.	Etat de l'art .....	16
1.	Intérêts et impacts de l'avatar .....	16
a.	Impact sur la charge cognitive .....	16
b.	Sentiment d'incarnation.....	18
c.	Variations en fonction du dispositif de réalité virtuelle .....	19
2.	Apparence de l'avatar.....	20
a.	Degrés de représentation et identification.....	20
b.	Appropriation des mains virtuelles .....	21
c.	Flexibilité de l'avatar .....	23
3.	Systèmes d'interaction .....	25
a.	Déplacement dans l'espace .....	26
b.	Interaction avec le monde virtuel .....	28
4.	Concurrence.....	29
5.	Synthèse.....	31
IV.	Matériel et méthodes.....	33
V.	Solutions proposées .....	37
1.	Préparation du projet .....	38
a.	Initialisation du projet .....	38
b.	Mise en place de la gestion de projet .....	43
c.	Organisation du projet .....	54
d.	Études et finalisation de la préparation.....	60
2.	Conception logicielle.....	67
a.	Définition de l'environnement de travail.....	67
b.	Gestion de versions .....	68
c.	Documentation et bonnes pratiques de codage .....	70
d.	Mise en place de la stratégie de test .....	70

e.	Spécification des user stories.....	72
f.	Architecture logicielle .....	72
g.	Structure hiérarchique du nouvel avatar dans Unity.....	73
h.	Utilisation de l'avatar par les développeurs .....	75
3.	Réalisation du nouvel avatar .....	76
b.	Création de la base de l'avatar.....	77
c.	Calibrage et positionnement.....	85
d.	Système d'interactions.....	89
e.	Variations visuelles de l'avatar.....	95
f.	Fonctionnalités mineures et extensions .....	98
g.	Communication avec le contrôleur my.c2.care .....	101
4.	Déploiement .....	103
a.	Intégration de l'avatar au flux de développement continu .....	103
b.	Stratégie de déploiement.....	104
VI.	Résultats et discussions .....	106
VII.	Conclusions.....	108
A.	Document d'évaluation de l'avatar existant pour faire l'objet d'une refonte .....	111
B.	Étude de l'avatar existant .....	112
C.	Carte heuristique des fonctionnalités du nouvel avatar .....	114
D.	« ROAM board » réalisé pour la gestion des risques.....	115
E.	Documentation d'un des risques du projet.....	116
F.	Budget détaillé prévisionnel .....	117
G.	Documentation technique : normes, standards et organisation .....	118
H.	Carte heuristique sur la structure de l'avatar.....	120
I.	Diagramme de classes UML sur la structure de code globale de l'avatar.....	121
J.	Diagramme de classes UML final .....	122
K.	Spécification d'une des user stories .....	123

# I. Introduction

---

## 1. L'entreprise C2Care

Depuis 2015, C2Care a pour objectif de développer et de mettre à la disposition des professionnels de santé, des logiciels thérapeutiques basés sur la réalité virtuelle. Les thérapies par exposition à la réalité virtuelle offrent la possibilité de confronter graduellement les patients à leurs phobies, dépendances, troubles du comportement alimentaire, troubles fonctionnels et vestibulaires.

En effet, à travers le casque de réalité virtuelle, les patients sont transportés dans des environnements immersifs dont les scénarios et protocoles ont été élaborés par un comité de recherche médicale composé de médecins universitaires, psychiatres, psychologues et kinésithérapeutes.

Les bénéfices de ces nouvelles thérapies sont nombreux : meilleure graduation des expositions, amélioration de la praticité (aucun déplacement en dehors du cabinet), possibilité de répéter des scènes à l'infini, diminution des incidents de sensibilisation et de tout problème de confidentialité.

Il est également essentiel de permettre d'effectuer un suivi régulier pour chaque séance et d'accompagner individuellement chacun des patients, pour cela, la technologie de C2Care brille par son adaptabilité. De nombreux environnements divers sont disponibles et personnalisables depuis l'interface my.c2.care.



Figure 1 : Les 10 principaux logiciels de réalité virtuelle de la suite C2Care

Aujourd'hui C2Care propose la gamme de logiciels thérapeutiques en réalité virtuelle la plus importante du marché, reconnus *Medical Device* de Classe 1. On compte plus de 10 applications dont 176 environnements 3D, pour des totaux de 2000 clients, 22 000 patients traités et 225 000 expositions lancées.

## 2. Thérapies par Exposition à la Réalité Virtuelle (TERV)

C2Care développe des logiciels utilisés dans le cadre de thérapies par réalité virtuelle. Les bénéfices de ce type de traitement ont été évoqués précédemment mais il est intéressant de présenter davantage le concept car il définit l'utilisation qui est faite des solutions développées par l'entreprise.

La thérapie par réalité virtuelle consiste en l'utilisation de plateformes immersives (visiocasques, ordinateur, traqueur de mouvement, etc.) et d'environnements virtuels créés par ordinateur pour traiter des individus souffrant de maladies, de troubles physiques ou de troubles mentaux.



*Figure 2 :* Exemple de thérapie par réalité virtuelle, avec à gauche le praticien dirigeant la séance et à droite le patient exposé à son trouble via la réalité virtuelle.

Lors d'une séance de TERV, le praticien utilise les logiciels de réalité virtuelle pour exposer le patient à sa phobie, son addiction, ou autres. Le thérapeute est toutefois pleinement impliqué dans la session car l'outil n'est qu'un complément qui n'a pas pour vocation de le remplacer.

Pendant la séance, le patient peut être physiquement assis ou debout. Dans l'environnement virtuel, il peut se déplacer si cela est permis et interagir avec les environnements. Dans la plupart des cas de TERV, le praticien a la possibilité de contrôler l'exposition avec une interface logicielle lui étant dédiée. Il peut s'agir d'un site web se connectant par internet au casque du patient, comme dans le cas de C2Care avec l'espace web *my.c2.care*.

Pour réussir une séance de TERV, il faut d'une part que le thérapeute dispose des outils nécessaires pour contrôler et observer l'exposition de son patient. D'une autre part, le niveau d'immersion du patient dans la réalité virtuelle doit être optimal. Ce second point sera abordé à maintes reprises tout au long de ce mémoire car il est au cœur du projet présenté.

### 3. Présentation de mon service

Globalement, l'organisation de C2Care est divisée en deux pôles majeurs :

- Le pôle commerce : responsable des aspects liés à la vente des logiciels.
- Le pôle développement : responsable de la mise à jour des applications de réalité virtuelle et des sites web de l'entreprise. Il s'agit du pôle dont je fais partie.

Le pôle développement est divisé lui-même en deux autres pôles : un pôle web et un pôle réalité virtuelle, auquel j'appartiens. Ce service est composé de développeurs, ingénieurs et graphistes spécialisés en Unity3D donnant vie aux environnements proposés aux professionnels de santé pour le traitement de leurs patients. L'équipe présentée est encadrée par un lead développeur ainsi que le président de l'entreprise.

Mes principales missions en tant qu'apprenti ingénieur au sein du service technique de C2Care ont été le développement de nouveaux environnements 3D, ou l'évolution d'environnements existants, en coordination avec les autres développeurs et graphistes de l'entreprise.

## 4. Notion d'avatar virtuel

L'avatar virtuel, sujet de ce mémoire, représente une forme d'interface entre l'utilisateur et l'expérience de réalité virtuelle dans laquelle il est immergé. Ce dispositif permet dans un premier temps de représenter virtuellement l'intégralité, ou des parties, du corps de l'utilisateur.



*Figure 3 : À gauche, un exemple de représentation minimaliste d'une main en réalité virtuelle. À droite, les bras de l'avatar virtuel d'un utilisateur tels qu'il les voit dans le jeu « Rumble ».*

Dans un second temps, l'avatar permet à l'utilisateur d'interagir avec les éléments qui composent le monde virtuel dans lequel il se trouve : attraper des objets, se déplacer, etc.

L'avatar existant de C2Care, faisant l'objet d'une refonte dans le cadre de ce projet, mérite déjà une brève introduction. Bien entendu, il sera amplement détaillé dans la suite du mémoire car il s'agit de l'élément central du projet présenté.

Pour ce qui est de son apparence, il existe actuellement sous deux formes : homme et femme. Il est donc possible de paramétrer le sexe de l'avatar, mais aussi sa taille et sa couleur de peau. L'avatar ne dispose pas de tête pour ne pas obstruer la vue de l'utilisateur. Un mode permettant de cacher entièrement le corps de l'avatar est également disponible.



*Figure 4 : Apparences homme et femme de l'avatar existant de C2Care.*

D'un point de vue interactions, l'avatar permet de :

- Répliquer le positionnement des membres du corps virtuel par rapport aux membres réels traqués par le casque et les manettes, si applicable.
- Se déplacer de manière réaliste : système de marche.
- Appuyer physiquement sur des boutons avec l'utilisation des mains virtuelles.
- Attraper et tenir en main des objets préalablement configurés à cet effet.
- Lancer les objets pris en main.

L'avatar virtuel, par la polyvalence de son utilisation dans la VR, est un élément essentiel dans les expositions à la réalité virtuelle. Par exemple, pour qu'un patient affronte sa phobie avec un maximum d'efficacité, il doit se sentir réellement présent dans l'environnement virtuel où il est immergé. De plus, l'avatar est un élément commun à tous les environnements 3D des applications de C2Care. Il a un impact direct sur le sentiment d'immersion des utilisateurs exposés, clé au succès des thérapies par réalité virtuelle.

## II. Problématique du PFE

---

L'avatar virtuel existant de C2Care est ancien. En effet, il date de 2016, peu après la création de l'entreprise. La réalité virtuelle a beaucoup évolué en ces quelques années : de nouveaux casques plus sophistiqués ont fait leur apparition sur le marché, tout comme de nouveaux systèmes d'interactions ou encore de normes ont été introduites.

Pour l'entreprise, il est devenu clair que l'avatar virtuel des applications C2Care doit évoluer afin de rester compétitif et à la pointe de la technologie. En effet, parmi les points négatifs de l'avatar actuel on compte : un déplacement peu fluide, attraper des objets nécessite une longue configuration pour chaque objet à saisir, les boutons utilisés pour chaque action ne sont pas clairement définis, la simulation des bras et jambes est peu réaliste en comparaison avec les nouvelles normes établies.

Faire évoluer l'avatar virtuel signifie en effectuer une refonte totale car l'existant est trop ancien et les nombreuses petites modifications réalisées à travers les années ont grandement endommagé la qualité du code.

Ainsi se pose la problématique suivante :

***Comment effectuer une refonte de l'avatar virtuel permettant d'améliorer les interactions de l'utilisateur avec les environnements de réalité virtuelle, tout en augmentant le sentiment de présence généré ?***

La refonte de l'avatar présente donc essentiellement de la valeur ajoutée pour les utilisateurs des logiciels de réalité virtuelle de C2Care. Toutefois, les avantages induits par ce travail pour l'équipe de développement de C2Care sont également conséquents : propreté du code, adaptabilité de l'avatar aux différents cas d'utilisation, fiabilité globale du dispositif.

Par le moyen d'une synthèse bibliographique, je présenterai dans un premier temps la notion d'avatar dans les expériences de réalité virtuelle dans l'optique d'identifier les éléments essentiels à la proposition d'une solution à la problématique de ce mémoire.

Les principaux composants logiciels employés lors du projet seront présentés afin d'introduire ensuite la présentation complète du processus de refonte de l'avatar mis en œuvre à C2Care. Chaque phase majeure du travail sera détaillée pour permettre une éventuelle reproduction de la solution.

Enfin, une analyse critique des résultats obtenus par rapport à la problématique initiale du projet permettra d'évaluer la solution tout en la resituant dans un cadre plus large.

### III. Etat de l'art

---

L'état de l'art scientifique présenté dans cette partie du mémoire propose la synthèse d'une étude bibliographique sur la notion d'avatar en réalité virtuelle. Ce travail théorique abordera dans un premier temps les intérêts et impacts de l'avatar, avant d'évoquer ses aspects fondamentaux : l'apparence et le système d'interactions induit. Enfin une étude concurrentielle sera présentée et une synthèse permettra de diriger les travaux du projet de ce mémoire.

#### 1. Intérêts et impacts de l'avatar

La notion d'avatar virtuel étant bien définie, il est essentiel de démontrer l'importance que présente l'avatar dans les expériences de réalité virtuelle. Cette partie s'appuiera sur des études dans la littérature scientifique afin de mettre en avant l'utilité d'un tel dispositif en réalité virtuelle, avant d'aborder ses spécificités en détail.

Cela permettra également de faire état des impacts positifs que le nouvel avatar de C2Care devra générer et les contraintes ou aspects négatifs qui devront être pris en compte et anticipés.

##### a. Impact sur la charge cognitive

La charge cognitive correspond à la quantité de ressources cognitives investies par un individu lors de la réalisation d'une tâche.

Dans les expériences de réalité virtuelle, la charge cognitive est souvent élevée car les utilisateurs peuvent se retrouver dans des situations où une forte concentration est demandée.

C'est le cas dans bon nombre de jeux en réalité virtuelle, mais aussi dans certaines applications de C2Care. En effet, une de ces applications est C2Brain, conçue pour offrir la possibilité de stimuler le fonctionnement cognitif des patients à travers des jeux de mémoire.

Il est ainsi d'autant plus essentiel de prendre en compte cet aspect dans la réalisation du nouvel avatar.

L'étude [1] soumet des sujets à un exercice de mémoire en réalité virtuelle, distinguant présence et absence de l'avatar sur les résultats obtenus. Le sentiment de présence est défini comme le sentiment authentique d'exister dans un monde autre que le monde physique où le corps se trouve :

*« La réalité virtuelle ne fait que généraliser ce principe qui consiste à offrir un produit privé de sa substance, privé de son noyau de réel, de résistance matérielle : tel le café décaféiné qui a le goût et l'odeur du café sans en être vraiment, la réalité virtuelle est une réalité qui ne l'est pas vraiment. »,* comme l'écrit Slavoj Žižek, philosophe et chercheur slovène, en 2002.

À l'issue de cette étude, les participants disposant d'un avatar et pouvant bouger leurs mains ont obtenu de meilleurs résultats sur l'exercice de mémoire que ceux ne disposant pas d'avatar. Une observation supplémentaire indique que l'absence d'avatar a pour cause de décourager les sujets à bouger leurs mains.

Cette étude démontre l'impact de l'avatar sur la charge cognitive en réalité virtuelle.

Il faut toutefois prendre en compte le fait qu'elle a été effectuée sur des sujets jeunes alors qu'une importante partie de la patientèle exposée aux applications de C2Care, dont C2Brain, est composée de personnes âgées.

Il est donc pertinent d'inclure un avatar dans les environnements de réalité virtuelle demandant une charge cognitive modérée ou importante. Cela est applicable à certains environnements virtuels de C2Care rentrant dans cette catégorie et qui ne comportent pas encore d'avatar.

## b. Sentiment d'incarnation

Lorsqu'un utilisateur porte un casque de réalité virtuelle et s'immerge dans un univers où il incarne un personnage autre que lui-même, la question du « sentiment d'incarnation » se pose :

*Quel effet cela fait-il de posséder, de contrôler et d'être à l'intérieur d'un corps ?*

J'ai défini précédemment le sentiment de présence. Celui-ci est fortement lié au sentiment d'incarnation, faisant l'objet de la publication [2]. Ce document apporte une définition à la notion de sentiment d'incarnation et élabore sur son application dans les expériences de réalité virtuelle.

Mel Slater, chercheur de l'Université de Barcelone et figure importante de la recherche dans le domaine de la réalité virtuelle apporte la définition suivante :

*« Le sentiment d'incarnation d'un être humain vers un corps B est le sentiment qui émerge lorsque les propriétés de B sont traitées comme s'il s'agissait des propriétés de son propre corps biologique. »*

La structure sous-jacente du sentiment d'incarnation est elle-même composée des trois éléments suivants :

- **Sentiment d'auto-localisation** : Volume déterminé dans l'espace, où l'on se sent situé.
- **Sentiment de pouvoir** : Suis-je en capacité d'agir ?
- **Sentiment de possession du corps** : L'autoattribution d'un corps qui n'est pas le sien.

Le sentiment d'incarnation a un impact considérable sur le sentiment de présence généré par l'expérience. Les applications de C2Care confrontant des patients à leurs phobies par exemple, se doivent d'en atteindre un niveau élevé afin d'obtenir les meilleurs résultats thérapeutiques possibles.

Le lien entre l'avatar en réalité virtuelle et le sentiment d'incarnation est développé dans l'étude [3]. Cette étude relève que la configuration de l'avatar ainsi que son apparence n'ont pas besoin d'atteindre un état de réalisme parfait pour générer un fort sentiment d'incarnation. Cette notion sera développée dans la suite de cet état de l'art.

### c. Variations en fonction du dispositif de réalité virtuelle

Au fil des années, les dispositifs de réalité virtuelle ont beaucoup évolué. De nombreux casques existent sur le marché et se distinguent par certaines caractéristiques techniques.

C2Care assure la compatibilité de ses programmes avec la majorité des principaux casques de réalité virtuelle du marché. En effet, plusieurs versions des logiciels sont disponibles et les spécificités de chaque plateforme supportée sont prises en considération. La réalisation du nouvel avatar devra donc prendre en compte les différences entre celles-ci.

La majeure divergence concerne la présence ou l'absence de manettes pour chaque main.



*Figure 5 : À gauche, l'Oculus Go. Comporte une seule manette dont seule la rotation est traquée. À droite, l'Oculus Quest. Comporte une manette pour chaque main de l'utilisateur.*

*En plus de la rotation, la position des manettes est traquée dans l'espace.*

Il est donc important de faire preuve de flexibilité autant dans la représentation corporelle de l'avatar que pour le système d'interaction qui sera mis en place.

Les parties suivantes de l'état de l'art se pencheront successivement sur ces deux aspects.

## 2. Apparence de l'avatar

Un des aspects les plus importants pour un avatar en réalité virtuelle concerne son apparence, c'est-à-dire la représentation virtuelle du corps de l'utilisateur. Cette notion est directement liée aux points clés présentés dans la partie précédente de cet état de l'art.

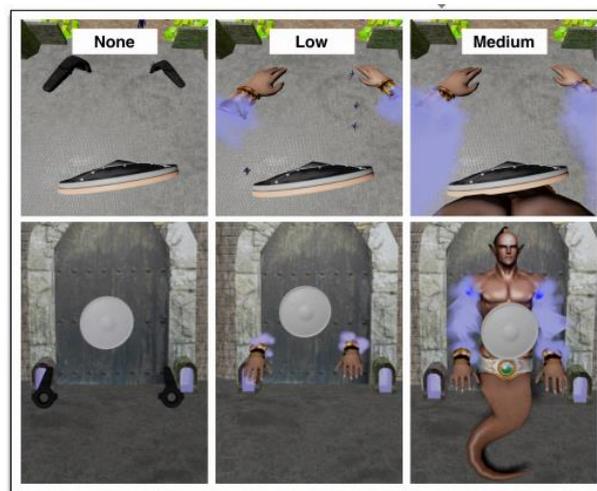
Ainsi, cette partie s'intéressera aux aspects propres à l'apparence de l'avatar en évoquant ses différents degrés de représentation, l'appropriation des mains virtuelles par l'utilisateur et l'importance de la flexibilité dont il doit disposer.

### a. Degrés de représentation et identification

L'avatar virtuel a pour rôle de représenter entièrement ou partiellement le corps de l'utilisateur. Il doit également se servir des données de position et rotation du casque et des manettes, si applicable, pour animer ce corps virtuel.

Afin de concevoir un avatar générant un niveau de présence élevé et donnant le sentiment d'incarnation, il est intéressant de connaître ses différents degrés de représentation ainsi que leurs effets respectifs sur ces sentiments.

L'étude [4] expose des sujets à 3 différents niveaux de représentation de l'avatar dans le cadre d'un jeu d'action en réalité virtuelle.



*Figure 6 : Trois niveaux de représentation de l'avatar dans le jeu iGod [4].*

Contrairement aux effets attendus par les chercheurs, les différents degrés de représentation de l'avatar n'ont pas eu d'impact significatif sur le sentiment d'incarnation. Toutefois, l'étude confirme que même dans un état minimaliste, disposer d'un avatar augmente de manière significative la sensation de présence pour l'utilisateur.

Cette étude diverge du cas de C2Care sur le point qu'elle a été effectuée sur un jeu vidéo. Les auteurs de la publication indiquent que dans le cas de jeux à forte dynamique, la concentration des utilisateurs sur leur tâche a pour effet de leur faire oublier partiellement leur corps.

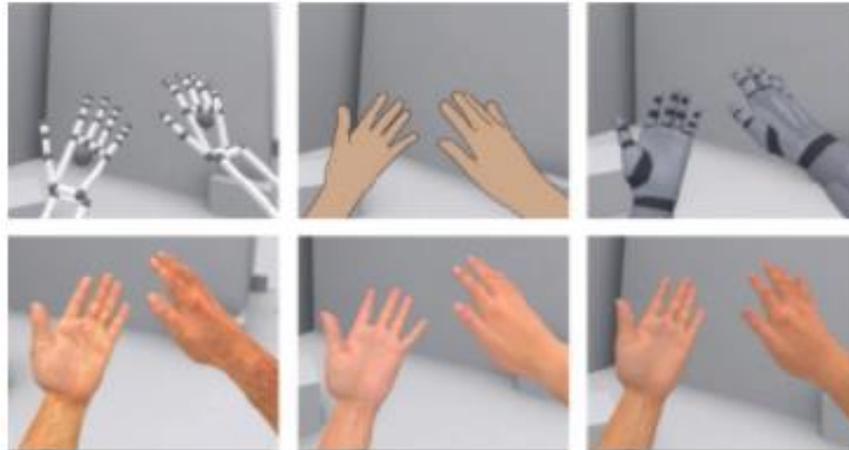
Globalement, cette étude permet quand même de rassurer sur le fait que l'avatar de C2Care pourra proposer plusieurs niveaux de représentation de l'avatar adaptés aux environnements sans pour autant perdre en présence générée. Par exemple, pour un environnement où le patient prend l'avion, une représentation entière de l'avatar est intéressante alors que pour un environnement de relaxation abstrait, la seule représentation des mains suffit.

## **b. Appropriation des mains virtuelles**

Dans la réalité virtuelle moderne, c'est-à-dire pour les casques de réalité virtuelle parus depuis 2016 et disposant de deux manettes, les mains de l'utilisateur sont virtuellement omniprésentes. En effet, les interactions diverses et immersives que celles-ci permettent sont des facteurs clés différenciant la réalité virtuelle de tout autre type d'expérience.

Pour continuer sur la thématique de l'identification et la représentation de l'avatar, il est essentiel de se pencher sur l'appropriation des mains de l'avatar. Cette sous-partie donnera des pistes qui permettront de guider la réalisation de cet aspect pour le futur avatar de C2Care.

Les applications de réalité virtuelle représentent les mains de l'utilisateur de manière plus ou moins réaliste, en fonction du sexe de l'utilisateur ou de ce qu'il est dans l'expérience : un robot par exemple. L'étude [5] s'intéresse à l'effet du sexe de l'utilisateur sur la perception de différentes représentations virtuelles de ses mains.



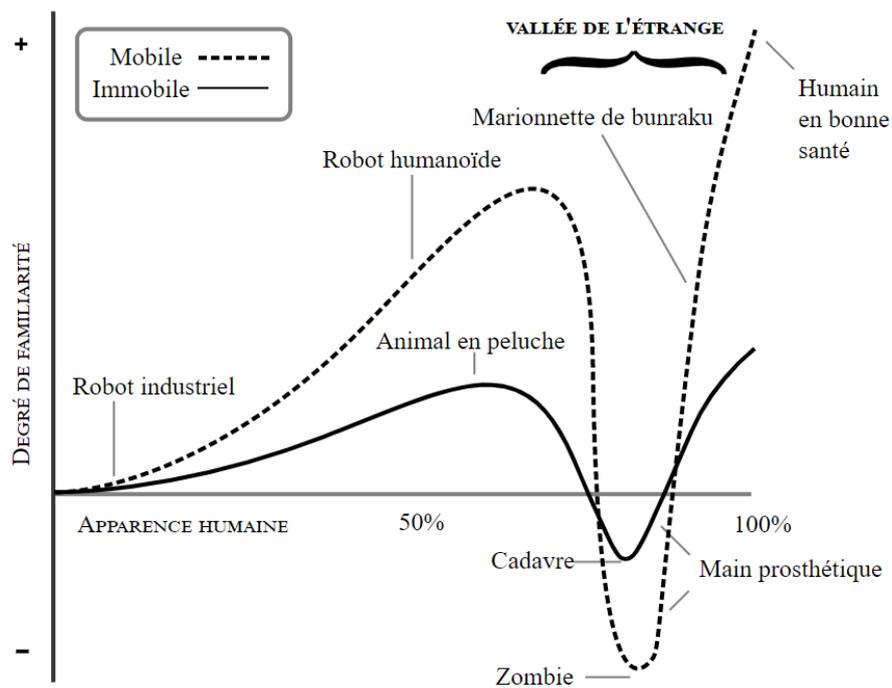
*Figure 7 : Six représentations de mains, respectivement abstraites, cartoon, robot, homme, femme et androgyne [5].*

Les résultats de cette étude donnent des pistes intéressantes pour la conception du nouvel avatar de C2Care. Les patients étant de sexe différent, l'issue de l'étude confirme l'importance de considérer la diversité des utilisateurs lors de la conception d'expériences de réalité virtuelle.

En effet, les femmes perçoivent un niveau de présence plus faible avec des mains d'homme alors que ceux-ci ne ressentent pas le même effet avec des mains de femmes. Les mains non humaines ont été mal reçues, particulièrement par les hommes. Offrir à l'utilisateur la possibilité d'avoir des mains virtuelles étant adaptées à son sexe offre un sens de possession du corps virtuel accru.

Cela confirme l'hypothèse que le nouvel avatar virtuel devra posséder une représentation homme ainsi qu'une représentation femme, comme c'est le cas avec l'avatar de C2Care existant.

Pour ce qui est du niveau de détail graphique des mains, un rendu réaliste est visé. Cependant, l'étude mentionnée met en garde sur le phénomène d'*Uncanny Valley* (ou Vallée de l'étrange). Il s'agit d'une hypothèse stipulant que les répliques anthropomorphiques se rapprochant trop de l'apparence et du comportement d'un vrai humain provoquent une réponse de rejet parmi les observateurs humains.



*Figure 8 : Uncanny Valley : réaction émotionnelle de sujets humains à l'apparence humaine d'un robot.*

Ce phénomène s'applique à la représentation graphique non seulement des mains de l'avatar mais de son intégralité. Ainsi, afin d'éviter cela pour le nouvel avatar, il faudra que sa représentation graphique trouve un juste milieu entre corps virtuel et réalisme.

### c. Flexibilité de l'avatar

Dans la sous-partie précédente, j'ai évoqué la notion de flexibilité de l'avatar virtuel en concluant qu'il doit disposer d'une représentation corporelle masculine et une représentation

féminine. Prendre en compte l'adaptabilité de l'avatar dans son processus de conception est primordial.

La publication de Mel Slater [6] synthétise les dernières trouvailles dans le domaine des neurosciences portant sur la flexibilité de notre représentation corporelle offerte par le cerveau. D'après les auteurs, les illusions multisensorielles fonctionnent bien en réalité virtuelle immersive : le cerveau se laisse influencer.

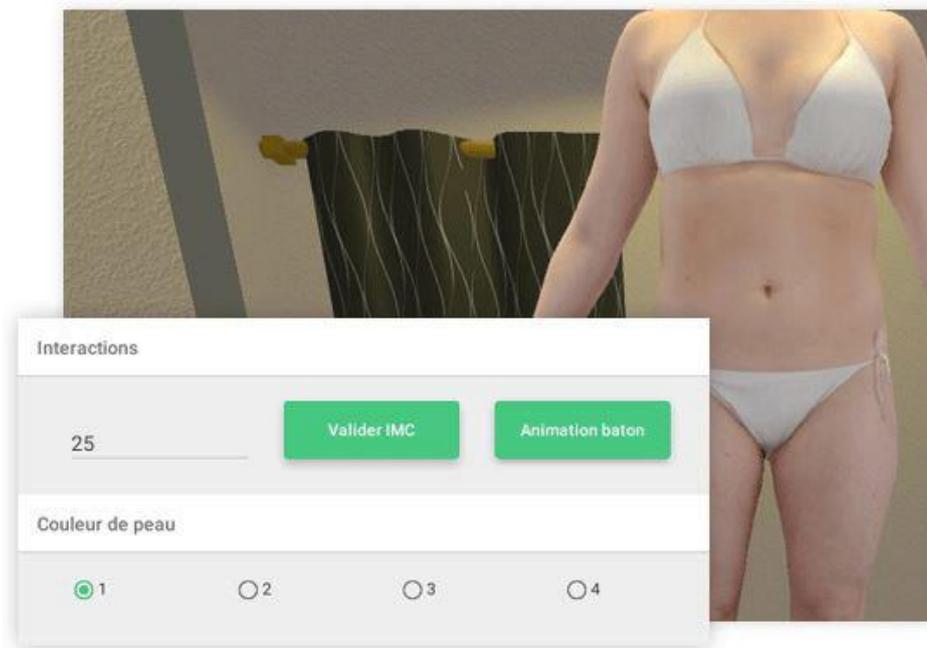
L'étude s'intéresse à la réaction d'utilisateurs à différents types de corps. Par exemple, mettre un adulte dans un avatar d'enfant, ou alors positionner l'utilisateur dans un corps de sexe ou d'ethnie différente. Il est observé que cette expérience permet aux utilisateurs de vivre le monde d'un autre point de vue et ouvre de nouvelles perspectives d'applications de la réalité virtuelle.

C2Care développant des applications pour le domaine de la santé et spécialisée dans de nombreux de ses sous-domaines, il peut être intéressant de profiter de cette opportunité afin de mieux répondre aux attentes des futurs projets thérapeutiques. Ainsi, en plus du sexe paramétrable, l'avatar devrait également proposer des options d'ethnie, de taille et de corpulence.

Pour prendre un exemple concret applicable à C2Care, il y a l'application de traitement des troubles alimentaires compulsifs : C2Nutri.

Un des environnements de cette application permet à l'utilisateur de se regarder dans un miroir alors que le thérapeute ajuste la corpulence de son avatar virtuel. Actuellement, deux avatars distincts et indépendants de l'avatar virtuel principal de C2Care sont utilisés.

Disposer d'un avatar à l'apparence flexible permettrait donc d'unifier et améliorer cette partie du logiciel.



*Figure 9 : Environnement virtuel « Dysmorphophobie Femme » de l'application C2Nutri.*

### 3. Systèmes d'interaction

Le second aspect fondamental concernant les avatars en réalité virtuelle est le système d'interaction qu'il apporte afin de permettre à l'utilisateur d'exister dans le monde virtuel et d'interagir avec celui-ci.

Cette troisième partie de l'état de l'art s'intéressera aux aspects fondamentaux liés aux interactions en réalité virtuelle. Ainsi, une étude des systèmes de déplacement dans l'espace virtuel sera présentée et les types d'interactions avec cet espace sont également confrontés dans l'optique de guider la conception du nouvel avatar de C2Care.

## a. Déplacement dans l'espace

Dans les expériences de réalité virtuelle, le déplacement dans l'espace est une des formes d'interactions fondamentales, qui doit presque toujours être implémentée. Pour certaines applications, par exemple une promenade immersive dans un lieu historique, il se peut même qu'il s'agisse de l'unique interaction offerte à l'utilisateur.

Avant de présenter les types de déplacement existants en réalité virtuelle, il est important de définir la notion de « motion sickness » ou cinétose. Il s'agit de l'ensemble des troubles induits par l'utilisation de la réalité virtuelle et se caractérise le plus souvent par une désorientation, des vertiges et des nausées. Le déplacement est la principale source de cinétose en réalité virtuelle : il y a discordance entre la perception visuelle de déplacement dans l'espace virtuel et celle du système vestibulaire.

L'étude [7] fait état des systèmes de déplacements existants et les compare en se basant sur les résultats obtenus auprès de ses sujets.

L'article distingue trois principaux types de déplacements en réalité virtuelle :

- **Marche sur place** : L'utilisateur marche sur place, ce qui est capté par le dispositif de réalité virtuelle et déclenche le déplacement de l'avatar dans la direction du regard.



*Figure 10 : Utilisateur se déplaçant avec le système de marche sur place [7].*

- **Déplacement analogique** : Déplacement similaire aux jeux vidéo sur consoles : l'utilisateur peut se déplacer dans toutes les directions à l'aide d'un stick analogique.



*Figure 11 : Déplacement analogique avec une manette du casque HTC Vive [7].*

- **Téléportation** : L'utilisateur maintient un bouton de la manette enfoncé et vise une zone dans l'espace virtuel. Une fois le bouton relâché, l'avatar est téléporté à la zone visée.



*Figure 12 : Visée d'une zone vers laquelle se téléporter avec une manette du casque HTC Vive [7].*

Les résultats de l'étude ont démontré que la technique de marche sur place était la plus immersive pour les utilisateurs, mais également la plus inconfortable d'un point de vue psychophysique. Le déplacement analogique est perçu comme le plus facile à prendre en main par les utilisateurs mais peut causer un niveau élevé de cinétose. La téléportation est considérée comme une méthode de déplacement efficace. Cependant, les « sauts » causés par la téléportation ont tendance à briser l'immersion des utilisateurs.

L'étude de ces trois systèmes de déplacement permet d'orienter le choix à effectuer pour le nouvel avatar de C2Care. Il est à noter que le système de déplacement de l'avatar existant est le déplacement analogique. Ceci sera à prendre en compte lors de la phase de conception du projet.

## **b. Interaction avec le monde virtuel**

En plus de pouvoir se déplacer dans un univers virtuel, l'utilisateur doit pouvoir interagir avec celui-ci. Dans les environnements des applications de C2Care, il doit pouvoir appuyer sur des boutons physiques (boutons d'ascenseur par exemple), attraper des objets et éventuellement les lancer, mais aussi interagir avec des menus. L'avatar virtuel doit donc proposer un système d'interactions adapté.

Mark R. Mine, avec son étude [8], identifie trois types d'interactions avec le monde virtuel :

- **Interactions directes de l'utilisateur** : Cela inclut l'utilisation du suivi des mains, la reconnaissance des gestes, le pointage, la direction du regard, etc. pour spécifier les paramètres de la tâche d'interaction. Il y a une correspondance naturelle entre l'action réelle de l'utilisateur et l'action résultante dans le monde virtuel.
- **Contrôles physiques** : Cela comprend les boutons, curseurs, manettes, volants et autres formes de dispositifs d'entrées. L'utilisation de contrôles physiques pour interagir avec un monde virtuel, comme un volant dans un simulateur de conduite,

peut grandement améliorer le sentiment de présence de l'utilisateur. Cependant, ils peuvent être difficiles à trouver lorsqu'on porte un casque de réalité virtuelle.

- **Contrôles virtuels** : Presque toute interaction imaginable peut être implémentée sous forme de contrôle virtuel. Cette grande flexibilité est le principal avantage de ce type d'interaction. Parmi les inconvénients, il y a le manque de retour d'information haptique et la difficulté générale d'interaction avec un objet virtuel. Il est essentiel de prêter une attention particulière à la conception du contrôle virtuel.

Pour C2Care, certaines actions telles que saisir physiquement des bouteilles d'alcool dans l'application C2Addict ne sont possibles qu'avec un mélange d'interactions directes et contrôles physiques. L'utilisateur devra déplacer sa main vers une bouteille comme pour la saisir réellement, puis appuyer sur un bouton de la manette pour signaler l'action de « prendre la bouteille ».

De plus, comme évoqué précédemment, les différents casques de réalité virtuelle supportés par C2Care n'offrent pas tous les mêmes systèmes d'entrées.

Ainsi, l'avatar devra certainement être conçu en intégrant ces trois types d'interactions. Pour chaque type d'actions de l'avatar, il faudra choisir le type d'interaction ou l'association adaptée tout en prenant en compte le fait que certains casques supportés par les applications ne disposent, par exemple, que d'une seule manette non traquée dans l'espace.

## 4. Concurrence

Bien que l'utilisation de la réalité virtuelle dans le domaine de la santé ne soit encore qu'un marché de niche, C2Care n'est pas la seule société actrice de cette révolution. Cependant, il est important de rappeler qu'elle se positionne en tant que leader du marché, notamment dans le domaine du traitement des phobies.

Dans l'optique de conclure cet état de l'art, quelques exemples d'approches par rapport à l'avatar chez la concurrence de C2Care vont être présentés et analysés.

La société **Psious** développe de nombreux environnements thérapeutiques en réalité virtuelle adaptés aux conditions de chaque patient. La majorité de ces environnements ne sont pas interactifs et ne présentent pas d'avatar virtuel. Certaines scènes font exception et proposent une représentation corporelle de l'utilisateur. Cependant, cet avatar est préalablement animé et ne permet aucune interaction de la part de l'utilisateur : les mains ne sont pas traquées, le mouvement des bras de l'avatar est animé par l'application.

La société **InVirtuo** propose plusieurs logiciels de réalité virtuelle (3D et interactifs) aux professionnels de la santé dans les domaines de l'anxiété sociale, généralisée, le stress post-traumatique, le trouble obsessionnel-compulsif et les phobies. InVirtuo a fait le choix d'utiliser un avatar uniquement dans les scènes où une interaction physique était nécessaire : boutons d'une machine à sous dans un casino. Il est visuellement représenté de manière minimaliste avec le seul rendu des mains. Certains des environnements 3D, bien que ne présentant pas d'avatar virtuel, utilisent le déplacement analogique pour permettre à l'utilisateur d'explorer les environnements.

La startup **KineQuantum** commercialise des jeux vidéo de rééducation en réalité virtuelle. Contrairement aux sociétés présentées précédemment et à C2Care, celle-ci se spécialise dans la kinésithérapie ce qui lui permet de se positionner en tant que 1<sup>er</sup> dispositif médical de rééducation fonctionnelle en réalité virtuelle. C2Care propose sa propre solution à cet effet avec le logiciel C2Physio. L'avatar virtuel est omniprésent dans les environnements KineQuantum, sous une forme minimaliste : seules les mains sont représentées. Afin de déclencher les mouvements nécessaires pour les exercices de rééducation, le système d'interactions directes de l'utilisateur est utilisé.

**HypnoVR**, tout comme KineQuantum, propose une solution logicielle spécialisée cette fois-ci dans l'hypnose médicale par réalité virtuelle. Les environnements sont contemplatifs et ne proposent aucune interaction. L'utilisateur se laisse déplacer automatiquement dans des lieux divers développés par la société. Il n'y a pas d'avatar dans les environnements.

Il est intéressant de constater qu'une partie de la concurrence n'a que peu implémenté l'avatar virtuel dans leurs applications. L'importance d'un tel composant dans les expériences

de réalité virtuelle ayant été démontrée plus tôt dans cet état de l'art, l'avatar virtuel de C2Care peut présenter un réel avantage concurrentiel. Une autre conclusion pouvant être tirée de cette étude concurrentielle est que l'avatar semble essentiel dans les environnements à forte interactivité, tels que ceux de rééducation par exemple. Dans des environnements contemplatifs où l'utilisateur n'est pas actif, la présence d'un avatar devient plus facultative.

## 5. Synthèse

Il est possible de tirer des conclusions de cet état de l'art par rapport à l'avatar de C2Care faisant l'objet d'une refonte.

L'avatar virtuel a un impact considérable sur la charge cognitive, cela signifie que son implémentation dans les environnements de C2Care où un certain travail mental est demandé, sera nécessaire. Insister sur la création du sentiment d'incarnation, lié au sentiment de présence, pour les utilisateurs permettra l'obtention de meilleurs résultats thérapeutiques. L'avatar devra bien prendre en compte les différents casques de réalité virtuelle supportés par les applications de C2Care.

D'un point de vue visuel, l'avatar devra représenter le corps de l'utilisateur en faisant preuve d'une grande flexibilité. Pour ce qui est des degrés de représentation de l'avatar, il semble pertinent d'avoir les deux suivants : minimaliste avec la seule représentation des mains, et complète avec un rendu animé procéduralement du corps entier. En effet, l'étude concurrentielle a permis d'observer la nécessité de certains degrés de représentation de l'avatar en fonction du type d'expériences. L'aspect graphique visé pour le corps de l'avatar sera réaliste, mais pas de manière excessive afin d'éviter le phénomène d'*Uncanny Valley*. L'importance de la flexibilité de l'avatar démontrée, celui-ci devra permettre le paramétrage du sexe, de la taille, la corpulence ainsi que la couleur de peau.

Pour ce qui est des interactions, le déplacement dans l'espace se fera avec le système de déplacement analogique. En effet, il s'agit du système de déplacement actuellement

implémenté et connu de la clientèle de C2Care. Les autres types de déplacement ne proposent pas d'avantages suffisamment importants pour imposer ce changement aux utilisateurs habitués. L'interaction avec le monde virtuel se fera idéalement avec les 3 types d'interactions présentés : interactions directes de l'utilisateur, contrôles physiques et virtuels. La phase de conception de l'avatar permettra d'associer un type de contrôle, ou une combinaison de types, à une interaction de l'avatar.

Ainsi, cette refonte de l'avatar permettra de mettre à jour ce composant essentiel de la suite d'applications de C2Care. Celui-ci contrairement à l'avatar précédent, sera conçu en accord avec une étude scientifique, et offrira à l'entreprise un avantage concurrentiel non négligeable.

## IV. Matériel et méthodes

---

La majeure partie de la réalisation du projet présenté dans ce mémoire sera effectuée au sein de l'environnement **Unity3D**, utilisé pour le développement de tous les logiciels de réalité virtuelle de C2Care. Développé par Unity Technologies, il s'agit de l'un des moteurs de jeux les plus répandus dans l'industrie du jeu vidéo.

Un **moteur de jeu** est un logiciel principalement conçu pour le développement de jeux vidéo. Il regroupe tout ce qui a trait au fonctionnement global du jeu, notamment l'ensemble des moteurs spécialisés nécessaires pour créer et développer un jeu vidéo :

- **Moteur graphique** : composant logiciel responsable du rendu graphique 2D ou 3D d'une application en interagissant avec la carte graphique d'un ordinateur.
- **Moteur son** : ensemble de code exécutable dans un programme informatique qui permet de gérer la partie audio du programme.
- **Moteur de physique** : bibliothèque logicielle indépendante appliquée à la résolution de problèmes de la mécanique classique, telles que les collisions, les chutes de corps, l'application de forces, la cinétique, etc.
- **Moteur d'intelligence artificielle** : ensemble d'outils logiciels permettant de construire un système d'intelligence artificielle.
- De nombreux autres composants tels qu'un moteur d'UI, débogage, animation, réseau, etc.

L'ensemble forme un simulateur en temps réel souple qui reproduit les caractéristiques des mondes imaginaires dans lesquels se déroulent les jeux.



*Figure 13 : Interface d'Unity3D pour un projet exemple.*

À l'image de la majorité des moteurs de jeu, Unity est multiplateforme. Cela signifie qu'un même projet peut être exporté sur de très nombreuses plateformes différentes, telles que PC, Mac OS, iOS, Android, consoles de jeu, etc. Il faut toutefois noter que plusieurs exports distincts et spécifiques à chaque plateforme sont souvent nécessaires. Le moteur est tout à fait adapté au développement d'applications en réalité virtuelle car il supporte nativement les principaux casques VR du marché. Des extensions créées par les développeurs des casques sont également mises à disposition pour avoir accès à des fonctionnalités avancées du support ainsi qu'à une intégration rapide.

Bien que la vocation première d'Unity soit d'offrir un environnement complet permettant de rapidement créer des jeux vidéo, il s'agit avant tout d'une plateforme de développement en temps réel offrant la possibilité de créer tout type d'application 2D, 3D ou même VR. Dans le cas de C2Care, il s'agit de l'outil idéal pour la réalisation d'environnements 3D en réalité virtuelle, permettant d'immerger un patient dans l'environnement de sa phobie, son addiction ou ses autres troubles.

Le langage de programmation associé à Unity est le C#, un langage de programmation orientée objet destiné à développer sur la plateforme .NET, un Framework créé par Microsoft. Au sein d'Unity, le code est organisé sous forme de « scripts » C# qui implémentent des bibliothèques de code du moteur, offrant ainsi la possibilité d'interagir avec les composants (graphiques, physiques, audio...) d'Unity.

Unity3D et C# constituent les principales technologies employées par le projet de ce mémoire. Pour ce qui est du matériel physique, il s'agit des casques de réalité virtuelle supportés par la suite logicielle de C2Care.

On distingue deux types de casques de réalité virtuelle : les casques dits « **autonomes** » et ceux qui ne le sont pas, aussi appelés « **filaire** ».

Les casques **filaire** sont les premiers casques VR ayant été créés et qui dominaient encore le marché jusqu'à l'apparition des casques autonomes en 2018. Ils sont connectés par un ou plusieurs fils à un ordinateur qui exécute les applications de réalité virtuelle et effectue le rendu visuel dans le casque.



*Figure 14 : Casque filaire Oculus Rift S avec ses deux manettes et le câble filaire nécessaire à son fonctionnement.*

Le système d'exploitation compatible avec les casques filaires est Windows. Il est important de préciser cela car les projets exportés depuis Unity pour ce type de casque devront être destinés à une exécution sur Windows.

Les casques **autonomes** constituent la dernière évolution majeure dans le domaine de la réalité virtuelle. Ces appareils ne dépendent pas d'un ordinateur externe pour fonctionner car ils disposent de composants de traitement intégrés. Il s'agit d'une révolution pour le grand public car ces casques sont pratiques et beaucoup plus abordables que les casques filaires qui eux, nécessitent l'achat d'un casque VR ainsi que d'un PC disposant d'une puissance de calcul élevée. Toutefois, les casques autonomes disposent d'une autonomie limitée et d'une puissance nettement inférieure à celle d'un ordinateur moderne équipé pour la réalité virtuelle.



*Figure 15 : Casque autonome Oculus Quest 2 avec ses deux manettes et un câble de rechargement.*

Le système d'exploitation se trouvant à la base de tous les casques autonomes est Android. Il est important de le préciser car l'export d'un projet depuis Unity pour ce type de casque sera complètement différent d'un export pour un casque filaire. C2Care supporte les 2 types de casques présentés ci-dessus. Lors du développement du nouvel avatar virtuel, il faudra s'assurer de sa compatibilité avec les casques VR supportés par les applications.

## V. Solutions proposées

---

Cette partie du mémoire a pour objectif de présenter, de manière précise et détaillée, les solutions que j'ai proposées pour répondre à la problématique du projet, ainsi que leur mise en œuvre concrète. Les éléments y figurant fournissent un recueil d'informations suffisamment complet pour permettre la compréhension du travail effectué et aider à sa réplique.

Les diverses phases majeures du projet seront amplement abordées, en commençant par la phase de préparation du projet avec son initialisation, la mise en place de la gestion de projet, l'organisation et planification du travail ainsi que des études annexes mais essentielles.

La phase de conception et de génie logiciel sera ensuite présentée dans l'optique d'introduire une importante description technique mais compréhensible de la réalisation même du projet. Chaque module principal du nouvel avatar virtuel de C2Care sera présenté de manière concise.

Une dernière partie dédiée à la validation et mise en exploitation de la solution développée conclura cette partie du mémoire.

## 1. Préparation du projet

Il s'agit ici de présenter comment la solution proposée pour répondre à la problématique de refonte de l'avatar existant s'est précisée, puis est devenue un projet concret. L'objectif de cette sous-partie est également d'illustrer comment le projet s'inscrit dans le processus de travail continu de l'entreprise, notamment pour ce qui est de la gestion de projet. Les diverses procédures, études et documentations que j'ai mises en place préalablement à la réalisation du projet seront énoncées et illustrées par des documents concrets disposés en annexes.

### a. Initialisation du projet

#### *i. Rappel sur le déclenchement du projet*

L'idée de retravailler l'avatar virtuel existant a émergé d'un constat commun aux membres de l'équipe de C2Care sur son état d'obsolescence en comparaison avec les nouveaux standards de la réalité virtuelle. Créé peu de temps après la fondation de l'entreprise en 2016, il était conforme aux casques de réalité virtuelle de sa génération ainsi qu'aux applications de l'entreprise. Cependant, les deux ont amplement évolué depuis et il est devenu évident pour tout le monde qu'une refonte de l'avatar s'impose.

Il est important de rappeler que ce projet, comme la majorité des travaux réalisés par l'équipe de développement, s'inscrit dans le processus d'évolution continue des logiciels existants de l'entreprise. Le commanditaire du projet est donc la société C2Care, physiquement représentée par son président Romain Streichemberger auprès duquel le travail sera défini, priorisé et validé.

Les autres parties prenantes du projet seront présentées dans une sous-partie dédiée.

## **ii. Expression des besoins**

Comme détaillé dans la problématique du mémoire ainsi que la synthèse de l'état de l'art, ce projet a vu le jour dans l'optique de répondre à des besoins devenus essentiels dans le cadre de l'amélioration des applications de réalité virtuelle de l'entreprise. Ces besoins découlent d'une problématique (celle de ce mémoire) abordée de manière récurrente lors des réunions portant sur le développement futur, au cours de l'année 2020. Ils ont été exprimés et recensés lors de multiples échanges, sans encore les transformer en projet concret.

On note ainsi les principaux besoins relevés, d'une part, pour les utilisateurs :

- Amélioration du sentiment de présence :
  - Représentation corporelle adaptable aux profils des patients : sexe, couleur de peau, taille, etc.
  - Mise à jour des corps homme et femme de base de l'avatar.
  - Proposer davantage d'options aux praticiens pour personnaliser l'expérience de leurs patients.
  - Ajout de davantage d'interactions dans les environnements de réalité virtuelle.

D'une autre part, pour l'équipe de développement de l'entreprise :

- Amélioration de l'environnement de travail :
  - Disposer d'un avatar flexible, largement paramétrable et évolutif.
  - Grandement améliorer la propreté du code des applications, fortement liée à l'avatar qui en est un composant commun.
- Compatibilité avec les nouveaux casques de réalité virtuelle du marché.

Le lancement du projet s'est concrétisé pour la première fois en novembre 2020, avec la création d'un premier document d'expression des besoins réalisé par un membre de l'équipe de développement de C2Care. Le temps d'attente avant de démarrer le projet s'explique par

le besoin de compléter certains projets importants déjà en cours et ayant pris du retard à cause de la pandémie covid-19.

Ce document (voir [Annexe A](#)) a été rédigé à mon intention pour que je puisse en prendre connaissance à mon retour en entreprise et l'affiner lors d'une réunion avec l'équipe. Le projet m'a été attribué car j'ai été à l'initiative de l'idée de refondre l'avatar. De plus, il s'agit d'un sujet de projet de fin d'étude très intéressant.

Le document énonce les principales fonctionnalités de l'avatar existant en associant à chacune : la solution actuellement employée, les inconvénients et problèmes potentiels observés, une idée de nouvelle solution à apporter ainsi qu'un indicateur de difficulté estimée pour sa réalisation.

Il ne s'agit pas exactement d'un document d'expression des besoins mais on y retrouve tout de même une formulation concise des problématiques posées par l'avatar actuel. Cela permet de cerner les points clés qui devront être pris en considération lors de la conception du nouvel avatar.

Bien que quelques fonctionnalités supplémentaires envisageables soient évoquées dans le document, celui-ci se concentre principalement sur la refonte de l'existant. Il ne propose pas réellement de nouveaux modules pouvant être intégrés au nouvel avatar.

### ***iii. Direction globale du projet***

Selon moi, effectuer un travail important pour la refonte de l'avatar constitue une opportunité parfaite pour penser à de nouvelles fonctionnalités qui pourraient potentiellement y être implémentées.

Ma vision s'est toutefois heurtée à celle de l'entreprise, qui comme énoncé précédemment, fait office de commanditaire du projet et est représentée par son président. Mon idée était de reconstruire l'avatar de zéro et d'intégrer directement dans sa conception, puis réalisation,

de potentielles nouvelles fonctionnalités en plus de l'existant. Pour l'entreprise, le projet était plutôt vu en deux temps : d'abord la refonte « à l'identique » de l'existant, mais avec un code plus propre et plus de flexibilité. Puis dans un second temps facultatif, l'ajout de nouvelles fonctionnalités à l'avatar créé.

Une réunion nous a permis de convenir d'une direction hybride entre nos deux visions pour le projet : un premier objectif de refonte de l'existant a été conservé mais certains modules ne seront pas exactement refaits « à l'identique ». Ils présenteront de légères améliorations notables dont la charge de travail ajoutée est faible.

#### ***iv. Étude de l'existant et brainstorm***

Les besoins ont été exprimés et une direction globale pour le projet a été adoptée par tout le monde. Ce n'est toutefois qu'à partir du 18 mai 2021 que le projet a réellement commencé lorsqu'on m'a demandé d'étudier l'avatar existant en détail. Ce délai entre un premier pas concret en novembre 2020 et le réel début du projet en mai 2021 s'explique par la présence d'une forte charge de travail prioritaire sur cette période.

Les objectifs de cette étude de l'avatar existant (voir [Annexe B](#)) sont multiples :

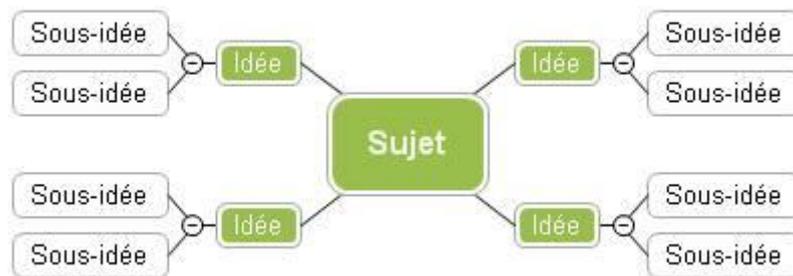
- Évaluer la structure hiérarchique de l'avatar existant dans Unity.
- Recenser ses composants et préciser leur utilité pour le fonctionnement de l'avatar : scripts C#, composants d'Unity pour la physique et les collisions par exemple, etc.

Globalement, il s'agit de prendre amplement connaissance de l'existant dont il faut effectuer une refonte.

Disposant d'une connaissance complète du système existant, ainsi que du savoir apporté par l'état de l'art effectué sur la notion d'avatar en réalité virtuelle, j'ai effectué un brainstorm sur les potentielles fonctionnalités du nouvel avatar. J'ai réalisé cet exercice dans l'optique de

déterminer chaque fonctionnalité que le nouvel avatar pourrait intégrer : que ce soient les fonctionnalités existantes intactes ou retravaillées, ou alors de nouvelles fonctionnalités.

Pour effectuer ce brainstorm, j'ai réalisé une carte heuristique, ou carte des idées, aussi appelée « mind map ». Il s'agit d'un schéma permettant de représenter visuellement des idées liées entre elles en reflétant le fonctionnement de la pensée. Une carte heuristique s'organise en suivant le schéma suivant :



*Figure 16 : Structure d'une carte heuristique.*

J'ai choisi d'utiliser ce système pour réaliser le brainstorm car je pense qu'il permet, en fixant le nouvel avatar comme sujet, d'ajouter, organiser et détailler ses différentes fonctionnalités potentielles. J'ai ainsi utilisé l'outil MindMup 2 : il s'agit d'un logiciel permettant de créer des cartes heuristiques dans un éditeur en ligne, seul ou à plusieurs collaborateurs simultanés, avec l'option de sauvegarder la création directement dans Google Drive. Utilisant déjà Google Drive pour stocker et organiser les documents du projet, MindMup 2 est donc un outil adapté.

Le produit de ce brainstorm (voir [Annexe C](#)) offre suffisamment de contenu pour mettre en place une gestion projet, puis ultérieurement, démarrer la phase de conception logicielle du projet.

## b. Mise en place de la gestion de projet

### i. Constitution de l'équipe de projet

Les fonctionnalités recensées dans la carte heuristique réalisée permettent déjà d'identifier une équipe de projet à partir de la nature des tâches.

Avant de constituer l'équipe de projet, il faut déjà visualiser l'ensemble de l'équipe technique de C2Care. Voici un organigramme du pôle développement de l'entreprise au moment du début de la réalisation du projet, en mai 2021 :

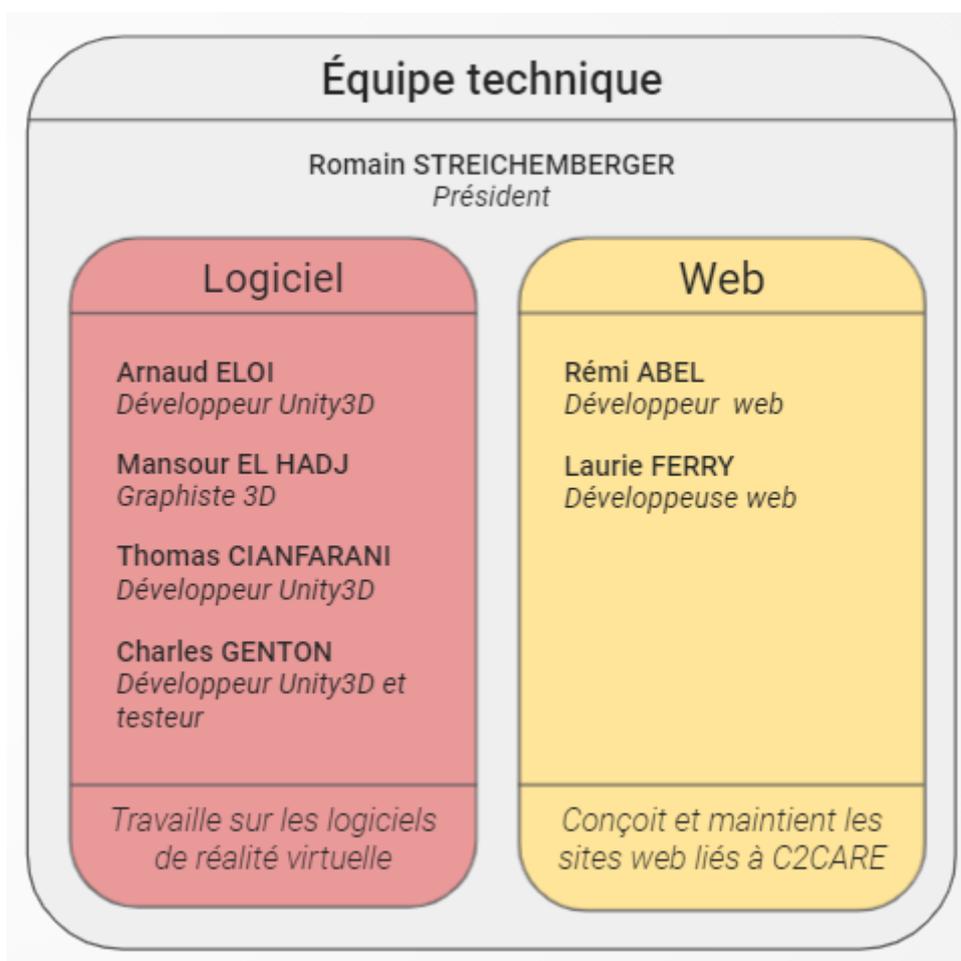


Figure 17 : Organigramme de l'équipe technique de C2Care, avec les pôles logiciels et web.

D'après la carte heuristique réalisée, on peut estimer les proportions de type de travail à fournir suivantes :

- **Développement Unity3D et C# : 80%**
- **Graphisme 3D : 15%**
- **Développement web : 5%**

Le développement Unity constitue la charge de travail la plus importante. Il semblerait donc logique d'affecter deux développeurs au projet. Cependant, l'équipe technique de C2Care est très petite en comparaison à la charge de travail liée au maintien et à l'évolution de plus de 10 applications de réalité virtuelle. De plus, celles-ci sont déployées sur 8 casques VR, le tout interagissant avec une interface web (my.c2.care) permettant aux praticiens de piloter, via internet, les applications dans le casque de leur patient.

Ainsi pour ce projet, j'assume l'intégralité du développement Unity. Pour ce qui est des tâches de graphisme 3D et de développement web, un individu pour chaque partie sera brièvement sollicité pour la réalisation de sa part de travail à un moment du projet. À ces 3 personnes, il faut ajouter un testeur qui interviendra principalement à la fin du projet pour valider le bon fonctionnement du nouvel avatar.

Enfin, le président de l'entreprise sera l'acteur qui évaluera le travail produit et s'il est conforme aux attentes de C2Care.

Pour récapituler, voici un tableau représentant les collaborateurs du projet ainsi que leur degré d'implication sur le projet :

Collaborateur	Fonction(s) principale(s) sur le projet	Degré d'implication
Thomas CIANFARANI	Développement Unity Gestion de projet Génie logiciel	Très élevé
Mansour EL HADJ	Graphisme 3D	Faible
Rémi ABEL	Développement web	Très faible
Charles GENTON	Test	Faible
Romain STREICHEMBERGER	Retours client	Faible

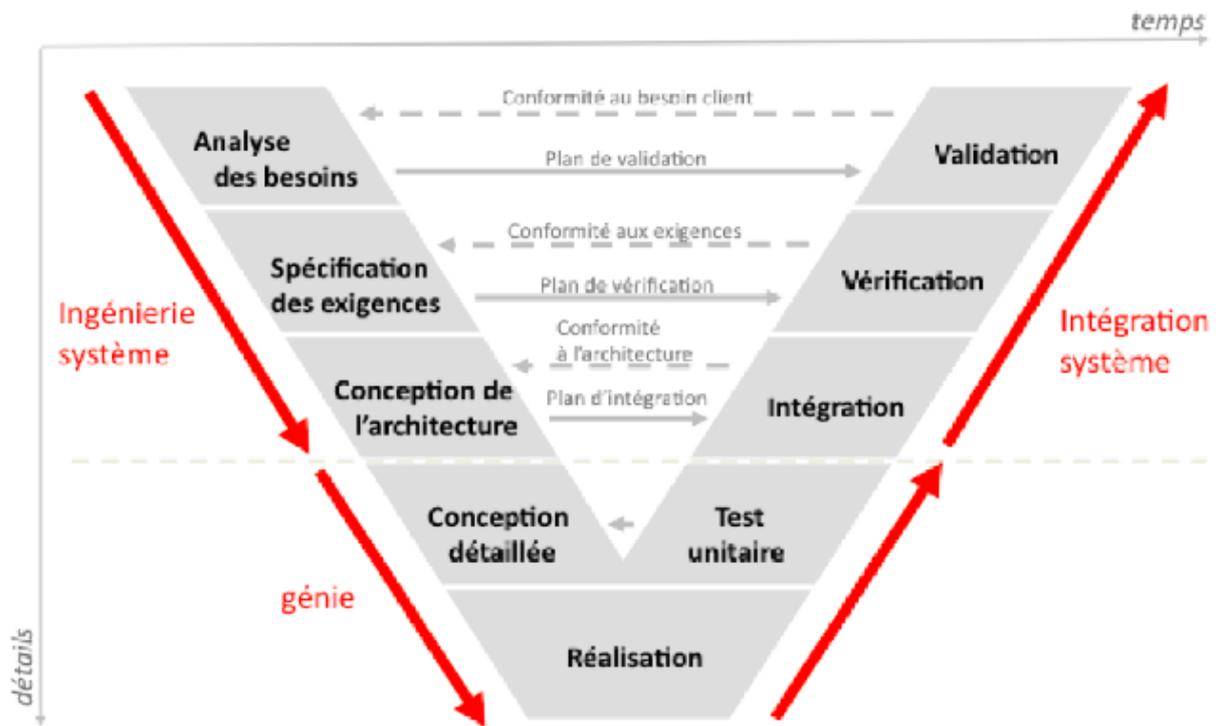
*Figure 18 : Équipe participant au projet de refonte de l'avatar, avec leur fonction et degré d'implication au sein de celui-ci.*

La constitution de cette équipe permet de créer le nouvel avatar tout en demandant une faible implication d'autres collaborateurs que moi, ce qui permet aux autres projets de C2Care de continuer d'avancer.

## **ii. Choix d'une méthode de gestion de projet**

Pour ce qui est de la gestion de projet, j'ai choisi de confronter deux méthodes différentes afin d'en choisir une à adopter pour le projet.

D'une part, il y a la gestion de projet en cycle en V. Il s'agit de la méthode en place pour les projets de l'entreprise.



*Figure 19 : Schéma représentatif de la gestion de projet en cycle en V.*

Le cycle en V est un modèle d'organisation des activités d'un projet qui se caractérise par un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation, et un flux ascendant, qui assemble le produit en vérifiant sa qualité.

À C2Care, la plupart des projets sont assez courts, avec une durée moyenne de moins d'une semaine. Il s'agit de la mise à jour des logiciels de réalité virtuelle : ajout de nouvelles fonctionnalités ou de nouveaux environnements, évolution de l'existant et corrections de bugs.

La durée de ces projets fait qu'il est fréquent que certaines étapes du cycle en V ne soient pas effectuées :

- La partie de définition du projet se résume à une description simple du travail attendu.
- L'implémentation se fait normalement.
- La partie de tests et d'intégration n'inclut en général que des tests fonctionnels, sans documentation.

Certains projets de plus grande envergure (par exemple, la création d'une nouvelle application entière) impliquent une utilisation plus complète du cycle en V.

D'une autre part, il y a la gestion de projet agile. Il s'agit d'une méthode que j'ai eu l'occasion de mettre en place dans de nombreux projets en dehors de l'entreprise et que j'ai toujours voulu faire découvrir à C2Care. Ce projet est la dernière opportunité de mon apprentissage pour pouvoir proposer cette gestion de projet à l'entreprise.

Les méthodes Agiles sont des outils de pilotage de projet de plus en plus utilisés dans les projets de développement logiciel. Le but de ces méthodes est d'éviter trop de planifications dans le développement d'un produit. On divise plutôt le projet en sous-projets de durées égales appelés « sprints » que l'équipe réalise successivement, en présentant le progrès aux clients à la fin de chaque itération et en effectuant une rétrospective interne.

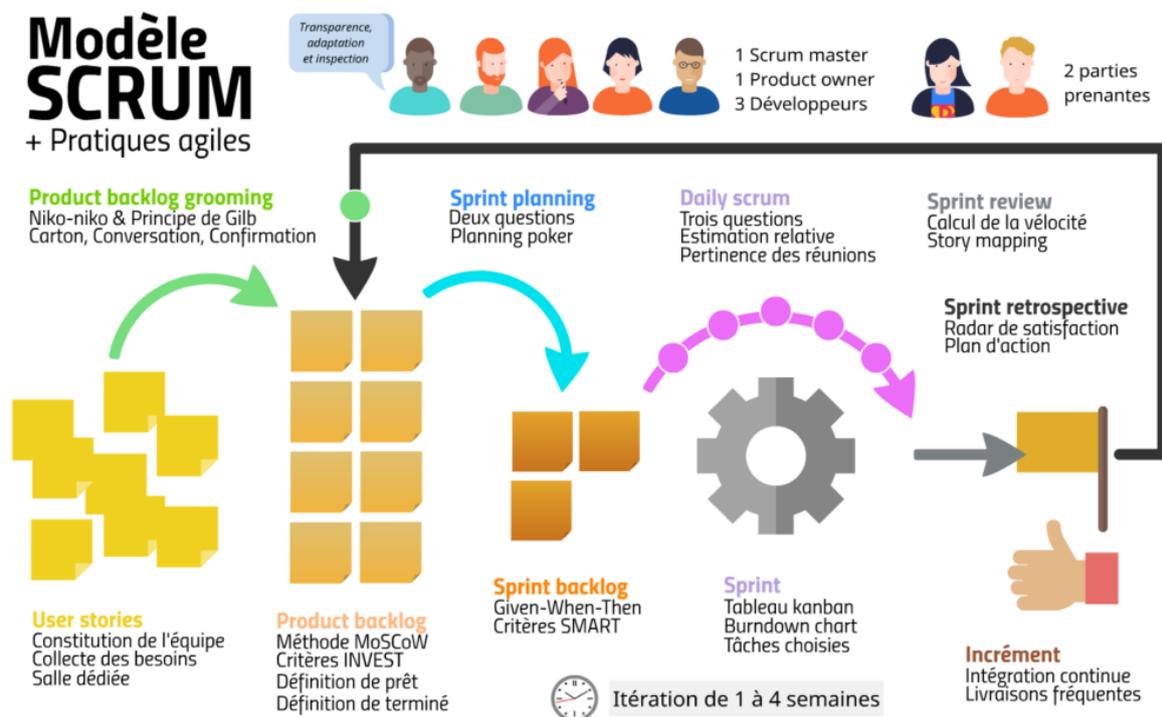


Figure 20 : Schéma représentatif de la gestion de projet agile, utilisant le modèle SCRUM.

Pour la refonte de l'avatar, il s'agit d'une méthode intéressante car le cadre du projet n'est pas clairement défini : on ne sait pas encore quelles fonctionnalités en plus de la recréation de l'existant seront développées.

Il y a d'une part, la refonte de l'avatar reprenant uniquement les fonctionnalités de l'existantes, avec d'éventuelles améliorations. D'une autre part, il y a tous les ajouts potentiels à effectuer sur l'avatar pour maximiser le sentiment de présence, la flexibilité globale de l'avatar, etc.

La méthode agile permet d'adapter le projet continuellement pour qu'il contienne les modules que le client désire vraiment dans son produit. Il se peut qu'à la fin du projet, l'entreprise se rende compte que certaines fonctionnalités supplémentaires prévues pour le nouvel avatar ne sont finalement pas nécessaires, ou pas intéressantes par rapport au coût qu'elles impliquent. L'agilité permet de prendre en compte cette éventualité.

À la suite d'une présentation des deux méthodes et des justifications propres à chacune, il a été accepté que le projet soit réalisé avec la méthode agile à condition que des dates clés soient données pour le projet :

- Une première date de fin de conception et début de réalisation du projet dans Unity.
- Une date marquant la fin de la refonte de l'avatar existant, à partir de laquelle il sera décidé quelles fonctionnalités supplémentaires seront retenues pour poursuivre le développement.

La durée d'un sprint a été fixée à 2 semaines afin de permettre à C2Care de visualiser l'évolution progressive de l'avatar. Il s'agit d'un projet beaucoup plus long que la moyenne de l'entreprise. Les points bimensuels permettront de l'affiner progressivement tout en effectuant une démonstration du travail produit.

Pour rappel, un sprint correspond à une plage temporelle de durée fixe, allant de 2 à 4 semaines, au cours de laquelle une partie du travail global sera effectuée.

### **iii. Attribution des rôles agiles**

La méthode agile SCRUM organise les équipes selon des rôles très spécifiques. Afin de respecter cette organisation, il a fallu attribuer les différents rôles agiles aux membres de l'équipe de projet constituée précédemment :

- **Le client** : l'entreprise C2Care.

La refonte de l'avatar est une commande interne à l'entreprise, dans l'optique d'améliorer à la fois la qualité du code pour optimiser le travail en interne, mais aussi offrir une meilleure interface avec la VR pour les utilisateurs.

- **Les utilisateurs finaux** : les patients des professionnels de santé clients de C2Care.

Les clients de C2Care sont des professionnels de santé : psychologues, psychiatres, thérapeutes spécialisés en hypnose, etc. Les individus exposés directement aux environnements de réalité virtuelle de C2Care sont les patients de ces thérapeutes.

- **Le Scrum master** : Thomas CIANFARANI.

*Le Scrum master s'assure que la méthode agile SCRUM est correctement appliquée. Il doit enseigner aux membres de l'équipe leurs rôle respectif et la bonne façon de l'exercer. Il doit également faciliter les interactions entre les membres et assurer la bonne productivité de l'équipe.*

Je suis à l'origine du projet dont je suis l'acteur principal, et je dispose de connaissances en gestion de projet agile.

- **Le product owner** : Romain STREICHEMBERGER.

*Le product owner porte la vision du produit. Il représente les utilisateurs et les commanditaires du projet. Il est donc avant tout chargé de maximiser la valeur du produit et le travail de l'équipe de développement.*

Dans la plupart des projets de C2Care, Romain, président de l'entreprise, est le référent pour la qualité des productions et l'atteinte des objectifs. Pour ce projet, il remplira le rôle d'interface entre le Scrum master et le client, C2Care.

- **L'équipe de développement :**
  - Thomas CIANFARANI : Scrum master et développeur Unity.
  - Mansour EL HADJ : graphiste 3D.
  - Charles GENTON : testeur.
  - Rémi ABEL : développeur web.

#### ***iv. Choix d'un outil de gestion de projet***

Afin de pouvoir débiter l'organisation du projet, notamment la création, priorisation et attribution des tâches de développement, il faut choisir un logiciel de gestion de projet.

Il est possible d'effectuer la gestion de projet agile sans passer par un logiciel mais aujourd'hui il existe de nombreuses solutions collaboratives en ligne, adaptées à la gestion de projet agile et offrant une structure et des outils réellement intéressants.

Tout comme le choix d'une méthode de gestion de projet, j'ai également mis en comparaison deux potentiels outils de gestion de projet utilisables dans le cadre de ce projet.



*Figure 21 : Logo du logiciel de gestion de projet Freedcamp.*

Il y a tout d'abord Freedcamp, l'outil de gestion de projet actuellement utilisé par C2Care. Il s'agit d'un outil de gestion de projet et un système de collaboration pour les équipes, disponible sur ordinateurs et appareils mobiles. Cet outil est gratuit pour un nombre illimité d'utilisateurs, fichiers et projets. Bien qu'il ne soit pas particulièrement adapté à la gestion de projet agile, il est possible de l'utiliser à cette fin.



*Figure 22 : Logo du logiciel de gestion de projet Jira.*

Jira est ma proposition personnelle. Il s'agit principalement d'un système de gestion de projets agiles développé par Atlassian. Cet outil est gratuit jusqu'à 10 utilisateurs, puis propose des plans progressifs pour les différentes tailles d'équipe. J'ai sélectionné parmi les autres outils de gestion de projet non utilisés à C2Care pour les raisons suivantes :

- Sa tarification par rapport à la taille de l'équipe de développement.
- Parfaitement adapté à la gestion de projet agile, notamment avec la méthodologie Scrum.
- Solution prouvée et reconnue : « L'outil de développement logiciel n°1 pour les équipes agiles ».
- Flexibilité d'utilisation et bons outils d'estimation.

J'ai réalisé le tableau comparatif suivant afin de proposer les deux outils, accompagnés d'arguments, au président de l'entreprise supervisant le projet :

			
<b>VERSION GRATUITE</b>			
Gratuit pour un nombre d'utilisateurs, projets illimités mais avec fonctionnalités limitées.		Gratuit jusqu'à 10 utilisateurs avec toutes les principales fonctionnalités de gestion de projet, à condition d'avoir 1 seule équipe.	
<b>AGILITÉ</b>			
Agilité possible mais demande une reconfiguration de l'espace Freedcamp.		Très nombreuses fonctionnalités natives pour utiliser la méthode Scrum.	
/		Interface adaptée, créations d'epics, sprints, backlog de produit.	
/		Outils de suivi et d'analyse agiles tels que la vélocité, burndown chart, etc.	
<b>RECONNAISSANCE</b>			
/		Multiples reconnaissances en 2020.	
<b>UTILISATION A C2CARE</b>			
Outil officiel de C2Care mais actuellement très peu utilisé.		Jamais utilisé auparavant.	

*Figure 23 : Tableau comparatif entre les logiciels de gestion de projet Freedcamp et Jira.*

À la suite d'une réunion présentant ce comparatif, la décision a été prise d'utiliser le logiciel de gestion de projet déjà en utilisation à C2Care : Freedcamp. Cette décision prise par la direction de C2Care est justifiée par le fait que l'entreprise paie déjà un abonnement pour Freedcamp et a pour objectif d'utiliser davantage le logiciel dans le futur. De plus, il y a une volonté de centralisation de la gestion de projet afin que toutes les équipes de C2Care (développement, marketing) organisent leurs projets dans Freedcamp.

Ce sont des arguments solides, bien mon opinion personnelle penchait vers Jira car au moment de la décision, l'outil actuel Freedcamp est très peu utilisé par les équipes. Jira est parfaitement adapté aux projets agiles contrairement à Freedcamp. J'avais espoir qu'avec l'utilisation d'un nouveau logiciel, avec une nouvelle méthode de gestion de projet donne une nouvelle impulsion positive dans l'organisation au sein des équipes de C2Care.

L'outil de gestion de projet sélectionné est donc Freedcamp.

#### ***v. Création du projet dans Freedcamp***

Afin de pouvoir débiter l'organisation et la planification du projet, j'ai créé un espace dédié au projet de refonte de l'avatar au sein de Freedcamp. Cet espace contiendra la majorité des éléments de gestion de projet et permettra d'effectuer un suivi de l'avancement du travail. Les différentes phases du projet y seront également distinguées et chaque tâche ajoutée au sein de l'outil sera enrichie d'informations telles que des descriptions, indicateurs de priorité, dates de réalisation, membres affectés, etc.

Comme décrit précédemment, Freedcamp n'est pas un outil spécifiquement dédié à la gestion de projet agile. La seule option de mise en forme d'un projet dans Freedcamp permettant d'appliquer la méthodologie Scrum est d'opter pour la configuration « Kanban ». On y retrouve ainsi les trois principales colonnes utilisées pour le suivi de l'avancement des tâches sur un projet agile : « en attente », « en cours », et « terminé ».

La configuration de Freedcamp pour le projet m'a permis de confirmer les premiers constats effectués lors de la comparaison du logiciel avec Jira. En effet, j'ai pu constater l'absence de certains termes spécifiques à la méthode agile Scrum, l'impossibilité d'ajouter des colonnes en plus des trois de base ou encore l'absence d'outils de suivi. Ces inconvénients seront à prendre en considération pour le bon déroulement de la gestion de projet, mais il faut toutefois souligner que les fonctionnalités essentielles pour la mise en place de la méthode agile Scrum sont présentes dans Freedcamp.

### **c. Organisation du projet**

#### ***i. Création des user stories pour constituer le product backlog entier du projet***

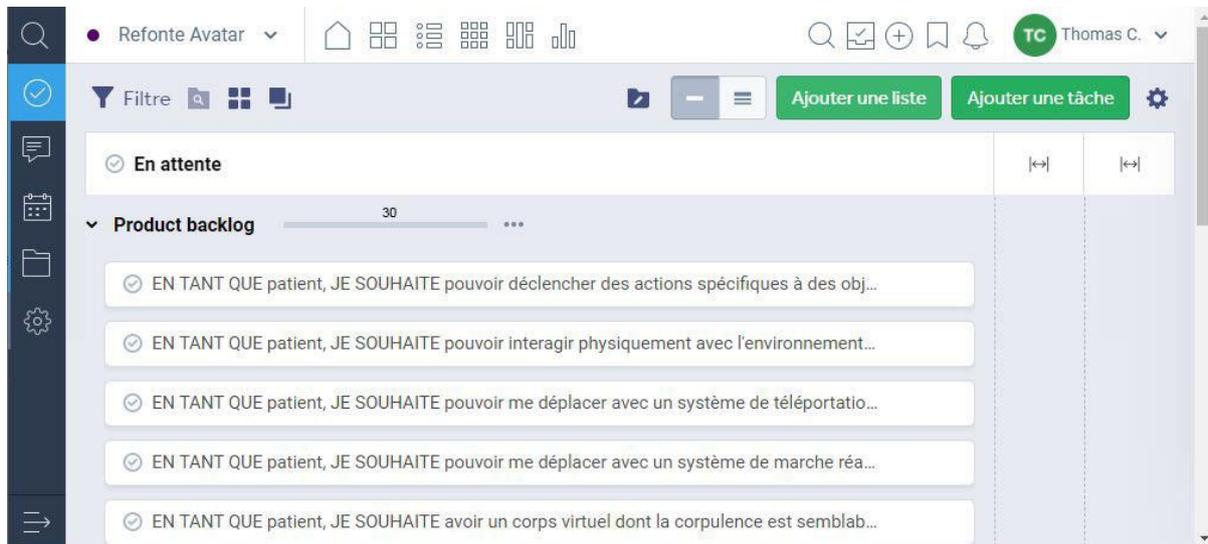
Tous les éléments nécessaires sont réunis pour démarrer l'organisation du projet en convertissant la carte heuristique réalisée lors du brainstorm en tâches réalisables par l'équipe de projet établie. Ces tâches doivent être intégrées dans l'outil de gestion de projet choisi, et évolueront tout au long du projet.

Dans le cadre d'un projet agile, ces tâches sont plutôt appelées « **user stories** ».

Les user stories sont de courtes descriptions de fonctionnalités racontées depuis la perspective de l'utilisateur. Ainsi, une user story correspond le plus souvent à une phrase simple, rédigée dans un langage courant, qui permet de décrire avec suffisamment de précision le contenu d'une fonctionnalité à développer. Ce format particulier vise à assurer que l'on délivre bien de la valeur pour les utilisateurs.

Au démarrage d'un projet agile, l'équipe doit réaliser un « product backlog », ou **backlog de produit**. Il s'agit d'une liste exhaustive d'user stories destinées à l'équipe de développement, couvrant l'intégralité du projet tel qu'il a été défini. Tout au long du projet, cette liste contiendra les fonctionnalités à réaliser pour atteindre la complétion du produit. Le backlog de produit est évolutif et affiné régulièrement par des échanges clients propres à la méthode agile.

Dans l'optique d'organiser le projet de refonte de l'avatar, j'ai constitué le backlog de produit du projet à partir des différents nœuds de la carte heuristique ayant été réalisée. Ainsi, pour chaque fonctionnalité existante ou à ajouter à l'avatar, j'ai créé une user story qui a été ensuite intégrée dans une liste Freedcamp intitulée « Product backlog ».



*Figure 24 : Capture d'écran du backlog de produit répertorié dans l'espace Freedcamp du projet.*

Trente user stories ont été créées pour l'intégralité du projet et répertoriées dans Freedcamp. Ce backlog de produit initial constitue donc une liste d'éléments de travail conséquente où chaque item peut être développé, attribué à un membre de l'équipe et planifié pour un sprint.

## **ii. Étude sur la pertinence des user stories : critères INVEST**

Avant de procéder à un traitement avancé du contenu du backlog de produit, j'ai fait le choix d'effectuer un passage sur chaque user story créée afin d'en assurer la qualité et la pertinence globale dans le cadre projet. Pour cela, j'ai utilisé les **critères INVEST**.

La grille des critères INVEST permet de juger de la qualité d'une user story ; elle conduit éventuellement à reformuler son énoncé, voire à modifier en profondeur la story.

Une bonne user story est :

- Indépendante des autres.
- **N**égociable initialement, plutôt qu'un engagement ferme.
- **V**erticale, ou ayant de la valeur en soi.
- **E**valuée en matière de complexité relative.
- **S**uffisamment petite (en anglais Small).
- **T**estable en principe, ce qu'on vérifie en écrivant un test.

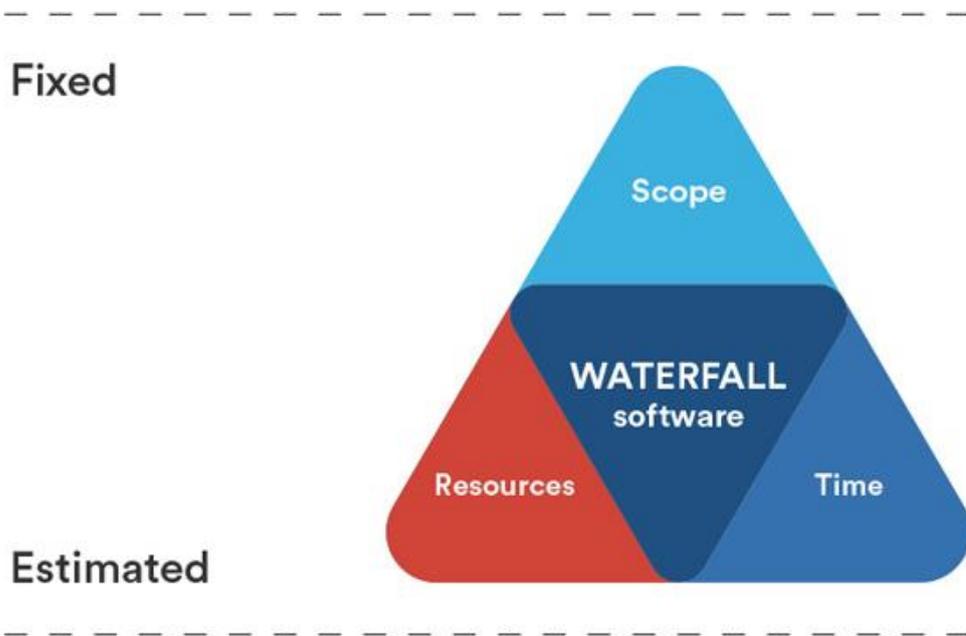
En appliquant cela aux user stories du projet, j'ai pu réduire leur nombre de 30 à 24. Ce travail a également permis d'améliorer la formulation de chaque story, bien qu'appliquer chaque critère de la grille soit difficilement réalisable.

### ***iii. Estimation de durée du projet***

Préalablement au démarrage des phases de conception logicielle et de réalisation du projet, il est essentiel de pouvoir estimer celui-ci en durée. C'est cette estimation, le plus souvent exprimée en jours-hommes, qui permet de déterminer un budget prévisionnel pour le projet. Ce budget estimé constitue un élément essentiel dont l'analyse peut amener à la poursuite du projet ou alors à sa réévaluation, voire son arrêt.

La refonte de l'avatar implique un travail conséquent. Il est donc nécessaire de pouvoir présenter une estimation de durée et de coût du projet à l'entreprise, faisant aussi office de client.

Dans le cadre d'une gestion de projet classique (cycle en V), on estime chaque tâche du projet en jours-homme. Un jour-homme correspond à un jour de travail d'un seul individu. Ainsi, une tâche estimée à dix jours-hommes peut être réalisée en dix jours par une seule personne, ou alors cinq jours par deux personnes. À l'issue de l'estimation, on peut ainsi proposer une durée totale du projet en jours-homme qui pourra être ensuite transformée par le chef de projet en une estimation plus poussée de durée et de coût. Cette approche est l'**estimation globale**, représentée par le triangle de fer du Dr. Martin Barnes.



*Figure 25 : Approche « Waterfall software », ou « triangle de fer » du Dr. Martin Barnes, 1969.*

Le périmètre du projet (scope) est fixe. Les ressources et le temps sont variables : on peut choisir d'ajouter plus de personnes ou de temps au projet. On peut ainsi obtenir une estimation en jours-hommes.

Comprendre l'estimation globale permet de mieux comprendre l'approche agile dans l'estimation d'un projet. Dans le cadre d'une gestion de projet agile, on considère que les estimations prédictives sont toujours fausses : chaque projet rencontre des obstacles et aléas. De plus, la gestion de projet agile ne s'applique pas à un périmètre de projet fixe. Tout est amené à évoluer au long du projet à la suite de réunions fréquentes avec le client. Ainsi est utilisée **l'estimation relative**.

Les fonctionnalités sont comparées entre elles afin de déterminer pour chaque élément, un indice de quantité de travail relatif plutôt qu'une estimation en jours-hommes.

La gestion de projet agile employée pour le travail de refonte de l'avatar implique alors d'utiliser la méthode d'estimation relative. Cependant, la direction de C2Care m'a imposé de proposer une durée totale du projet, avec notamment deux dates importantes :

- Une date pour la refonte de l'avatar avec ses fonctionnalités existantes.
- Une autre pour la fin totale du projet avec l'ajout des fonctionnalités supplémentaires et le déploiement dans tous les logiciels de C2Care.

L'estimation relative ne me permet pas de donner des dates aussi précises. Ainsi, j'ai fait le choix de combiner estimation relative et estimation globale afin de pouvoir donner les informations qui m'ont été demandées tout en essayant de conserver une certaine flexibilité agile pour le projet.

J'ai décidé d'utiliser des tailles de vêtements pour effectuer une estimation relative. Habituellement, la suite de Fibonacci est utilisée, mais on peut y associer des tailles de vêtements pour une meilleure compréhension.

XXS	XS	S	M	L	XL	XXL
1	2	3	5	8	13	21

*Figure 26 : Tailles de T-Shirt et suite de Fibonacci.*

Afin de pouvoir aussi donner une estimation globale du projet, j'ai également associé à la taille « M » une durée en jours-homme de 3 jours.

Disposant d'un backlog de produit complet, comportant des user stories indépendantes et distinctes, j'ai d'abord choisi une story d'étalonnage à laquelle j'ai affecté la taille de vêtement « M » : « En tant que patient, je souhaite avoir un corps virtuel animé dynamiquement afin d'avoir une représentation fidèle de mes membres dans réalité virtuelle ». À partir de cela, j'ai estimé toutes les autres user stories du projet relativement à celle-ci. Une user story de

durée « XS » est bien plus courte que la story d'étalonnage alors qu'une de durée « L » sera un peu plus longue.

Ayant estimé relativement les 24 user stories du backlog de produit, j'ai ensuite converti les durées relatives en durée jours-homme ce qui me permettra d'estimer une durée totale du projet.

XXS	XS	S	M	L	XL	XXL
<1	1	2	3	4	5	>5

*Figure 27 : Tailles de T-Shirt et estimation en jours-hommes associée.*

L'estimation finale obtenue est de 53 jours-hommes pour l'intégralité du projet.

#### **iv. Priorisation des user stories**

Dans l'optique de distinguer les user stories concernant la refonte de l'avatar existant et celles concernant des ajouts plus ou moins importants pour le client, j'ai priorisé chaque story du backlog de produit en utilisant le système intégré à Freedcamp, comportant 3 simples labels distinctifs :

- **Priorité haute** : toutes les fonctionnalités à refondre que l'avatar existant possède.
- **Priorité moyenne** : les fonctionnalités supplémentaires ou autres tâches prioritaires à la suite de la refonte de l'existant.
- **Priorité faible** : tout élément du backlog de produit restant dont l'éventuelle réalisation sera décidée ultérieurement dans le projet.

Tout au long du projet, il sera ainsi simple de distinguer les éléments de travail prioritaires. La distinction majeure entre les tâches de priorité haute et les autres permet, à l'aide de

l'estimation réalisée précédemment sur chaque user story, de donner les deux dates clés du projet demandées par le client.

L'équipe dispose d'un backlog de produit complet, dont les composants sont estimés et priorisés. Toutefois avant de se prononcer sur des dates clés et conclure la préparation du projet, j'ai choisi d'effectuer quelques études supplémentaires que j'estime nécessaires pour un projet de cette ampleur.

#### **d. Études et finalisation de la préparation**

##### ***i. Étude de risques sur le projet***

La solution proposée dans le cadre de ce projet de fin d'études se doit de répondre à la problématique du projet tout en respectant un certain nombre de contraintes liées à l'entreprise et à ses environnements applicatifs. La gestion de risque sur un tel projet est nécessaire pour assurer, dans la mesure du possible, son bon déroulement et une création de valeur efficace.

Un risque est une part d'incertitude sur la réalisation d'un objectif. La plupart des risques sont identifiables, d'où l'intérêt d'effectuer cette étude. On peut également les classer par ordre de probabilité et de gravité.

En se basant sur ce modèle, j'ai dans un premier temps tenté d'identifier tous les risques du projet. J'ai associé à chaque risque des indices de probabilité et de gravité allant de 1 à 3 :

1. Faible
2. Moyen
3. Élevé

Cela m'a permis d'obtenir le tableau suivant :

Risque	Probabilité	Gravité
Premier projet agile : vélocité inconnue	3	1
L'avatar devient trop complexe	1	3
Problème de compatibilité de l'avatar avec un des casques supportés	1	1
Environnement de test de l'avatar insuffisant	1	1
Certains environnements très spécifiques vont demander d'adapter l'avatar en cours de déploiement	3	3
Déploiement bloqué par les versions d'Unity divergeant sur les projets	2	1
Difficulté de mettre à jour la partie my.c2.care liée à l'avatar	1	2
Manque d'expérience par rapport à certaines tâches	2	2
Incapacité d'analyser l'évolution du projet	2	1

*Figure 28 : Risques identifiés sur le projet, avec indices de probabilité et gravité.*

J'ai ensuite classé ces risques par ordre d'importance en me basant sur la formule suivante :

$$\text{indice d'importance} = \text{probabilité} * \text{gravité}$$

ID	Risque	Importance
1	Certains environnements très spécifiques vont demander d'adapter l'avatar en cours de déploiement	9
2	Manque d'expérience par rapport à certaines tâches	4
3	L'avatar devient trop complexe	3
4	Premier projet agile : vitesse inconnue	3
5	Déploiement bloqué par les versions d'Unity divergeant sur les projets	2
6	Difficulté de mettre à jour la partie my.c2.care liée à l'avatar	2
7	Incapacité d'analyser l'évolution du projet	2
8	Problème de compatibilité de l'avatar avec un des casques supportés	1
9	Environnement de test de l'avatar insuffisant	1

*Figure 29 : Risques identifiés sur le projet, classés par ordre d'importance.*

Connaître l'importance de chaque risque est essentiel pour pouvoir prendre conscience de son impact potentiel sur le projet, et le traiter en priorité. Disposer d'une liste de risques classés par ordre de priorité est la première étape pour la gestion des risques du projet. Il faut ensuite en effectuer un suivi continu en leur associant des étiquettes par rapport à leur état de traitement. Le « ROAM board » est un modèle particulièrement adapté aux méthodes agiles, permettant de maintenir une gestion des risques efficace tout au long du projet.

Il s'agit d'un tableau composé de quatre cases dans lesquelles les risques identifiés sur le projet sont classés en fonction de leur état de traitement :

- **Resolved** (résolu) : le risque est clairement identifié et une action peut être mise en place pour le contourner ou pour l'éliminer.

- **Owned** (pris en charge) : un membre de l'équipe a accepté la responsabilité de traiter ce risque et de mettre en place un certain nombre d'actions en ce sens.
- **Accepted** (accepté) : le risque a été clairement identifié ainsi que ses conséquences potentielles. Néanmoins, l'ensemble de l'équipe est tombé d'accord pour ne pas le traiter et ne rien faire. Cela peut être le cas si le risque identifié semble tout à fait mineur et que même si un problème survient, il n'aura que peu d'impact sur le déroulement du projet.
- **Mitigated** (atténué) : une action a été mise en place de façon que la probabilité qu'un problème survienne soit grandement diminuée, et que dans ce cas, les conséquences soient minimisées.

J'ai donc réalisé ce tableau sur Google Sheets, disponible en [Annexe D](#).

Pour conclure cette étude de risque, j'ai ensuite repris chaque risque individuellement en y associant des détails, ses potentielles conséquences, probabilité et gravité du risque justifiées et enfin une éventuelle solution correspondant à l'état du risque dans le ROAM board. Un exemple de cette rédaction pour un des neuf risques identifiés est lisible en [Annexe E](#).

Cette étude de risque complète a permis à l'équipe de se préparer à des éventualités qui ont été réellement rencontrés lors de la réalisation du projet et son déploiement. Cette étape a donc été cruciale pour ce projet.

## *ii. Étude budgétaire : budget prévisionnel*

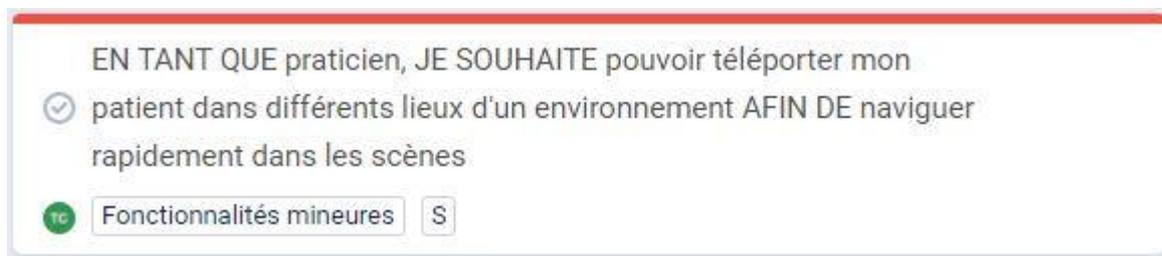
La majorité des éléments nécessaires à une estimation budgétaire ont été préparés. Comme dit précédemment, c'est cette appréciation prévisionnelle sera présentée au président de l'entreprise étant à la fois client et commanditaire du projet. Elle permettra donc la validation du projet ou une réévaluation de son périmètre.

Avant de procéder à la réalisation de l'estimation budgétaire, on peut découper l'intégralité du projet en sous-parties auxquelles seront associées une estimation de temps, puis de coût. Ce sont également ces sous-parties qui seront détaillées plus tard dans ce mémoire.

On distingue ainsi les étapes suivantes :

- Préparation du projet
- Conception initiale
  - Conception globale du projet
  - Préparation de l'environnement de test
- Réalisation
  - Création de la base de l'avatar
  - Système d'interactions
  - Représentation visuelle
  - Calibrage et positionnement
  - Fonctionnalités mineures
- Post-projet
  - Intégration dans les projets
- Bilan du projet

Dans Freedcamp, en plus des informations de durée et priorité, j'ai ajouté à chaque user story un label correspondant à l'étape à laquelle elle correspond.



*Figure 30 : User story enrichie du backlog de produit dans Freedcamp.*

L'estimation budgétaire prévisionnelle sera faite à partir du nombre de jours hommes associé à chaque étape du projet. Pour connaître la durée d'une étape, il suffit donc de faire la somme de la durée en jours-homme associée à chacune des user stories la composant.

Afin d'estimer un coût pour chaque étape, également pour le projet dans son intégralité, il faut appliquer la formule suivante :

$$\text{Budget nécessaire} = \text{Temps de réalisation estimé} * \text{Coût journalier}$$

Le temps de réalisation estimé étant connu, il reste à déterminer le coût journalier. J'ai fait la requête auprès des ressources humaines de C2Care. Le taux journalier moyen facturé à des clients dans le cadre de projets impliquant la même équipe est de 350€ par jour.

Disposant de toutes ces informations, j'ai ainsi réalisé l'estimation budgétaire prévisionnelle détaillée disponible en [Annexe F](#).

Pour la réalisation de l'intégralité du projet, soit pour 53 jours-hommes à un taux moyen de 350 € par jour, le budget prévisionnel total estimé est de 18 550 €.

À l'issue du projet, le budget prévisionnel sera confronté au budget réel dépensé.

### **iii. Présentation et validation**

À l'issue de la préparation du projet, une réunion a été organisée impliquant tous les acteurs du projet. Bien que des échanges aient eu lieu tout au long de la mise en place du projet, ce rassemblement représente un passage clé pour faire la transition vers la réalisation de la solution.

J'ai réalisé et présenté un PowerPoint abordant tous les points de la préparation du projet précédemment détaillés dans ce mémoire, dans l'optique de :

- Présenter le projet dans sa globalité : objectifs et fonctionnalités.
- Expliquer la méthode de gestion de projet et les rôles spécifiques de chacun des membres de l'équipe.
- Justifier certains choix d'orientation du projet.
- Proposer les estimations temporelles et budgétaires.
- Obtenir une validation du projet et donc un ordre de démarrage.

Cette réunion a ouvert des discussions sur plusieurs points, notamment sur les aspects techniques de l'avatar à refondre alors que l'objectif n'était pas d'aborder la conception. Un point intéressant concerne la nécessité d'avoir une vision globale sur le projet d'un point de vue conception logicielle. C'est-à-dire préparer l'architecture logicielle du projet avant de démarrer la réalisation. Nous avons donc fait le choix de dédier un premier sprint un peu plus court que les autres à la conception logicielle de la solution. Globalement, les autres sprints auront une durée de deux semaines au bout desquelles une présentation client sera effectuée comme le veut la méthode agile Scrum.

Cette réunion a eu lieu le 25 mai 2021, après cinq jours de préparation du projet.

Les deux dates clés demandées par le client, que j'ai proposées lors de la réunion sont :

- **Lundi 5 juillet 2021** : fin de la refonte de l'avatar existant.
- **Vendredi 30 juillet 2021** : fin de la réalisation des fonctionnalités supplémentaires de l'avatar et déploiement dans tous les logiciels de réalité virtuelle de C2Care effectué.

Le projet est validé par tous les acteurs. Une première phase de conception et de génie logiciel préparera ensuite la réalisation du nouvel avatar dans un environnement de travail dédié. L'avatar sera ensuite déployé dans toutes les applications ce qui clôturera le projet.

## 2. Conception logicielle

Ainsi que démontré précédemment, la refonte de l'avatar virtuel de C2Care est un projet de taille conséquente. De plus, un des objectifs du nouvel avatar est d'offrir une structure logicielle plus ordonnée et évolutive à ce composant essentiel aux logiciels de réalité virtuelle de l'entreprise. Il est alors d'autant plus nécessaire d'installer un processus de génie logiciel important, couvrant pour le projet : stratégie de tests, architecture logicielle, structure du nouvel avatar, documentation et bonnes pratiques de codage.

### a. Définition de l'environnement de travail

La réalisation de la refonte doit être réalisée dans un environnement de travail adapté au travail de réalisation du nouvel avatar, puis à son déploiement dans les autres logiciels de C2Care. Comme énoncé précédemment dans le mémoire, les deux technologies essentielles qui sont employées pour la réalisation du projet sont le moteur de jeu Unity3D et son langage de programmation associé, le C#.

C2Care propose à ses clients plus de 10 logiciels de réalité virtuelles différents, possédant la nomenclature « C2 + acronyme de l'application » : C2Phobia, C2Addict, etc. Chacune de ces applications correspond à un projet Unity3D distinct. Chaque projet possède ses propres « Assets », c'est-à-dire des ressources diverses centralisées dans le projet Unity. Comme assets, on compte par exemple des fichiers images, vidéos, modèles 3D, ou alors des fichiers générés par Unity : scènes, configuration du projet. Chaque projet Unity de C2Care contient tout de fois une base commune et présente partout. Cette base correspond à un ensemble d'assets, réunis dans un dossier nomencluré « C2Base ». On y retrouve l'avatar, mais aussi tous les éléments liés au démarrage des applications : vérification de licence, système de réseau, scripts C# génériques, etc.

Pour effectuer la refonte de l'avatar, il faut donc modifier C2Base.

Le dossier C2Base est présent dans un projet Unity particulier : C2Packages. Il s'agit d'un projet Unity uniquement utilisé pour gérer tous les éléments communs aux autres applications. Lorsqu'il faut mettre à jour la base des applications, on le fait dans C2Packages, puis la base est exportée dans un seul fichier ayant pour extension « .unitypackage ». Il s'agit d'une archive propre au moteur de jeu. On peut ainsi intégrer son contenu dans d'autres projets Unity en effectuant un simple glisser-déposer dans l'interface Unity d'un projet.

Ainsi, la refonte de l'avatar virtuel s'effectuera dans le projet C2Packages. Une fois l'avatar terminé, un « .unitypackage » sera exporté et intégré à tous les logiciels de C2Care.

## **b. Gestion de versions**

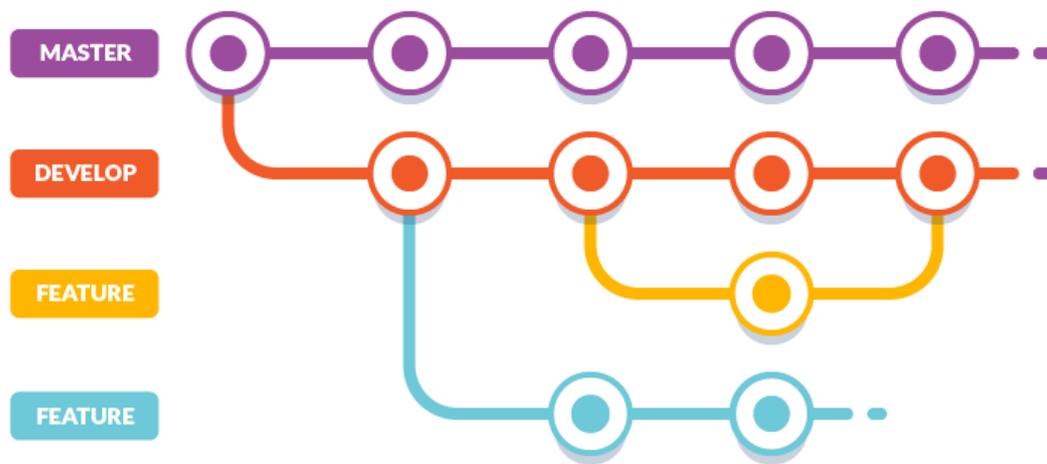
La gestion de versions est une activité qui consiste à maintenir l'ensemble des versions d'un logiciel tout au long de son développement. Il est indispensable d'utiliser un logiciel dédié à cette fonction. Combiné avec un hébergeur en ligne adapté, il s'agit également de l'outil permettant à tous les collaborateurs de l'équipe de développement de collaborer sur le projet.

Le système de gestion de versions utilisé à C2Care est l'association du logiciel Git et de l'hébergeur en ligne GitHub :

- **Git** est un logiciel de gestion de versions décentralisé, permettant une collaboration avancée sur du code et des fichiers.
- **GitHub** est un service web d'hébergement et gestion de développement de logiciels utilisant Git. Il permet, couplé avec Git, d'apporter une meilleure visibilité de la gestion de versions. Il s'agit également d'un serveur distant sécurisé sur lequel les fichiers des projets y sont conservés et partagés avec les équipes.

Pour chaque projet Unity de C2Care, un projet Git hébergé sur GitHub est associé. Cela permet de versionner les fichiers des projets tout en les rendant accessibles à tous les membres de l'équipe.

Il existe différentes approches pour l'utilisation de Git dans les projets logiciels. L'approche la plus commune, intitulée **Git-Flow** est utilisée à C2Care : un flux commun contenant la dernière version d'un projet est maintenu et chaque collaborateur travaille dans une copie de ce flux, appelée « branche », pour y développer sa fonctionnalité. La branche est ensuite fusionnée avec le flux de développement principal lorsque la fonctionnalité est terminée.



*Figure 31 : Schéma de la méthode « Git-Flow ». Le flux de développement principal correspond à la branche « develop ». Des branches « feature » naissent de develop puis y sont fusionnées une fois le développement achevé.*

Tout comme les projets « produits » de C2Care, le projet C2Packages dans lequel l'avatar doit être réalisé est versionné en suivant ce même principe. Ainsi, pour sa réalisation, une branche « feature » sera créée à partir du flux principal de développement du projet, soit la branche « develop ». Une fois le développement terminé, la branche créée sera à nouveau intégrée à develop.

### c. Documentation et bonnes pratiques de codage

La réalisation du nouvel avatar implique une grande quantité de travail de programmation. De plus, le nouvel avatar doit répondre à contraintes de qualité de code et de pérennité. Pour y répondre, j'ai alors créé une documentation technique liée au projet. Sous la forme d'un Google Doc accessible par tous les membres de l'équipe de développement, j'ai recensé les normes, standards et consignes d'organisation à adopter pour le projet (voir [Annexe G](#)).

### d. Mise en place de la stratégie de test

La stratégie de tests pour un projet logiciel est un élément indispensable du génie logiciel. L'avatar virtuel est un composant partagé par toutes les applications de C2Care. Cela signifie qu'en cas de dysfonctionnement, ce sont tous les produits qui sont concernés, ce qui peut entraîner des conséquences très graves pour l'entreprise. Il est donc primordial de mettre en place une stratégie de tests efficace et complète pour l'avatar.

La stratégie pour laquelle j'ai opté consiste en la création et l'exécution de tests tout au long du développement. Il s'agit là de tester continuellement les fonctionnalités développées lorsque celles-ci viennent d'être achevées.

Occurrence	Action(s)
Fin de la réalisation d'une user story	<ul style="list-style-type: none"><li>• Création d'un cas de test associé</li><li>• Exécution successive de tous les tests réalisés jusqu'à présent dans Unity</li></ul>
Fin d'un sprint	<ul style="list-style-type: none"><li>• Exécution successive de tous les tests réalisés jusqu'à présent sur tous les casques de C2Care</li></ul>

*Figure 32 : Organisation des tests tout au long du projet.*

Il faut que pour chaque user story développée, un **mini-scénario de test** soit créé.

Il doit être :

- Reproductible.
- Indépendant d'autres tests.
- Court en durée.
- Rapide d'implémentation.

Par exemple, pour l'user story « En tant que patient, je souhaite pouvoir me déplacer directionnellement afin d'explorer les environnements 3D », le mini-scénario associé demandera au testeur de se déplacer pour atteindre un cube à quelques mètres de lui. S'il y parvient, alors le test sera validé. Sinon, le testeur aura l'option de faire échouer le test manuellement avec un bouton de la manette, après avoir constaté un problème sur le déplacement de l'avatar.

L'ensemble de ces mini-scénarios permettra de constituer **un scénario de test complet** qui pourra finalement être exécuté afin de valider le fonctionnement de l'avatar en 5-10 minutes. L'ensemble du scénario sera intégré dans une seule scène Unity et devra être suffisamment explicite (instructions, simplicité du test, etc.) pour que quelqu'un qui ne connaît pas le logiciel puisse valider l'avatar.

Procéder à une création de tests de manière continue permettra également d'avoir un environnement de test global de l'avatar grandissant en même temps que le projet avancera. Il sera ainsi possible, par exemple à la fin de chaque sprint, de tester simplement l'avatar dans son état sur tous les casques de réalité virtuelle supportés par les logiciels.

Afin de rendre cette stratégie réalisable, le développement d'un framework de test sera nécessaire préalablement au début de la réalisation même du nouvel avatar. Ce framework devra être utilisable pour d'autres scénarios de tests que celui de l'avatar. Il sera réalisé en suivant les principes suivants :

- Créer une scène pour effectuer des tests sur un thème global : l’avatar par exemple.
- Définir plusieurs tests dans la scène avec un script “TestCase” pour chaque test :
  - Définir une instruction au démarrage du test.
  - Définir plusieurs étapes pour le test :
    - Instructions.
    - Actions au début et à la fin de l’étape.
    - Nombre de validations pour passer l’étape.
- Ajouter tous les tests (“TestCase\_*nom-du-test*”) réalisés à un script “TestRunner” qui va gérer leur exécution successive et l’affichage des résultats des tests à la fin.

### e. Spécification des user stories

À chaque user story créée dans Freedcamp, j’ai associé une description suivant le format suivant :

- **Actions** : fonctionnalités apportées.
- **Moyens** : ressources ou méthodes à utiliser.
- **Contraintes** : éléments à prendre en considération lors de la réalisation de la story.
- **Paramètres** : variables de configuration à intégrer.

Cette documentation de chaque user story permet de conserver de manière écrite, les détails liés à celle-ci. L’intégration directement dans Freedcamp est d’autant plus avantageuse pour aisément consulter la description au moment de l’affectation de la story au statut « en cours » du kanban agile. Voir un exemple de spécification d’une user story en [Annexe K](#).

### f. Architecture logicielle

L’architecture logicielle est la phase de conception demandant certainement le plus de réflexion. Il s’agit de décrire de manière symbolique et schématique les différents éléments d’un ou plusieurs systèmes informatiques, leurs interrelations et leurs interactions. Dans le

cadre de la refonte de l'avatar, il s'agit de concevoir une structure de code solide et organisée, permettant à l'avatar de disposer de toutes les fonctionnalités désirées pour le projet. Cette phase consiste à réaliser presque entièrement le projet sous une forme abstraite avant la production effective.

Le document d'étude de l'avatar existant (disponible en [Annexe B](#)) est indispensable pour identifier la structure existante et la retravailler. Pour cela, j'ai dans un premier temps réalisé une nouvelle carte heuristique (disponible en [Annexe H](#)) en tentant de distinguer et regrouper les composants de l'avatar par module, dans l'optique d'en déduire une première structure de code.

Pour la conception même de l'architecture logicielle de l'avatar, j'ai décidé de réaliser un diagramme de classes UML. Il s'agit d'un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations.

J'ai d'abord réalisé un premier diagramme simple à partir de la carte heuristique créée sur la structure de l'avatar. Ce diagramme, visible en [Annexe I](#), est une première ébauche de la structure de code du nouvel avatar que j'ai par la suite validée avec l'équipe.

Enfin, à partir de ces éléments et de mon expérience sur des projets Unity, j'ai conçu le diagramme de classes UML final disponible en [Annexe J](#). Il s'agit de l'architecture logicielle complète du projet, couvrant l'ensemble des fonctionnalités à réaliser tout en offrant un nouvel ensemble de code propre et organisé.

### **g. Structure hiérarchique du nouvel avatar dans Unity**

Dans le cadre de projets Unity, l'intégration de la programmation au sein du moteur se fait par des scripts. Ce sont des classes C#, héritant obligatoirement de MonoBehaviour, la classe de base du moteur de jeu. Cela permet aux scripts d'appeler des méthodes propres au moteur, telles que *Start()* appelée au démarrage d'une scène et *Update()* appelée juste avant le rendu de chaque image du jeu.

Les scripts héritant de MonoBehaviour doivent être placés sur des objets présents dans le moteur Unity. Voici une voiture dans une scène Unity.



*Figure 33 : Exemple de scène Unity simple, avec une voiture contrôlable par le joueur.*

Pour déplacer avec une manette de jeu par exemple, il faut lui ajouter un script C# pouvant s'intituler « CarController.cs ». Au lancement de la scène, ce script sera exécuté et appliqué à la voiture. En la dupliquant, plusieurs instances du script seront exécutées simultanément pour contrôler chaque voiture.

Le même principe s'applique à l'avatar. Les classes composant le diagramme de classes UML réalisé pour l'architecture logicielle du projet doivent être présentes sur les objets visibles ou invisibles dans la hiérarchie de la scène Unity. Ainsi, j'ai schématisé la hiérarchie d'objets Unity suivante :



*Figure 34 : Structure hiérarchique du nouvel avatar dans Unity.*

Chaque item de la liste présentée ci-dessus est appelé « GameObject » ou objet. Dans le cadre de l'exemple de la voiture, le véhicule est un GameObject lui-même composé de sous-objets pour le châssis, les roues, etc.

Pour l'avatar, on retrouvera ainsi sur le GameObject « Avatar », les scripts principaux issus du diagramme de classes, tels que *VRPlayer*. Pour le système de téléportation par exemple, on retrouvera simplement le script *Teleporter* sur le GameObject « Teleporter » de l'avatar.

## h. Utilisation de l'avatar par les développeurs

Une des contraintes de la refonte de l'avatar est de permettre une utilisation et configuration simple de l'avatar dans les environnements 3D. Pour cela, des « prefabs » de l'avatar seront mis à disposition des développeurs. Un « prefab » est un semblable à un objet « préfabriqué » que l'on peut simplement intégrer dans une scène Unity en effectuant un glisser-déposer. Pour l'avatar, il s'agit de proposer deux avatars « préfabriqués » pouvant être rapidement intégrés dans les environnements :

- **Prefab « Avatar\_CameraOnly »** : il s'agit uniquement du système de caméra de l'avatar avec les fonctionnalités de calibrage et fondu de transition. L'utilisateur de l'application peut ainsi seulement observer son environnement en réalité virtuelle.
- **Prefab « Avatar »** : il s'agit de l'avatar entier, dont la structure est celle de la figure 34. En l'intégrant dans une scène, on permet à l'utilisateur de se déplacer, de disposer d'un corps virtuel, etc. L'avatar est préconfiguré avec des paramètres d'apparence et interactions possibles par défaut.

Ainsi les développeurs disposent d'avatars préconfigurés qu'ils peuvent aisément intégrer aux environnements, grâce au système de prefab d'Unity. Un autre avantage de ce système est qu'en effectuant une mise à jour d'un des prefabs de l'avatar, toutes ses instances dans tous les environnements de l'application seront mises à jour. Par exemple, si on veut changer l'apparence corporelle masculine de l'avatar, il suffit de modifier le prefab. Il n'est pas

nécessaire de passer par chaque scène du projet pour y modifier l'instance de l'avatar ajoutée par les développeurs.

Une fois intégré dans une scène, l'avatar peut être modifié individuellement pour l'adapter à l'environnement. Pour cela, le développeur doit modifier les paramètres visibles dans Unity du script *VRPlayer*, se trouvant sur le GameObject racine « Avatar » (voir figure 34). Ce choix de conception vise à proposer aux développeurs un seul point d'entrée pour configurer l'avatar dans une scène. Une fois implémenté sous sa forme préconfigurée, de nombreuses options seront offertes pour paramétrer spécifiquement l'avatar, par exemple pour l'empêcher de se déplacer dans un environnement spécifique.

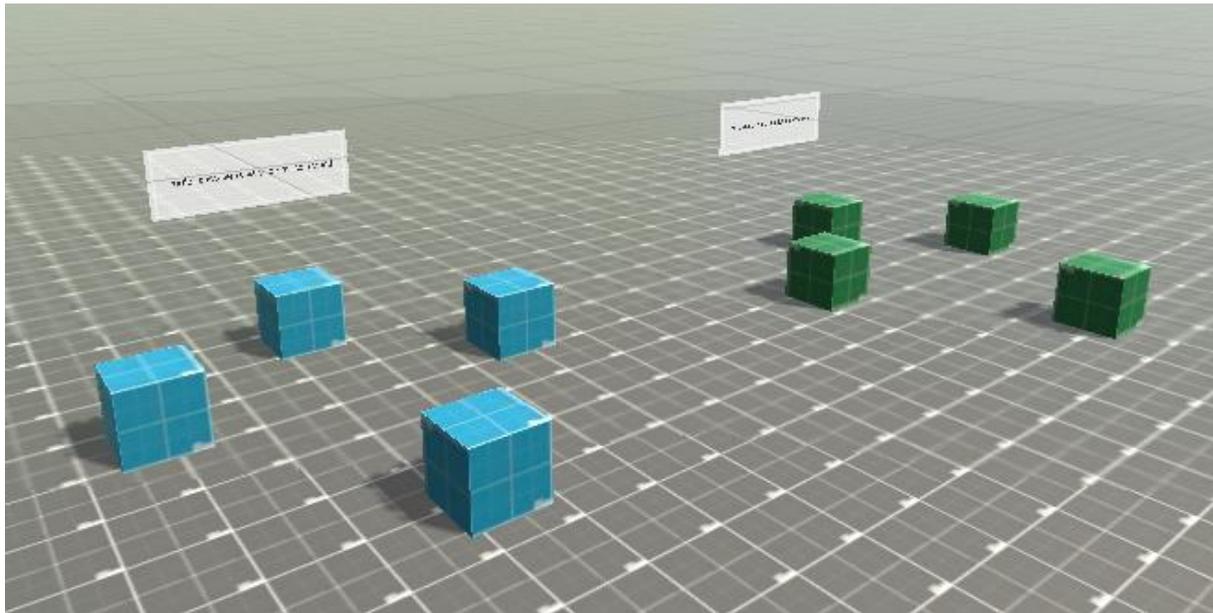
### **3. Réalisation du nouvel avatar**

La phase de conception effectuée permet de cadrer et organiser la réalisation du nouvel avatar. Dans cette partie, je vais présenter les procédés employés pour la réalisation de chaque grande étape de la production de la solution proposée.

#### **a. Préparation de l'environnement de travail**

Comme énoncé précédemment, j'ai choisi de créer le nouvel avatar dans un environnement 3D de test, au sein du projet *C2Packages* étant dédié à la réalisation des composants communs aux applications. J'ai donc créé, au sein du projet, une scène Unity intitulée « AvatarTester » dans laquelle l'avatar sera développé et où chaque fonctionnalité sera associée à un cas de test exécutable.

En suivant les instructions établies lors de la phase de conception, j'ai également développé un framework de test générique qui sera utilisé pour créer un petit cas de test pour chaque fonctionnalité. Un script C# nommé « TestRunner » recense tous les cas de tests et les exécute successivement, suivant le principe de la stratégie de tests détaillé précédemment.



*Figure 35 : Environnement de test créé pour l'avatar.*

Dans la figure ci-dessus, on peut voir un environnement de test générique avec deux cas de tests distincts démontrant simplement comment se servir du framework. Pour chaque test, on associe un nombre de validations requises. Pour le cas des tests de la figure ci-dessus, il y a 4 validations nécessaires : une pour chaque cube. Passer à travers chaque cube bleu permet de réussir le premier test. Il en est de même pour le second test exemple. Un bouton permet sinon de valider ou échouer manuellement un cas de test.

Après avoir créé le framework, j'ai conçu un premier environnement de test des « inputs », correspondant aux entrées provenant des boutons des manettes des casques de réalité virtuelle, pour valider le fonctionnement du framework. Il sera également utile pour valider la compatibilité de l'avatar avec les différents casques VR supportés par C2Care.

## **b. Création de la base de l'avatar**

La création de la base de l'avatar concerne la réalisation des fonctionnalités lui étant essentielles, permettant ensuite de développer la plupart des autres modules de l'avatar.

### ***i. Déplacement dans l'espace virtuel***

Il s'agit certainement de la fonctionnalité la plus importante de l'avatar. La moitié des environnements de C2Care proposent à l'utilisateur se déplacer dans un environnement virtuel. L'avatar doit être dirigeable simplement par l'utilisateur en utilisant les manettes des casques de réalité virtuelle. Parmi les 3 systèmes de déplacement étudiés dans l'état de l'art du projet, le déplacement continu a été retenu car il s'agit de celui étant le plus adapté à un avatar entièrement représenté. De plus, il a été constaté qu'il a impact positif sur le sentiment de présence des utilisateurs, un des points de la problématique du projet.

Ainsi pour le déplacement continu de l'avatar, il faut déplacer la caméra dans Unity, correspondant aux yeux de l'utilisateur. Le système sera associé aux joysticks des manettes :

- **Joystick manette gauche** : translation de l'avatar (incluant la caméra) dans la direction vers laquelle le joystick est orienté.
- **Joystick manette droite** : rotation par à-coups de l'avatar (incluant la caméra), de 35°, vers la droite ou vers la gauche en fonction de l'orientation du joystick. Cette fonctionnalité permet de tourner la tête virtuellement sans le faire physiquement.

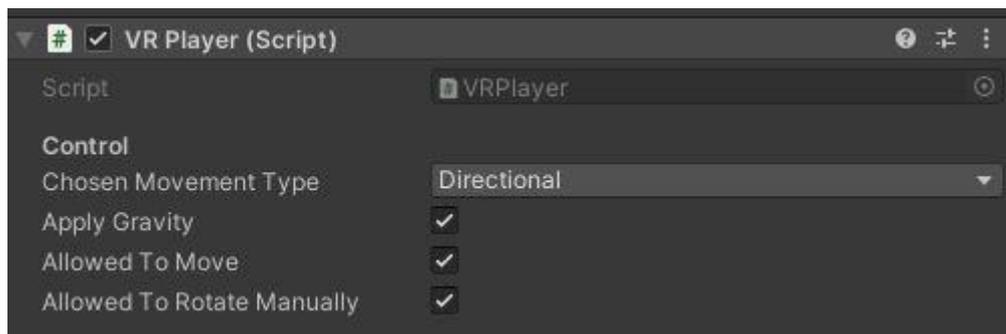
Le système implémenté pour l'avatar est similaire au système employé dans les jeux vidéo à la première personne, tels que les jeux de tir.

Afin de reproduire un mouvement réaliste, la vitesse de déplacement est ajustée en fonction de la direction de déplacement : si on recule le déplacement sera ralenti de moitié, si on marche vers la droite ou la gauche, le déplacement sera ralenti d'un tiers de la vitesse normale.

J'ai programmé tout ce qui est évoqué ci-dessus dans le script dédié *VRPlayerController*, en association avec le composant d'Unity : « *CharacterController* ». Il s'agit d'une brique logicielle propre au moteur de jeu, permettant de gérer des fonctions avancées liées au

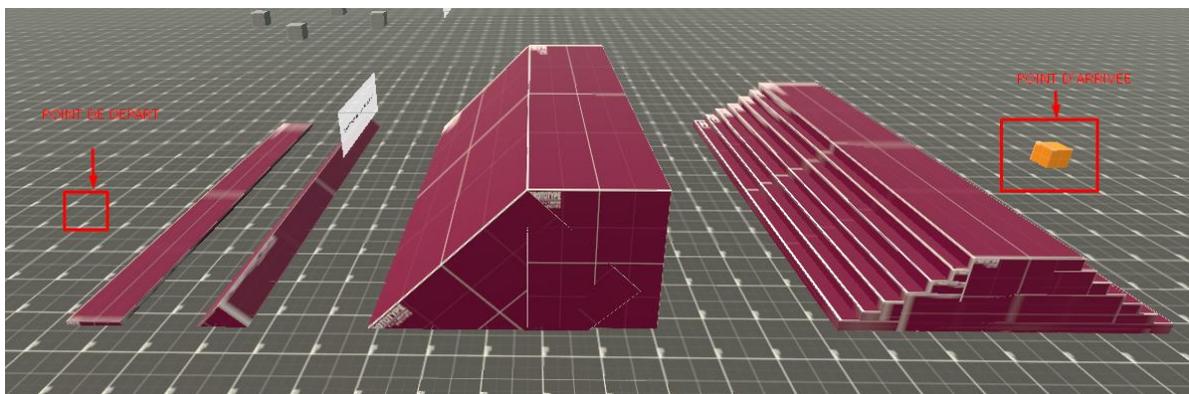
déplacement d'un objet : gestion de collisions, pouvoir monter ou descendre des pentes et escaliers, etc.

Enfin, j'ai ajouté les paramètres prévus lors de la spécification de la fonctionnalité. Ces paramètres sont accessibles directement dans le script *VRPlayer* à la racine de l'avatar pour être modifiés en fonction de l'environnement dans lequel se trouve l'avatar.



*Figure 36 : Script VRPlayer positionné à la racine de l'avatar, configurable en modifiant les paramètres visibles.*

Le script *VRPlayer* évoluera progressivement avec l'ajout de nouvelles fonctionnalités. Afin d'illustrer une dernière fois la stratégie de tests mise en place pour le projet, voici une capture du cas de test réalisé pour l'user story liée au déplacement de l'avatar :



*Figure 37 : Cas de test du déplacement de l'avatar : parcours à traverser.*

En suivant le même principe, des cas de tests seront réalisés tout au long du projet pour les fonctionnalités, mais ne seront pas présentés pour concentrer le contenu du mémoire sur les procédés mis en œuvre lors de la réalisation du nouvel avatar.

## ***ii. Représentation corporelle***

Un autre point essentiel pour la réalisation du nouvel avatar est sa représentation corporelle. Afin d'améliorer le sentiment de présence des utilisateurs, le choix a été fait de proposer une représentation corporelle entière de l'avatar. Avant de s'intéresser à ses variations visuelles (couleur de peau, sexe, degré de représentation), il faut se pencher sur la base.

Dans un premier temps, il faut changer le corps représentant l'utilisateur dans la réalité virtuelle car le corps existant a été jugé techniquement et visuellement en retard sur la technologie. Le corps de l'avatar doit être un modèle 3D spécifiquement préparé pour pouvoir être animé.

C2Care dispose d'une bibliothèque de modèles 3D photoréalistes peuplant les environnements des applications. Il y a une trentaine de personnages virtuels réalistes, hommes, femmes et enfants qui sont utilisés en tant que PNJ (personnages non joueurs) dans les scènes. De plus, ces corps disposent tous d'un squelette, c'est-à-dire d'une structure associant chaque partie du corps à un « os », permettant ainsi de les animer.

Pour choisir un nouveau corps pour l'avatar, j'ai donc étudié les modèles présents dans cette liste, hommes dans un premier temps. J'en ai ensuite sélectionné un, suffisamment générique et dont l'apparence surpasse nettement celle de l'ancien avatar.



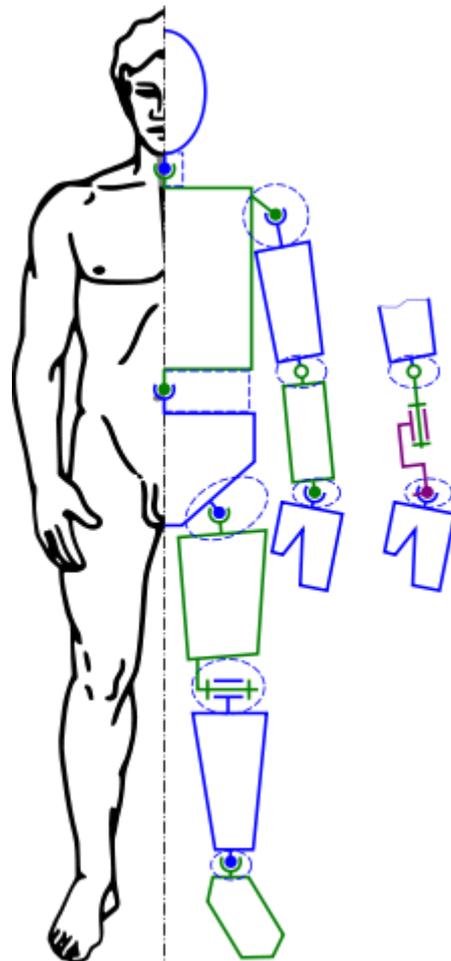
*Figure 38 : À gauche, le modèle de l'ancien avatar masculin.  
À droite, le nouveau modèle choisi.*

Disposer d'un corps n'est que la première étape. Il faut que celui-ci soit animé dynamiquement en fonction de la position et rotation des manettes tenues par l'utilisateur et du casque qu'il porte sur la tête. En tenant la manette droite dans la main droite et tendant le bras, l'utilisateur doit voir son corps virtuel correspondre presque parfaitement à son corps réel.

Animer dynamiquement le corps est une tâche extrêmement complexe si on veut le réaliser entièrement soi-même. Il faut deviner la position et rotation de chaque os du corps pour positionner la main virtuelle au niveau de la main réelle.

Unity propose une solution intégrée au moteur de jeu afin de gérer l'animation dynamique d'un corps. Cette solution s'appuie sur la **cinématique inverse**, désignant l'ensemble des

méthodes de calcul des positions et rotations d'un modèle articulaire afin d'obtenir une pose désirée. La cinématique inverse (souvent abrégée « IK ») permet pour un modèle humain de déterminer la torsion des poignets, des coudes, épaules mais aussi genoux et hanches, le tout en vue d'atteindre une ou des poses données pour certains membres du corps. Dans le cas de l'avatar, il s'agit de la tête dont la position et rotation sont mises à jour en fonction du casque de réalité virtuelle, puis des mains localisées dans l'espace par les manettes.



*Figure 39 : Exemple de chaîne cinématique du corps humain.*

La solution proposée par le moteur de jeu Unity pour appliquer cela à l'avatar est intéressante, cependant, C2Care dispose d'un outil externe acheté, dédié à la cinématique inverse dans Unity. J'ai fait le choix d'utiliser ce second outil car il est beaucoup plus complet que celui proposé nativement par le moteur. De plus, il propose un script intitulé *VR IK* dédié à l'usage de la cinématique inverse pour les avatars de réalité virtuelle. Cet outil est utilisé

dans les plus grandes productions de jeux de réalité virtuelle et propose un support que j'ai sollicité plusieurs fois au cours de l'intégration dans Unity.

Ainsi, en ajoutant le script *VRIK* au nouveau corps de l'avatar, et en associant les os des mains et de la tête aux positions réelles du casque et des manettes dans l'espace, je suis parvenu à obtenir une représentation corporelle correcte. *VRIK* propose de nombreux paramètres qui seront modifiés tout au long du projet pour essayer d'obtenir la représentation corporelle dynamique la plus fidèle de l'avatar par rapport au corps réel.

### ***iii. Animation de déplacement***

Le déplacement de l'avatar et sa représentation corporelle doivent être synchronisés pour animer le corps lorsque l'avatar est en mouvement. L'utilisateur regardant ses pieds lorsqu'il se déplace doit pouvoir constater que son corps virtuel est bien en train de marcher et qu'il n'est pas statique en train de glisser sur le sol. Pour réaliser cela, j'ai étudié deux approches.

Dans un premier temps, j'ai tenté d'utiliser le système de *VRIK* utilisé pour l'animation corporelle dynamique de l'avatar. En effet, ce système propose une option permettant de simuler la position des pieds en fonction d'un changement de position du corps, par exemple lié à un déplacement. Ce système se base sur la cinématique inverse, et propose de nombreux paramètres permettant de configurer une animation de marche procédurale. Toutefois, après de nombreux tests et tentatives de paramétrage, l'équipe et moi avons jugé que le système ne permettait pas d'obtenir un résultat suffisamment réaliste : la position des pieds était quelques fois hasardeuse, l'enchaînement entre les pas n'était pas fidèle à la réalité, etc.

Ainsi, une seconde piste a été explorée, concernant l'utilisation d'un système plus traditionnel : l'animation des jambes du modèle 3D en fonction de la vitesse de déplacement de l'avatar. Ce système permet d'appliquer à l'avatar des animations déjà réalisées en motion capture pour reproduire fidèlement la marche d'un être humain. L'ancien avatar utilisait ce système mais il était peu flexible et peu réaliste : une seule animation de marche vers l'avant était utilisée, même lorsqu'il s'agissait de faire des pas sur les côtés ou de reculer.

Afin de mettre en place ce système pour le nouvel avatar, j'ai déjà identifié la nécessité d'intégrer trois animations pour la marche :

- Marche vers l'avant.
- Marche de côté.
- Marche vers l'arrière.

J'ai ensuite parcouru le catalogue du site Mixamo.com où se trouvent de nombreuses animations de motion capture gratuites et faciles d'intégration dans Unity. J'en ai sélectionné une pour chaque type de marche. Je les ai ensuite connectées à l'avatar par le système d'animation d'Unity et en utilisant un « Blend Tree ». Il s'agit d'un composant d'Unity permettant d'effectuer des transitions progressives entre plusieurs animations, en se basant sur plusieurs paramètres. Dans le cadre de l'avatar, ces paramètres sont sa vitesse sur l'axe X (sur les côtés) et sa vitesse sur l'axe Z (vers l'avant). Ces deux paramètres, vont de -1 à 1 en fonction de la direction et de la vitesse du déplacement de l'avatar, créant ainsi le système suivant :

- Si la vitesse sur l'axe Z est supérieure à 0, alors l'animation de marche vers l'avant est jouée.
- Si la vitesse sur l'axe Z est inférieure à 0, alors l'animation de marche vers l'arrière est jouée.
- Si la vitesse sur l'axe X est supérieure à 0, alors l'animation de marche vers la droite est jouée.
- Si la vitesse sur l'axe X est inférieure à 0, alors l'animation de marche vers la gauche est jouée (animation de marche vers la droite inversée).

L'implémentation de ce système dans Unity permet d'effectuer ces transitions dynamiquement. De plus, grâce au Blend Tree, un mélange d'animations est effectué si les vitesses sur X et Z sont différentes de zéro en même temps, par exemple dans le cas où l'avatar se déplace en diagonale.

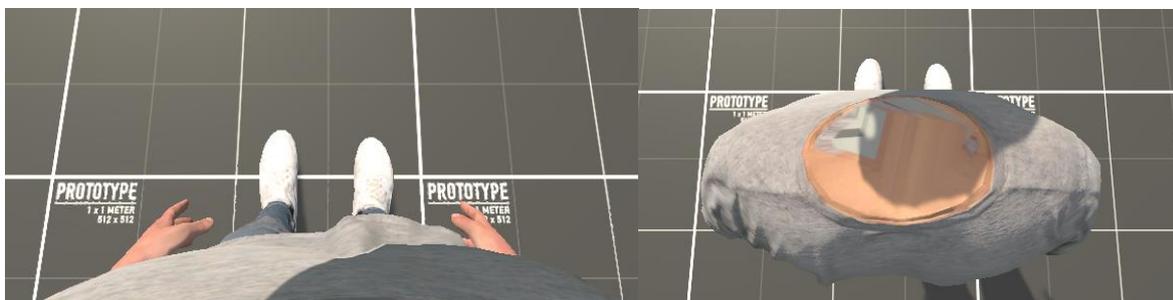
Le résultat de ce système propose une animation de marche satisfaisante bien qu'il faille noter que de nombreux tests, paramétrages et changement d'animations ont été nécessaires avant d'obtenir un résultat final. En effet parmi les difficultés rencontrées, il y a eu la difficulté de parfaitement adapter la vitesse de l'animation à la vitesse de déplacement.

### c. Calibrage et positionnement

Cette partie est dédiée aux fonctionnalités de calibrage de l'avatar et à ses variations de position. Il s'agit là de présenter les procédés mis en œuvre afin de permettre l'alignement de la caméra (ce que voit l'utilisateur avec le casque) par rapport au corps virtuel de l'avatar, le paramétrage de la taille de l'avatar et enfin le système poses permettant, par exemple, d'asseoir l'avatar.

#### i. Calibrage

Le calibrage est une fonctionnalité essentielle aux expériences de réalité virtuelle. En effet, les casques VR ont leur position traquée dans l'espace réel puis reproduite dans l'espace virtuel. Ainsi, si on avance d'un mètre dans le monde réel, on aura avancé d'un mètre dans le monde virtuel. Cela implique que si on démarre une scène en étant assis puis qu'on se lève, alors la hauteur de la caméra par rapport au sol virtuel aura aussi augmenté. Cela peut causer comme effet de se retrouver plusieurs dizaines de centimètres au-dessus du corps virtuel.



*Figure 40 : À gauche ce que voit un utilisateur bien calibré regardant ses pieds.*

*À droite, ce que peut voir un utilisateur mal calibré.*

Pour répondre à cette problématique, il faut mettre en place un système de calibrage de la caméra virtuel efficace pour qu'en effectuant cette action, la vue de l'utilisateur soit recentrée au bon endroit (au-dessus du cou du corps de l'avatar) et orientée dans la bonne direction (où l'orientation du corps est cohérente par rapport à la vue : une épaule de chaque côté et le corps bien dans l'axe si l'utilisateur regarde ses pieds).

La structure hiérarchique de l'avatar dans Unity permet de connaître les écarts de position et de rotation réalisés par la caméra, à chaque frame (image rendue par seconde), par rapport à la position et rotation normale la caméra au niveau de la tête de l'avatar. Une des contraintes de la réalité virtuelle dans Unity est que la position et la rotation de la caméra sont écrasées par les valeurs du casque VR. Il est alors impossible de directement modifier la caméra, cependant on peut le faire en modifiant ses objets parents. Ainsi, pour recalibrer la caméra, on compense les écarts de position et de rotation de la caméra en positionnant le parent de la caméra à l'opposé de la caméra. On applique le même principe à la rotation de la caméra.

Dans le cas de la deuxième capture de la figure précédente, on peut imaginer qu'il y a un écart de position de 2 unités sur l'axe Y et pour la rotation de  $90^\circ$  sur l'axe X. Alors on affecte sur Y la position -2 au parent la caméra, ce qui va donc replacer la caméra au niveau du cou de l'avatar. Pour la rotation, on applique  $-90^\circ$  au parent de la caméra sur l'axe X ce qui va orienter la vue vers l'horizon.

J'ai également pris en compte certains cas de calibration particuliers, avec par exemple la calibration horizontale. Il s'agit d'appliquer le même principe que pour la calibration normale, à l'exception que l'horizon est placé pour être toujours face à la vue après un recalibrage, au lieu de le placer au niveau de l'horizon réel. J'ai aussi pris le temps de mettre en place un système de fondu qui permettra de recalibrer la caméra pendant un court écran noir et ainsi rendre la transition plus agréable pour l'utilisateur.

## ii. *Systeme de taille*

L'avatar virtuel a pour objectif de s'adapter à tous les utilisateurs. Il est déjà important de mettre en place un système de taille permettant de faire correspondre les proportions du corps virtuel aux proportions du corps réel de l'utilisateur.

La taille est un paramètre qui peut être modifié à tout moment pour l'avatar. Changer la taille de l'avatar implique plusieurs choses :

- Modifier l'échelle du corps virtuel.
- Ajuster la position de la caméra pour correspondre à la nouvelle position des yeux.
- Mettre à jour les paramètres du « CharacterController », composant présenté précédemment et notamment lié à la gestion des collisions de l'avatar.

Afin de mettre à jour la taille de l'avatar à l'aide d'un simple paramètre reçu, voici la démarche mise en œuvre. Les paramètres initiaux des composants concernés par un changement de taille sont enregistrés au lancement de l'environnement dans l'application. L'échelle des corps est modifiée en appliquant la formule suivante :

$$\text{nouvelle échelle} = \frac{\text{nouvelle taille} * \text{échelle initiale}}{\text{taille initiale}}$$

Pour ce qui est de la position de la caméra, sa position locale (position au sein de l'objet parent) sur l'axe Y est simplement affectée à la nouvelle taille. Par exemple, si l'avatar passe à 1m95 alors on affectera une position locale sur Y de la caméra à 1,95. Comme dit précédemment, on ne peut pas modifier directement la caméra. La nouvelle position sera donc affectée à son objet parent.

Enfin, pour ce qui est du « CharacterController », gérant les collisions de l'avatar sous la forme d'une capsule invisible l'entourant, il faut ajuster rayon, taille et centre de cette capsule en fonction de la taille. Pour ce qui est de la taille de la capsule, on lui affecte simplement la nouvelle taille de l'avatar.

Pour ce qui est de son rayon, on applique la formule suivante :

$$\text{nouveau rayon} = \frac{\text{nouvelle taille} * \text{rayon initial}}{\text{taille initiale}}$$

Pour bien positionner le centre de la capsule afin que son point le plus bas soit bien au niveau du sol, on modifie sa position sur l'axe Y en appliquant la formule suivante :

$$\text{nouvelle position sur Y} = \frac{\text{nouvelle taille} * \text{position initiale sur Y}}{\text{taille initiale}}$$

### iii. Poses de l'avatar

Le système de poses de l'avatar vise à permettre, sans chargement nécessaire, de modifier la pose dans laquelle se trouve l'avatar. Par exemple, on peut avancer vers une chaise, puis déclencher un changement de pose pour asseoir l'avatar en modifiant son animation.

Un tel système implique beaucoup de paramétrage. Par exemple, l'utilisateur ne doit pas pouvoir se déplacer une fois en pose assise. De plus, ce système doit permettre d'ajouter de nouvelles poses facilement. La solution mise en place concerne l'utilisation du système d'animation d'Unity. Un composant « Animator » permet d'éditer un graphique organisant plusieurs animations entre elles. Pour chaque nouvelle pose à ajouter à l'avatar, il faudra ajouter un élément à ce graphe.

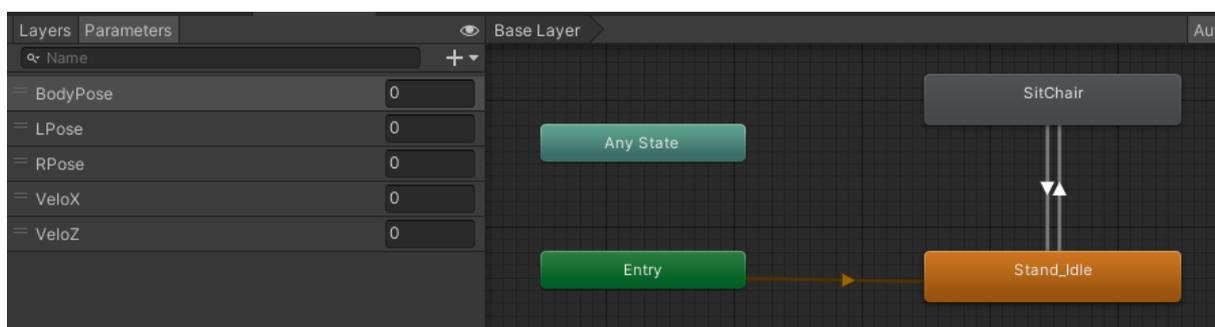
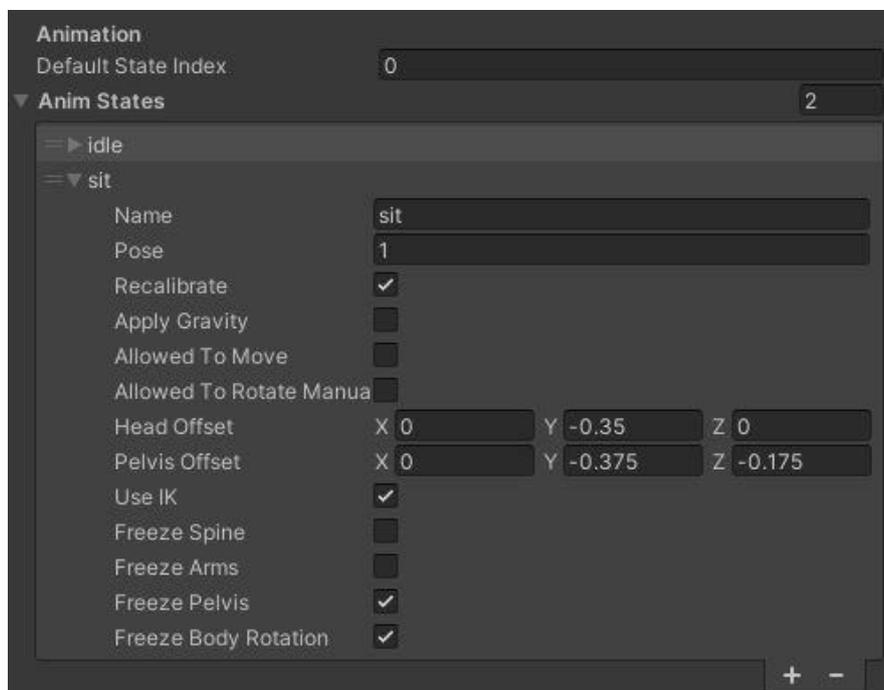


Figure 41 : Partie dédiée au système de poses de l'Animator de l'avatar.

Dans chaque état ajouté, on doit associer un clip d'animation correspondant à la pose à appliquer. Ensuite, il faut créer une transition entre les états se basant sur un entier : « BodyPose ». À chaque état est ensuite associée une valeur, en commençant par 0 pour l'état par défaut, puis 1, 2, etc. En modifiant ce paramètre depuis un script, la transition d'une pose à une autre sera faite automatiquement.

Pour ce qui est des paramètres spécifiques à une pose, tels que la possibilité de se déplacer, la pose devra être clairement définie dans *VRPlayer*. Tous ces paramètres seront appliqués à l'avatar au moment de jouer la pose.



*Figure 42 : Paramètres de la pose « sit » correspondant à l'avatar assis sur une chaise.*

#### **d. Système d'interactions**

Le système d'interactions de l'avatar se trouve au cœur de la problématique à laquelle il tente de répondre. Il a été constaté par l'étude bibliographique de l'avatar que les interactions en réalité virtuelle représentent un des facteurs majeurs influant sur le sentiment de présence généré pour les utilisateurs. Plusieurs types d'interactions doivent être implémentés pour

l'avatar : les interactions non physiques (interagir avec des menus ou des objets par le moyen d'un laser), les interactions physiques (pouvoir appuyer sur les boutons avec les mains de l'avatar par exemple) et le système de « grab », permettant de prendre des objets en main et de les lancer.

Globalement, l'objectif est de permettre à l'avatar d'utiliser les types d'interactions évoqués dans l'état de l'art du projet : interactions directes de l'utilisateur, contrôles physiques, contrôles virtuels.

### ***i. Interactions « non physiques »***

Les interactions non physiques sont particulièrement utiles pour permettre à l'utilisateur d'interagir avec des menus. À l'aide d'un laser déjà réalisé par l'équipe de C2Care plusieurs mois avant le début de ce projet, il est possible de pointer des options d'interface utilisateur avec précision et de les actionner.

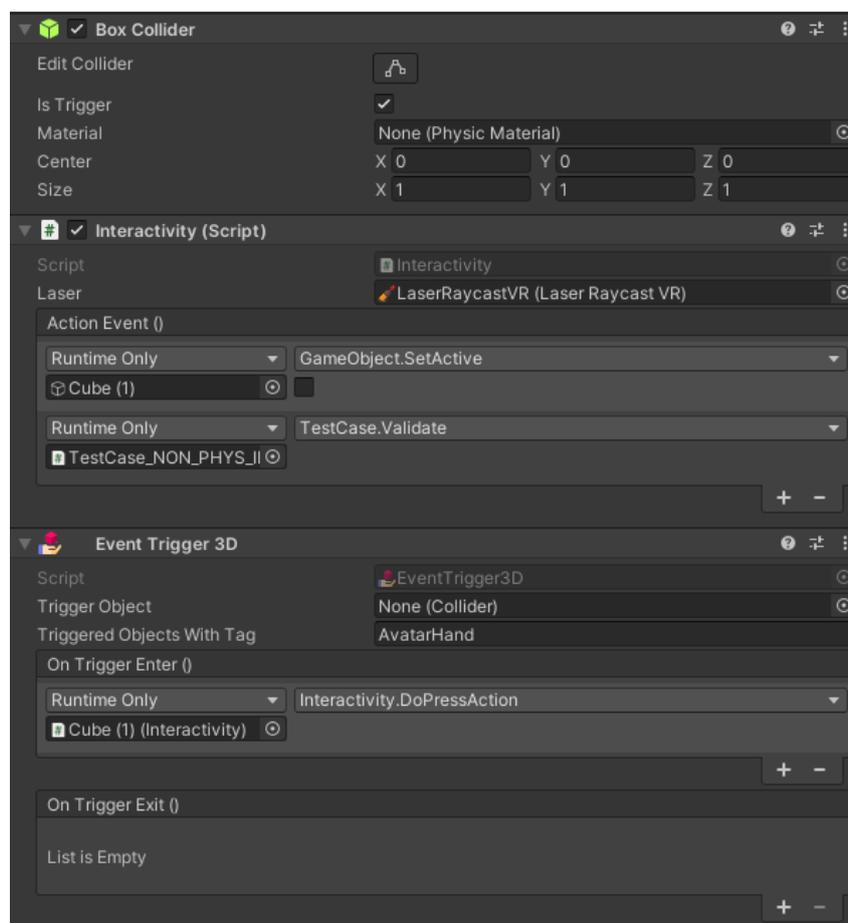
Pour cette partie, je n'ai pas eu beaucoup de travail à faire. Il a suffi d'intégrer le système de laser existant aux scènes où se trouve l'avatar. Le laser fonctionne globalement avec le rendu d'une ligne droite partant d'une des deux manettes. Appuyer sur un des boutons de la manette permet d'envoyer un signal d'action à l'objet se trouvant sous le pointeur. Si celui-ci prend en compte le système de laser, il réceptionnera l'action et effectuera les traitements pour lesquels il a été programmé.

### ***ii. Interactions « physiques »***

Les interactions physiques concernent davantage l'avatar car ce sont les mains du corps virtuel qui vont être au centre du système. Ici, l'idée est de permettre aux utilisateurs de reproduire des interactions « réelles » avec leur avatar virtuel. L'exemple le plus parlant est l'action d'appeler l'ascenseur pour les environnements de claustrophobie de C2Care.

Pour mettre cela en place, j'ai réutilisé et adapté le système en place pour l'avatar précédent. Dans un premier temps, il faut ajouter des boîtes de collisions aux mains de l'avatar, ce qui permettra leur détection par d'autres objets. Il s'agit de la seule configuration à effectuer sur l'avatar même. Dans un second temps, il faut configurer les objets avec lesquels on veut que l'avatar puisse interagir. Pour cela, deux scripts C# sont utilisés en corrélation avec la boîte de collisions de l'objet :

- **EventTrigger3D** : script générique de C2Care permettant de déclencher des actions lorsqu'un objet entre dans la boîte de collisions de l'objet sur lequel le script se trouve.
- **Interactivity** : autre script générique, permettant de définir une ou plusieurs actions à effectuer lors d'un appel à une des méthodes du script.



*Figure 43 : Configuration d'un objet avec lequel on peut interagir physiquement.*

Ainsi, lorsqu'une des mains de l'avatar entre dans la boîte de collisions de l'objet avec lequel interagir, les actions définies dans le script *Interactivity* sont déclenchées. Ce système permet également d'unifier les interactions physiques et non physiques de l'avatar en référençant le laser de la scène dans le script *Interactivity*. Ainsi, la même action sera déclenchée si on touche physiquement l'objet ou si on le pointe avec le laser et on appuie sur un des boutons d'action.

### **iii. Attraper et lancer des objets**

En plus des deux systèmes d'interactions, l'avatar doit permettre à l'utilisateur d'attraper des objets préconfigurés à cet effet, et de les lancer de manière réaliste. Cette fonctionnalité, souvent appelée « grab » est au cœur des expériences de réalité virtuelle depuis la sortie des premiers casques disposant de manettes traquées dans l'espace. Il s'agit également d'un axe d'amélioration majeur par rapport à l'ancien avatar, tant pour les utilisateurs que pour les développeurs souhaitant ajouter la possibilité d'attraper des objets dans les scènes 3D.

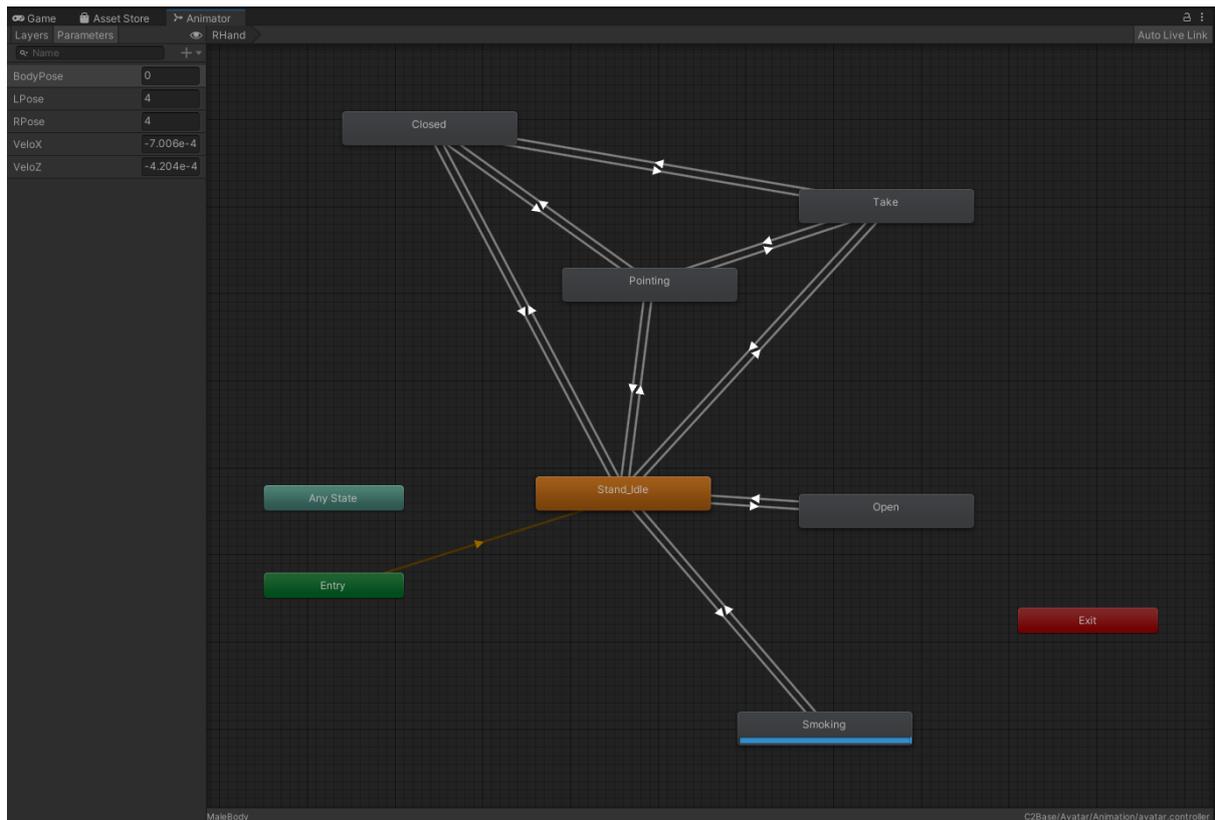
Chaque main de l'avatar est gérée par un script *VRHand* qui permet de la traquer dans l'espace. Ce script permet également de mettre en place le système de « grab ».

Tout d'abord, j'ai créé un système de configuration d'objets attrapables représenté par le script *VRGrabbable*, fonctionnant en corrélation avec une boîte de collisions et un « RigidBody ». Ce dernier composant natif d'Unity permet d'appliquer de la physique aux objets. Grâce à *VRGrabbable*, on associe éventuellement des paramètres de position et de rotation à adopter dans la main de l'utilisateur, puis on choisit une pose pour la main à appliquer au moment où on attrape l'objet. Lorsque l'utilisateur appuie sur un bouton à une distance donnée d'un objet attrapable, on amène celui-ci vers la main d'où provient l'interaction, puis la pose choisie de la main est jouée.



*Figure 44 : Avatar tenant une cigarette configurée en tant qu'objet attrapable .*

Tout comme le système de pose de l'avatar entier présenté précédemment, permettant par exemple d'asseoir l'avatar, le composant « Animator » d'Unity associé à l'avatar est une fois de plus utilisé. Le même principe est appliqué : un paramètre propre à chaque main permet d'appliquer la pose souhaitée. Afin de modifier chaque main indépendamment, un « masque de couche » a été appliqué pour que l'animation affectée à chaque pose ne s'applique qu'à la main souhaitée.

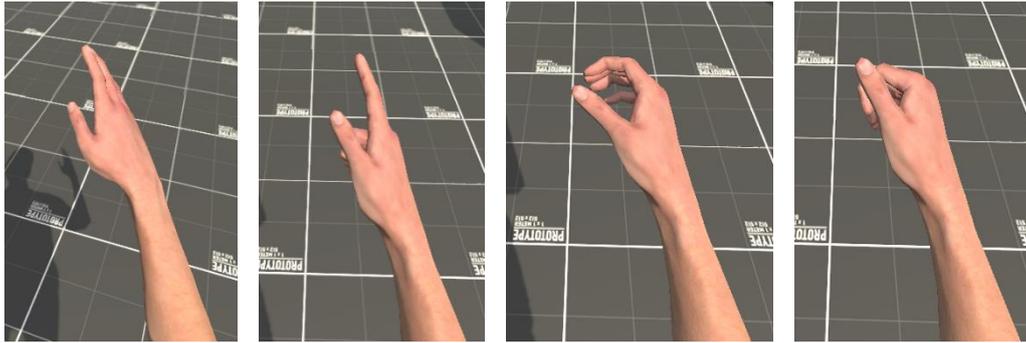


*Figure 45 : Système d'Animator employé pour gérer les différentes poses des mains.*

Pareillement au système de pose de l'avatar entier, ajouter de nouvelles poses pour les mains afin de les adapter à de nouveaux types d'objets attrapables doit être simple. Dans ce cas, il suffit d'ajouter un état à l'Animator, puis de lui affecter l'animation souhaitée et enfin, ajouter les transitions avec une nouvelle valeur du paramètre « LPose » et « RPose ».

#### **iv. Animation des mains**

Dans l'optique d'ajouter un degré de réalisme et d'interactivité supplémentaire, j'ai ajouté la possibilité de changer la pose de la main en fonction des boutons sur lesquels l'utilisateur appuie. Les poses des mains démontrent à l'utilisateur qu'il peut interagir avec l'environnement, et permettent en plus de faire correspondre davantage les mains virtuelles à ses mains réelles.



*Figure 46 : Poses d'une main lors d'inputs utilisateur, respectivement : aucun bouton, gâchette intérieure seulement, gâchette extérieure seulement et les deux en même temps.*

### **e. Variations visuelles de l'avatar**

Ce projet a pour but de proposer un nouvel avatar configurable pour ressembler le plus fidèlement possible aux utilisateurs. Cette notion est importante à développer pour assurer que l'utilisateur puisse s'identifier à son corps virtuel. L'état de l'art a démontré l'importance de distinguer des avatars hommes et femmes, mais aussi de proposer plusieurs couleurs de peau ainsi que plusieurs degrés de représentation de l'avatar.

Un des points abordés lors de la conception était la corpulence de l'avatar. Cela n'a pas été réalisé pour des raisons de complexité et donc, de délais et de budget.

#### ***i. Genre de l'avatar***

D'après l'étude bibliographique présentée au début de ce mémoire, il est difficile pour les hommes de s'identifier à un corps virtuel féminin. L'inverse s'applique également. Il faut alors proposer, tel que l'ancien avatar l'a fait, une représentation corporelle masculine et féminine.

Tout comme pour le choix d'un avatar masculin, j'ai parcouru les différents corps virtuels photoréalistes à disposition dans les dossiers de C2Care, avant d'enfin en sélectionner un.

Toutefois, je me suis vite heurté à un problème : les bras et mains des deux corps n'étaient pas aux mêmes proportions. Ainsi, positionner correctement un objet dans la main de l'avatar homme le positionnait de manière décalée pour l'avatar femme. À cause de cela, ainsi que d'autres problèmes notamment liés à la qualité des mains du modèle, j'ai dû effectuer un autre choix en comparant à chaque modèle féminin testé, les proportions des membres avec l'avatar masculin. Je suis enfin parvenu à trouver un modèle d'apparence suffisamment générique pour faire office d'avatar femme, et aux proportions correctes.



*Figure 47 : À gauche, le corps féminin de l'ancien avatar. À droite, celui du nouvel avatar.*

Les têtes des modèles ont été retirées pour les mêmes raisons que cela avait été fait pour l'ancien avatar : ils obstruent la vue provenant de la caméra s'y trouvant.

## **ii. Couleurs de peau**

Proposer plusieurs couleurs de peau pour les avatars est un pas de plus pour l'adapter au plus grand nombre d'utilisateurs possible. L'ancien avatar proposait quatre couleurs de peau. Le nouveau en propose désormais six. Le procédé employé est le même que pour l'ancien avatar. Les modèles 3D des corps homme et femme utilisent une texture : une image enveloppée tout autour des modèles afin d'y ajouter des couleurs. Pour chaque corps, 6 textures ont été créées par le graphiste de l'entreprise, que j'ai ensuite intégrées dans un système permettant de modifier la texture à appliquer à l'avatar en temps réel.

Chaque texture est associée à une couleur peau. La divergence se situe au niveau des zones où se trouve la peau de l'avatar. Une teinte a été appliquée afin d'en modifier la couleur.



*Figure 48 : Les six couleurs de peaux disponibles appliquées à l'avatar homme.*

## **iii. Degrés de représentation**

Le choix du degré de représentation de l'avatar s'est porté sur l'avatar complet. Toutefois, certaines expériences ont été jugées comme adaptées à une représentation des mains seulement, alors que d'autres nécessitent simplement de ne pas du tout représenter l'avatar.

J'ai ajouté trois degrés de représentation de l'avatar, avec la possibilité de les changer en temps réel :

- Avatar entier.
- Avatar avec mains seulement.
- Aucun avatar.

Pour ce qui est de la réalisation des degrés « avatar entier » et « aucun avatar » : pour le premier, le système est déjà en place alors que pour le second, il suffit de désactiver les corps dans Unity depuis un script.

Pour ce qui est du mode « mains seulement », j'ai d'abord demandé au graphiste de C2Care de retravailler les modèles des corps de l'avatar pour en découper les mains tout comme cela avait été fait pour la tête. Ensuite, en passant en mode « mains seulement », le rendu de l'avatar global est caché pour ne laisser apparaître que les mains flottantes du modèle.



*Figure 49 : Degré de représentation « mains seulement » de l'avatar homme.*

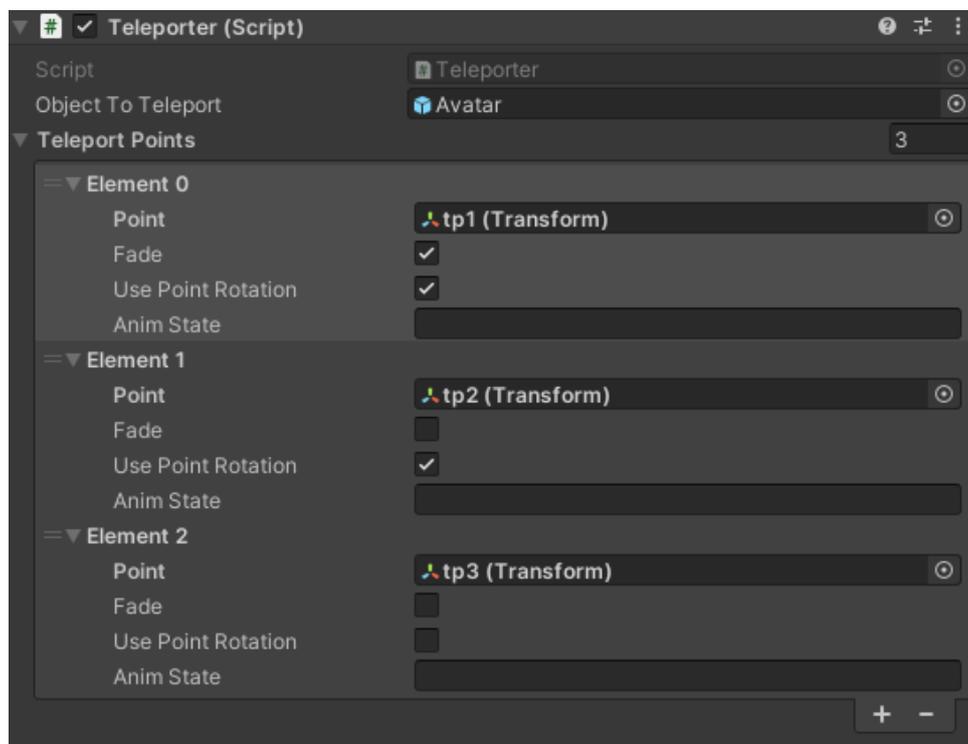
## **f. Fonctionnalités mineures et extensions**

Quelques fonctionnalités mineures et extensions présentes sur l'ancien avatar doivent être ajoutées au nouvel avatar. Le seul véritable changement ici se trouvera dans la structure d'intégration de ces éléments au sein de l'avatar. Toutefois, j'ai effectué une refonte du système de téléportation de l'avatar afin de l'adapter davantage au nouvel avatar ainsi qu'aux environnements ayant évolués.

## *i. Système de téléportation*

Le système de téléportation est le dispositif qui permet de transporter instantanément l'utilisateur d'un point à un autre d'un environnement, lorsque cela a été préalablement configuré. J'ai effectué une refonte de ce système en simplifiant et clarifiant le code dans un premier temps. Améliorer l'avatar et ses composants pour les développeurs de C2Care est une des problématiques importantes du projet. J'ai alors essayé d'y répondre le mieux possible tout au long du projet, y compris sur le téléporteur en le retravaillant.

L'objet principal de la refonte, en plus d'améliorer la qualité du code, est de rendre le système plus flexible en permettant de paramétrer davantage chaque point de téléportation. Ainsi, avant où le paramétrage d'un point de téléportation était limité à une position dans l'environnement, associé à un index de tableau non modifiable, de nouvelles options ont été ajoutées.

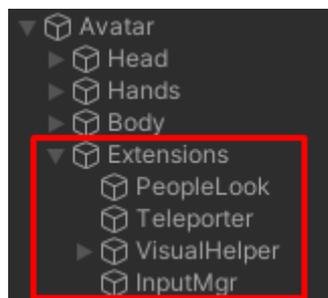


*Figure 50 : Paramétrage proposé par le nouveau système de téléportation.*

La téléportation s'effectue en modifiant la position de l'avatar dans l'environnement, éventuellement avec un fondu de transition, en appliquant ou non la rotation du point de téléportation et enfin, en appliquant une pose d'avatar entier si le champ correspondant est rempli.

## ii. *Intégration des extensions*

Pour ce qui est des extensions et autres fonctionnalités mineures, je n'ai pas eu à effectuer beaucoup de modifications de code. J'ai juste retravaillé leur intégration au sein du nouvel avatar.



*Figure 51 : Intégration des extensions au sein de la hiérarchie de l'avatar dans Unity.*

Parmi ces extensions, on compte :

- **PeopleLook** : système permettant de déclencher le regard des PNJ vers l'avatar lorsqu'il passe à côté d'eux.
- **Teleport** : système de téléportation de l'avatar.
- **VisualHelper** : affichage d'aides visuelles à la respiration avec un motif apparaissant dans le casque, accompagné d'effets de particules synchronisés avec un rythme de respiration.
- **InputMgr** : système de détection d'inputs (entrées utilisateur provenant des manettes principalement).

J'ai toutefois légèrement optimisé le système de *PeopleLook* pour alléger l'avatar d'une boîte de collisions non nécessaire.

### g. Communication avec le contrôleur my.c2.care

Les applications de réalité virtuelle peuvent être connectées au site web *my.c2.care* pour piloter les expositions à distance. Ainsi lors d'une séance, le praticien paramètre l'environnement de réalité virtuelle dans lequel est exposé son patient. Il peut alors influencer en temps réel sur :

- L'environnement dans lequel le patient se trouve.
- Sa position dans l'environnement.
- Le déclenchement d'actions spécifiques aux environnements.
- La configuration de paramètres communs : aide visuelle, insistance des regards, etc.
- Enfin, la configuration de l'avatar virtuel du patient.

L'interface dédiée à la configuration de l'avatar permet de choisir son sexe, sa taille ainsi que sa couleur de peau. Le nouvel avatar ajoute quelques options supplémentaires : un mode de représentation « mains seulement » de l'avatar, ainsi que l'ajout de deux nouvelles couleurs de peau. J'ai donc réalisé une maquette que j'ai communiquée à un développeur web de C2Care afin qu'il effectue les ajouts demandés par rapport à l'avatar.



*Figure 52 : Interface de configuration de l'avatar mise à jour dans my.c2.care.*

Le système de communication entre les applications de réalité virtuelle et le site web se base sur un système réseau complexe qui ne sera pas présenté en détail ici. On peut toutefois noter que la finalité de ce système offre un mode de communication simplifié se basant sur l'envoi de messages d'un appareil à l'autre. Chaque message est identifié par une clé : une chaîne de caractères dont la nomenclature symbolise au mieux l'objet du message. À chaque clé est ensuite associée une valeur pouvant être de trois types différents : booléen (« bool »), chaîne de caractères (« string ») ou nombre flottant (« float »).

Par exemple, dans le cas de la configuration de l'avatar avec l'interface *my.c2.care*, les messages suivants sont communiqués :

Paramètre à modifier	Clé	Type de valeur	Valeurs possibles
Personnage	avatar_visual	string	Une des valeurs suivantes : <ul style="list-style-type: none"> <li>• woman</li> <li>• man</li> <li>• woman_hands</li> <li>• man_hands</li> <li>• none</li> </ul>
Taille	avatar_size	float	1,75 pour 1 mètre 75
Couleur de peau	avatar_skin	float	Flottant converti en entier allant de 0 à 5 pour chaque couleur de peau

*Figure 53 : Détail des messages échangés entre my.c2.care et les applications VR au sujet de l'avatar.*

Ces messages sont interprétés par le nouvel avatar avec le script *VRRemoteHandler*. Ce script fait appel au système de réseau existant pour « écouter » les messages ayant les clés

associées à l'avatar. Lorsqu'un message est reçu, une méthode de *VRPlayer* est appelée pour mettre l'avatar à jour par rapport à la clé et à la valeur reçue.

## 4. Déploiement

Une fois l'avatar réalisé dans C2Packages, son intégration à toutes les applications où il doit être présent est la dernière étape du projet. Dans un premier temps, le développement effectué sur le nouvel avatar doit être intégré au flux de développement continu de C2Packages, avant d'être exporté puis importé dans les projets Unity concernés.

### a. Intégration de l'avatar au flux de développement continu

La réalisation de l'avatar s'est effectuée dans C2Packages, où l'intégralité du développement a été versionné dans une branche nommée « feat/newAvatar ». Comme expliqué plus tôt dans le mémoire, cette méthode de gestion de versions m'a permis de concevoir l'avatar sans bloquer pour autant le flux de développement continu, représenté par la branche « develop ».

L'avatar terminé doit toutefois être intégré à ce flux de développement continu avant de faire l'objet d'un export vers les autres projets où il doit être intégré.

À la suite d'un passage de documentation et nettoyage du code, j'ai donc créé une « Pull request », correspondant à une demande de fusion de mon travail avec le flux commun sur GitHub où le projet est hébergé.



*Figure 54 : Pull request effectuée pour intégrer « feat/newAvatar » à « develop ».*

On compte 211 « commits » correspondant à tout mon travail sur la réalisation du nouvel avatar dans Unity, débutant du 1<sup>er</sup> juin 2021 au 12 juillet 2021. Chaque « commit » correspond à une sauvegarde du projet dans son état au moment de sa réalisation.

La fusion s’est effectuée sans soucis particuliers, préparant ainsi l’export du nouvel avatar.

## **b. Stratégie de déploiement**

Dix applications doivent être mises à jour pour intégrer le nouvel avatar.

Je l’ai déployé dans les applications de C2Care en suivant la stratégie de déploiement de l’avatar suivante :

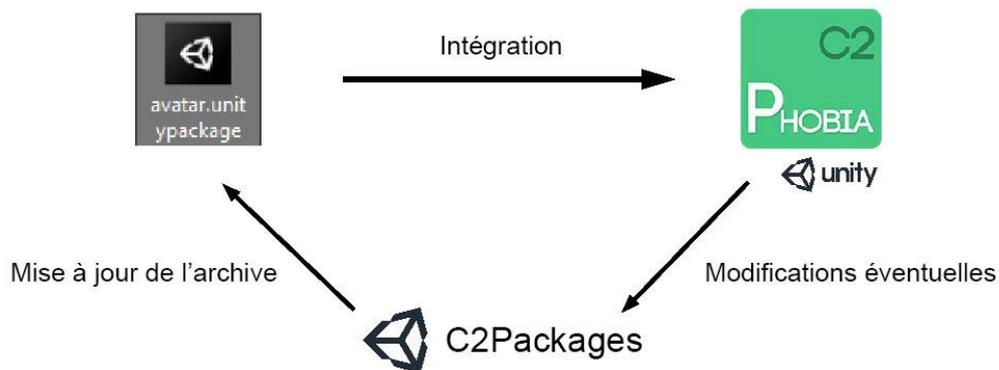
1) Test complet de l'avatar avec la scène "AvatarTester"



2) Export de l'avatar dans une archive "avatar.unitypackage"



3) Intégration dans C2Phobia, la plus grande application de C2Care



4) Intégration dans toutes les autres applications

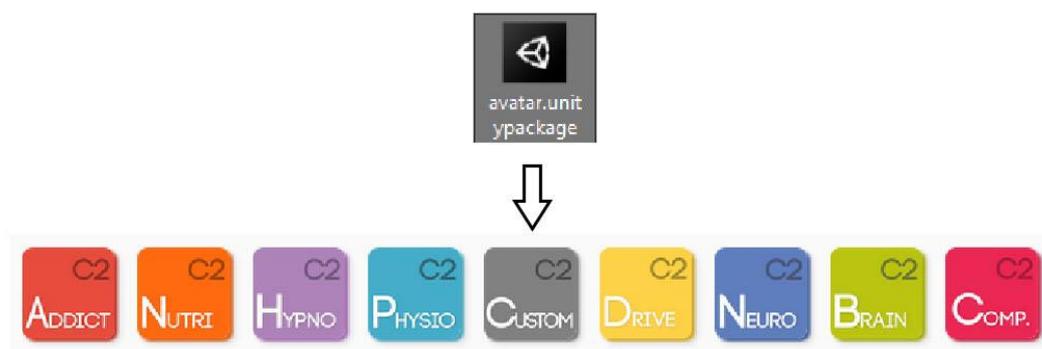


Figure 55 : Schéma récapitulatif de la stratégie de déploiement de l'avatar.

## VI. Résultats et discussions

---

Le plan initial du projet était de créer un avatar virtuel reprenant l'existant, en y apportant des améliorations diverses principalement destinées aux utilisateurs mais aussi aux développeurs. En m'appuyant sur un constat personnel et sur la recherche bibliographique effectuée, j'estime que les objectifs ont été atteints.

Du point de vue développement, le nouvel avatar bénéficie d'une structure de code solide, flexible et bien documentée. Un travail conséquent a été effectué pour permettre l'intégration et la configuration simple de l'avatar dans tout environnement 3D souhaité. La refonte de certains systèmes internes permet de nouveaux comportements auparavant indisponibles. Parmi cela, on compte par exemple un système de poses du corps virtuel dynamique, un système de téléportation plus flexible, la prise d'objets ayant entièrement été recréée pour une configuration plus simple, la création d'un point d'entrée unique pour paramétrer l'avatar en temps réel ou préalablement à l'exécution d'une scène, etc.

Du point de vue utilisateur, il propose une amélioration globale du rendu du corps virtuel, bénéficiant également d'un niveau d'adaptabilité encore supérieur. Il est possible de modifier le sexe, la couleur de peau et la taille de l'avatar. En plus du système d'animation dynamique des membres du corps et d'un rendu réaliste de la marche, l'appropriation de l'avatar par les utilisateurs est fortement améliorée.

Le nouvel avatar a été testé par plusieurs membres de C2Care ne faisant pas partie de l'équipe de développement mais ayant déjà utilisé la solution disposant de l'ancien avatar. Plusieurs constats ont pu être faits :

- La prise en main du nouvel avatar semble instantanée. Les utilisateurs commencent par regarder leurs bras virtuels et sont impressionnés par le réalisme des mouvements du corps virtuel. L'hypothèse faite dans l'état de l'art concernant l'impact de l'avatar dans la réalité virtuelle s'est confirmée.

- Attraper plusieurs objets en même temps avec un changement de pose dynamique des mains a été l'amélioration ayant eu le plus de succès. L'ancien avatar avait de véritables lacunes qui ont été comblées à ce niveau-là.
- L'animation de marche semble laisser relativement indifférent. Il s'agit de quelque chose de « normal » pour les utilisateurs, là où le mouvement dynamique des bras semble être beaucoup plus impressionnant. Ce constat de normalité de la fonctionnalité est une bonne chose car cela peut signifier que l'utilisateur s'est approprié le mouvement du corps sans que cela ne nuise à son immersion.
- Le degré de représentation de l'avatar « mains seulement » n'a pas reçu l'accueil attendu par les utilisateurs. Après avoir fait l'expérience du corps entier de l'avatar, le mode « mains seulement » leur est paru bien moins satisfaisant. Il avait pourtant été fait pour hypothèse que les deux modes de représentation pouvaient générer un sentiment de présence à peu près égal car la différence réelle se trouvait entre l'absence totale d'avatar et la présence d'au moins quelque chose de minimaliste, tel que les mains. Il se peut que l'avatar présenté dans l'étude bibliographique ne se démarquait pas suffisamment de la seule présence des mains, alors que cela peut être le cas avec le nouvel avatar de C2Care.

Il faudrait étendre ces tests à un public plus grand afin d'essayer de confirmer les résultats obtenus. Cela permettrait ainsi de retravailler éventuellement certaines parties de l'avatar mais aussi confirmer davantage les hypothèses littéraires étudiées lors de la réalisation de l'état de l'art du projet.

## VII. Conclusions

---

La refonte de l'avatar s'est effectuée sur une période totale de deux mois et demi, débutant par un travail préparatif le 18 mai 2021 et s'achevant avec la fin de son déploiement le 30 juillet 2021. Le nouvel avatar est présent dans toutes les applications de réalité virtuelle de C2Care.

La solution développée a pour objectif de répondre à la problématique de ce mémoire :

***Comment effectuer une refonte de l'avatar virtuel permettant d'améliorer les interactions de l'utilisateur avec les environnements de réalité virtuelle, tout en augmentant le sentiment de présence généré ?***

Cet objectif a été atteint avec le développement d'un nouvel avatar se basant sur une étude bibliographique orientée sur les deux points majeurs de la problématique : l'amélioration des interactions de l'utilisateur et l'augmentation du sentiment de présence.

Le nouveau système d'interactions offre à l'utilisateur plusieurs approches adaptées au contenu avec lequel il doit interagir. La refonte du système permettant d'attraper des objets permet de mettre à jour cette fonctionnalité majeure des expériences de réalité virtuelle avec les normes actuelles. Le sentiment de présence a été grandement amélioré par rapport à l'avatar précédent. Il est possible d'affirmer cela en comparant les solutions mises en place avec les résultats constatés. Une représentation plus fidèle du corps et de ses mouvements, couplée à l'apport de variations visuelles spécifiques, permettent à l'utilisateur de s'identifier à son corps virtuel.

Toutefois, la réalité virtuelle évolue très vite et l'avatar devra continuer d'évoluer avec. Il est tout à fait possible que de nouveaux moyens d'interactions fassent leur apparition et deviennent la norme dans le domaine. La qualité de rendu de l'image au sein des casques s'améliorant, il se peut que certaines contraintes disparaissent et que d'autres naissent. Une veille technologique et littéraire sera, selon moi, à effectuer sur les années à venir. La nouvelle

base sur laquelle l'avatar évoluera est solide et apporte une valeur réelle, mais il est toujours possible de faire mieux en étudiant encore le sujet davantage et en poursuivant l'évolution du dispositif.

Personnellement, il s'agit du projet le plus complet sur lequel j'ai eu l'occasion de travailler. Assumer la responsabilité d'un projet où mes tâches ont été aussi variées, comprenant gestion de projet, conception, réalisation, semble être la conclusion parfaite à trois années d'apprentissage. Si je devais refaire le projet avec l'expérience que j'ai acquise, j'aborderais peut-être certaines parties différemment mais globalement le cœur du travail resterait le même.

# Bibliographie

---

1. Steed, A., Pan, Y., Zisch, F., & Steptoe, W. (2016, March). The impact of a self-avatar on cognitive load in immersive virtual reality. In *2016 IEEE virtual reality (VR)* (pp. 67-76). IEEE.
2. Kilteni, K., Groten, R., & Slater, M. (2012). The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4), 373-387.
3. Fribourg, R., Argelaguet, F., Lécuyer, A., & Hoyet, L. (2020). Avatar and sense of embodiment: Studying the relative preference between appearance, control and point of view. *IEEE transactions on visualization and computer graphics*, 26(5), 2062-2072.
4. Lugin, J. L., Ertl, M., Krop, P., Klüpfel, R., Stierstorfer, S., Weisz, B., ... & Latoschik, M. E. (2018, March). Any "body" there? avatar visibility effects in a virtual reality game. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 17-24). IEEE.
5. Schwind, V., Knierim, P., Tasci, C., Franczak, P., Haas, N., & Henze, N. (2017, May). "These are not my hands!" Effect of Gender on the Perception of Avatar Hands in Virtual Reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 1577-1582).
6. Slater, M., & Sanchez-Vives, M. V. (2014). Transcending the self in immersive virtual reality. *Computer*, 47(7), 24-30.
7. Boletsis, C., & Cedergren, J. E. (2019). VR locomotion in the new era of virtual reality: an empirical comparison of prevalent techniques. *Advances in Human-Computer Interaction*, 2019.
8. Mine, M. R. (1995). Virtual environment interaction techniques. *UNC Chapel Hill CS Dept.*

# Annexes

## A. Document d'évaluation de l'avatar existant pour faire l'objet d'une refonte

Fonctionnalité	Solution existante	Inconvénients / Problèmes	Nouvelle solution	difficulté estimé [1-5]
Avatar immobile 3DOF	une option possible à définir dans la scène (ne pouvant pas être modifiées par le contrôleur): - bloquer la rotation du corps.  Dans le cas contraire une rotation du corps est effectuée lors de la rotation de la tête une fois un certain angle atteint	Rotation du corps sans animation.	conserver l'existant, probablement la meilleure solution, mais la rotation sera fluidifiée par les modifications liées au déplacement ( <i>Avatar 6DOF en mouvement</i> )	1
Avatar immobile 6DOF	Même comportement que <i>Avatar immobile 3DOF</i>  Dans le cas d'un <b>déplacement de la tête in vivo</b> , le corps se penche pour suivre le mouvement de celle-ci.  Dans le cas du <b>déplacement des mains in vivo</b> , il est possible de dépasser la distance visuelle des mains in virtuo.	Actuellement l'option pour bloquer la tête (freezeHeadVR / passer en mode 3DOF) ne fonctionne pas.  Problème de positionnement des mains de l'avatar par rapport à la position réelle des mains in vivo.	fix de l'existant  Positionnements des mains virtuelles toujours fidèles aux positions réelles	3
Avatar 6DOF en mouvement in virtuo (joystick/bouton)	Déplacement avec animation	pas de fluidité dans le déplacement	Mettre en place le déplacement avec IK	4
Avatar 6DOF en mouvement in vivo	Dans le cas d'un <b>déplacement de la tête in vivo</b> , le corps se penche pour suivre le mouvement de celle-ci. En cas de dépassement d'une certaine distance le corps se repositionne pour se placer dessous la tête, à moins que la position du corps ne soit bloquée.	rigidité du modèle en fonction des mouvements (impossible de différencier si le corps se déplace ou si l'utilisateur se penche).	Différencier une rotation de la tête seul et de la tête + le corps (manettes)  Savoir si on se penche ou si on se déplace (par rapport à la position des mains)	3
Avatar assis	possibilité de bouger la tête si 6DOF, le tronc suit le mouvement de la tête		conserver l'existant	2
Caméra seul (avatar invisible)	cache le visuel du corps		conserver l'existant	1
Visuel de la tête	Actuellement le modèle 3D n'a pas de tête mais une sphère uniquement pour le rendu de l'ombre de l'avatar.	Rendu visuel très médiocre (sphère flottante dessus le corps ou encastré dans les épaules).	Remplacé par une vraie ombre de tête liée au corps, pas nécessairement liée à la position de la caméra.	3
Main uniquement	Inexistant		Proposer la possibilité d'afficher uniquement le visuel des mains dans l'environnement (il faut probablement faire des mains séparées en plus de l'avatar). Récupérant les mêmes paramètres que l'avatar (sexe, couleur...)	3
Grab	Main dans une forme prédéfinie pour le grab et hiérarchisation des objets grabbés.	Visuellement toujours en position du grab.  Pose certains problèmes du fait de changer le parent de l'objet.	Ne pas hiérarchiser les objets grabbés.  Lerp du visuel de la main en fonction de l'objet qui va être grabbé.	4
Actions	Un trigger sur l'objet de l'action et la main qui passe physiquement à travers pour effectuer l'action.	Pas de visuelle spécifique de la main pour l'action.  On passe à travers les objets.	Ajouter une collision pour bloquer le déplacement des mains à travers les objets.  Lerp du visuel pour adapter la main à l'action.	4

### Fonctionnalités supplémentaires envisageables

- S'asseoir: passer d'un avatar debout à une position assise (et inversement) avec lerp d'animation.
- Se baisser: détecter le fait que l'utilisateur se baisse in vivo pour une cohérence visuelle avec l'avatar (probablement automatisé avec les bons outils).
- Configuration avancée (Corpulence ?)

### Récapitulatif général

L'objectif est donc de réaliser un nouvel avatar (from scratch) avec les options actuelles présentes dans le contrôleur, à savoir:

- couleur de peau
- taille
- sexe

Il faudra un seul avatar et non des variantes en fonction des besoins comme l'avatar grab etc... et une cohérence entre l'avatar homme et femme (même taille/placement des bones...)

Il inclura les fonctions de base liées à la caméra tel quel le recalibrage (qui ne devrait pas nécessiter d'être exécuté au lancement).

Il faudra revoir le système de téléportation actuellement inclus dans l'avatar pour permettre la conservation de la rotation de la caméra (si voulu) lors d'une téléportation et même d'un changement de scène.

Le résultat du développement devra comprendre une scène de test complète sur le projet c2packages avec toutes les actions et options possibles.

## B. Étude de l'avatar existant

### Structure



### A la racine de l'avatar :

- **Character controller** : utilisé pour le déplacement de l'avatar, à l'aide de sa méthode "Move()"
- **AvatarMgr.cs** : responsable du mouvement, recalibrage, sexe de l'avatar, animations, freeze de la position/rotation/tête
- **Nav Mesh Obstacle** : fait de l'avatar un obstacle mouvant sur une navmesh, pour que les PNJ l'évitent sur leur route
- **Shake.cs** : permet de faire des shakes sur la caméra
- **MainManager.cs** : responsable des téléportations
- **EventAvatar.cs** : gère les messages reçus du contrôleur pour mettre à jour la taille de l'avatar, sa couleur de peau, son sexe, l'insistance des regards
- **InputMgr.cs** : gère les inputs

### Visual helper :

- **Visualhelper.cs** : responsable de l'affichage des visuels de battements de coeur, respiration, particules bulles & buée
- **EventInspector.cs** : artefact de debug à supprimer...
- **Enfants :**
  - **GameObject - Bulles** ⇒ ParticleSystem
  - **GameObject - Buée** ⇒ ParticleSystem
  - **GameObject - UI** ⇒ Canvas avec enfants pour afficher

### GraphicMan & GraphicWoman :

Tous les GameObjects pour faire un rendu de l'avatar homme & femme.

- **Animator :**
  - **Idle** : animation d'avatar immobile
  - **Assis** : animation d'avatar assis
  - **Marche** : animation de marche de l'avatar
- **BipedIK** : Inverse Kinematics pour les bras et le corps

## TargetDummies :

- **PhysicsDummiesMgr.cs** : met à jour la position et rotation de ses enfants par rapport aux données spatiales du casques VR et des manettes
- Enfants :
  - **DummyRightHand** : gameobject main droite, avec un rigidbody et un collider trigger
  - **DummyLeftHand** : gameobject main gauche, avec un rigidbody et un collider trigger
  - **DummyHead** : gameobject pour la position de la tête, pour câbler le haut du corps à la caméra VR
  - **DummyPelvis** : à investiguer

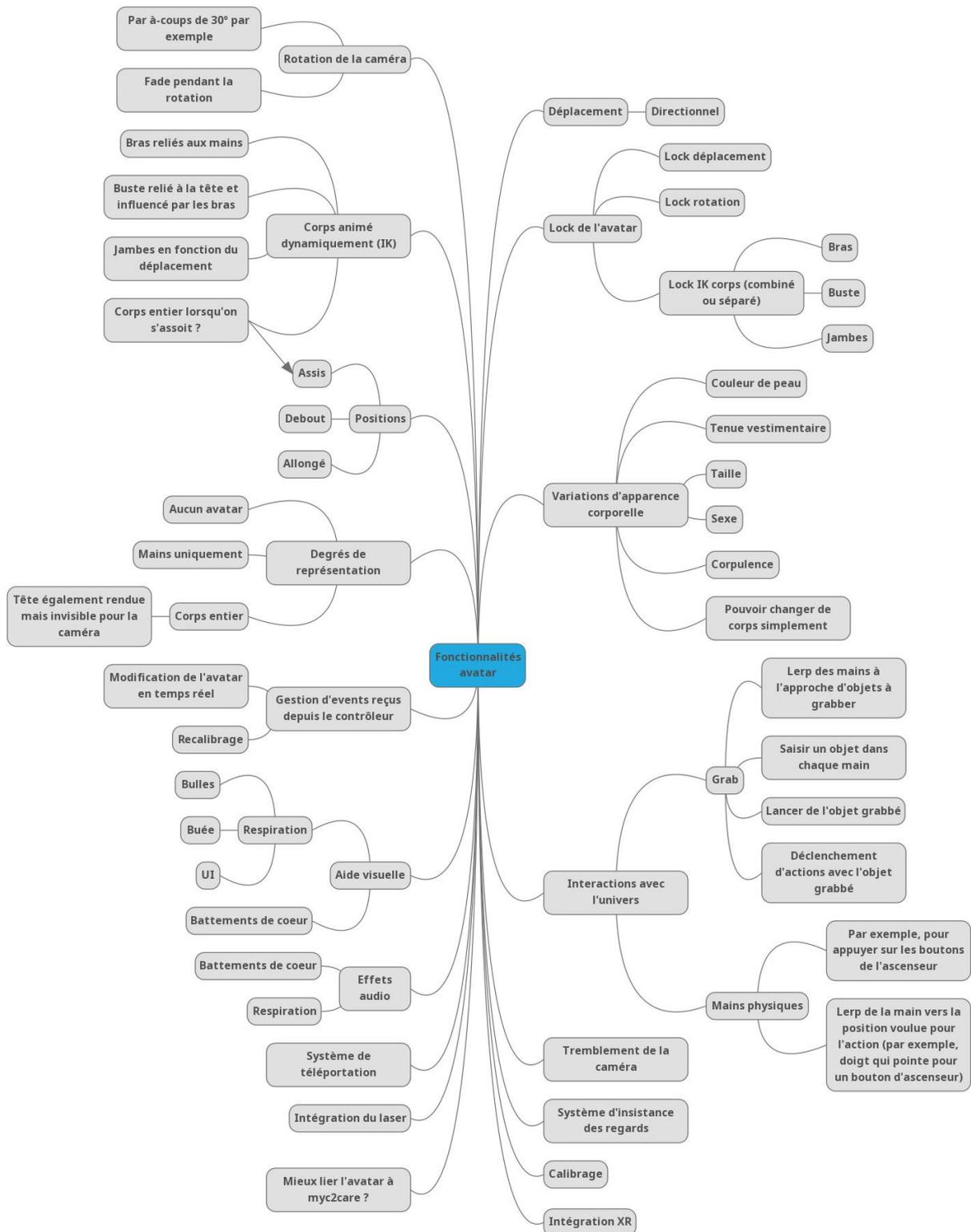
## PeopleLook :

- **BoxCollider** : trigger dans lequel les PNJ doivent entrer pour déclencher le "look"
- **PeopleLook.cs** : déclenche l'insistance des regards à l'aide du trigger associé

## XR Rig :

- **XRRig** : intégration VR d'Unity, pour gérer le calibrage
- **PXR\_Manager** : script de compatibilité Pico
- Enfants :
  - **CamParent**
  - Enfants :
    - **LeftHandController** : XR controllers, interactors
    - **RightHandController** : XR controllers, interactors
    - **Camera** : main camera overridee par le casque VR

## C. Carte heuristique des fonctionnalités du nouvel avatar





## E. Documentation d'un des risques du projet

Risque n°5 par ordre d'importance :

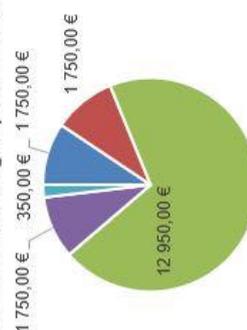
« Déploiement bloqué par les versions d'Unity divergeant sur les projets »

- Détails :  
Les différents projets de C2Care sont sur des versions d'Unity différentes, où le fonctionnement global diverge beaucoup. Le nouvel avatar sera créé sur la version d'Unity supporté par la majorité des applications de C2Care : Unity 2018. Il faudra que toutes les applications soient dessus pour pouvoir compléter le déploiement de l'avatar dans les projets.
- Conséquences :  
Mise en pause du déploiement de l'avatar au moment de passer sur des projets se trouvant dans des versions d'Unity inférieures.
- Probabilité : moyenne  
Il se peut que les projets soient tous migrés vers la dernière version d'Unity supportée d'ici la fin de la réalisation du nouvel avatar et du début du déploiement.
- Gravité : faible  
Le seul problème est la mise en pause du déploiement. Ce n'est pas grave car l'avatar aura été achevé et son intégration se fera progressivement, application par application.
- Solution : pris en charge  
Identifier les appels à l'avatar dans les applications pour les uniformiser. Il faut aussi documenter l'intégration de l'avatar. Globalement, le prévoir à une migration future vers une nouvelle version d'Unity (avec les nouveaux systèmes).

## F. Budget détaillé prévisionnel

Refonte Avatar		BUDGET DETAILLE											
Référence des coûts journaliers		Coût moyen journalier RH :		350,00 €		Coût moyen journalier Sous-traitance :				Date de MàJ		26/05/2021	
Phases et étapes du projet	Charge RH (en J*H)		Ressources internes		Coût RH (en €)		Pas de sous-traitance		Ressources matérielles et achats		Coût Total		
	Prévu	Actualisé	Prévu	Actualisé	Prévu	Actualisé	Prévu	Actualisé	Prévu	Actualisé	Prévu	Actualisé	
Préparation du projet	5	0	1 750,00 €	- €	- €	- €	5	0	1 750,00 €	- €	5	0	1 750,00 €
Conception initiale	5	0	1 750,00 €	- €	- €	- €	5	0	1 750,00 €	- €	5	0	1 750,00 €
Conception globale du projet	4		1 400,00 €	- €	- €	- €	4	0	1 400,00 €	- €	4	0	1 400,00 €
Préparation de l'environnement de test	1		350,00 €	- €	- €	- €	1	0	350,00 €	- €	1	0	350,00 €
Réalisation	37	0	12 950,00 €	- €	- €	- €	37	0	12 950,00 €	- €	37	0	12 950,00 €
Création de la base de l'avatar	5		1 750,00 €	- €	- €	- €	5	0	1 750,00 €	- €	5	0	1 750,00 €
Système d'interactions	12		4 200,00 €	- €	- €	- €	12	0	4 200,00 €	- €	12	0	4 200,00 €
Représentation visuelle	5		1 750,00 €	- €	- €	- €	5	0	1 750,00 €	- €	5	0	1 750,00 €
Calibrage et positionnement	6		2 100,00 €	- €	- €	- €	6	0	2 100,00 €	- €	6	0	2 100,00 €
Fonctionnalités mineures	9		3 150,00 €	- €	- €	- €	9	0	3 150,00 €	- €	9	0	3 150,00 €
Post-projet	5	0	1 750,00 €	- €	- €	- €	5	0	1 750,00 €	- €	5	0	1 750,00 €
Intégration dans tous les projets	5		1 750,00 €	- €	- €	- €	5	0	1 750,00 €	- €	5	0	1 750,00 €
Bilan du projet	1		350,00 €	- €	- €	- €	1	0	350,00 €	- €	1	0	350,00 €
Ensemble du projet	53	0	18 550,00 €	- €	- €	- €	53	0	18 550,00 €	- €	53	0	18 550,00 €

Répartition du budget prévisionnel



- Préparation du projet ■ Conception initiale ■ Réalisation
- Post-projet ■ Bilan du projet

## G. Documentation technique : normes, standards et organisation

### Partie C#

Les conventions de codage C# à respecter sont les conventions officielles proposées par Microsoft sur le langage C# : <https://docs.microsoft.com/fr-fr/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>

Les règles essentielles à respecter :

- Les **variables, fonctions et commentaires** sont en **anglais**.
- Le **nom des variables et fonctions** doivent être **explicites, quitte à être longs**.
- Les **variables** s'écrivent en **lowerCamelCase**.
- Les **fonctions** s'écrivent en **UpperCamelCase**.
- Les **constantes** s'écrivent **LIKE\_THAT**.
- Ajouter des **commentaires <summary>** **au-dessus de chaque fonction**, et y écrire une description sur celle-ci, ses paramètres ainsi que ce qu'elle retourne.
- **Éviter l'usage de variables publiques**, préférer la protection de variables privées avec des **properties**.

### Partie Unity

Il s'agit ici principalement de respecter une organisation des dossiers se trouvant dans le dossier Assets de chaque projet.

Légende pour la hiérarchie ci-dessous :

- **en gras** = dossier
- normal = description du contenu du dossier

Hiérarchie à respecter :

- **\_Scenes**
  - Scènes de C2Base adaptées pour le projet
  - **Environments**
    - Scènes spécifiques au projet

**Animations**

- **Anim1**
  - Animation clips & animator controllers, spécifiques au projet

#### **Biblio**

- **GFXElement1 (idéalement présent comme ci-dessous)**
  - Prefabs à la racine créés à partir des sources de l'élément
  - **Sources**
    - Fichiers composant l'élément (fbx, textures, materials...)

#### **C2Base**

- **Biblio**
  - Equivalent du Biblio du projet mais commun à tous les projets
- **Animations**
  - Animations communes à tous les projets
- **Scripts**
  - Scripts communs à tous les projets
- **Scenes**
  - Scènes communes à tous les projets
- **Prefabs**
  - Prefabs communes.
- ...

#### **Oculus**

- Contient les fichiers récupérés depuis Oculus Integration

#### **Editor**

- Scripts pour l'éditeur (achetés ou persos)

#### **Gizmos**

- Fichiers utilisés pour les Gizmos dans la Scene View

#### **Plugins**

#### **Resources**

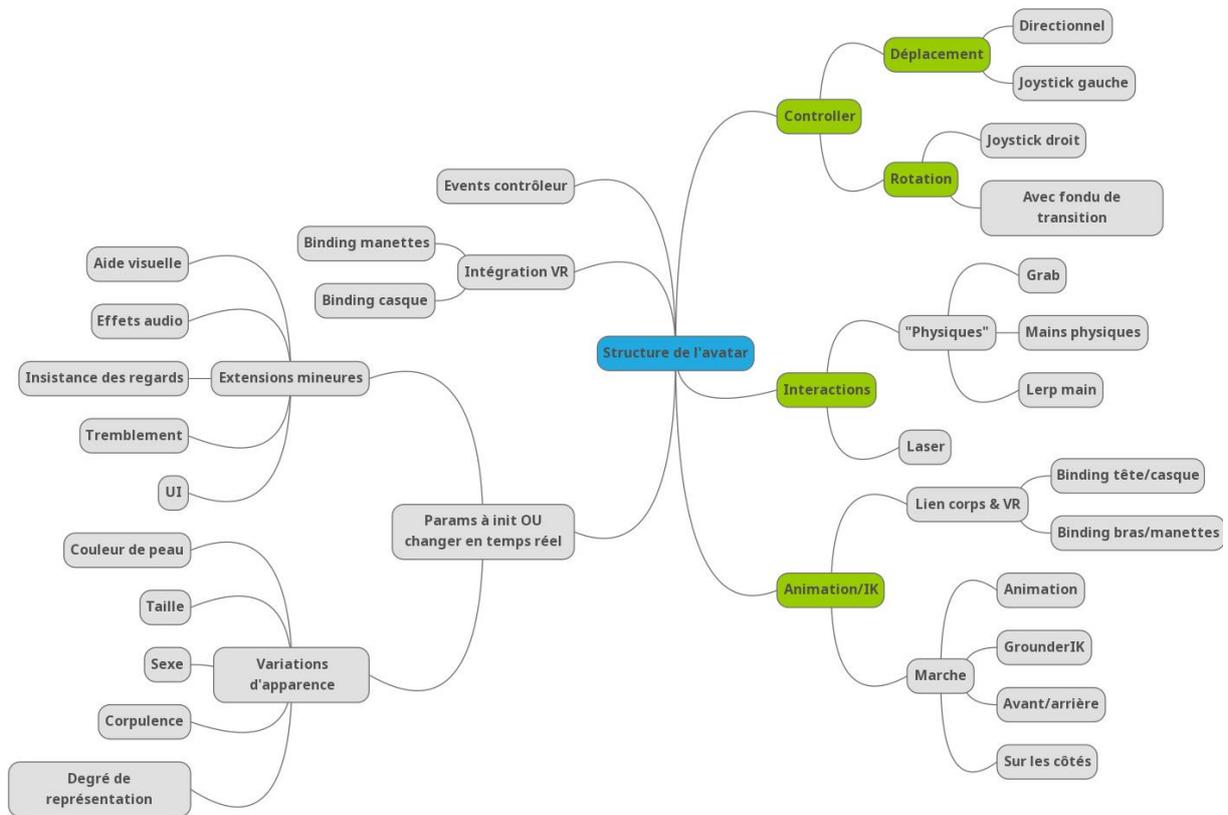
#### **Scripts**

- Scripts spécifiques au projet

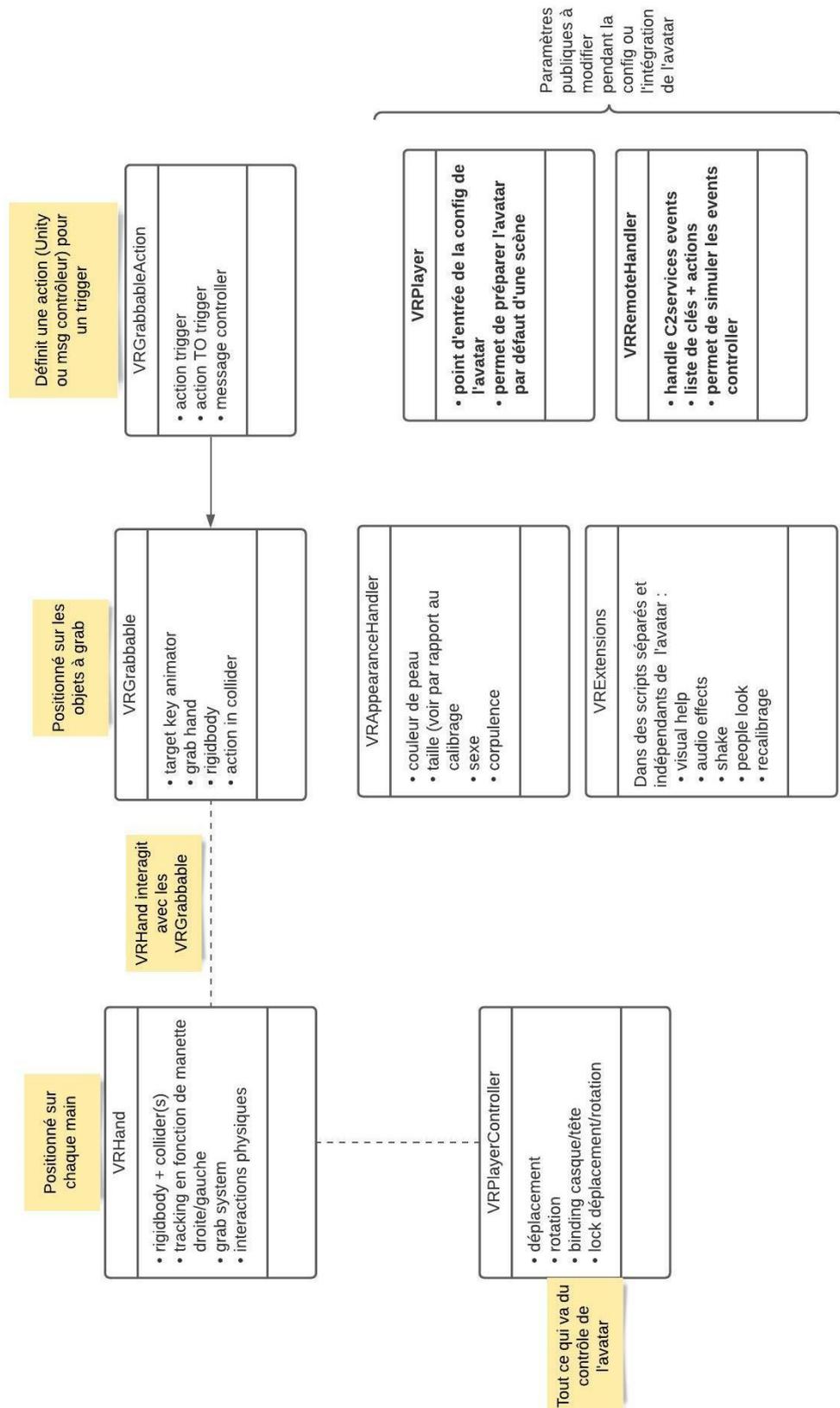
#### **Sounds**

- Sons spécifiques à l'application

## H. Carte heuristique sur la structure de l'avatar



# I. Diagramme de classes UML sur la structure de code globale de l'avatar





## K. Spécification d'une des user stories

Sprint 1  
TC EN TANT QUE patient, JE SOUHAITE pouvoir recalibrer ma vue AFIN DE être à la bonne hauteur et regarder dans la bonne direction Recommencer la tâche ☺ ...

Créé par TC Thomas C. le Mai 21

Attribué à TC Thomas C.

Commencez / Date d'échéance **Jun 8** – **Jun 9**

Priorité »» **Haute**

Statut ✓ **Terminé Jun 9 100%**

**Tests :** test case CALIBRATION dans AvatarTester.

**Actions :**

- Calibrer la vue en remettant la caméra à la bonne hauteur et en la tournant dans la bonne direction.

**Moyens :**

- Système de calibration existant et fonctionnel.

**Contraintes :**

- Calibrer la position de la caméra en fonction de la taille de l'avatar.
- Calibrer la rotation de la caméra en fonction des données enregistrées par le casque au démarrage de la scène.
- Option pour calibrer avec un effet de fondu rapide.
- Option pour choisir de calibrer automatiquement au lancement d'une scène. Par défaut, on calibre. cas où on ne veut pas de calibrage auto : passage de niveaux dans Brain.
- Pouvoir bien calibrer par rapport à l'horizon.

**Paramètres :**

- Bool : calibration avec fondu.
- Float : temps de fondu en secondes.

## Résumé et mots clés

---

### Résumé :

Ce mémoire propose une étude synthétique sur la notion d'avatar dans les expériences de réalité virtuelle. Un état de l'art identifie les différents leviers de présence liés aux avatars virtuels et une solution est proposée pour répondre à la problématique de C2Care concernant la refonte de l'avatar existant pour générer davantage de présence chez les utilisateurs tout en améliorant l'environnement de développement global. Ainsi, toutes les étapes du projet sont présentées pour permettre au lecteur d'en comprendre les enjeux et les solutions apportées pour tenter d'y répondre. Les résultats constatés permettent de souligner l'importance d'intégrer un système d'interactions complet aux avatars virtuels et d'en proposer une variété de représentations corporelles adaptables aux caractéristiques physiques des utilisateurs.

**Mots-clés :** Réalité virtuelle, avatar, représentation de soi, interactions, déplacement, présence

### Summary :

This thesis proposes a synthetic study on the concept of avatars in virtual reality experiences. A state of the art identifies the different presence levers related to virtual avatars and a solution is provided to answer C2Care's problem concerning the redesign of the existing avatar to generate more presence among users while improving the overall development environment. Thus, all the steps of the project are presented to allow the reader to understand the stakes and the solutions provided to try to answer them. The results show the importance of integrating a complete interaction system to virtual avatars and of proposing a variety of body representations adaptable to the physical characteristics of the users.

**Keywords :** Virtual reality, avatar, self-representation, interactions, movement, presence