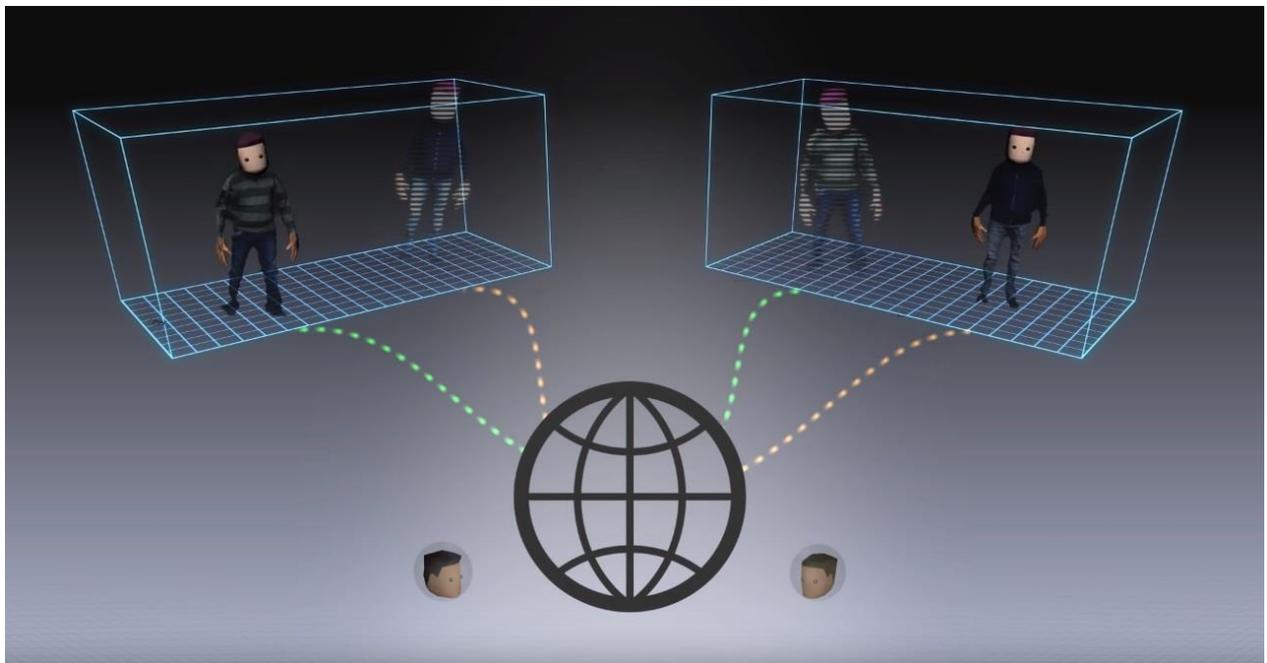


Travail de Bachelor 2018

La réalité virtuelle collaborative connectée



Source : <https://www.indiedb.com/news/this-htc-vive-prototype-game-projects-live-players-into-vr-with-kinect>

Étudiant : Jeff Zufferey

Professeure : Anne-Dominique Salamin

Déposé le : 30 juillet 2018

www.hevs.ch

Résumé

Au vu de son potentiel divers et varié, la réalité virtuelle a récemment gagné en importance. Ainsi, Cyberlearn, le centre e-learning de la HES-SO, a commencé à réaliser diverses expériences dans ce domaine afin de déterminer l'éventuel apport de la réalité virtuelle pour l'apprentissage. Le centre, proposant bon nombre d'outils d'e-learning, s'interroge sur une utilisation collaborative et connectée de cette technologie.

Est-il possible de créer des applications collaboratives connectées en réalité virtuelle ? Cette problématique a donné suite à la question de recherche suivante : les *frameworks* open source permettent-ils de développer des applications collaboratives connectées en réalité virtuelle ? Notre hypothèse soutient que les *frameworks* open source permettent effectivement la création d'applications collaboratives connectées en réalité virtuelle.

Afin de vérifier cette hypothèse, une analyse des *frameworks* open source a été réalisée, démontrant ainsi qu'Unity pouvait utiliser deux méthodes pour concevoir une application en réalité virtuelle connectée : Unet ou Photon.

En utilisant ces *frameworks*, nous avons pu développer une application en réalité virtuelle collaborative connectée sur la base de deux casques HTC VIVE. Deux utilisateurs possédant ce casque, deux ordinateurs compatibles et l'application peuvent se connecter dans le même environnement virtuel, interagir et communiquer ensemble pour atteindre un objectif commun, bien que le contenu de l'application de réalité virtuelle reste perfectible.

Grâce à l'application fonctionnelle développée pour ce travail nous avons prouvé que l'affirmation émise par notre hypothèse est fondée.

Mots-clés : réalité virtuelle, Unity, Unet, Photon, application, connectée, collaborative

AVANT-PROPOS

Dans le but de terminer sa formation à la Haute école spécialisée de Suisse occidentale Valais-Wallis, chaque étudiant de la filière Informatique de gestion doit rédiger un travail de Bachelor. Ce rapport écrit, réalisé durant trois mois, sera présenté lors d'une défense orale qui aura pour but d'expliquer le contenu, l'environnement de travail ainsi que la présentation des résultats obtenus.

L'origine du thème de ce travail vient, d'une part, de la fascination que j'éprouve pour le domaine de la réalité virtuelle, et, d'autre part, d'une conférence québécoise nommée « L'Été Indien » pour laquelle j'ai réalisé une application de réalité virtuelle fonctionnelle dans le cadre d'un stage chez Cyberlearn. L'équipe Cyberlearn a présenté cette application durant cet événement international se déroulant entre la France, le Québec et la Suisse, tout en me proposant d'y participer. Lors de ce séjour, plusieurs idées sont venues à l'esprit de l'équipe, telles que l'apprentissage d'une langue étrangère dans un environnement virtuel mettant en scène plusieurs personnes ou la réalisation d'une opération chirurgicale virtuelle dans un but d'amélioration de la formation médicale. J'ai alors contacté la Professeure Anne-Dominique Salamin dans le but d'aborder le thème de la réalité virtuelle connectée. La Prof. Salamin a montré un fort intérêt pour cette thématique technique, tout en ajoutant l'aspect collaboratif que la réalité virtuelle peut offrir.

Grâce au centre d'e-learning Cyberlearn, la thématique de la réalité virtuelle collaborative connectée a pu voir le jour. L'objectif de ce document est d'expliquer chacun des éléments composant cet aspect et d'amener à la prise de conscience de leur importance, puis d'explorer leur synergie et leur potentiel. Le centre Cyberlearn souhaiterait qu'un prototype fonctionnel soit développé afin de pouvoir en faire une étude par la suite et d'utiliser la technologie mise en place pour la réalisation d'autres cas d'exécution.

Concernant les technologies et les logiciels mentionnés, il ne s'agit en aucun cas d'une liste exhaustive des moyens existants ou utilisés actuellement. Ceux-ci seront mentionnés pour exprimer une notion particulière dans la démarche scientifique.

Il est nécessaire d'avoir des connaissances basiques en programmation afin de bien comprendre la partie de ce travail ayant trait au développement de l'application.

La liste des références aux tableaux et aux illustrations se trouve en fin de document.

Ce rapport a été écrit conformément aux normes de mise en forme de l'*American Psychology Association*.

REMERCIEMENTS

Je remercie toutes les personnes qui m'ont soutenu durant la réalisation de ce travail de Bachelor.

J'adresse spécifiquement mes remerciements les plus sincères aux personnes suivantes :

- Madame la Professeure Anne-Dominique Salamin, pour l'encadrement de ce projet ainsi que le suivi effectué durant le temps de la réalisation du travail,
- L'équipe du centre e-learning Cyberlearn, pour l'aide et le support technique concernant le développement ainsi que pour la disponibilité du matériel de réalité virtuelle.
- Tous les lecteurs ayant eu la gentillesse et la patience de lire ce travail pour l'améliorer.
- Monsieur Robert Seligmann qui a eu le courage et la patience de tester l'application à chaque phase du développement.

Un dernier merci à ma famille et à mes amis pour leur soutien durant cette formation de niveau tertiaire.

TABLE DES MATIÈRES

LEXIQUE	XII
1. INTRODUCTION	1
1.1. CONTEXTE DU TRAVAIL	1
1.2. CYBERLEARN	1
1.3. STRUCTURE DE TRAVAIL	2
2. ASPECTS MÉTHODOLOGIQUES	2
2.1. MÉTHODOLOGIE DE RECHERCHE	2
2.2. MÉTHODOLOGIE DE TRAVAIL	3
2.2.1. <i>Scrum appliqué</i>	3
3. ÉTAT DE L'ART	3
3.1. INTRODUCTION DES THÈMES	3
3.1.1. <i>Collaboration/coopération</i>	4
3.1.2. <i>Réalité virtuelle</i>	5
3.2. SOLUTIONS EXISTANTES	7
3.2.1. <i>Escape room</i>	7
3.2.2. <i>Jeux VR</i>	8
3.2.3. <i>VR en entreprise</i>	9
3.2.4. <i>Apprentissage VR</i>	10
3.2.5. <i>Simulation VR</i>	11
3.3. PRISE DE POSITION.....	12
4. DÉFINITION DES BESOINS	12
4.1. SCÉNARIO	12
4.2. RÉSEAU	12
4.3. INTERACTIONS DES UTILISATEURS.....	13
4.4. ENCADREMENT DE L'APPLICATION	13
4.5. <i>LOGS</i>	13
4.6. RÉCAPITULATIF DES CARACTÉRISTIQUES TECHNIQUES	14
5. ANALYSE DES OUTILS	14
5.1. CASQUES VR	14
5.2. DÉTERMINATION DU MEILLEUR LOGICIEL DE CONCEPTION VR	16
5.2.1. <i>Unity</i>	16
5.2.2. <i>Unreal Engine</i>	17
5.2.3. <i>Cry Engine</i>	17

5.2.4.	<i>Résultat du classement des logiciels.....</i>	19
5.2.5.	<i>Détermination des assets à utiliser pour Unity.....</i>	19
5.2.6.	<i>Présentation des assets open sources existants</i>	19
5.2.7.	<i>Comparatif des moyens de conception d'un réseau</i>	21
5.2.8.	<i>Résultat du classement.....</i>	22
5.2.9.	<i>Assets sélectionnés</i>	23
6.	QUELQUES BASES D'UNITY	23
7.	DÉVELOPPEMENT	24
7.1.	ENVIRONNEMENT DE DÉVELOPPEMENT.....	25
7.2.	PROCESSUS DE CONCEPTION	25
7.3.	TYPES DE SERVEURS	27
7.3.1.	<i>Serveur avec autorité.....</i>	27
7.3.2.	<i>Serveur sans autorité</i>	28
7.3.3.	<i>La notion de latence</i>	28
7.3.4.	<i>LAN & WAN</i>	28
7.4.	DÉPLOIEMENT D'UNET ET DE VRTK.....	29
7.4.1.	<i>Unet</i>	29
7.4.2.	<i>Étapes fonctionnelles réalisées.....</i>	29
7.4.3.	<i>Obstacles rencontrés</i>	30
7.4.4.	<i>Conclusion sur le projet Unet + VRTK.....</i>	37
7.5.	DÉPLOIEMENT DE UNET EN UTILISANT STEAMVR.....	38
7.5.1.	<i>Comparaison des solutions aux problèmes non résolus :</i>	38
7.5.2.	<i>Obstacles rencontrés</i>	40
7.5.3.	<i>Conclusion sur le projet à base d'Unet + SteamVR</i>	40
7.6.	CONCLUSION SUR UNET.....	41
7.7.	DÉPLOIEMENT DE PHOTON + VRTK SUR UNITY.....	42
7.7.1.	<i>Photon</i>	42
7.7.2.	<i>Déroulement de conception.....</i>	42
7.7.3.	<i>Implémentation de Photon sur Unity :.....</i>	42
7.8.	OBSTACLES RENCONTRÉS	45
7.9.	CONCLUSION SUR PHOTON	48
8.	VÉRIFICATION DE L'HYPOTHÈSE.....	49
9.	RECOMMANDATIONS POUR LA SUITE	49
10.	CONCLUSION	50

11.	DÉCLARATION SUR L'HONNEUR	52
12.	RÉFÉRENCES DES TABLEAUX	58
13.	RÉFÉRENCES DES ILLUSTRATIONS	59
14.	ANNEXES	62
14.1.	LE CAHIER DES CHARGES	62
14.2.	PRODUCT BACKLOG	64
14.3.	TABLEAU DE BORD.....	65
14.4.	RAPPORTS HEBDOMADAIRES	66
14.5.	PROCÈS-VERBAUX.....	79

Liste des tableaux

Tableau 1 : Tableau récapitulatif des caractéristiques des notions de collaboration et de coopération	4
Tableau 2 : Comparatif réalisé par l'auteur selon des sources multiples	15
Tableau 3 : Comparaison des logiciels de conception VR.....	18
Tableau 4 : Matrice de comparaison	22
Tableau 5 : Récapitulatif des fonctionnalités réalisées avec Unet et VRTK (réalisé par l'auteur)	37
Tableau 6 : Récapitulatif des fonctionnalités réalisées avec Unet et SteamVR (réalisé par l'auteur)	41
Tableau 7 : Récapitulatif des fonctionnalités réalisées avec Photon.....	49

Liste des illustrations

Figure 1 : Logo de Cyberlearn	2
Figure 2 : Illustration de la réalité virtuelle	5
Figure 3 : Illustration de la réalité augmentée.....	5
Figure 4 : Illustration de la pièce inversée du scénario « Vice Versa » de Trap Game.....	8
Figure 5 : Illustration du jeu « Beat Saber »	9
Figure 6 : Illustration d'une bombe à désamorcer du jeu «Keep talking and nobody explodes » ...	9
Figure 7 : Système Cave	9
Figure 8 : Conception de maquette virtuelle Range Rover 2011	9
Figure 9 : Conception du <i>level design</i> directement en réalité virtuelle.	10
Figure 10 : Capture du jeu « Job Simulator », assignant à l'utilisateur un rôle de vendeur dans un fast-food.....	11
Figure 11 : <i>Use case</i> des interactions par utilisateur (réalisée par l'auteur)	13
Figure 12 : Illustration du « Room Scale » du HTC VIVE	16
Figure 13 : Logo d'Unity	17
Figure 14 : Interface Pro d'Unity	17
Figure 15 : Logo d'Unreal Engine	17
Figure 16 : Interface d'Unreal Engine.....	17
Figure 17 : Logo de Cry Engine	17
Figure 18 : Interface de Cry Engine	17
Figure 19 : Logo du <i>plugin</i> Steam VR.....	20
Figure 20 : Logo de Virutal Reality Tool Kit (VRTK)	20
Figure 21 : Logo de Photon Engine	20
Figure 22 : Logo de Forge Networking	21
Figure 23 : Logo de Master Server Framework.....	21
Figure 24 : Logo de Dark Rift 2	21
Figure 25 : Explication de l'interface à partir d'une capture d'écran de l'auteur.	23
Figure 26 : Processus de conception (réalisé par l'auteur via Lucidchart.com)	26
Figure 27 : Fonctionnement d'un serveur avec autorité (réalisé par l'auteur)	27
Figure 28 : Fonctionnement d'un serveur n'ayant pas d'autorité (réalisé par l'auteur).....	28
Figure 29 : Configuration de Unet sur Unity	29
Figure 30 : Menu de création d'une instance Unet.....	29
Figure 31 : Succès du test d'implémentation d'Unet.....	30
Figure 32 : Environnement 3D Unet + VRTK.....	30
Figure 33 : Tentative d'utilisation d'adresse spécifique réalisée par l'auteur sur Unity.....	31
Figure 34 : Utilisation du « Network Discovery » réalisée par l'auteur sur Unity.....	32
Figure 35 : L'ordinateur peut créer une <i>Room</i> sur le réseau	32
Figure 36 : Le client peut rejoindre la <i>Room</i> créée par le serveur	32

Figure 37 : Illustration des fonctions [Command] et [ClientRPC] réalisée par l'auteur selon (Byl, 2018)	35
Figure 38 Utilisateur client voyant le joueur hôte manipuler un cube, vu de l'éditeur (réalisé par l'auteur)	40
Figure 39 : Utilisateur serveur ne voyant aucun changement du client, vu de l'éditeur (réalisé par l'auteur)	40
Figure 40 : Paramétrage de l'Appld de Photon pour le <i>cloud</i>	42
Figure 41 : Succès de la première étape avec Photon	43
Figure 42 : Décors 3D pour Photon réalisés avec ProBuilder	44
Figure 43 : Paramétrage du Appld de Photon pour la voix	44
Figure 44 : Système de <i>log</i> de position	45
Figure 45 : Exemple de <i>logs</i> de positions.....	45
Figure 46 : État actuel de l'application	48
Figure 47 : Product Backlog.....	64
Figure 48 : Tableau de bord (réalisé par l'auteur)	65
Figure 49 : Rapport hebdomadaire de la 1 ^{ère} semaine (réalisé par l'auteur).....	66
Figure 50 : Rapport hebdomadaire de la 2 ^{ème} semaine (réalisé par l'auteur)	67
Figure 51 : Rapport hebdomadaire de la 3 ^{ème} semaine (réalisé par l'auteur)	68
Figure 52 : Rapport hebdomadaire de la 4 ^{ème} semaine (réalisé par l'auteur)	69
Figure 53 : Rapport hebdomadaire de la 5 ^{ème} semaine (réalisé par l'auteur)	70
Figure 54 : Rapport hebdomadaire de la 6 ^{ème} semaine (réalisé par l'auteur)	71
Figure 55 : Rapport hebdomadaire de la 7 ^{ème} semaine (réalisé par l'auteur)	72
Figure 56 : Rapport hebdomadaire de la 8 ^{ème} semaine (réalisé par l'auteur)	73
Figure 57 : Rapport hebdomadaire de la 9 ^{ème} semaine (réalisé par l'auteur)	74
Figure 58 : Rapport hebdomadaire de la 10 ^{ème} semaine (réalisé par l'auteur).....	75
Figure 59 : Rapport hebdomadaire de la 11 ^{ème} semaine (réalisé par l'auteur).....	76
Figure 60 : Rapport hebdomadaire de la 12 ^{ème} semaine (réalisé par l'auteur).....	77
Figure 61 : Rapport hebdomadaire de la 13 ^{ème} semaine (réalisé par l'auteur).....	78

Lexique

<i>Framework</i>	Ensemble d'outils logiciel indépendants.
<i>Add-On</i>	Extension d'un logiciel.
<i>Product Backlog</i>	Tableau contenant des <i>users stories</i> , permettant la planification du travail.
<i>User stories</i>	Fonctionnalité d'une application, décrite sous le format « En tant que..., je veux..., afin que... ».
<i>Story Point</i>	Point de mesure pour appréhender la complexité, l'effort à fournir.
<i>MoSCoW</i>	Méthodologie de priorisation des tâches. <i>Must</i> doit être réalisé, <i>Should</i> devrait l'être, <i>Could</i> pourrait l'être, <i>Won't</i> ne le sera pas dans l'immédiat.
<i>CAVE</i>	Acronyme de <i>Cave Automatic Virtual Environment</i> , signifiant un espace pour un environnement virtuel automatique.
<i>Hackathon</i>	Évènement rassemblant des personnes, des informaticiens et des designers pour résoudre ensemble une problématique métier dans un laps de temps d'environ 48 heures.
<i>Plugin</i>	Module d'extension permettant à un logiciel d'obtenir de nouveaux outils.
<i>Room</i>	Lieu virtuel dans lequel sont rassemblés plusieurs utilisateurs.
<i>Looby</i>	Salle d'attente pour les utilisateurs où ils peuvent choisir une <i>Room</i> .
<i>Mesh</i>	Terme de conception d'objet 3D. Ensemble de points constituant une forme 3D.
<i>Retour haptique</i>	Technologie permettant un retour sur un appareil électronique à la suite d'une action de l'utilisateur.
<i>Assets</i>	Ressource Informatique.

1. Introduction

Dans un monde toujours plus proche du digital, la réalité virtuelle est une technologie plaçant l'utilisateur au cœur de cette numérisation. Immergé dans un univers fictif, les possibilités d'exécutions sont sans limites ; c'est pourquoi, depuis l'arrivée des casques de réalité virtuelle, de nombreuses entreprises ont été créées afin d'explorer et d'extraire ce potentiel virtuel infini. Toutefois, si bénéficier de cette technologie est enrichissant, quel est l'intérêt sur le long terme si l'expérience reste confinée à une utilisation solitaire ? Durant toutes les époques de la civilisation, le partage a été la clé pour améliorer la vie des hommes. Ce qu'il manque à la technologie de réalité virtuelle est justement le partage, d'autant plus dans le monde d'aujourd'hui où la collaboration est devenue un facteur clé de réussite. Cependant, est-il désormais possible de créer des applications collaboratives connectées en réalité virtuelle ? Partager l'émotion des mondes virtuels, la connaissance de tous et pouvoir créer des contenus avec son entourage - qu'il s'agisse de membres de sa propre famille ou de collègues professionnels - serait merveilleux et permettrait à tous de profiter de ces bienfaits. Pour concevoir cet idéal de partage en réalité virtuelle connectée, il est nécessaire de se poser la question suivante : les *frameworks* open source sont-ils adaptés ? C'est ce que nous allons déterminer dans ce document, en commençant par résumer ce qui existe déjà sur le marché et en analysant divers *frameworks*.

1.1. Contexte du travail

Comme énoncé dans l'avant-propos, ce travail est réalisé au terme de la formation d'informaticien de gestion. Il a pour but de démontrer les compétences acquises durant le processus d'apprentissage. Au cours de la troisième année, les étudiants ont pu choisir un thème donné par un professeur de l'institut de recherche ou proposer leur propre thématique de travail de Bachelor. Dans ce dernier cas, il était de leur devoir de trouver une entreprise intéressée à les suivre dans l'écriture dudit travail. La contrainte principale est que le sujet doit comprendre une partie de recherche, une partie d'analyse et une partie de développement. La durée de ce travail est estimée à 360 heures. Dans le cas de ce travail, le thème a été proposé par l'auteur au centre e-learning Cyberlearn. La réalisation du travail a débuté le 1^{er} mai 2018 et s'achèvera le 30 juillet 2018. Le travail sera défendu lors de la dernière semaine d'août 2018.

1.2. Cyberlearn

Cyberlearn est le centre e-learning de la HES-SO. Il utilise la plateforme Moodle pour permettre une gestion facilitée des cours entre les professeurs et les étudiants. De ce fait, Cyberlearn offre une alternative à la gestion des cours standards permettant de centraliser ceux-ci et diversifiant leur contenu en intégrant de nombreux documents tels que des ressources multimédias ou des quizz personnalisés. Le centre accompagne les professeurs dans ce nouvel environnement afin de les

familiariser avec son interface et les aide dans divers projets e-learning tels que des cours en ligne massifs appelés MOOCs. Il œuvre aussi dans la recherche et le développement afin d'améliorer et d'innover l'e-learning actuel. Ces résultats profitent à l'ensemble de la HES-SO, qui utilise le système Moodle mis en place par Cyberlearn.

Cyberlearn est donc présent sur de nombreux domaines, tant pour les étudiants que pour les professeurs. Ces domaines comprennent notamment :

- Des MOOCs
- Des applications Web
- Des applications mobiles

Depuis 2016, le centre a débuté des recherches sur la réalité virtuelle et sur la réalité augmentée, dans le but d'étudier ces nouveaux outils numériques afin de déterminer s'il est possible de les intégrer comme des outils d'apprentissage viables.



Figure 1 : Logo de Cyberlearn

1.3. Structure de travail

Afin de permettre une meilleure lecture de ce document, une structure a été définie. Tout d'abord, nous allons aborder les aspects méthodologiques afin de comprendre la méthode de recherche et le rythme d'avancement de ce travail. Ensuite, l'état de l'art sera divisé en plusieurs parties. La première servira d'introduction aux éléments nécessaires à la bonne compréhension de la problématique, à savoir les notions de collaboration et de coopération, ainsi que de réalité virtuelle. La seconde montrera la mise en pratique de ces notions dans les solutions existantes. Puis, l'analyse se concentrera sur l'environnement de développement avec les casques de réalité virtuelle, les logiciels de conception et les *assets* nécessaires à la conception d'un réseau. Enfin, nous continuerons avec une mise en pratique dans la partie développement, mettant en avant les *assets* et les logiciels retenus. Avant de conclure, les résultats obtenus par la partie applicative seront confrontés à l'hypothèse soutenue. Le document se terminera par les recommandations pour la suite du projet et par diverses conclusions.

2. Aspects méthodologiques

Dans cette partie, nous allons évoquer l'objectif de travail à réaliser ainsi que la façon dont le projet est organisé entre l'auteur et Cyberlearn.

2.1. Méthodologie de recherche

Aujourd'hui, de nombreuses entreprises développent des applications en réalité virtuelle et en réalité augmentée : le financement dans ces domaines est déjà estimé à plus de trois milliards de dollars (Digi-capital, 2018). Est-il possible désormais de créer des applications collaboratives connectées en réalité virtuelle ? Pour répondre à cette question, une analyse du marché est nécessaire, ainsi qu'une partie de développement afin de mettre en pratique la problématique suivante : les *frameworks* open source permettent-ils de développer des applications collaboratives connectées en réalité virtuelle ? Le résultat de ce développement sera comparé à notre hypothèse, qui est la suivante : les *frameworks* open source permettent la création d'applications collaboratives connectées en réalité virtuelle.

2.2. Méthodologie de travail

Pour la réalisation de ce projet, un dérivé de Scrum, une des méthodologies agiles, a été utilisé. L'agile comporte quatre valeurs importantes (Pekka Abrahamsson, 2017) :

- Les individus et les interactions sont au-dessus des processus et des outils.
- Le logiciel compte plus que la documentation.
- La collaboration avec le client est plus importante que les termes du contrat.
- La flexibilité à suivre les demandes du client et ne pas être figé par un plan défini en amont.

En accord avec cette approche, l'auteur et Cyberlearn ont effectué plusieurs séances abordant l'aspect technique avec Christophe Hadorn, responsable du développement de réalité virtuelle, ainsi qu'avec Anne-Dominique Salamin pour les aspects relatifs à la gestion du projet.

2.2.1. Scrum appliqué

La méthode Scrum implique un *Product backlog* intégrant tous les aspects du projet sous forme d'*User Stories* avec des critères d'acceptance, des priorités, des *Story Points*, le MOSCOW, la date à laquelle la tâche a été acceptée et terminée ainsi que l'état d'achèvement de celle-ci. Les besoins du projet ont été définis avant le commencement de celui-ci ; cependant, à chaque sprint, des besoins ont pu être modifiés afin de correspondre à ce projet. L'exécution de ces tâches se retrouve répartie en plusieurs cycles appelés sprint, chacun d'entre eux durant une semaine.

3. État de l'art

Ici, nous expliquerons chaque notion pour améliorer et mieux appréhender la thématique, puis nous examinerons ce qui existe déjà sur le marché actuel.

3.1. Introduction des thèmes

Comme énoncé précédemment, nous allons approfondir les notions de base et déterminer ce que sont la collaboration/coopération et la réalité virtuelle.

3.1.1. Collaboration/coopération

Pour commencer, il est important de bien discerner les valeurs de la coopération et de la collaboration.

La coopération

La coopération dans la sphère professionnelle est une organisation collective du travail œuvrant à satisfaire un objectif général segmenté par des sous-tâches. Ces dernières sont ensuite attribuées à la bonne ressource pour qu'elle soit correctement réalisée. Cette segmentation impacte sur la ressource puisqu'elle sait depuis l'attribution de sa sous-tâche avec quelle ressource elle doit partager et communiquer des informations afin d'atteindre l'objectif. Lorsque toutes les ressources accomplissent leurs sous-tâches, cela forme l'objectif général. En résumé, la coopération est la division d'un objectif en sous-tâches réalisées individuellement et dont l'assemblage des résultats complète ledit objectif. (Piquet, 2009)

La collaboration

La collaboration, à l'inverse de la coopération qui responsabilise la personne à qui est attribuée la sous-tâche, responsabilise le groupe entier. Puisqu'il s'agit d'atteindre des objectifs communs qu'une ressource seule ne pourrait atteindre, la synergie de plusieurs ressources permet de les atteindre. La communication et le partage ne se réalisent plus entre les personnes concernées, mais bien avec tout le groupe pour gravir les obstacles et connaître l'avancée du travail. (Piquet, 2009)

Tableau 1 : Tableau récapitulatif des caractéristiques des notions de collaboration et de coopération

Le travail coopératif	Le travail collaboratif
Le travail se fait par addition de travaux individuels.	Le travail se fait par fusion et modification permanentes.
Les rapports sont orientés vers la hiérarchie.	Les rapports sont orientés vers l'équipe de travail.
Le mode de communication est plutôt asynchrone, même si le travail synchrone n'est pas impossible.	Le mode de communication alterne entre le synchrone et l'asynchrone.
Le travail individuel effectué est facilement identifiable à la fin et la responsabilité des acteurs est engagée.	Le travail individuel est difficilement identifiable à la fin et la responsabilité est constamment partagée.

Source 1 : Tableaux adaptés selon (Piquet, 2009)

La différence entre ces deux notions influe sur le type de ressource dans une entreprise. Dans le premier cas, les personnes restreignent la communication et le partage d'informations au strict nécessaire, la motivation étant personnelle. Dans le deuxième cas, les membres d'un groupe doivent être motivés au-delà de l'objectif personnel ; cette motivation est souvent associée à l'objectif lui-même. Les relations interpersonnelles sont nécessaires et ne sont plus une obligation hiérarchique, mais un sentiment d'appartenance à un groupe (Piquet, 2009).

Ces notions sont déterminantes pour les interactions à développer par la suite. Elles doivent être conçues pour rassembler ces deux profils.

3.1.2. Réalité virtuelle

Dans ce sous-chapitre, nous allons déterminer la notion de réalité virtuelle, la perception de la réalité, les périphériques améliorant l'immersion et l'intérêt de la réalité virtuelle.

DÉFINITION

La réalité virtuelle est l'immersion visuelle à 360 degrés d'un utilisateur dans un monde conceptualisé virtuellement par des logiciels informatiques. Le but de cette technologie est de rendre possible une activité sensorimotrice et cognitive pour un utilisateur dans un monde artificiel créé numériquement (Philippe Fuchs, 2006). Cependant, il ne faut pas confondre cette technologie à la réalité augmentée, car cette dernière n'immerge pas l'utilisateur dans un monde virtuel, mais ajoute du contenu virtuel au-dessus la réalité par le biais de caméras ou d'écrans.



Figure 2 : Illustration de la réalité virtuelle



Figure 3 : Illustration de la réalité augmentée

FONCTIONNEMENT

Pour comprendre comment la réalité virtuelle interagit avec le corps humain, nous allons aborder la perception du corps humain, regarder les différents visiocasques et quelques périphériques.

La perception du corps humain

L'homme perçoit la réalité qui l'entoure grâce aux cinq sens (la vue, l'ouïe, le toucher, le goût et l'odorat). Sa présence dans cette réalité par la proprioception est élevée au rang de sixième sens selon le Traité de la réalité virtuelle (Philippe Fuchs, 2006). Ce dernier sens comprend sa conscience

du mouvement par les informations provenant du système musculaire, de l'effort exercé pour interagir avec l'extérieur, de la mobilité des articulations, de l'étirement des tendons et l'état du corps par rapport à l'extérieur, présenté sous trois systèmes fonctionnant en synergie (Philippe Fuchs, 2006) :

- Le système visuel interprétant les informations reçues par la vision.
- Le système vestibulaire déterminant sa verticalité, à savoir s'il est debout, assis ou en mouvement. Nous pouvons imaginer ce système comme un baromètre. Il est souvent confondu avec l'oreille interne gérant la notion d'équilibre.
- Le système égo-centré est la représentation interne pour une personne de la position des objets l'entourant.

Lorsque ces trois systèmes sont désynchronisés, cela peut créer des vertiges, des nausées, voire des vomissements. Ce phénomène est appelé « cinétose » ou, en anglais, « *motion sickness* ».

Pour percevoir le monde virtuel comme le réel, il faut tromper ses sens. Les caques de réalité virtuelle troublent la vue, l'ouïe et une partie de la proprioception. Afin de perturber les autres sens, des dispositifs émergents seront décrits dans les périphériques VR.

Les périphériques VR

En 2018, de nombreux périphériques sont apparus sur le marché commercial, alors que d'autres se trouvent toujours au stade de prototype. Ils renforcent l'immersion de l'utilisateur dans le monde virtuel. Des gants existent désormais pour capter la position des mains et les matérialiser, à l'image de « CaptoGlove » (CaptoGlove, 2018). Le « Hi5 » possède des retours haptiques sous forme de vibrations (Noitom, 2018), alors que des gants encore au stade de prototype comme les « VR Gluv » permettent de ressentir la forme et la texture des objets (VRGLUV, 2018). Actuellement, les contrôleurs du HTC VIVE fournissent le moyen d'immerger l'utilisateur, car ils représentent les mains de l'utilisateur ; cependant, cela reste un composant électronique servant d'interface et, bien qu'il soit possible d'ajouter des modèles 3D dans les applications pour l'illusion de l'utilisation de mains virtuelles, ce ne sont pas les mouvements des doigts qui sont retranscrits. Tous ces gants augmentent la sensation réelle du touché, brisant encore plus la barrière entre le réel et le virtuel. D'autres appareils tels que des plastrons « HardLight VR Suit » (HardLight VR, 2018) permettent de ressentir l'environnement lorsque cela implique le torse du personnage virtuel. Un costume complet de « Tesla Suit » (Teslasuit, 2018) va plus loin en capturant le corps entier et en y intégrant des ventilateurs avec une variation de 10 degrés afin de ressentir la chaleur du décor environnant. D'autres encore utilisent les nerfs et les muscles pour capturer l'information d'un mouvement et pour en faire des actions prédéfinies dans des applications, comme le bracelet « MYO » (Bernhardt, 2015). La marque Taclim a, elle, conçu des chaussures utilisant un système de retour haptique offrant la sensation du sol fictif sur lequel marche l'utilisateur (Taclim, 2018). Ces périphériques créent un écosystème d'interface homme-machine renforçant l'illusion exercé sur les six sens de l'être humain.

L'intérêt

La plus grande force de la réalité virtuelle est d'immerger l'utilisateur dans un monde artificiel (Philippe Fuchs, 2006). Cette technologie d'immersion est renforcée par des périphériques trompant la perception du corps et qui permettent de rendre possible des situations impossibles ou difficilement reproductibles dans le monde réel, ainsi que de créer et de concevoir des objets en étant directement au cœur de l'application ou de s'évader par l'esprit tout en restant au même endroit physique. La réalité virtuelle permet donc la matérialisation de concepts abstraits.

3.2. Solutions existantes

Les solutions suivantes mettent en pratique des éléments vus lors de l'introduction. Il s'agit d'une liste non exhaustive d'applications.

3.2.1. *Escape room*

L'*escape room* (traduit littéralement par « salle d'évasion ») est un concept d'énigmes à résoudre en groupe dans le but de s'échapper de la pièce dans laquelle des personnes se retrouvent bloquées. Le travail d'équipe, la collaboration et la cohésion de groupe sont les éléments nécessaires à la réussite de l'évasion. Un groupe d'individus placé dans un milieu inconnu et ayant pour but de résoudre des casse-têtes se doit d'utiliser la connaissance de chacun de ses membres ainsi que de collaborer pour trouver la solution à une énigme donnée. Ce concept est né au Japon en 2007 ; il a commencé à s'étendre à partir de 2012 en Europe et dans le reste de l'Asie. Initialement basé sur les jeux vidéo, il avait l'ambition de se démarquer pour réaliser un jeu de rôle non pas virtuel, mais réel. Depuis lors, la complexité de ces énigmes a beaucoup évolué, allant de jeux dont le but est de sortir d'une pièce en résolvant quelques énigmes à des scénarios poussés nécessitant de s'échapper de plusieurs pièces, voire en résolvant des énigmes sous-marines. (Cain, 2015)

Un exemple d'*escape room* est proposé par Trap Game (Trap Game, 2018), une entreprise valaisanne spécialisée dans ce domaine. Voici le témoignage de Lionel Engel, étudiant HES-So en filière d'informatique de gestion ayant pratiqué le scénario intitulé « Vice Versa » : « *Six personnes sont coincées dans une salle où tous les objets sont fixés de manière inversée à la gravité, donnant l'impression de marcher à l'envers dans la pièce. Les énigmes se distinguent par la réflexion et la technologie puisqu'il fallait lire des messages via un miroir afin de trouver les codes d'ouverture pour une application de tablette qui actionnait l'ouverture de la pièce.* »



Figure 4 : Illustration de la pièce inversée du scénario « Vice Versa » de Trap Game.

L'entreprise de « *Virtual Room* » a sorti son propre concept d'escape room VR. Ils n'ont pour l'heure que deux scénarios (*Virtual Room*, 2018). La citation suivante est le regroupement du témoignage de Robert Seligmann et de Loïc Berthod, tous deux étudiants HES-SO en filière informatique de gestion ayant participé au premier scénario. « *Physiquement, la salle est construite par plusieurs cubes de deux mètres carrés accueillant chacun un utilisateur. L'expérience utilisait le casque HTC VIVE et se jouait en collaboration à deux ou à quatre joueurs. Étant séparés, les joueurs se réunissaient avec leurs avatars dans une seule pièce virtuelle. Le contexte était de trouver les ennemis qui parcouraient le temps et détraquaient le présent en laissant certains objets. La tâche à effectuer était de retrouver ces objets et de déterminer la prochaine destination temporelle. Les mécanismes sont simplifiés puisque le déplacement se réalisait par la marche limitée dans la surface du cube et les actions constituaient à trouver la solution à l'énigme, partager l'information et/ou partager l'objet pour réaliser l'objectif. Ainsi, dans le premier monde à l'image de douve d'un château, les joueurs devaient comprendre qu'il fallait chacun insérer une barre métallique dans des rouages dentés sur le sol afin de déverrouiller un mécanisme faisant descendre des cages en plein centre de la pièce dans lesquelles les informations pour la prochaine destination se trouvaient. Dans le deuxième monde, cette fois-ci sur la lune, chacun devait ouvrir une caisse à l'aide d'une visseuse et se passer les objets correspondant à leurs énigmes en prenant garde à la gravité lors du transfert. Le troisième monde se situait dans une pyramide à l'abord d'un tombeau et nécessitait de résoudre un mécanisme de réflexion de miroir afin d'ouvrir la voie. Dans le dernier monde se déroulant à l'époque préhistorique, il fallait à l'aide d'arcs et de flèches survivre à une attaque de T-Rex et de ptérodactyles. »*

3.2.2. Jeux VR

Les jeux vidéo actuels en réalité virtuelle proposent du contenu divers et varié, allant du *gameplay* solo comme le jeu « Beats Saber » qui met des sabres tout droit sortis de l'univers « Star Wars » (Beat Games, 2018) à des jeux collaboratifs comme «Keep talking and nobody explodes » où un joueur possédant un visiocasque doit désarmer une bombe en communiquant des informations sur celle-ci à un autre joueur hors casque qui, en fonction des descriptions reçues, doit trouver les bonnes séquences de désamorçage et les transmettre au premier joueur. (Steel Crate Games, 2018).



Figure 5 : Illustration du jeu « Beat Saber »



Figure 6 : Illustration d'une bombe à désamorcer du jeu «Keep talking and nobody explodes »

3.2.3. VR en entreprise

La réalité virtuelle peut se révéler très utile pour la conception de maquettes virtuelles ainsi que pour le marketing. Des architectes emploient déjà la réalité virtuelle pour réaliser des visites virtuelles de maisons ou pour montrer l'avancement de la réalisation de maisons à leurs clients et aux futurs propriétaires (Archdaily, 2018). Cependant, la conception de maquettes virtuelles a débuté bien avant cette émergence de casque pour le grand public. Dans l'industrie automobile par exemple, Volkswagen utilisait déjà la réalité virtuelle, mais à l'aide de la technologie CAVE (Volkswagen, 2018). Au lieu de miniaturiser les écrans pour les porter comme des lunettes, les images étaient projetées sur des toiles. En plaçant la personne au centre d'une pièce, puis en utilisant des algorithmes associant les appareils de projection au programme avec un moyen d'interaction tel qu'une télécommande, le décor projeté varie en fonction des actions réalisées.

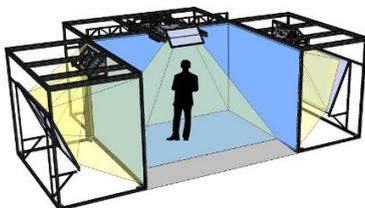


Figure 7 : Système Cave



Figure 8 : Conception de maquette virtuelle Range Rover 2011

L'entreprise automobile Jaguar a même réalisé, lors sa présentation à Los Angeles (Jaguar, 2016), une démonstration en réalité virtuelle d'I-PACE, une nouvelle voiture rentrant en compétition avec Tesla Model X. Celle-ci fut présentée virtuellement, les invités à cet événement ayant pu apprécier les différents angles de vue de la voiture ainsi que ses fonctionnalités de manière virtuelle. Lors de cette visite, ils étaient accompagnés par un employé virtuel de Jaguar servant de guide et d'instructeur sur l'aspect technique de l'automobile (Nafarrete, 2016).

Epic Games, développeur et éditeur de jeux vidéo, utilise son propre logiciel de conception, Unreal Engine, afin de concevoir la partie graphique de certains jeux vidéo (Sweeney, 2016).



Figure 9 : Conception du *level design* directement en réalité virtuelle.

La médecine est une discipline favorisant l'utilisation de la VR ; elle l'emploie notamment pour guérir des troubles psychiatriques et le traitement de l'anxiété (Jessica L. Maples-Keller, 2017) . La Dre Maples-Keller, conjointement avec ses collègues, entend par là les phobies spécifiques à un objet ou à une situation, les troubles d'anxiété sociale, le stress post-traumatique ainsi que d'autres maladies psychiques. L'utilisation de la réalité virtuelle peut améliorer l'état des patients, voire les guérir complètement dans certains cas mais doit être adaptée au besoin du patient sous forme de thérapies dynamiques (Jessica L. Maples-Keller, 2017).

Le sport utilise cette technologie via le *live streaming*. Utilisant le format mobile de la réalité virtuelle avec les dispositifs Cardboards, Samsung Gear ou Oculus Go, le *live streaming* permet de prendre la position de caméra dédiée dans l'infrastructure d'un match de sport, donnant une interface d'une salle VIP pour l'utilisateur avec en grand écran la diffusion en direct d'un événement sportif ou culturel (Live like vr, 2017).

Le tourisme est fortement touché par le potentiel du marketing 4.0. Cette version dite 4.0 construit de nouveaux indicateurs basés sur le parcours du client, se divisant en quatre stratégies : augmenter l'attraction, susciter la curiosité, accroître l'engagement et renforcer l'affinité avec la marque (Philip Kotler, 2017). Comme dit précédemment, la réalité virtuelle permet la visite de lieux et d'événements en immergeant le consommateur, tout en lui donnant envie d'y être pour de vrai.

3.2.4. Apprentissage VR

Plusieurs applications ayant un but d'apprentissage en réalité virtuelle font leur apparition sur le marché. Il peut s'agir d'applications destinées aux particuliers, à des écoles ou à des institutions formatrices.

« Job Simulator » (Owlchemy Labs, 2018) est un jeu vidéo VR de divertissement qui introduit une personne dans un métier. Même si celui-ci est représenté dans sa version minimale, il permet une immersion totale dans un environnement. De ce fait, le divertissement s’allie à l’apprentissage d’une vocation par certains gestes primaires. De plus, l’application étant conceptualisée en anglais, les utilisateurs n’étant pas familiers avec cette langue pourront parfaire leur vocabulaire adapté aux différentes situations de travail.



Figure 10 : Capture du jeu « Job Simulator », assignant à l'utilisateur un rôle de vendeur dans un fast-food.

Le domaine de la science utilise également la réalité virtuelle. « MEL Chemistry VR » est une application permettant d’aborder la chimie d’une nouvelle manière. Elle permet de visualiser des éléments chimiques en 3D directement dans un élément du décor (Mel Science, 2018). « SuperChem VR » invite l’étudiant dans un laboratoire de chimie virtuel afin d’y réaliser des expériences chimiques. Ce procédé permet l’éducation de la matière sans pour autant exposer l’étudiant à un danger dû à certains mélanges. De surcroît, les coûts de développement sont minimes en comparaison avec la construction d’un laboratoire tout équipé (James, 2017).

3.2.5. Simulation VR

Lorsqu’il s’agit de simuler un comportement ou un événement trop coûteux ou impossible à réaliser avant un certain moment, la réalité virtuelle prend tout son sens. SimForHealth crée du contenu VR sur mesure pour l’éducation médicale (SimforHealth, 2018). UbiSim a réalisé un module VR complémentaire pour la prise de sang (UbiSim, 2018). Ce module est utilisé à la Haute École de Santé La Source à Lausanne. Selon un entretien oral avec Mme Emmanuelle Mazzitti, assistante de projet à la Haute Haute École soutient l’ambition d’intégrer des formations VR dans le cursus pour les soins infirmiers. En mars 2018, dans le cadre d’un hackaton sur le thème de la santé, un groupe d’étudiants de la filière Informatique de gestion de la HES-SO a réalisé un prototype de simulation VR d’intervention médicale dans des environnements extrêmes pour diminuer les coûts de la formation des urgentistes du GRIMM(groupe d’intervention médicale en montage présent en Valais). (Hackathon, 2018).

Le domaine militaire utilise également la réalité virtuelle dans le but de diminuer les coûts. La VR permet à l’armée américaine de concevoir des villes temporaires servant à la simulation d’interventions sur le terrain (Virtual reality combat simulation, 2017). La VR est également utilisée

pour l'apprentissage et pour exercer plusieurs scénarios de véhicules, notamment avec des avions (Virtual reality air force training, 2017).

3.3. Prise de position

Dans cet état de l'art, nous avons pu comprendre ce qu'est la réalité virtuelle, positionner sa relation avec les sens humains, analyser l'intérêt de son utilisation ainsi que démontrer un certain nombre de domaines d'application :

- La santé
- La psychologie
- L'éducation
- La simulation
- Le tourisme
- La conception virtuelle
- Le sport
- Les jeux vidéo

Au-delà de ces aspects, nous avons constaté que la réalité virtuelle est devenue un sérieux outil pour l'immersion et l'interactivité, permettant de collaborer, comme nous l'avons vu dans les jeux, la conception virtuelle ou le domaine militaire. À partir de ce postulat, nous allons déterminer le *framework* le plus adapté et réaliser une application VR collaborative connectée.

4. Définition des besoins

Dans ce chapitre, nous allons aborder le scénario et les besoins pour l'application VR collaborative connectée qui sera créée plus tard. Globalement, il s'agit de concevoir une scène dans laquelle deux personnes pourraient se voir, interagir entre elles et communiquer en réseau alors qu'elles seraient physiquement et géographiquement séparées. Des *logs* seront enregistrés pour l'évaluation future de la qualité de la collaboration.

4.1. Scénario

Le binôme se retrouve à l'intérieur d'une salle dans laquelle il n'y a qu'une seule échappatoire. Leur objectif est de la trouver, puis de l'atteindre dans un temps imparti. Pour se faire, ils devront coopérer afin de construire un passage et collaborer par le biais de la communication orale dans le but de déterminer le meilleur procédé pour y arriver. Un compte à rebours permettra de limiter l'application dans le temps et de déterminer si l'objectif est atteint, c'est-à-dire si tous deux ont réussi à rejoindre la sortie dans le temps imparti.

4.2. Réseau

Le réseau devra répondre à deux critères :

1. Connexion de deux utilisateurs à une scène.
2. Synchronisation des données sur le réseau.

4.3. Interactions des utilisateurs

Les *use cases* suivantes servent à déterminer les actions pouvant être réalisées par les utilisateurs pour atteindre l'objectif.

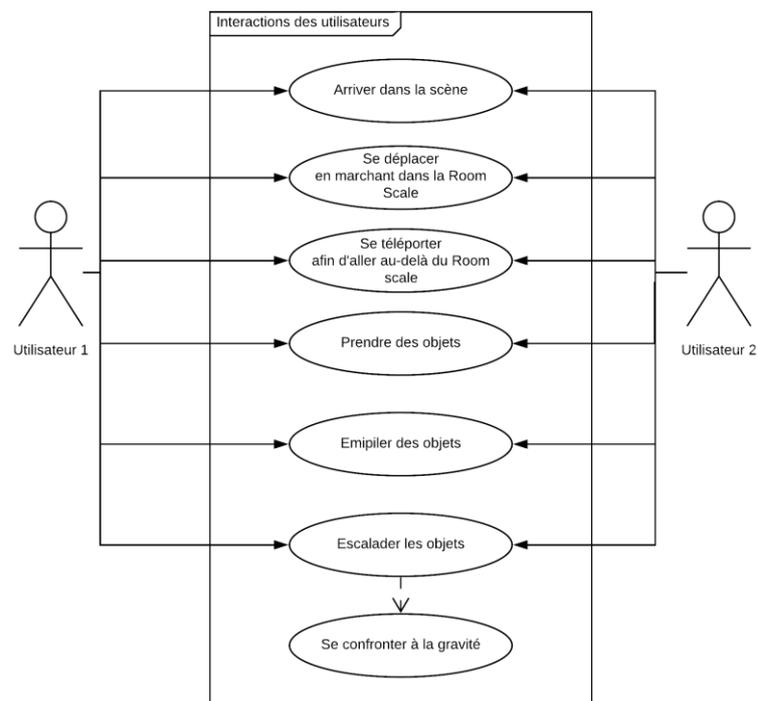


Figure 11 : *Use case* des interactions par utilisateur (réalisée par l'auteur)

4.4. Encadrement de l'application

Pour garder un contrôle sur le temps de l'application et le résultat de l'objectif, les points suivants sont nécessaires :

- Synchronisation d'un compte à rebours entre les instances.
- Système de fin d'application.

4.5. Logs

La création de *logs* permettra à Cyberlearn de réaliser, ultérieurement à ce travail de Bachelor, des tests et des analyses. Il est nécessaire de prendre en compte :

- Les *logs* des actions.
- Les *logs* des déplacements.
- Les *logs* des échanges vocaux.

4.6. Récapitulatif des caractéristiques techniques

La partie de développement devra répondre aux caractéristiques suivantes :

1. Réseau :
 - a. Connexion de deux utilisateurs à une scène.
 - b. Avatars visibles.
 - c. Synchronisation des données sur le réseau.
2. Utilisateurs :
 - a. Capacité à prendre des objets.
 - b. Capacité à construire.
 - c. Soumission à la gravité et capacité d'escalade.
 - d. Capacité à se déplacer et à se téléporter.
3. Communication :
 - a. Capacité à communiquer entre les utilisateurs.
4. Encadrement de l'application :
 - a. Synchronisation d'un compte à rebours entre les instances.
 - b. Système de fin d'application.
5. *Logs* :
 - a. *Logs* des actions.
 - b. *Logs* des déplacements.
 - c. *Logs* des échanges vocaux.

Ce récapitulatif permettra d'évaluer quelle fonctionnalité a bien été implémentée et de démontrer quelle partie n'aurait pas été respectée.

5. Analyse des outils

Dans le chapitre qui suit, il s'agira d'analyser les casques VR et les logiciels de conception dans le but de définir l'environnement matériel ainsi que le logiciel le plus adapté à notre utilisation. Puis, en fonction de cet environnement, nous évaluerons les différents procédés pour réaliser une application.

5.1. Casques VR

Une comparaison sur les différents casques sortis durant l'année 2016 est effectuée. 2016 correspond à l'année durant laquelle la réalité virtuelle est devenue accessible au grand public avec l'arrivée du

casque d'Oculus. C'est également en 2016 que Cyberlearn a fait l'acquisition du casque de réalité virtuelle HTC VIVE. En ajoutant le casque PSVR, les principaux acteurs pionniers de ce marché alors émergent auront été cités.

Tableau 2 : Comparatif réalisé par l'auteur selon des sources multiples

Casque	Oculus Rift	HTC Vive	PSVR
Constructeur	Oculus	HTC-Valve	Sony
Fonctionne sur la/les plateforme(s)	Oculus Home (Ordinateur)	SteamVR, Viveport (Ordinateur)	Playstation Network (Console PS4)
Image par seconde	90Hz	90Hz	90/120Hz
Résolution (par œil)	1080x1200 px	1080x1200 px	960x1080 px
FOV	110°	110°	100°
Poids	470g	555g	610g
Position d'utilisation	Assis	Debout	Assis
Nombre de caméras	1	2	1
Aire de jeu	Max 3.9m ²	Min 3m ² · Max 12m ²	Max 4.56m ²
Contrôleur	Manette de la console Xbox One	Contrôleur VIVE	Manette de la console PS4
Prix de départ	699€	799€	399€
Prérequis	Configuration PC requise : Intel i5-4590, AMD Ryzen 5 1500X ou meilleur NVIDIA GTX 970, AMD Radeon R9 290 ou meilleur Ou la certification « VR Ready » ¹	Configuration PC requise : Intel Core i5-4590/AMD FX 8350 ou meilleur NVIDIA GeForce GTX 1060, AMD Radeon RX 480 ou meilleur Ou la certification « VR Ready »	Console PS4 PlayStation Camera

¹ Un ordinateur ayant le logo ou la mention de VR Ready signifie qu'il est certifié compatible à l'utilisation de casques et d'application VR. (NVidia, 2016)

Frais supplémentaires	Contrôleur Oculus Touch	Trépied pour caméra	Contrôleur PS Move

Le casque HTC VIVE possède une résolution d'écran équivalente à celle de l'Oculus Rift et un champ de vision couvrant presque les 120° de la vision binoculaire (Philippe Fuchs, 2006). La fréquence des images capturées par l'œil est de 25 images par seconde pour un mouvement fluide. Pour des écrans monoscopiques tels que des téléviseurs ou des smartphones, une fréquence de 50Hz sera suffisante ; cependant, pour les écrans stéréoscopiques employés pour les applications en réalité virtuelle, le double sera nécessaire afin de garder ce même nombre par œil (Philippe Fuchs, 2006). Ainsi, le PSVR possède davantage d'images par seconde. Concernant la surface d'application, HTC surpasse largement ses concurrents et se démarque par son système de déplacement nommé « Room Scale ». Grâce aux deux bases disposées en diagonale de la surface de déplacement, le casque ainsi que les deux contrôleurs VIVE reçoivent des lasers infrarouges localisant leur position spatiale. L'utilisateur peut donc marcher dans la surface couverte par les bases, s'accroupir ou tourner sur lui-même (skyfunvr, 2018). De plus, le PSVR ne peut pas directement entrer dans la catégorie des moyens d'apprentissage puisqu'il est associé à une console de jeux. De ce fait, le HTC VIVE paraît être le meilleur casque VR pour expérimenter de nouveaux outils d'apprentissage VR et d'expérimentation.

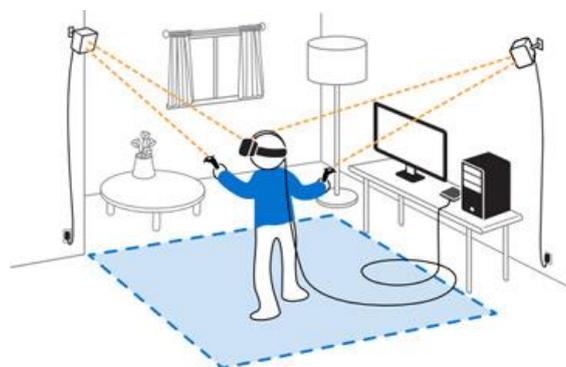


Figure 12 : Illustration du « Room Scale » du HTC VIVE

5.2. Détermination du meilleur logiciel de conception VR

Précédemment, nous avons pu apercevoir plusieurs utilisations de la réalité virtuelle dans différents domaines. À présent, nous allons comparer les différents logiciels de conception avec lesquels il est possible de créer des applications VR.

5.2.1. Unity



Figure 13 : Logo d'Unity

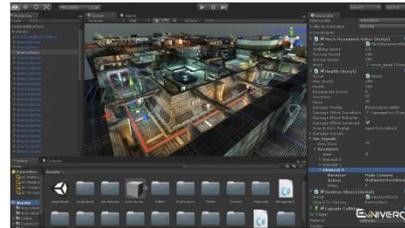


Figure 14 : Interface Pro d'Unity

La philosophie de cette entreprise est « *At Unity, we work every day to democratise development, solve hard problems and enable success... and we're not done yet.* » (Unity, 2018)

5.2.2. Unreal Engine



Figure 15 : Logo d'Unreal Engine

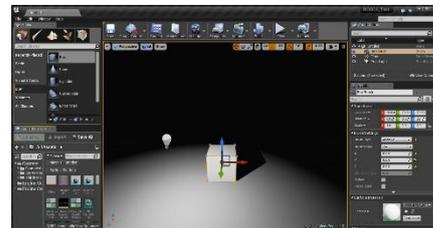


Figure 16 : Interface d'Unreal Engine

Unreal Engine se considère comme le moteur de jeux vidéo le plus complet avec la plus puissante qualité graphique.

5.2.3. Cry Engine



Figure 17 : Logo de Cry Engine



Figure 18 : Interface de Cry Engine

Comme ses deux principaux concurrents, CryEngine possède un but précis : « *to create the most powerful game engine in the industry, and to give creators all across the globe the tools to harness this power to create world-class gaming experiences, no matter their budget or team size.* » (Crytek, 2018)

Dans le tableau suivant, nous comparons les trois programmes de conceptions précités.

Tableau 3 : Comparaison des logiciels de conception VR

Logiciel	Unity	Unreal Engine	Cry Engine
Consoles supportées	Xbox 360, Xbox One, PlayStation 3, PlayStation 4, PlayStation Vita, Wii U, Nintendo Switch, Nintendo 3DS	PlayStation 4, Xbox One, Nintendo Switch	Xbox One, PlayStation 4
Systèmes d'exploitation supportés	Windows, macOS, Linux	Windows, macOS, Linux, SteamOS, HTML5	Windows, Linux
Systèmes d'exploitation mobiles supportés	Windows Phone, iOS, Android, BlackBerry 10, Tizen	iOS, Android	iOS, Android
Langages supportés	C#, Javascript	C++, Blueprints	C++, C#, Lua
AR supportée	Hololens	Hololens	Hololens
VR supportées	Oculus Rift, Gear VR, Google Daydream, Cardboard, SteamVR/HTC Vive, Oculus GO, Casque Microsoft de réalité mixte	SteamVR/HTC Vive, Oculus Rift, OSVR, Google VR/Daydream, Samsung Gear VR, Oculus Go, Casque Microsoft de réalité mixte en cours d'intégration	Oculus Rift, HTC VIVE
Qualité graphique	Graphiques simples, cartoon	Graphiques réalistes	Graphiques réalistes
Communauté	Forum Tutoriel Documentation officielle Blog	Forum Tutoriel Documentation officielle Blog	Forum Tutoriel Documentation officielle Blog
Prix de licence	Gratuit - Personnel ~25\$/ mois - Plus	Gratuit - Personnel	Gratuit - Personnel

	125\$/mois - Pro	Entreprise - 5% de <i>royalties</i> sur l'application.	Entreprise - Package <i>customisable</i>
--	------------------	--	---

Le centre e-learning Cyberlearn a besoin de flexibilité puisqu'il cherche constamment de nouveaux moyens d'apprentissage ; un outil trop sélectif restreindrait donc son champ d'action. L'accessibilité à des ressources pour une meilleure appréhension du logiciel est tout aussi importante. En revanche, la qualité du produit est indifférente, pour autant qu'elle soit présentable et facilement identifiable par les utilisateurs. L'aspect le plus important réside sur le nombre de plateformes différentes en réalité virtuelle.

5.2.4. *Résultat du classement des logiciels*

Le classement suivant résume, par ordre de préférence, les programmes répondant au mieux aux critères :

1. Unity
2. Unreal Engine
3. Cry Engine

Pour créer des applications mobiles de réalité virtuelle ou augmentée sur Cardboard ou Hololens, Unity et Unreal Engine paraissent équivalentes, contrairement à Cry Engine qui ne permet pas cela. Tous deux sont suivis par une grande communauté et comportent un grand nombre de ressources disponibles. De plus, il est couramment admis qu'Unreal Engine est plus complexe à prendre en main qu'Unity pour les novices. Puisque la création de jeu vidéo n'est pas le but premier de Cyberlearn et que ce dernier ne vise pas un graphisme ultra réaliste, Unity, qui supporte le plus grand nombre de *devices*, convient parfaitement à ses besoins.

5.2.5. *Détermination des assets à utiliser pour Unity*

À présent que le choix de la plateforme de conception et du casque a été établi, nous allons analyser les *add-ons* permettant à Unity de concevoir des applications de réalité virtuelle connectée.

5.2.6. *Présentation des assets open sources existants*

Assets de création d'interfaces compatibles avec le HTC VIVE :

- ***Plugin Steam VR***



Figure 19 : Logo du *plugin* Steam VR

Il s'agit de la librairie native pour programmer sur le casque de HTC.

- **Virtual Reality Tool Kit**



Figure 20 : Logo de Virtual Reality Tool Kit (VRTK)

Comme son nom l'indique, il s'agit d'un *toolkit* se basant sur le *plugin* Steam VR. Il est capable de réaliser des cas basiques de comportement VR tels que différents types de téléportation, de saisies d'objet ou d'autres interactions facilitant la prise en main de la gestion d'interface virtuelle (VRTK, 2018).

Assets permettant de concevoir un réseau sur Unity :

- Unity Network

Unity Network est le *plugin* standard d'Unity pour la création rapide de contenu multijoueur. Il est hautement personnalisable. Unity fournit également les serveurs pour héberger ces applications. Il permet la création de lobbies, la gestion des joueurs ainsi que le transfert d'informations d'une application connectée. Il ne possède officiellement qu'un seul *chat* écrit (Unity, 2018).

- Photon Engine



Figure 21 : Logo de Photon Engine

Moteur de réseau indépendant d'Unity simplifiant l'utilisation du réseau, Photon est indépendant des logiciels et par conséquent compatible avec ceux listés précédemment. Il est capable de générer des lobbies dans lesquels les utilisateurs peuvent se connecter ainsi que de gérer la communication d'informations dans l'application et la communication écrite et orale entre joueurs (Photon Engine, 2018).

- Forge Networking



Figure 22 : Logo de Forge Networking

Il s'agit d'un *plug-in* comprenant plusieurs bibliothèques ayant pour but de mettre en place un réseau communicant entre plusieurs appareils. L'équipe de développement apprécie l'aide de sa petite communauté pour améliorer ce système afin qu'il soit sans failles. Étant toujours en bêta depuis juillet 2017, les développeurs encouragent son utilisation par le plus grand nombre possible et le signalement des potentiels problèmes que les joueurs pourraient trouver (Forge, 2018).

- Master Server Framework



Figure 23 : Logo de Master Server Framework

Ce *framework* se concentre sur la création d'un serveur permettant l'authentification des utilisateurs, la création d'un profil sans utilisation de base de données, l'hébergement d'une application Unity, un système de création de scènes ainsi qu'un système de communication. Il s'agit d'un module complémentaire de gestion de serveur qui ne comporte aucun comportement pour le transfert d'informations dans un jeu, ce rôle revenant à Unet (Poškevičius, 2017).

- DarkRift 2



Figure 24 : Logo de Dark Rift 2

Ce *plug-in* est basé sur son prédécesseur, Dark Rift 1. Il ne fournit aucune gestion de lobby et sa fonctionnalité principale se concentre sur l'optimisation des échanges d'informations dans l'application elle-même (DarkRift Networking, 2018).

5.2.7. Comparatif des moyens de conception d'un réseau

En reprenant les besoins nécessaires à la partie développement évoquée au point 4.1, nous pouvons établir les critères suivants :

- *Cloud* : Possibilité d'héberger l'application sur un espace en ligne gratuitement.
- Gestion de scènes : un environnement prêt à recevoir des joueurs.
- Gestion des joueurs : la gestion des joueurs entrants et sortants ainsi que leur position respective et leurs actions.
- Gestion de la communication : un moyen de communiquer.
- Communauté : taille de la communauté pour trouver des informations et taille de la documentation.
- Documentation officielle : la qualité des documents explicatifs et des tutoriels.
- Autre documentation : la diversité des documents, tutoriels ou des autres moyens d'apprentissage.
- SteamVR supporté : le *network* prend en charge l'utilisation de cette implémentation.
- VRTK supporté : le *network* prend en charge l'utilisation de cette implémentation.

Nous utiliserons la pondération suivante :

- 0 → ne correspond pas du tout au critère.
- 1 → correspond partiellement au critère.
- 2 → correspond parfaitement au critère.

Tableau 4 : Matrice de comparaison

Critère	Coefficient	Unet	Photon	Forge Networking	Master Server Framework	DarkRift 2
Cloud	3	2	2	1	1	1
Gestion de scènes	4	2	2	1	1	1
Gestion des joueurs	4	2	2	1	1	2
Gestion de la communication	3	1	2	1	1	0
Communauté	2	2	1	1	1	1
Documentation officielle	1	2	2	2	1	1
Autre documentation	2	2	1	2	1	1
SteamVR supporté	4	2	2	0	0	0
VRTK supporté	5	2	2	0	0	0
Résultat		53	50	22	19	20

5.2.8. Résultat du classement

Selon la matrice de comparaison listée ci-dessus, voici, dans l'ordre, les meilleurs moyens pour réaliser un *network* sur Unity :

1. Unet
2. Photon
3. Forge Networking
4. DarkRift 2
5. Master Server Framework

Unet possède un net avantage en raison des nombreux tutoriels existants et de sa grande communauté passablement active. Il ne possède pas de *chat* vocal à l'image de celui de Photon, mais sa large communauté compense ce point. À partir du troisième moyen de conception listé dans notre classement, nous pouvons affirmer que ces moyens ne sont pas adaptés pour l'utilisation de la VR.

5.2.9. Assets sélectionnés

Au vu des différents éléments précités, l'*asset* sélectionné pour concevoir un réseau sur Unity est Unet.

6. Quelques bases d'Unity

Avant d'entrer dans le vif du sujet, à savoir la partie liée au développement, nous présentons ici une brève documentation d'Unity dans le but de mieux comprendre les étapes réalisées par la suite. Unity sert à manipuler des objets 3D et ne comporte pas d'éditeur de script dédié.

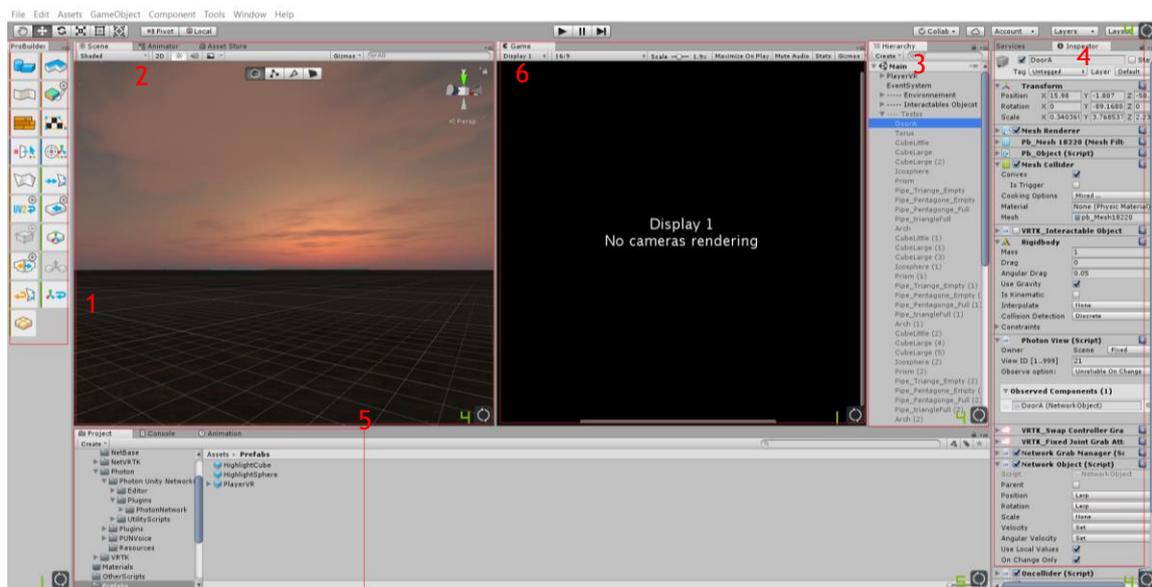


Figure 25 : Explication de l'interface à partir d'une capture d'écran de l'auteur.

L'interface est entièrement modulable.

1. La palette d'outil pour la conception 3D de « Probuilder ».
2. L'onglet « Scène » représente l'éditeur. C'est dans celle-ci que nous ajoutons et manipulons les objets 3D.
3. La « Hiérarchie » est la liste des *Game Objects* que contient la scène. Un *Game Object* est l'appellation d'un objet dans Unity.
4. L'« Inspecteur » affiche les *components* (modules) d'un objet sélectionné ainsi que les scripts qui y sont rattachés.
5. L'onglet « Projet » représente l'arborescence des fichiers du projet. Ici sont montrés les *prefabs* ; il s'agit d'un « Game Object » que nous convertissons en tant que *prefab*. Il généralise l'ensemble de l'objet pour faciliter sa manipulation entre les scènes. Une modification du *prefab* entraînera une modification dans toutes les scènes où il est implémenté, à l'inverse de l'objet qui lui reste un objet indépendant dans chaque scène et pour qui la modification est manuelle.
6. L'onglet « Game » est la vue finalisée de notre scène par l'aperçu d'une caméra. Lors d'un développement VR, cette dernière est inaccessible tant que nous n'exécutons pas le programme.

Quelques *components* :

Transform : module servant à localiser la position spatiale de l'objet, sa rotation ainsi que sa taille.

Collider : module matérialisant l'aspect physique de l'objet virtuel sur la base d'une forme simple ou calquée sur des *meshs*. Dans le monde réel, cela équivaut au toucher de la matière. Il nous est ainsi impossible de traverser des objets physiques tels qu'une table par exemple.

Trigger : module fonctionnant de manière opposée au précédent ; il génère une zone dans laquelle il peut être déterminé si un objet est à l'intérieur ou à l'extérieur de ladite zone. Il est possible de le traverser.

RigidBody : module servant à ajouter de la gravité à un objet.

Mesh renderer : module permettant de présenter visuellement un objet 3D. En le désactivant, la forme deviendra invisible, mais existera via le *Collider* ou le *Trigger*.

7. Développement

Dans cette partie consacrée à la conception de l'application VR collaborative connectée, nous allons définir les outils qui sont utilisés et le processus de conception, puis décrire l'implémentation du réseau sur Unity.

7.1. Environnement de développement

Deux ordinateurs portables ayant la certification « VR Ready » sont nécessaires, de même que deux casques HTC VIVE. Concernant les logiciels utilisés, nous développons l'application sur le moteur de jeu Unity, ce en raison des différents éléments précités. L'écriture des scripts sera faite en C# sur le logiciel Visual Studio Community. Ce dernier est le plus adéquat puisqu'il possède des fonctionnalités d'autocomplétion et de gestion des erreurs, contrairement à MonoDevelop, éditeur souvent associé à Unity. Notons toutefois que l'éditeur de script n'a pas d'importance et n'affecte en rien le comportement d'Unity.

7.2. Processus de conception

Dans le but d'avoir une ligne directrice uniforme, nous avons réalisé le processus de développement suivant :

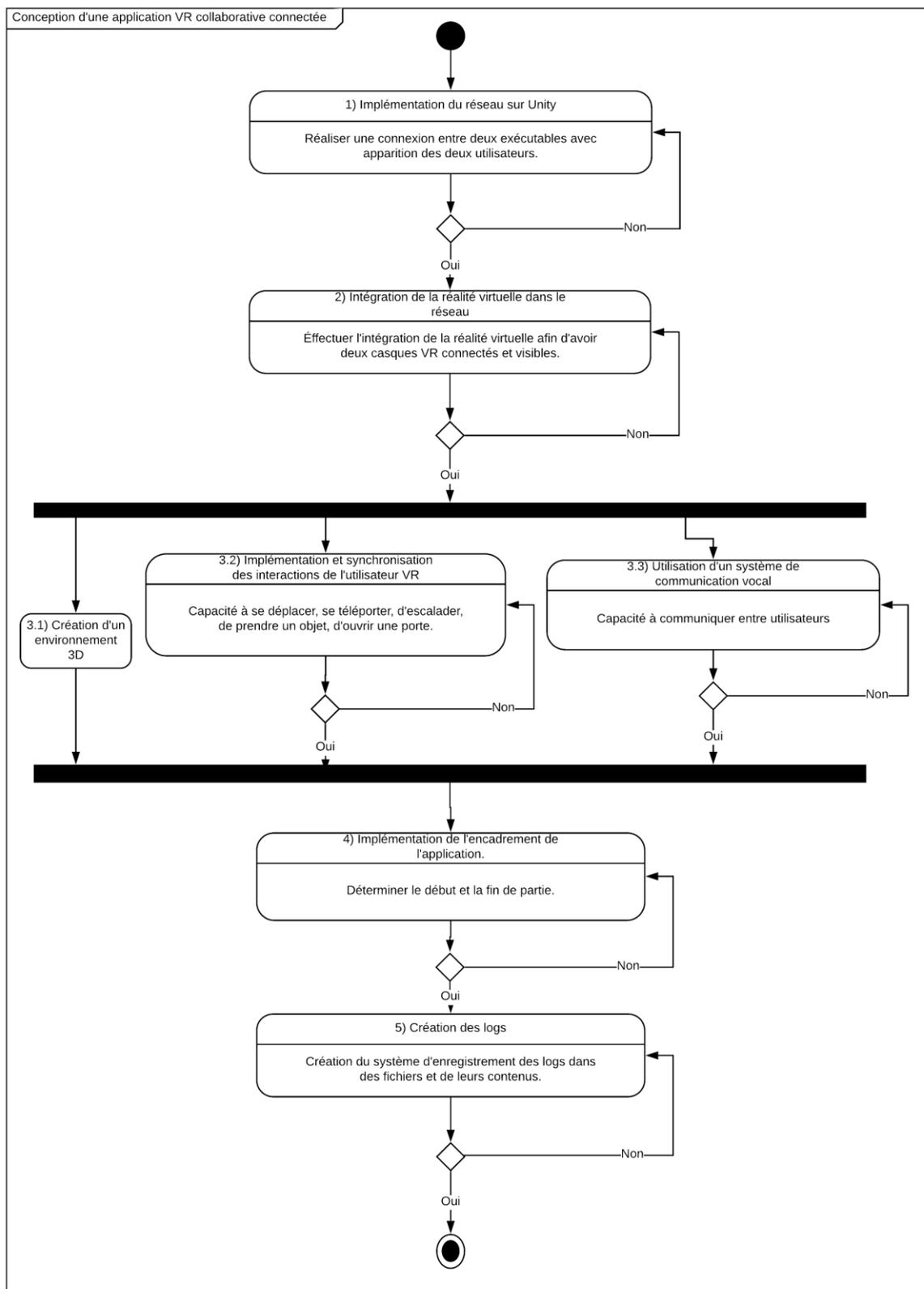


Figure 26 : Processus de conception (réalisé par l'auteur via Lucidchart.com)

Étape 1 : Vérifier que l'outil utilisé pour implémenter le réseau fonctionne sur une application basique. Deux utilisateurs se connectent à la même scène et s'aperçoivent sous forme d'avatars.

Étape 2 : Lorsque l'outil est vérifié, nous implémentons le moyen choisi pour intégrer l'interface virtuelle. Il doit correspondre aux mêmes résultats que la première étape.

Étape 3 : Génération du corps de l'application. Ces étapes sont indépendantes les unes des autres.

3.1) Créer un décor minimaliste.

3.2) Intégrer toutes les fonctionnalités des utilisateurs et les synchroniser sur le réseau.

3.3) Intégrer un moyen de communication interne ou externe.

Étape 4 : Un compte à rebours doit pouvoir être synchronisé dans toutes les instances en utilisant le même procédé de synchronisation de données que le point 3.2. Afin de déterminer le résultat de l'application, un déclencheur pour le compte à rebours doit exister, une zone de fin basée sur le décor validant un succès et un dépassement du temps conduisant à un échec.

Étape 5 : Les *logs* se servent de toutes les étapes précédentes. Ils utilisent non seulement le compte à rebours comme facteur de liaison entre les déplacements effectués, mais également les actions réalisées et les échanges vocaux.

7.3. Types de serveurs

Dans l'univers des jeux vidéo, il existe plusieurs types de serveurs dont l'utilisation varie en fonction des besoins du jeu développé.

7.3.1. Serveur avec autorité

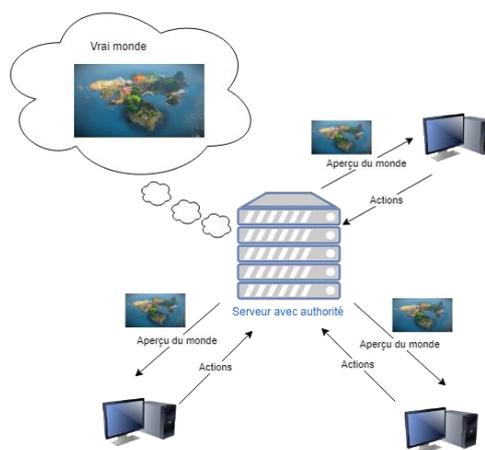


Figure 27 : Fonctionnement d'un serveur avec autorité (réalisé par l'auteur)

Le serveur est le maître de toutes les autorités et possède le vrai monde du jeu, y compris toutes ses composantes. Les clients connectés ne possèdent qu'un aperçu de ce monde ; autrement dit, ils voient ce monde par une fenêtre et avec un décalage. L'application tourne sur le serveur et reçoit les informations du client qui modifie le monde. Puis, le monde se met à jour et renvoie un aperçu de celui-ci à tous les clients (Byl, 2018).

7.3.2. Serveur sans autorité

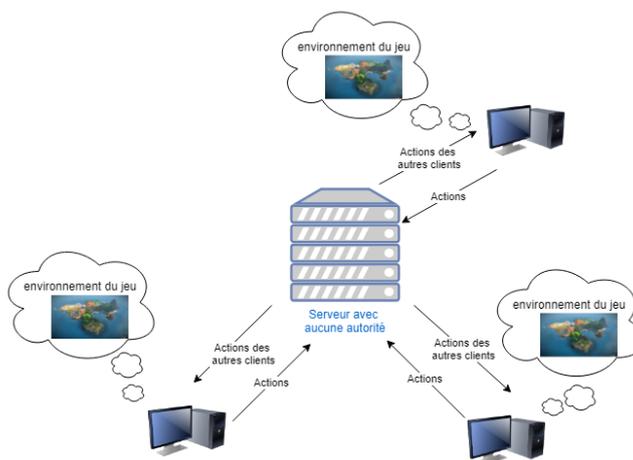


Figure 28 : Fonctionnement d'un serveur n'ayant pas d'autorité (réalisé par l'auteur)

À l'inverse du précédent serveur, ici, le vrai monde du jeu tourne en simultané sur tous les clients. La latence ne peut survenir que dans les informations échangées avec l'extérieur, comme par exemple le déplacement des autres joueurs et leurs actions (Byl, 2018).

7.3.3. La notion de latence

La notion de latence signifie qu'il y a un temps de retard dans la réception des informations du serveur. Cela dépend de la bande passante, de la vitesse du réseau, du nombre de joueurs connectés ainsi que du nombre de messages envoyés. Sur le premier type de réseau, l'ensemble du jeu souffrira de ce désagrément, puisque le réseau envoie aux clients une version antérieure au serveur. En revanche, sur le deuxième, seules les informations provenant des autres terminaux seront affectées par ce phénomène (Byl, 2018).

7.3.4. LAN & WAN

Il existe plusieurs façons de créer un réseau. Dans ce document, nous évoquerons uniquement le LAN et le WAN.

LAN ou *Local Area Network* est une connexion locale entre plusieurs ordinateurs câblés à un serveur ou à un routeur.

WAN ou *Wide Area Network* est une connexion à distance utilisant Internet pour rassembler les terminaux aux serveurs.

7.4. Déploiement d'Unet et de VRTK

Dans ce premier projet, nous souhaitons compléter les critères de l'application connectée listés au point 4.6. Contrairement à VRTK, Unet ne nécessite pas d'importation, puisqu'il est intégré de base à Unity.

7.4.1. Unet

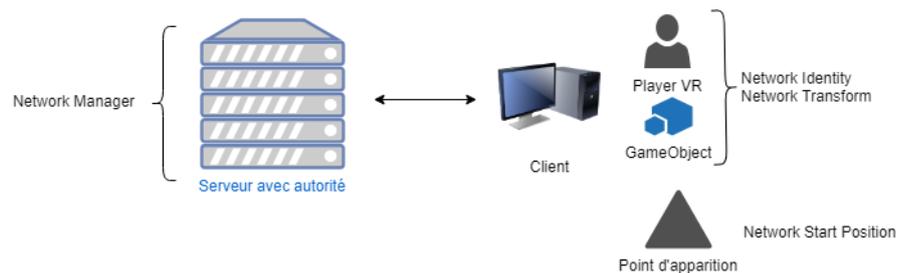


Figure 29 : Configuration de Unet sur Unity

Le serveur est géré par le script de « Network Manager ». Tous les paramètres IP y résident, ainsi que le *prefab* du « Player VR » qui permet de générer une copie sur le réseau pour chaque nouvelle connexion d'un utilisateur. Chaque *Game Object* devant se mouvoir et transmettre sa position doit comporter un « Network Identity » afin d'obtenir un *id* propre à Unet, et un « Network Transform » pour la synchronisation dans la scène. Le « Network Start Position » communique au serveur que ce *Game Object* sert à la génération d'objets sur son réseau.

7.4.2. Étapes fonctionnelles réalisées

1) Implémentation d'Unet sur Unity :

La gestion d'une scène accueillant des utilisateurs se construit par le biais de *Rooms*. Le « Network Manager » gère cette partie avec efficacité. Le premier joueur connecté a le choix de créer une partie en local, en LAN ou sur Internet.



Figure 30 : Menu de création d'une instance Unet

Dans un premier temps, nous avons utilisé le « LAN Host » sur Unity et généré un exécutable se connectant sur « LAN Client(c) ».

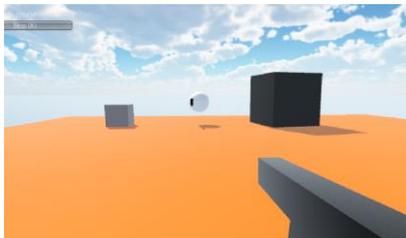


Figure 31: Succès du test d'implémentation d'Unet

Par la suite, quelques difficultés sont survenues lors de cette première étape et les étapes 2, 3.2, 3.3. Ces complications seront présentées dans la section suivante. L'étape suivante est celle de la création du décor. C'est pourquoi nous arrivons directement à l'étape suivante du processus.

3.1) Création d'un environnement 3D :

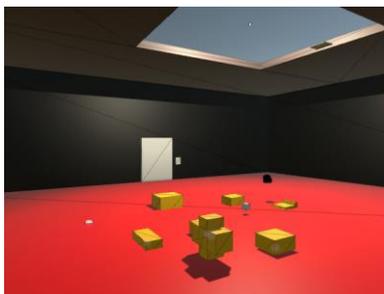


Figure 32 : Environnement 3D Unet + VRTK

Nous avons amélioré le décor minimaliste afin d'effectuer une meilleure représentation visuelle correspondant au scénario de l'application collaborative connectée.

7.4.3. *Obstacles rencontrés*

Nous avons rencontré de multiples difficultés à chaque échelon du développement. Celles-ci seront présentées par étapes.

1) Implémentation d'Unet sur Unity :

1. Incapacité à utiliser un LAN pour connecter deux ordinateurs.

Après avoir testé en local le serveur Unet utilisant des avatars simplifiés pour les utilisateurs dans une simple application PC utilisant le clavier et la souris, nous savions pertinemment que l'exécution de l'application devait s'effectuer sur des terminaux différents pour les tests à venir, car un ordinateur ne peut être branché qu'à un seul casque VR. Le LAN permet de combler ce besoin par l'utilisation de câbles RJ45 entre deux PC.

Solutions testées :

- a. Vérification de la connexion entre les ordinateurs sur le réseau.

Nous avons vérifié que les adresses IP des deux ordinateurs étaient correctes via la commande « ifconfig » dans une console sur Windows. Pour vérifier qu'ils pouvaient communiquer effectivement, nous avons envoyé un « ping » à l'adresse IP de l'autre terminal. Tout fonctionnait.

- b. Modification des paramètres du *component* « Network Manager ».

Selon le tutoriel suivi sur Udemy (Byl, 2018), il est possible de forcer l'utilisation d'une adresse IP pour connecter les instances. Nous avons donc inscrit l'adresse IP du terminal servant d'hôte pour le LAN. Le deuxième ordinateur se connectant au premier par sa connexion LAN aurait dû s'y connecter. Ce ne fut toutefois pas le cas.

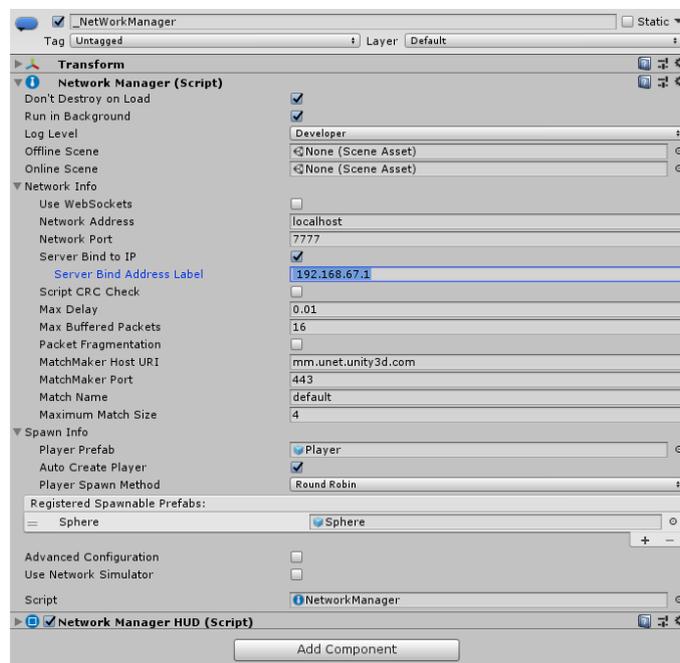


Figure 33 : Tentative d'utilisation d'adresse spécifique réalisée par l'auteur sur Unity

- c. Vérification du pare-feu et de l'antivirus.

Nous avons vérifié si le blocage venait du pare-feu en ajoutant une règle autorisant l'ouverture du port 7777 en entrée/sortie. Suspectant également l'antivirus de créer une interférence, nous l'avons désactivé. Malgré cela, lorsque nous exécutons les deux instances sur leur terminal, les utilisateurs ne sont pas réunis.

- d. Utilisation du *component* « Network Discovery ».

Suivant les indications de la documentation officielle d'Unity, nous avons utilisé « Network Discovery », outil censé scanner les ports du logiciel ouvert et connecter les instances lorsqu'il les découvre (NetworkDiscovery, 2018). Nous

n'avons pas connu davantage de succès avec cette méthode. Nous avons répété la manœuvre C sur ces nouveaux ports, mais le résultat resta toujours le même.

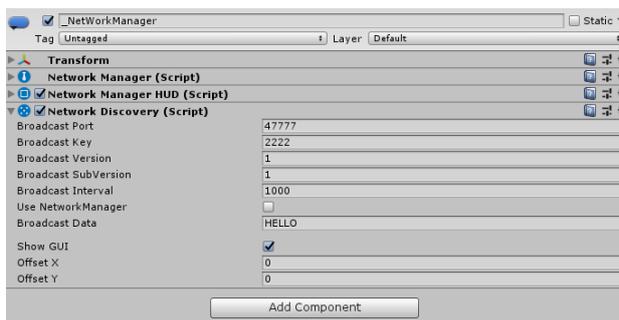


Figure 34 : Utilisation du « Network Discovery » réalisée par l'auteur sur Unity

Résolution du problème menant à une solution :

e. Utilisation du WAN

Après de multiples tentatives durant lesquelles nous avons scrupuleusement suivi les tutoriels et la documentation officielle d'Unity, nous sommes arrivés à la conclusion qu'il ne s'agissait pas du meilleur moyen pour créer une application prototype réseau. Nous sommes donc passés par le cloud d'Unity. Après avoir inscrit notre projet sur le service multijoueur, nous avons réussi à avoir une connexion entre deux instances sur différents terminaux.



Figure 35 : L'ordinateur peut créer une Room sur le réseau



Figure 36 : Le client peut rejoindre la Room créée par le serveur

2) Intégration de la réalité virtuelle dans Unet en utilisant VRTK :

1. Avatar invisible.

Malgré le fait que nous avons ajouté au joueur une capsule faisant office de corps temporaire, les deux utilisateurs ne peuvent pas se voir lorsque l'instance tourne.

Solutions testées :

Cette difficulté n'a pu être résolue puisque nous avons effectué le même procédé que pour les sphères servant précédemment de corps (voir figure 30) ainsi que pour les objets à déplacer. Autrement dit, ajouter aux *Game Objects* servant de corps un « Network Identity » avec l'attribut « autorité locale » signifiant que ce *Game Object*

ne dépend pas du serveur pour se déplacer, mais bien de l'instance locale et d'un « Network Transform ».

2. Problème d'apparitions des utilisateurs.

Lorsque nous utilisons les sphères avec un simple script de déplacement et des *Game Objects* possédant le composants « Network Start Position », les utilisateurs apparaissaient bien sur leur point de réapparition convenu. Lors de l'intégration de la réalité virtuelle, les deux utilisateurs apparaissent au même endroit en dehors des points désignés.

Solutions testées :

Lorsque nous utilisons le simulateur VR fourni par VRTK permettant de tester l'instance avec le clavier et la souris, les utilisateurs réapparaissent toujours en plein centre de la pièce, précisément là où le *prefab* du « playerVR » a été enregistré. Le même phénomène se passe lorsque nous lançons les instances avec les deux casques. Il n'existe que deux moyens de créer un point d'apparition (Spawning and Respawning, 2018) :

a. Points d'apparition dans la scène.

La première solution consiste à créer, de manière séquentielle, un *Game Object* dans une scène en lui attribuant le « Network Start Position » et en s'assurant que la méthode utilisée par le « Network Manager » n'est pas aléatoire, ceci pour éviter qu'ils n'apparaissent l'un sur l'autre. En limitant le nombre d'utilisateurs pouvant se connecter au même nombre de points d'apparition, nous prévenons le problème d'apparition chevauchée. Malgré tout, la réapparition se fait toujours au centre de la pièce.

b. Gestionnaire de points d'apparitions sur le réseau.

L'utilisation d'un *Game Object* permet de créer les points d'apparitions sur le réseau grâce à ce script :

```
public class NetworkSpwanSetAuthority : NetworkBehaviour {  
  
    public GameObject prefab;  
  
    //Method call when the scene is lauching  
    void Start ()  
    {  
        CmdSpawn();  
    }  
}
```

```
}  
//Method giving order to server to create the GameObject in networking by copy  
the prefab  
[Command]  
public void CmdSpawn()  
{  
    GameObject go = (GameObject)Instantiate(prefab, transform.position + new  
Vector3(0, 8, 0), Quaternion.identity);  
    NetworkServer.Spawn(go);  
}
```

« CmdSpawn() » crée un duplicata du *prefab* copié, puis utilise la position du *Game Object* attaché à ce script en ajoutant huit unités de hauteur ainsi que sa rotation. Ce *duplicate* est généré sur le serveur par « Network.Spawn() » ajoutant l'autorité du serveur et son existence sur toutes les instances. Le résultat de ce deuxième moyen fait écho à la précédente solution.

3.2) Implémentation et synchronisation des interactions de l'utilisateur VR :

1. Latence au niveau des objets à synchroniser.

Lorsque le serveur effectue un mouvement avec un objet, le client voit le déplacement de celui-ci de manière saccadée, voire de manière instable puisqu'il effectue des comportements non demandés.

Solutions testées :

a. Changement de réseau Internet.

En effectuant un test réseau pour vérifier la vitesse de celui-ci, nous nous sommes aperçus qu'elle était trop basse. Nous avons donc pris comme réseau le point d'accès d'un smartphone qui lui était supérieur. Nous avons noté une amélioration notable au niveau du mouvement des objets, bien que cela ne fût toujours pas parfaitement fluide.

b. Modification des paramètres du « Network Transform ».

Il comporte les propriétés suivantes :

- Le nombre d'informations envoyées par seconde
- La distance à parcourir pour envoyer une nouvelle information
- Le facteur d'interpolation, déterminant l'apparition de la prochaine position de l'objet par rapport à la position actuelle.

En les dosant correctement, nous sommes parvenus à améliorer la fluidité du mouvement. Malgré tout, les saccades n'ont pas totalement cessé.

2. Connexion unidirectionnelle, synchronisation des interactions.

Lors de l'exécution de l'instance, l'ordinateur servant de serveur arrive à déplacer des objets à la vue du client. Cependant, lorsque ce dernier bouge un objet, le serveur n'observe aucun changement. Les actions du client ne sont pas prises en compte. Le logiciel ne montre aucune erreur malgré la non-synchronisation des données.

Solutions testées :

a. Création des objets directement sur le réseau

En réutilisant le procédé de l'étape précédente pour la gestion des points d'apparition (problème 2, solution b), l'objet généré obtient les droits du serveur. Ainsi, si le serveur reçoit l'information d'interagir avec celui-ci, il pourra le faire. Dans notre cas, le client ne peut toujours pas déplacer des objets dans le côté serveur.

b. Utilisation des fonctions pour serveur sur l'objet à déplacer

Comme dans les étapes précédentes, nous utilisons les fonctions de serveur [Command] ou [ClientRPC].

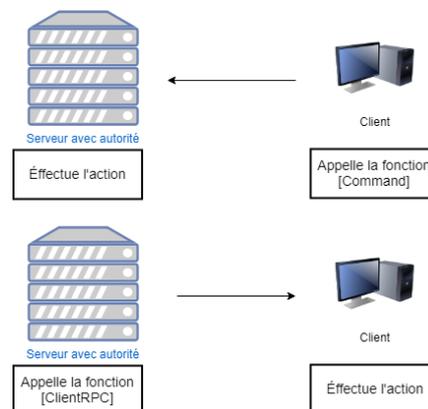


Figure 37 : Illustration des fonctions [Command] et [ClientRPC] réalisée par l'auteur selon (Byl, 2018)

Nous avons dû créer une méthode de synchronisation du « Transform » d'un *Game Object* pour remplacer le *component* « Network Transform ». De plus, nous l'avons ajouté à chaque élément devant se synchroniser, ce qui a donné le script suivant :

```
[SyncVar] Vector3 position;
[SyncVar] Quaternion rotation;

//Method play each frame
private void Update()
{
    CmdSetPos(this.gameObject.transform.position,
    this.gameObject.transform.rotation);
}
//Call the server to update position
```

```
[Command]
public void CmdSetPos(Vector3 position, Quaternion
rotation)
{
transform.position = position;
transform.rotation = rotation;
RPCGetPos(position, rotation);
}

//Call all client to update position
[ClientRpc]
public void RPCGetPos(Vector3 position, Quaternion
rotation)
{
this.position = position;
this.rotation = rotation;
}
```

À chaque changement d'image, le serveur est sollicité par « CmdSetPos() » pour mettre à jour la position et la rotation du *Game Object*, puis appelle tous les clients avec « RPCGetPos » pour que ceux-ci se mettent également à jour. Cette solution a toutefois échoué.

c. Utilisation des fonctions pour serveur appelées depuis le « PlayerVR »

Le « PlayerVR » possédant la notification de « player » d'Unity a toutes les autorités. Nous avons changé le script précédent pour ne laisser que la méthode [Command]. Un script a été ajouté au contrôleur du « playerVR » dans le but de déterminer s'il saisit un objet, auquel cas il appelle cette commande serveur. Une nouvelle fois, ce fut un échec. Le logiciel nous avertit que le client tente d'accéder à une méthode [Command] en ne possédant pas les autorisations.

d. Attribution des droits à l'objet par l'extension de ceux accordés à l'utilisateur (Glomby, 2016)

Lorsque le « playerVR » prend un objet, nous transmettons son « Network Identity » et l'assignons à celui de l'objet saisi afin qu'il puisse bénéficier des mêmes droits pour appeler la commande de synchronisation sur le serveur. Grâce à des *logs*, nous avons pu confirmer que nous étions bien en possession des paramètres demandés. Toutefois, le résultat est le même que la tentative A. L'erreur se focalisait cependant sur la « CMDSetAuthority(...) ».

3.4) Utilisation d'un système de communication vocal :

1. Unet ne possède qu'un chat écrit :

Unity ne comporte pas d'asset officiel pour la communication orale entre joueurs connectés.

Solutions envisagées :

a. Utilisation de texte prédéfini.

En utilisant le chat écrit d'Unet, nous aurions pu prédéfinir une arborescence de phrases textuelles à sélectionner soit par l'usage d'une interface virtuelle apparaissant lors de la pression d'une touche, soit par l'assignation de textes sélectionnés arbitrairement à plusieurs touches du contrôleur. Cette solution n'a pas été retenue par manque de liberté et de spontanéité dans la communication ; elle occasionnerait une perte de temps à transmettre des informations basiques au lieu de se focaliser sur la réalisation de l'objectif principal.

b. Déléguer la fonctionnalité à un logiciel extérieur dédié.

« Discord » est un logiciel de communication gratuit et *online* qui permet de sous-traiter la gestion de la communication vocale.

Solution retenue :

c. Utilisation d'un script manuel (A Free Script For Voice Chat in uNet, 2018).

Cette solution non officielle permet de communiquer de manière orale en utilisant les *channels* dédiés au chat écrit d'Unet. Le test fut concluant.

7.4.4. Conclusion sur le projet Unet + VRTK

Puisque l'étape 3.2 n'a pu être validée, l'étape 4 n'a pas été développée. En effet, cette dernière repose sur le même principe de synchronisation de données. La dernière étape ne comportant ni compte à rebours, ni système de fin, il est impossible de créer des *logs* de valeurs et de les écrire de manière automatique dans des fichiers au terme de l'application.

Tableau 5 : Récapitulatif des fonctionnalités réalisées avec Unet et VRTK (réalisé par l'auteur)

Fonctionnalité	Statut
Réseaux	
Connexion de deux utilisateurs à une scène	Fonctionnelle
Avatars visibles	Non fonctionnelle
Synchronisation des données sur le réseau	Partiellement fonctionnelle
Utilisateurs	
Capacité à prendre des objets	Fonctionnelle
Capacité à construire	Fonctionnelle
Soumission à la gravité et capacité d'escalade	Fonctionnelle
Capacité à se déplacer et à se téléporter	Fonctionnelle
Communication	
Capacité à communiquer entre utilisateurs	Fonctionnelle

Encadrement de l'application	
Synchronisation d'un compte à rebours entre les instances	Non implémentée
Système de fin d'application	Non implémentée
<i>Logs</i>	
Logs des actions	Non implémentée
Logs de déplacements	Non implémentée
Logs des échanges vocaux	Non implémentée

En se référant à la documentation officielle et à l'utilisation de nombreux tutoriels, les utilisateurs ne peuvent pas se voir et l'hébergeur de la partie est le seul à pouvoir déplacer des objets et les empiler à la vue du client. Toutefois, les interactions des usagers sont toutes effectuées et fonctionnelles. Seule la partie du réseau ne fonctionne pas de manière correcte.

7.5. Déploiement de Unet en utilisant SteamVR

Afin de mieux comprendre nos erreurs, nous avons acheté un projet Unity pour comparer l'implémentation du réseau et pouvoir l'utiliser comme base pour l'application à développer. Nous allons donc comparer les parties non fonctionnelles du projet précédent. La nature de ce projet est un paintball multijoueur.

7.5.1. Comparaison des solutions aux problèmes non résolus :

Dans cette partie, nous allons comparer les problèmes rencontrés dans le premier projet à ceux de ce deuxième projet lors des étapes du processus établi.

1) Intégration de la réalité virtuelle dans Unet en utilisant VRTK :

1. Résolution du problème de l'avatar invisible.

Ce projet intègre un « Local Player » et un « Player Avatar » qui se créent sur le réseau. Le premier lui permet de capturer les mouvements de l'utilisateur qu'il lie au second généré en ligne. Cela signifie que, sur le réseau, ils génèrent leur propre avatar et ID respectif, les rendant visibles pour l'autre utilisateur. Cela est semblable à ce que nous avons implémenté en utilisant directement les « Local Player » pour être visibles sur le réseau.

De plus, en n'utilisant pas le « Network Transform » d'Unet, mais son propre système de transfert qui capture les mêmes données que dans notre script de « Synchronisation » concernant la position et la rotation de la tête ainsi que les contrôleurs, un script les

convertit en octets, puis les envoie via un « `ByteWriter` » et les reçoit via un « `ByteReader` ». En ce qui concerne la synchronisation entre serveur et client rencontré dans ce projet, elle procède aux mêmes étapes que le projet précédent :

```
[Command]
private void CmdSendNetworkPlayerTransformDataToServer(byte[] data)
{
    mostRecentPlayerTransformData =
NetworkPlayerTransformWriter.FromBytes(data);
    RpcSendNetworkPlayerTransformDataToClients(data);
}

[ClientRpc]
private void RpcSendNetworkPlayerTransformDataToClients(byte[] data)
{
    if (isLocalPlayer)
        return;

    mostRecentPlayerTransformData =
networkPlayerTransformWriter.FromBytes(data);

networkPlayer.SetTransformsToNetworkValues(mostRecentPlayerTransformData);
}

[Client]
private void Update()
{
    if (ShouldSendPlayerTransformDataToServer())
    {
        networkSendTimer = 1f / (float)networkUpdatesPerSecond;

        var playerTransformDataToSendServer =
networkPlayer.GetLocalPlayerTransformData();

        var data =
networkPlayerTransformWriter.GetBytes(playerTransformDataToSendServer);

        CmdSendNetworkPlayerTransformDataToServer(data);
    }
}
```

La fonction *update* est dédiée pour les [Client] alors que nous avons utilisé [ClientRPC]. Cette commande client active la méthode *update* seulement s'il s'agit d'un client et non d'un serveur. Ainsi, le deuxième utilisateur devient visible par le serveur. Pour compenser ce fonctionnement que ne possède pas le [ClientRPC], nous avons testé si le joueur connecté était le serveur ; s'il ne l'est pas, il utilise les méthodes dédiées au serveur.

2. Résolution du problème d'apparition des utilisateurs :

Nous effectuons le même paramétrage en utilisant la solution B : des gestionnaires de points d'apparition sur le réseau. Cela confirme que l'erreur provenait de l'utilisation de VRTK. La génération d'un « `Player VR` » par le biais d'un script lui donne une position

identique au *Game Object* posé dans l'éditeur d'Unity. Lorsque Unet crée l'avatar d'un joueur, il l'instancie au bon endroit, Le problème est que le script de VRTK rentre en conflit en n'utilisant pas les nouvelles positions attribuées par Unet, mais celles enregistrées dans son *prefab*.

2) Implémentation et synchronisation des interactions VR avec SteamVR :

1. Résolution de la latence au niveau des objets à synchroniser.

Le développeur de ce projet utilise une synchronisation manuelle des informations. C'est ce procédé qui a été utilisé pour la gestion du corps de l'« Avatar Player ». De cette façon, le transfert est plus rapide que celui proposé par le « Network Transform » d'Unet.

7.5.2. Obstacles rencontrés

2) Implémentation et synchronisation des interactions VR avec SteamVR :

1. Connexion unidirectionnelle.

- a. La seule interaction de ce projet est le tir de projectiles. Il reprend la solution A, « Création des objets directement en réseau ». En répliquant la méthode fonctionnelle de synchronisation du corps de l'« Avatar Player » sur les objets à synchroniser, nous avons obtenu la même erreur. Le client tente d'accéder à une fonction du serveur pour laquelle il ne possède pas les droits.

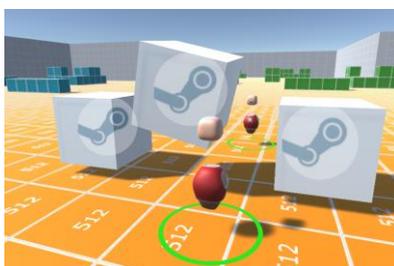


Figure 38 Utilisateur client voyant le joueur hôte manipuler un cube, vu de l'éditeur (réalisé par l'auteur)

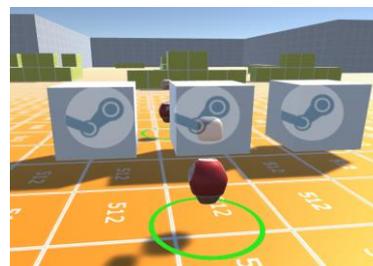


Figure 39 : Utilisateur serveur ne voyant aucun changement du client, vu de l'éditeur (réalisé par l'auteur)

2. Plusieurs interactions insatisfaisantes.

- La téléportation ne peut plus s'effectuer sur des objets du décor.
- L'utilisateur n'est pas soumis à la gravité et il ne peut pas escalader.

Solutions testées :

- a. Adaptation des méthodes utilisées dans VRTK pour reproduire ces comportements. Ce fut un échec, les codes de VRTK et de STEAMVR étant entrés en conflit.

7.5.3. Conclusion sur le projet à base d'Unet + SteamVR

Tout comme pour le précédent projet, les étapes 4 et 5 n'ont pas pu être implémentées, et ce pour les mêmes raisons. En reprenant cette application, nous avons pu mieux cerner le problème de l'avatar invisible. Cependant, les objets ne peuvent toujours pas être synchronisés entre le client et le serveur. De plus, l'utilisateur ne possède plus les facultés suivantes :

- Soumission à la gravité
- Capacité à escalader
- Capacité à se téléporter sur les objets
- Capacité à se téléporter librement

Le tableau ci-dessous récapitule les caractéristiques techniques accomplies dans le deuxième projet.

Tableau 6 : Récapitulatif des fonctionnalités réalisées avec Unet et SteamVR (réalisé par l'auteur)

Fonctionnalité	Statut
Réseaux	
Connexion de deux utilisateurs à une scène	Fonctionnelle
Avatars visibles	Fonctionnelle
Synchronisation des données sur le réseau	Partiellement fonctionnelle
Utilisateurs	
Capacité à prendre des objets	Fonctionnelle
Capacité à construire	Fonctionnelle
Soumission à la gravité et capacité d'escalade	Non fonctionnelle
Capacité à se déplacer et à se téléporter	Partiellement fonctionnelle
Communication	
Capacité à communiquer entre utilisateurs	Fonctionnelle
Encadrement de l'application	
Synchronisation d'un compte à rebours entre les instances	Non implémentée
Système de fin d'application	Non implémentée
Logs	
Logs des actions.	Non implémentée
Logs de déplacements.	Non implémentée
Logs des échanges vocaux.	Non implémentée

Nous constatons qu'une nouvelle fois, ce projet ne couvre pas toutes les fonctionnalités du réseau.

7.6. Conclusion sur Unet

En conclusion, Unet ne concrétise pas de manière stable une application collaborative connectée. Sur deux projets construits avec deux approches différentes, mais ayant des procédés semblables sur le fond, il ne remplit pas la synchronisation des interactions, une fonctionnalité pourtant importante dans un réseau. Nous élaborons alors un troisième projet basé sur le deuxième moyen de concevoir un réseau sur Unity : Photon.

7.7. Déploiement de Photon + VRTK sur Unity

Photon étant un moteur de réseau, nous avons créé un compte auprès de cette plateforme afin de bénéficier des offres « Photon Unity Networking » et « Photon Voice ». Nous avons ensuite installé les *assets* de Photon et ceux de VRTK. Photon possède deux manières de communiquer des données : de manière constante, avec la méthode « OnPhotonSerializeView », ou lors de l'appel d'une fonction serveur nommée [PunRPC].

7.7.1. Photon

Le « PhotonView » attribue un ID unique sur le réseau. Il est indispensable pour instancier un *Game Object* et permet d'observer les objets, les *components* comme « Transform » ou « Rigidbody » et les scripts à synchroniser.

7.7.2. Déroulement de conception

À l'inverse des deux projets précédents, nous suivons tout le déroulement selon les étapes du processus.

7.7.3. Implémentation de Photon sur Unity :

Après avoir créé un compte, nous devons paramétrer le *cloud* Photon avec l'ID de l'application lié à notre compte.

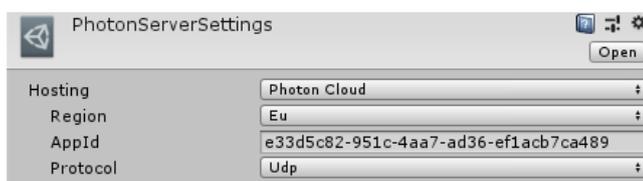


Figure 40 : Paramétrage de l'AppId de Photon pour le *cloud*

Pour ce faire, nous avons utilisé la même scène primitive qu'utilisée pour le premier projet. La gestion des connexions dans une scène est manuelle et nous devons donc gérer tous les cas :

- La connexion du *master client* qui possède des droits supplémentaires par rapport à un client standard et le changement de client pour le *master client* si celui-ci se déconnecte.
- La création de la « Room », la rejoindre et la quitter.
- La connexion d'un joueur et sa déconnexion.

Le « Network Manager » d'Unet gère automatiquement cette partie via l'interface présentée dans la figure 30 et nécessite le même paramétrage manuel dans le cas d'une connexion automatique. Nous avons donc créé une « Room » pour la scène, puis ajouté sur les sphères le PhotonView.

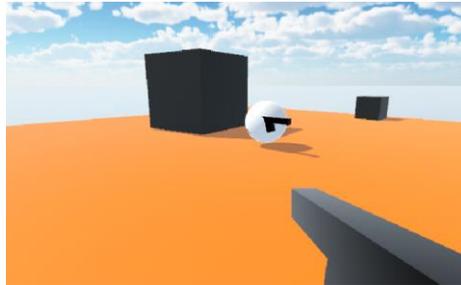


Figure 41 : Succès de la première étape avec Photon

Comme il est possible de le constater en haut à gauche de la figure 41, il n'y a plus de texte spécifiant le « port 7777 », comme c'était le cas dans la figure 31.

1) Intégration de la réalité virtuelle dans Photon :

Pour cette partie, nous avons tenu compte des problèmes rencontrés lors des deux autres projets. Ainsi, nous avons un *Game object* « Player Local » nommé « Player VR » et le « Player Avatar » qui est généré sur le réseau. Toutefois le *prefab* de ce « player » à générer doit se trouver dans un dossier spécifique dans Photon. Il doit se trouver dans le dossier « ressources » contenu dans le dossier « photon », alors pourtant que pour Unet son lieu de stockage importait peu. Nous avons rencontré le problème de liaison entre les deux « Player ». La solution offerte dans le deuxième projet était trop spécifique et n'a pas pu être adaptée. Enfin, nous retrouvons un problème déjà présent dans le premier projet : le corps de l'avatar ne se synchronise pas.

3) Implémentation du corps de l'application.

1. Création d'un environnement 3D :

Afin de créer un décor pour un environnement test proche d'une version finale, nous avons utilisé l'extension « ProBuilder » qui génère des outils de conception 3D tels que proposés par « Maya » ou « Blender ». Il n'est pas aussi puissant que ces derniers, mais permet de réaliser rapidement des maquettes.

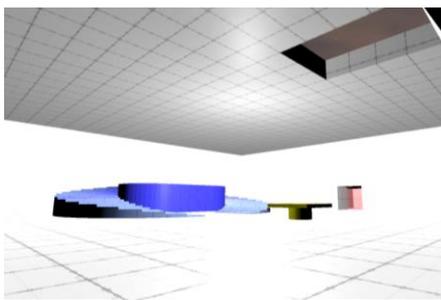


Figure 42 : Décors 3D pour Photon réalisés avec ProBuilder

2. Implémentation et synchronisation des interactions de l'utilisateur VR :

En utilisant les fonctionnalités de VRTK, cela fut assez rapide à créer pour un seul utilisateur. Toutes les actions telles que se déplacer dans la « Room Scale », se téléporter ou escalader, sont synchronisées par le « PhotonView » du « Player Local » synchronisant le corps de l'avatar par le script « NetworkObject » sur son avatar *online*. Ce même procédé a été employé afin de synchroniser les interactions de prise des objets de construction d'artefacts.

3. Utilisation d'un système de communication vocal :

Contrairement à Unet, Photon propose son propre système de communication vocal, « Photon Voice ». Tout comme pour PUN, nous devons obtenir un ID pour l'application à utiliser dans le paramètre du *cloud* pour la voix.

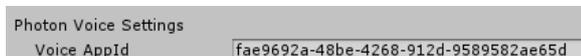


Figure 43 : Paramétrage du AppId de Photon pour la voix

Grâce à cela nous disposons d'un *component* « PUN Voice » trackant l'utilisation du micro d'Unity. Malgré le fait que nous ayons ce traqueur, nous éprouvons des difficultés à transmettre l'information via le réseau. Ceci sera développé à partir du point 7.7.4.

4) Implémentation de l'encadrement de l'application :

Afin de déterminer le début et la fin de l'application et de définir le résultat, nous voulons démarrer un compte à rebours synchronisé. Si le duo accède à la zone de sortie dans le temps imparti, il réussit l'épreuve. S'il est toujours dans la salle à la fin du temps réglementaire, il perd. Nous avons ici aussi rencontré des obstacles qui seront décrits plus loin.

5) Création des logs

Nous assignons un chemin spécifique dans Unity pour atteindre un dossier préfini pour la récupération des *logs*. À la fin de l'application, le script génère automatiquement des fichiers *logs* de l'utilisateur local.

- *Log* de la position du parcours réalisé par le joueur : nous voulions éviter de comparer une multitude de lignes composées des propriétés de son « Transform » dans le format « X : 23.43, Y : 1.80, Z : 50.01 ». Une nouvelle écriture avec ce format pour chaque *frame* lorsque la personne bouge de quelques secondes devient illisible. Par conséquent, nous avons séparé la pièce en plusieurs lignes de cube utilisant des « Triggers » et en supprimant les « Mesh Render ».

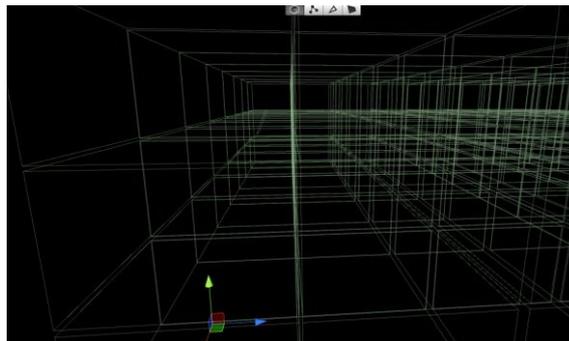


Figure 44 : Système de log de position

Lorsque l'un des cubes invisibles détecte que le joueur est à l'intérieur, il inscrit l'heure du passage et son nom dans une liste qui s'enregistre à la fin du jeu dans un fichier CSV. Pour l'instant, le nom de chaque cube est composé de trois chiffres déterminant l'étage, représenté par la flèche verte de la figure ci-dessus, la rangée par la flèche rouge et la ligne par la flèche bleue.

VR2C Data from	7/21/2018 1:21:44 PM	
0:0.130	positin player at	123
0:10.290	positin player at	133
0:11.432	positin player at	123
0:14.81	positin player at	123
0:17.785	positin player at	123
0:19.237	positin player at	132
0:20.15	positin player at	131

Figure 45 : Exemple de logs de positions.

- *Log* des échanges vocaux : actuellement, nous utilisons l'enregistreur vocal de l'ordinateur allié à un bruit sonore lors de l'activation de l'application. Cela permet de synchroniser le début de l'application avec la bande sonore.
- *Log* des actions : nous rencontrons des soucis pour déterminer quel objet prend l'utilisateur. Cela sera détaillé dans le point suivant.

7.8. Obstacles rencontrés

Nous aborderons ici les problèmes selon leur étape respective du processus.

1) Intégration de la réalité virtuelle dans Photon

Aucune liaison ne s'effectue entre le « Player Local » et le « Player Avatar ».

Solutions résolvant le problème :

1. Liaison entre les deux « Player ».

Nous attribuons un script de liaison entre ces deux « Player » en utilisant une méthode spécifique au VRTK (VRTK_SDKSetup, 2018) :

```
void Awake () {
    if (!photonView.isMine) {
        return;
    }
    // Move the SDK player local to player online
    VRTK_SDKManager sdk = VRTK_SDKManager.instance;
    sdk.loadedSetup.actualBoundaries.transform.position = transform.position;
    sdk.loadedSetup.actualBoundaries.transform.rotation = transform.rotation;
}
```

Le script vérifie si l'ID de « PhotonView » associé au « Player avatar » appartient bien à l'instance exécutée sur la machine et associe le « SDK Manager » du « Player Local » au « Player Avatar ». Pour le moment, nous avons juste lié les deux « Player », mais nous ne les avons pas encore synchronisés.

2. Aucune synchronisation du corps de l'avatar :

Pour ce faire, nous avons appliqué un script de synchronisation d'objet basé sur la méthode « OnPhotonSerializeView » (quintesse, NetworkObject, 2017).

3.3) Utilisation d'un système de communication vocal :

Incapacité à transmettre du contenu audio :

Solutions résolvant le problème :

1. Utilisation d'un script « AudioManager ».

Nous avons compris qu'il fallait capturer le son venant du composants « Pun Voice » et l'envoyer par un « PhotonView » en utilisant cette fois un appel au serveur par [PunRPC]. Toutefois, pour le contenu à envoyer, nous utilisons un script générant une liste de son découpant l'enregistrement, puis le serveur prépare les parties pour la transmission (quintesse, AudioManager, 2017).

4) Implémentation de l'encadrement de l'application :

1. Synchronisation d'un compte à rebours entre plusieurs instances

Pour le compte à rebours, nous avons un panneau sur lequel est écrit un texte qui est mis à jour à chaque seconde pour afficher les minutes sous le format « minute : seconde »

a) Synchronisation du compte à rebours par la méthode « OnPhotonSerializeView ».

Nous voulons générer un seul compte à rebours chez le serveur pour le transmettre au client.

```
void OnPhotonSerializeView(PhotonStream stream, PhotonMessageInfo info)
{
    if (stream.isWriting)
    {
        stream.SendNext(time);
    }
    else
    {
        time = (int)stream.ReceiveNext();
    }
}
```

Cette méthode vérifie si le *stream* est en cours d'écriture. Dans ce cas-là, il envoie l'information du compte à rebours en secondes, sinon il la reçoit et il remplace la valeur du « time » par la nouvelle. Cette solution a échoué.

b) Synchronisation du compte à rebours par la méthode [PunRPC].

Même principe que la solution précédente, mais en utilisant l'autre méthode de transfert.

```
void Update()
{
    photonView.RPC("updateTimer", PhotonTargets.All);
}

[PunRPC]
void updateTimer()
{
    timeLeft -= Time.deltaTime;
    text.text = "Time Left:" + Mathf.Round(timeLeft);
    if (timeLeft < 0)
    {
        Debug.Log("Timer Finished");
    }
}
```

Ici, nous déterminons un nombre en secondes « timeLeft » duquel seront décomptées les secondes en temps réel par « Time.deltaTime ». Le texte du panneau est également affiché afin qu'il soit synchronisé automatiquement. La solution a échoué car aucune information ne se transmettait.

Solution résolvant le problème :

- c) Déterminer lorsque les deux utilisateurs sont connectés.

En simplifiant le problème, nous avons trouvé que l'application se lance dès que les deux utilisateurs se connectent dans la scène. C'est pourquoi nous avons cherché à détecter le nombre de « Player » dans la partie. Si deux « Player » sont connectés, alors un booléen instancié initialement à faux devient vrai et déclenche individuellement sur toutes les instances le compte à rebours.

2. Aucune synchronisation sur le nombre de « Player » présent dans la zone de fin.

- a. Détecter le nombre d'instances de « Player ».

Par le biais d'un compteur, nous augmentons un nombre pour chaque « Player » dans la zone et le diminuons à chaque sortie d'un « Player ». Pour se faire, nous utilisons la même solution que celle réalisée en a et en b pour la synchronisation d'un compte à rebours. Le résultat obtenu est également le même.

- 5) Création des *logs* :

Solution testée :

Pour la gestion des *logs* d'actions, nous utilisons un script attribué à chaque objet pouvant être déplacé. Lorsque celui-ci est saisi par un utilisateur, son nom et sa position sont capturés. Le problème est le suivant : l'utilisateur prend un objet, mais celui-ci ne se considère pas comme saisi. Le cas est toujours non résolu.

7.9. Conclusion sur Photon

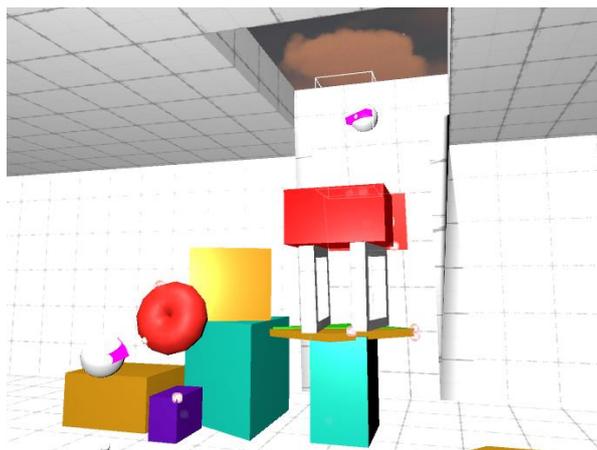


Figure 46 : État actuel de l'application

Ce troisième et dernier projet couvre presque la totalité des fonctionnalités. Comme visible sur la figure ci-dessus, nous avons deux avatars de tête avec des lunettes roses ayant construit une sorte

d'escalier afin d'atteindre la sortie située en hauteur. Un utilisateur donne un dernier objet cylindrique à l'autre pour qu'il puisse se hisser en hauteur par le biais de la téléportation. Photon a réussi là où Unet a échoué. Nous pouvons affirmer que Photon est stable.

Tableau 7 : Récapitulatif des fonctionnalités réalisées avec Photon

Fonctionnalité	Statut
Réseaux	
Connexion de deux utilisateurs à une scène	Fonctionnelle
Avatars visibles	Fonctionnelle
Synchronisation des données sur le réseau	Fonctionnelle
Utilisateurs	
Capacité à prendre des objets	Fonctionnelle
Capacités à construire	Fonctionnelle
Soumission à la gravité et capacité d'escalade	Fonctionnelle
Capacité à se déplacer et à se téléporter	Fonctionnelle
Communication	
Capacité à communiquer entre utilisateurs	Fonctionnelle
Encadrement de l'application	
Synchronisation d'un compte à rebours entre les instances	Fonctionnelle
Système de fin d'application	Non fonctionnelle
Logs	
Logs des actions.	Non fonctionnelle
Logs de déplacements.	Fonctionnelle
Logs des échanges vocaux.	Fonctionnelle

Bien qu'un système de fin et une gestion des *logs* ne soient toujours pas fonctionnels, Photon a tout de même permis de créer une application collaborative connectée.

8. Vérification de l'hypothèse

Nous affirmons au début du document que l'utilisation d'*assets* open source permettait la création d'applications collaboratives connectées en réalité virtuelle. Même si deux projets sur trois nous ont permis de douter de la qualité des applications collaboratives connectées, le dernier projet prouve qu'il est possible de créer une application en réalité virtuelle collaborative connectée. Notre hypothèse est donc validée.

9. Recommandations pour la suite

Dans le but d'effectuer des analyses de collaboration en réalité virtuelle, il serait convenable de résoudre les problèmes non résolus dans ce travail, à savoir déterminer une zone de fin détectant tous les joueurs ainsi qu'enregistrer des *logs* d'actions. De plus, l'application possédant son propre compte à rebours, il serait judicieux d'y intégrer un système d'enregistrement vocal afin d'avoir une synchronisation parfaite dans l'ensemble des *logs* et de ne plus dépendre d'un logiciel externe.

L'environnement de test est composé de deux postes ayant chacun un ordinateur « VR Ready », un HTC VIVE, une connexion Internet et l'exécutable de cette application. Les tests devront s'effectuer dans des salles distinctes et sous la surveillance d'un accompagnateur. L'application permet une séparation physique entre les utilisateurs. Pour éviter les désagréments découverts lors du développement dus à l'utilisation du casque HTC VIVE et de son « Room Scale », les points suivants sont à noter :

- Une transmission audio et visuelle filaire reliant le casque à l'ordinateur peut perturber l'expérience en s'enroulant autour des pieds de l'utilisateur.
- Lorsque deux surfaces de « Room Scale » et leurs composants sont trop proches l'une de l'autre, cela crée des conflits. Les contrôleurs changent de caméra et perdent la connexion avec leur *setup* initial. L'utilisateur subit alors des effets indésirables tels que des sauts de caméra et des comportements inadéquats du contrôleur s'éloignant de sa position. Cela représente un parfait mélange pour engendrer la cinétose.

Il est recommandé de s'adresser à des profils de personnes variés issus de tous horizons afin d'obtenir des données de qualité.

10. Conclusion

À travers ce document, nous avons non seulement déterminé que la réalité virtuelle est présente dans de nombreux domaines, mais également que la collaboration virtuelle existe. Grâce à la partie d'analyse, nous avons trouvé le meilleur environnement de travail et déterminé les *assets* les plus adaptés dans le but de vérifier notre hypothèse. La partie de développement a validé cette dernière, à savoir que l'utilisation d'*assets* open source permet la création d'applications collaboratives connectées en réalité virtuelle. L'application développée permet d'intégrer deux casques HTC VIVE dans une scène commune hébergée dans un *cloud*. Les possesseurs de l'exécutable de l'application peuvent s'y retrouver pour interagir, construire et communiquer.

Notons toutefois que le cahier des charges n'a pas été entièrement validé. En effet, il manque un système de *logs* pour les actions réalisées par une personne ainsi qu'une zone de fin détectant tous les joueurs.

Ce travail a été une formidable opportunité pour moi de découvrir les bases de la conception d'une application en réalité virtuelle multijoueur. Ma maîtrise du réseau étant l'un de mes points faibles,

mes compétences ont été mises à rude épreuve lors de la partie consacrée au développement. De surcroît, la synchronisation d'objets 3D s'est avérée également d'une grande complexité. J'ai toutefois su surmonter ces obstacles pour arriver à un résultat que j'estime honorable.

11. Déclaration sur l'honneur

Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- Anne-Dominique Salamin
- Professeur HES
- HES-SO // Valais

Lieu et Date

Signature

(Jeff Zufferey, Étudiant HES-SO)

Références

- Angers, M. (1996). *Initiation pratique à la méthodologie des sciences humaines, deuxième édition*. Anjou: Les éditions CEC inc.
- Archdaily. (2018, Juillet 14). Récupéré sur archdaily: <https://www.archdaily.com/tag/virtual-reality-for-architects>
- Beat Games. (2018, Juillet 14). Récupéré sur beatsaber: <http://www.beatsaber.com/>
- Bernhardt, P. (2015, Avril 23). */experience-vr-using-the-myo-armband/*. Récupéré sur developerblog.: <https://developerblog.myo.com>
- Byl, P. d. (2018). *Unity Networking From Scratch for (Unity 5 to Unity 2018+)*. Récupéré sur Udemey: https://www.udemy.com/unet_intro/learn/v4/t/lecture/6158108?start=0
- Cain, B. L. (2015, Septembre 8). Récupéré sur lefigaro: <http://www.lefigaro.fr/actualite-france/2015/08/29/01016-20150829ARTFIG00013-l-escape-game-un-phenomene-mondial-qui-seduit-un-public-varie.php>
- CaptoGlove. (2018, Juillet 11). *CaptoGlove*. Récupéré sur CaptoGlove: <https://www.captoglove.com/>
- Crytek. (2018). Récupéré sur CryEngine: <https://www.cryengine.com/>
- Crytek. (2018, - -). *Features*. Récupéré sur Cry Engine: <https://www.cryengine.com/features>
- DarkRift Networking. (2018). *DarkRift 2*. Récupéré sur DarkRift Networking: <https://darkriftnetworking.com/DarkRift2>
- Digi-capital. (2018, Janvier). *Record over \$3B AR/VR investment in 2017 (\$1.5B+ in Q4)*. Récupéré sur Digi-capital: <https://www.digi-capital.com/news/2018/01/record-over-3b-ar-vr-investment-in-2017-1-5b-in-q4/#.W02uPNL7RPY>
- Epic Games. (2018, - -). *Gameplay*. Récupéré sur Unreal Engine: <https://docs.unrealengine.com/en-us/Engine>
- Forge. (2018). *Welcome to Forge Networking Developers!* Récupéré sur Forge Developpers: <https://developers.forgepowered.com/>
- Glomby. (2016, 11 16). *UNET: How do I properly handle Client Authority with interactable Objects?* Récupéré sur Forum Unity: <https://answers.unity.com/questions/1245341/unet-how-do-i-properly-handle-client-authority-wit.html>

- Hackathon. (2018, Mars 25). #11 *Simuler des interventions médicales en milieux extrêmes // Simulate medical interventions in extreme environments*. Récupéré sur Innobaord: <http://www.innobaord.ch/arkathon/hhvalais2018/challenges/25/solutions/35>
- HardLight VR. (2018, Juillet 14). *home*. Récupéré sur HardLight VR: <http://www.hardlightvr.com/>
- HTC Corporation. (2018). *Quelle est la configuration système ?* Récupéré sur Vive: https://www.vive.com/fr/support/vive/category_howto/what-are-the-system-requirements.html
- Jaguar. (2016). *virtual reality and how we use it*. Récupéré sur www.jaguar.co.uk: <https://www.jaguar.co.uk/about-jaguar/jaguar-stories/virtual-reality-and-how-we-use-it.html>
- James, P. (2017, May 6). *'SuperChem VR' Shows Impressive, Engaging Gamification of Chemistry Lab Education*. Récupéré sur Road To VR: <https://www.roadtovr.com/watch-superchem-vr-cool-virtual-lab-fun-immersive-learning/>
- Jessica L. Maples-Keller, P. B.-J. (2017). *The Use of Virtual Reality Technology*. Hard.
- Lancelin-Golbery, M. (2016, 02 29). *MWC 2016 : HTC Vive, prix et date de sortie officiellement dévoilés*. Récupéré sur phonandroid: <http://www.phonandroid.com/mwc-2016-htc-vive-prix-et-date-de-sortie-officiellement-devoiles.html>
- Live like vr. (2017). Récupéré sur Live like VR: <http://www.livelikeyr.com/>
- Martindale, J. (2018, 04 04). *Oculus Rift vs. HTC Vive*. Récupéré sur digitaltrends: <https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/>
- Mel Science. (2018). *Atoms in solids*. Récupéré sur Mel Science: <https://melscience.com/vr/lessons/atoms-in-solids/>
- Merli, N. (2016, janvier 6). *Oculus Rift : date de sortie, prix et caractéristiques du casque de réalité virtuelle*. Récupéré sur gaming gentside: https://gaming.gentside.com/oculus-rift/oculus-rift-date-de-sortie-prix-et-caracteristiques-du-casque-de-realite-virtuelle_art5636.html
- Merli, N. (2016, Novembre 15). *PlayStation VR (PS4) : date de sortie, prix et caractéristiques du casque de réalité virtuelle de Sony*. Récupéré sur <http://gaming.gentside.com>: http://gaming.gentside.com/playstation-vr/playstation-vr-ps4-date-de-sortie-prix-et-caracteristiques-du-casque-de-realite-virtuelle-de-sony_art7982.html
- Nafarrete, J. (2016, Novembre 14). *Jaguar Ipace car launch Htc Vive*. Récupéré sur VRScout: <https://vrscout.com/news/jaguar-ipace-car-launch-htc-vive/>

- Noitom. (2018, juillet 11). *Noitom VR Glove*. Récupéré sur Noitom Vr Glove: <https://hi5vrglove.com/>
- NVidia. (2016, - -). */object/vr-ready-program-fr.html*. Récupéré sur Nvidia: <http://www.nvidia.fr/object/vr-ready-program-fr.html>
- Oculus. (2018, Juillet 11). *setup*. Récupéré sur Oculus Rift: <https://www.oculus.com>
- Owlchemy Labs. (2018). *Home*. Récupéré sur jobsimulatorgame: <https://jobsimulatorgame.com/>
- Pekka Abrahamsson, O. S. (2017). *Agile Software Development Methods: Agile software development methods: Review and analysis*. . VTT Technical Research Centre of Finland.
- Philip Kotler, H. K. (2017). *Marketing 4.0. Moving from Traditional to Digital by Philip Kotler, Hermawan Kartajaya and Iwan Setiawan*. Hoboken: John Wiley & Sons.
- Philippe Fuchs, e. a. (2006). *Le traité de la réalité virtuelle Volume 1 : L'homme et l'environnement virtuel* (Vol. 1). France: Transvalor - Presses des mines.
- Photon Engine. (2018). *Home*. Récupéré sur Photon Engine: <https://www.photonengine.com/>
- Picard, D. A. (2007). Récupéré sur <http://ludicine.ca/>: <http://goo.gl/FKDhO5>
- Piquet, A. (2009). *Guide pratique du travail collaboratif: Théories, méthodes, et outils au service de la colalboration*. Brest.
- Poškevičius, A. (2017, Mars 17). *Is It For You?* Récupéré sur Github: <https://github.com/alvyxaz/barebones-masterserver/wiki/Is-It-For-You%3F>
- Prommer-Games. (2018, Janvier 27). *A Free Script For Voice Chat in uNet*. Récupéré sur Unity: <https://forum.unity.com/threads/a-free-script-for-voice-chat-in-unet.514497/>
- quintesse. (2017, Mars 22). *AudioManager*. Récupéré sur Github: <https://github.com/quintesse/PlayoVR/blob/master/Assets/Scripts/AudioManager.cs>
- quintesse. (2017, Août 12). *NetworkObject*. Récupéré sur Github: <https://github.com/quintesse/PlayoVR/blob/master/Assets/Libraries/NetBase/NetworkObject.cs>
- SimforHealth. (2018). *Nos solutions*. Récupéré sur SimForHealth: <https://simforhealth.fr/nos-solutions/>
- skyfunvr. (2018, Juillet 14). */vr-positioning-and-head-tracking-in-htc-vive-oculus-rift-and-playstation/*. Récupéré sur skyfunvr: <https://www.skyfunvr.com>

Slant. (2018, Juillet). *Unity vs cryengine*. Récupéré sur Slant:
https://www.slant.co/versus/1047/5125/-unity_vs_cryengine

Slant. (2018, Juillet -). *Unity vs unreal engine 4*. Récupéré sur slant:
https://www.slant.co/versus/1047/5128/-unity_vs_unreal-engine-4

sony entertainment. (2018, Juillet 11). */fr-ch/explore/playstation-vr/*. Récupéré sur playstation:
<https://www.playstation.com>

Steel Crate Games. (2018, Juillet 14). *KeepTalkinggame*. Récupéré sur KeepTalkinggame:
<http://www.keeptalkinggame.com/>

Sweeney, T. (2016, Avril 2). *build fo rvr invr*. Récupéré sur www.unrealengine.com:
<https://www.unrealengine.com/en-US/blog/build-for-vr-in-vr>

Taclim. (2018, Juillet 14). *Home*. Récupéré sur Taclim: <https://taclim.cerevo.com/en/>

Teslasuit. (2018, Juillet 14). <https://teslasuit.io/>. Récupéré sur Teslasuit: <https://teslasuit.io/>

thestonefox. (2018, - -). *VRTK_SDKSetup*. Récupéré sur VRTK:
https://vrtoolkit.readme.io/docs/vrtk_sdksetup

Trap Game. (2018, Juillet 14). *Vice-Versa*. Récupéré sur trapgame:
<http://www.trapgame.ch/fr/Rooms/11-Vice-Versa/>

UbiSim. (2018). Récupéré sur Ubisim: <http://www.ubisim.ch/?lang=fr>

Unity. (2018). *Multiplayer*. Récupéré sur Unity 3D:
<https://unity3d.com/fr/unity/features/multiplayer>

Unity. (2018). *Network Authority*. Récupéré sur Unity Documentation:
<https://docs.unity3d.com/Manual/UNetAuthority.html>

Unity. (2018). *NetworkDiscovery*. Récupéré sur Unity Documentation:
<https://docs.unity3d.com/ScriptReference/Networking.NetworkDiscovery.html>

Unity. (2018). *Public Relations*. Récupéré sur Unity: <https://unity3d.com/fr/public-relations>

Unity. (2018). *Spawning and Respawning*. Récupéré sur Unity:
<https://unity3d.com/fr/learn/tutorials/topics/multiplayer-networking/spawning-and-respawning>

Unity. (2018, - -). *Unity*. Récupéré sur Unity: <https://unity3d.com/fr/unity>

Virtual Reality society. (2017). *Virtual reality air force training*. Récupéré sur Virtual Reality society: <https://www.vrs.org.uk/virtual-reality-military/air-force-training.html>

Virtual reality society. (2017). *Virtual reality combat simulation*. Récupéré sur Virtual reality society: <https://www.vrs.org.uk/virtual-reality-military/combat-simulation.html>

Virtual Room. (2018, Juillet 14). *Experience*. Récupéré sur VirtualRoom: <https://www.geneva.virtual-room.com/#experience>

Vive. (2018, Juillet 11). *Product*. Récupéré sur Vive: <https://www.vive.com/fr/product/>

Volkswagen. (2018, Juillet 14). *Virtual technologie*. Récupéré sur www.volkswagen.com: <https://www.volkswagen.com/en/group/research/virtual-technologies.html>

VRGLUV. (2018, juillet 11). *VR Gluv*. Récupéré sur VR Gluv: <https://vrgluv.com/>

VRTK. (2018). *Welcome to VRTK*. Récupéré sur VRTK: <https://vrtoolkit.readme.io/>

12. Références des tableaux

Tableau 1 réalisé par l'auteur selon le modèle de récapitulatif de collaboration/coopération (Picard, 2007)

Tableau 2 réalisé par l'auteur selon les sources suivantes :

- a. Éléments concernant le PSVR (Merli, 2016)
- b. Éléments concernant le HTC VIVE (Lancelin-Golbery, 2016)
- c. Éléments concernant le Rift (Merli, Oculus Rift : date de sortie, prix et caractéristiques du casque de réalité virtuelle, 2016)
- d. Autres Éléments (Martindale, 2018)

Tableau 3 réalisé par l'auteur selon les sources suivantes :

- a. Éléments concernant Unity (Unity, 2018)
- b. Éléments concernant Unreal Engine (Epic Games, 2018)
- c. Éléments concernant Cry Engine (Crytek, 2018)
- d. Éléments concernant Unity vs Unreal Engine (Slant, 2018)
- e. Éléments concernant Unity vs Cry Engine (Slant, 2018)

Tableau 4 réalisé par l'auteur.

Tableau 5 réalisé par l'auteur.

Tableau 6 réalisé par l'auteur.

13. Références des illustrations

Figure 1 aperçue sur : <http://blog.cyberlearn.ch/?p=2351>

Figure 2 aperçue sur : <https://www.dailydot.com/parsec/vive-mixed-reality-demo/>

Figure 3 aperçue sur : <https://www.wired.com/story/ikea-place-ar-kit-augmented-reality/>

Figure 4 aperçue sur : <http://www.trapgame.ch/fr/Rooms/11-Vice-Versa/>

Figure 5 aperçue sur : <https://www.roadtovr.com/hands-beat-saber/>

Figure 6 aperçue sur : <http://www.keeptalkinggame.com/>

Figure 7 aperçue sur : <http://www.visbox.com/products/cave/viscube-c4/>

Figure 8 aperçue sur : <http://voituredefutur.blogspot.com/2011/12/les-outils-virtuels-revolutionnent-la.html>

Figure 9 aperçue sur : <https://www.unrealengine.com/en-US/blog/build-for-vr-in-vr>

Figure 10 aperçue sur :

https://upload.wikimedia.org/wikipedia/commons/3/38/Job_Simulator_screenshot_-_Constore_02.png

Figure 11 réalisée par l'auteur sur www.lucidchart.com

Figure 12 aperçue sur : <https://www.octopusrift.com/setup-your-room-for-vr/>

Figure 13 aperçue sur : https://commons.wikimedia.org/wiki/File:Official_unity_logo.png

Figure 14 aperçue sur : <https://eunivers.net/2017/02/unity/>

Figure 15 aperçue sur : https://commons.wikimedia.org/wiki/File:Unreal_Engine_Logo.svg

Figure 16 aperçue sur : <https://docs.unrealengine.com/en-us/Programming/Slate/Overview>

Figure 17 aperçue sur : [https://commons.wikimedia.org/wiki/File:CryEngine_Nex-Gen\(4th_Generation\).png](https://commons.wikimedia.org/wiki/File:CryEngine_Nex-Gen(4th_Generation).png)

Figure 18 aperçue sur : <https://www.cryengine.com/features/sandbox-tools>

Figure 19 aperçue sur :

https://store.steampowered.com/app/323910/SteamVR_Performance_Test/?l=french

Figure 20 aperçue sur : https://twitter.com/vr_toolkitLogo Photon : <https://playfab.com/add-ons/photon/>

Figure 21 aperçue sur : <https://playfab.com/add-ons/photon/>

Figure 22 aperçue sur : <https://assetstore.unity.com/packages/tools/network/forge-networking-remastered-38344>

Figure 23 aperçue sur : <https://assetstore.unity.com/packages/tools/network/master-server-framework-71391>

Figure 24 aperçue sur : <https://assetstore.unity.com/packages/tools/network/darkrift-networking-2-95309>

Figure 25 réalisée par l'auteur sur Unity.

Figure 26 réalisée par l'auteur sur www.lucidchart.com

Figure 27 réalisée par l'auteur sur www.draw.io

Figure 28 réalisée par l'auteur sur www.draw.io

Figure 29 réalisée par l'auteur sur www.draw.io

Figure 30 réalisée par l'auteur sur Unity.

Figure 31 réalisée par l'auteur sur Unity.

Figure 32 réalisée par l'auteur sur Unity.

Figure 33 réalisée par l'auteur sur Unity.

Figure 34 réalisée par l'auteur sur Unity.

Figure 35 réalisée par l'auteur sur Unity.

Figure 36 réalisée par l'auteur sur Unity.

Figure 37 réalisée par l'auteur sur www.draw.io

Figure 38 réalisée par l'auteur sur Unity.

Figure 39 réalisée par l'auteur sur Unity.

Figure 40 réalisée par l'auteur sur Unity.

Figure 41 réalisée par l'auteur sur Unity.

Figure 42 réalisée par l'auteur sur Unity.

Figure 43 réalisée par l'auteur sur Unity.

Figure 44 réalisée par l'auteur sur Unity.

Figure 45 aperçue d'un fichier csv généré par Unity.

Figure 46 réalisée par l'auteur sur Unity.

Figure 47 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 48 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 49 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 50 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 51 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 52 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 53 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 54 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 55 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 56 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 57 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 58 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 59 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 60 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

Figure 61 réalisée par l'auteur sur la base du fichier excel « JeffZufferey_RapportHebdomadaire »

14. Annexes

14.1. Le cahier des charges

La réalité virtuelle collaborative connectée - Cahier des charges

Contexte

Ce projet est réalisé dans le cadre du travail de Bachelor 2018 au terme de la formation en Haute École Spécialisée dans la filière d'informatique de gestion. Il s'agit de créer une application collaborative connectée en réalité virtuelle avec des logiciels et des *frameworks* open source.

Objectif du travail

L'objectif est de créer une application de réalité virtuelle collaborative connectée à partir de *frameworks* et de logiciels open source. Deux utilisateurs, utilisant des casques de réalité virtuelle, pourront se retrouver dans un même espace virtuel. Ils devront communiquer pour œuvrer à l'exécution de tâches collaboratives ou coopératives.

Contexte de l'application

Les deux utilisateurs pourront se connecter au même espace virtuel. Ils pourront communiquer et effectuer des tâches en coopération. L'espace représente une salle close avec une échappatoire dont la réussite résulte de la collaboration.

Contraintes

Voici les contraintes demandées par Cyberlearn pour l'application développée :

- Utilisation de la réalité virtuelle.
- Intégration de deux personnes dans une même application.
- Séparation physique des utilisateurs pour réaliser l'expérience.
- Établissement de la communication.
- Intégration de la notion de collaboration.
- Détermination du *framework* et *assets* à utiliser pour la création de l'application.
- Logs des actions.

Fonctionnalités

Les fonctionnalités développées sont les suivantes :

- Capacité à pouvoir prendre des objets, à se déplacer grâce à la téléportation et à escalader.
- Capacité d'accueillir plusieurs utilisateurs dans une même application.
- Création d'interactions collaboratives.
- Utilisation d'un chat vocal pour communiquer.

- Logs des déplacements.
- Logs des échanges vocaux.

Lu et approuvé, le 29 mai 2018

Prof. Anne-Dominique Salamin



14.2. Product backlog

N° US	Descriptif	Critères d'acceptances	Priorité	Story Point	Date Tâches acceptées	MoSC oW	Status	Date terminés
1	Elaboration du scénario	Avoir une situation de départ, de fin avec toutes les interactions définies.	1000	13		must	●	05.06.2018
2	Définition de la Gamification du Scénario	Définir un système de gamification en cas de succès ou d'erreur	200	5		could	●	05.06.2018
3	Création du décor et avatar du personnage	Avoir un trou dans un décor avec un mont de charge puis un avatar avec une couleur attribué de manière random	900	8		must	●	05.06.2018
4	Création des objets pour les interactions	Avoir les objets pour correspondant aux interactions défini ci-avant	800	8		must	●	05.06.2018
5	Création des interactions en réalité virtuelle	Pouvoir avec une personne mettre les objets en place selon le scénario	700	13		must	●	05.06.2018
6	Création d'une application en réseau	Accueillir deux personnes dans la même scène, qui peuvent se voir	600	21		must	●	05.06.2018
7	Modifier les interactions pour les mettre en collaboration	Realiser les etapes du scenarion a 2 personnes	500	8		must	●	05.06.2018
8	Création du chat vocat	Pouvoir discuter avec deux casques	400	21		must	●	05.06.2018
9	Créations des logs	Capturer chaque déplacement, interaction et échanges vocales	300	5		must	●	05.06.2018
10	Réaliser la Gamification du scénario	Définir un système de gamification en cas de suceps ou d'erreur	100	5		could	●	05.06.2018
17	Création du réseau pour accueillir les clients	Permette à deux player de se voir	1000	5	05.06.2018	must	●	17.06.2018
18	Rendre indépendant les actions des players	Les deux joueurs peuvent se déplacer inpedemment de l'autre	900	13	05.06.2018	must	●	24.06.2018
11	Créer l'environnement test	Pouvoir avoir une zone de test	875	5	05.06.2018	must	●	09.06.2018
12	Créer l'interation de la porte	Lorsque la carte touche le bouton d'accès la porte s'ouvre	850	5	05.06.2018	must	●	09.06.2018
13	Créer l'interaction des cubes à superposer	Pouvoir prendre des cubes et les placers les un sur les autres	825	5	05.06.2018	must	●	09.06.2018
14	Créer l'interaction du grappin	Pouvoir prendre, jeter un grappin, puis qu'il s'accroche à un rebord	820	5	05.06.2018	must	●	15.06.2018
15	Créer le corps du joueur afin qu'il soit soumis à la gravité	Pouvoir monter sur les cubes et tomber	810	5	05.06.2018	msut	●	11.06.2018
16	Le joueur doit pouvoir se déplacer et se téléporter	Le joueur peut marcher et se téléporter sur les cubes mais pas en dehors du décor	800	5	05.06.2018	must	●	11.06.2018
28	Synchronisation des données en réseau	Synchroniser les donnés des interactions des utilisateurs	750	23	05.07.2018	must	●	08.07.2018
20	Créer le chat vocale	Les deux joueurs peuvent communiquer via le microphone du kit main libre	700	21	05.06.2018	must	●	08.07.2018
24	Stockage des voix	Avoir un fichier mp3 qui retrace la communication à la fin de l'application	600	8	05.06.2018	msut	●	08.07.2018
21	Création du timer	Créer l'objet 3D puis le décompte	500	5	05.06.2018	must	●	09.07.2018
27	Création des logs d'actions	Générer un document par utilisateur donnant le timer, l'objet saisi	400	13	06.06.2018	must	●	
19	Créer les logs de déplacement	Permet de situer les déplacements de chaque personne	300	13	05.06.2018	must	●	08.07.2018
26	test de l'application	vérifier que toutes les interactions puissent être exécuté, que les loges s'enregistre ainsi que la voix	200	21	05.06.2018	must	●	09.07.2018
22	Créer l'ambiance de l'urgence du moment	A l'aide d'un son et de lampes montrer l'urgence de sortir	150	5	05.06.2018	could	●	
23	Création du système de gamification	Gagne des secondes lorsqu'on parle et lors de la construction un artefact	100	13	05.06.2018	could	●	
25	Création de l'avatar du joueur et de ces mains	Que les joueurs puissent se voir et voir le main	50	8	05.06.2018	could	●	08.07.2018

Figure 47 : Product Backlog

14.3. Tableau de bord

HES-SO Valais		Report Summary													
Title															
Student		Zufferey Jeff													
Period		30.04.18 - 31.07.18													
		Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Total
Recherche/Lecture		0.00	8.00	6.00	4.50	2.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	21.00
Analyse		0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
Séance		1.50	0.00	0.00	0.00	1.75	0.00	2.00	1.00	0.00	2.00	1.00	0.00	0.00	9.25
Installation/Préparation/EcriturePV		6.00	0.00	0.50	0.00	1.00	1.20	0.00	0.00	0.00	0.00	0.00	5.00	0.00	13.70
Développement		0.00	0.00	8.00	6.00	0.00	8.00	28.00	16.00	4.00	31.00	7.00	0.00	0.00	108.00
Rédaction		2.00	2.50	4.00	18.00	14.00	0.00	7.00	0.00	32.00	14.00	40.00	65.00	20.00	218.50
Cours		0.00	4.50	0.00	0.00	0.00	0.00	2.00	3.00	0.00	0.00	0.00	0.00	0.00	9.50
Total		9.50	15.00	19.50	28.50	19.25	9.20	39.00	20.00	36.00	47.00	48.00	70.00	20.00	380.95

Week	Objectif de Développement	Week	Objectif de D'écriture
1	Préparation de l'environnement de travail	1	Préparation des documents (tables des matières, structure)
2	Avoir une meilleure perception du travail en réseau	2	Terminer la table des matières
3	Définition du décor et du scénario	3	Recherche sur la collaboration
4	Définition du scénario établissement du cahiers de cahrges	4	Ecriture du cahiers des charges
5	Définition du scénario et document pour base de travail	5	Refonte des documents demandés + écriture de la partie VR
6	Décor test, insteraction solo	6	Terminer l'introduction
7	Mise en place du réseaux LAN	7	Avancement de l'introduction des thèmes
8	Correction Réseau + Voice Chat	8	inition de l'introduction des thèmes + Avancement des solutions existantes
9	Correction réseau + Gamification + logs + test	9	Avancement des solutions existantes
10	Correctifs du réseau	10	Mettre à jour l'état de l'art
11	Système d'encadrement + création des logs	11	Refonte du document entier
12	-	12	Terminer le rapport
13	-	13	Correction et mise en page du rapport

Figure 48 : Tableau de bord (réalisé par l'auteur)

HES-SO Valais		Weekly Report							2
Title									
Name		Zufferey Jeff							
Week		07.05.18 - 13.05.18							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Recherche/Lecture			4.00		4.00				8.00
Analyse									0.00
Séance									0.00
Installation/Préparation/EcriturePV									0.00
Développement									0.00
Rédaction		2.50							2.50
Cours		4.50							4.50
Total		7.00	4.00	0.00	4.00	0.00	0.00	0.00	15.00
Date	Note								
07.05.2018	Finition de la table des matières + dépôt dans la dropbox								
07.05.2018	Tutoriel fonctionnement du cloud Unity								
08.05.2018	Recherche pdf collaboration, sorte de collaboration								
10.05.2018	Recherche pdf immersion, pédagogie								
Date	Signature Professor								
Comment									

Objectif Dev Avoir une meilleure perception du travail en réseau
Objectif TB Terminer la table des matières

N°US	Tâches	Temps estimé	Temps réalisé	Statut
6	Tutoriel pour mise en place de 2 personnages en multi	4	4.5	

Figure 50 : Rapport hebdomadaire de la 2^{ème} semaine (réalisé par l'auteur)

HES-SO Valais	Weekly Report							4
Title								
Name Zufferey Jeff								
Week 21.05.18 - 27.05.18								
	Mo	Tu	We	Th	Fr	Sa	So	Total
Recherche/Lecture		1.50	1.50	1.50				4.50
Analyse								0.00
Séance								0.00
Installation/Préparation/EcriturePV								0.00
Développement	4.00				2.00			6.00
Rédaction	5.00		5.00	5.00	3.00			18.00
Cours								0.00
Total	9.00	1.50	6.50	6.50	5.00	0.00	0.00	28.50
Working Report Details								
Date	Note							
21.05.2018	Finitions du décors + rédaction des documents rendus. Cahier des charges sur words + 1ère correction, rapport d'avancement, Scénario. Réécriture du scénario							
23.05.2018	lecture, Rattraper cours de SAP + Correction cahiers des charges et du scénario, continuation d'écriture							
24.05.2018	Correction Cahier des charges							
25.05.2018	Correction Cahier des charges + début de la tache conceptin des interactions.							
Comment								
22.05.2018	Rattrapage cours de SAP + lecture							

Objectif Dev Définition du scénario établissement du cahiers de cahrges
Objectif TB Ecriture du cahiers des charges

N°US	Tâches	Temps estimé	Temps réalisé	Statut
3	Création du décor	8	2	●
4	Création des objets d'interaction	5	3	●
1	Définition du scénario	5	3	●
5.1.	Création de l'interaction pour la porte	5	2	●

Figure 52 : Rapport hebdomadaire de la 4^{ème} semaine (réalisé par l'auteur)

HES-SO Valais	Weekly Report								6
Title	Zufferey Jeff								
Name	Zufferey Jeff								
Week	04.06.18 - 10.06.18								
	Mo	Tu	We	Th	Fr	Sa	So	Total	
Recherche/Lecture								0.00	
Analyse								0.00	
Séance								0.00	
Installation/Préparation/EcriturePV			1.20					1.20	
Développement						4.00	4.00	8.00	
Rédaction								0.00	
Cours								0.00	
Total	0.00	0.00	1.20	0.00	0.00	4.00	4.00	9.20	
Working Report Details									
Date	Note								
04.06.2018	Correctif du projet du 646 VR pour le Vrforum								
05.06.2018	Séance avec AnneDo pour validation de la table des matières des								
Commen									
05.06.2018	Ajout de fonctionnalité pour le projet du arkathon								
06.06.2018	Rattrape dernier dernier et projet pour les cours 616 et 636.								
07.06.2018	Représentant de l'HES-SO FIG dans les stands de l'hes au vrforum								
08.06.2018	Représentant de l'HES-SO FIG dans les stands de l'hes au vrforum								
09.06.2018	Monte en tant visiteur pour le vrforum 2018								
10.06.2018	Représentant de l'HES-SO FIG dans les stands de l'hes au vrforum								

Objectif Dev Décor test, interaction solo
Objectif TB Terminer l'introduction

N°US	Tâches	Temps estimé	Temps réalisé	Statut
11	Créer l'environnement test	1	1	●
12	Créer l'interaction de la porte	1	2	●
13	Créer l'interaction des cubes à empiler	1	1	●
14.1	Créer une corde flexible soumis à la gravité	8	5	●

Figure 54 : Rapport hebdomadaire de la 6^{ème} semaine (réalisé par l'auteur)

HES-SO Valais	Weekly Report								7
Title									
Name Zufferey Jeff									
Week 11.06.2018 - 17.06.2018									
	Mo	Tu	We	Th	Fr	Sa	So	Total	
Recherche/Lecture								0.00	
Analyse								0.00	
Séance					2.00			2.00	
Installation/Préparation/EcriturePV								0.00	
Développement	6.00	2.00	8.00	3.00	1.00	4.00	4.00	28.00	
Rédaction		7.00						7.00	
Cours	2.00							2.00	
Total	8.00	9.00	8.00	3.00	3.00	4.00	4.00	39.00	
Working Report Details									
Date	Note								
11.06.2018	Correctif du grappin + système d'accroche à une corniche + plus physic body pour player								
12.06.2018	Correctif de la table des matières + maj du rapportHebdomadaire								
13.06.2018	Découvert du bug réseau en développement recherche de solution en LAN								
14.06.2018	Recherche solution d'une connection en LAN								
15.06.2018	RDV avec Christophe pour parler du problème de la corde et problème du réseau LAN + RDV avec AnneDo								
16.06.2018	Recherche solution d'une connection en LAN								
17.06.2018	Recherche solution d'une connection en LAN + solution trouvé UNITY n'accepte pas deux adresse publiques								
Comment									
Connection Réseaux rectifié, mais besoin d'un deuxième setup pour calibrer les paramètres réseaux (fluidifier, interactions,									

Objectif Dev Mise en place du réseaux LAN
Objectif TB Avancement de l'introduction des thèmes

N°US	Tâches	Temps estim	Temps réalisé	Statut
14.2	Attacher le grappin à l'encoche	5	3	
15	Créer le corps du joueur afin qu'il soit soumis à la gravité	1	1	
14.3	Pouvoir prendre la le grappin n'importe ou	2	3	
16	Le joueur doit pouvoir se déplacer et se téléporter	1	1	
17	Création du réseau pour accueillir les clients	10	21	
18	Rendre indépendant les actions des players	5	2	

Figure 55 : Rapport hebdomadaire de la 7^{ème} semaine (réalisé par l'auteur)

HES-SO Valais	Weekly Report							8
Title								
Name	Zufferey Jeff							
Week	18.06.18-24.06.18							
	Mo	Tu	We	Th	Fr	Sa	So	Total
Recherche/Lecture								0.00
Analyse								0.00
Séance				1.00				1.00
Installation/Préparation/EcriturePV								0.00
Développement	5.00		4.00			3.00	4.00	16.00
Rédaction								0.00
Cours						3.00		3.00
Total	5.00	0.00	4.00	1.00	0.00	6.00	4.00	20.00
Working Report Details								
Date	Note							
18.06.2018	Mise en place du système réseau en mode VR. Gestion des spawns, mettre les objets visible par le réseau,							
20.06.2018	Mise en place de l'environnement de test à deux casques VR. Test, révèle que les deux joueurs ne sont pas							
21.06.2018	Rdv avec Annedo pour la prévenir de l'avancé de l'application							
23.06.2018	Recherche d'autre tuto, pour changer la mécanique du réseau. Partir depuis une application fonctionnel.							
24.06.2018	Mettre un collider sur les joueurs pour qu'ils puissent grimper sur les cubes.							
Comment								
21.06.2018	Examen finaux							
22.06.2018	Examen finaux							

Objectif Dev Correction Réseau + Voice Chat

Objectif TB Finition de l'introduction des thèmes + Avancement des solutions existantes

N°US	Tâches	Temps estimé	Temps réalisé	Statut
18	Rendre indépendant les actions des players	8	2	
19	log de déplacement par joueur	5		
20	Création du chat vocale	10	5	
21	Stockage des voix par joueurs	5		
18.1	Avatar visible	5	5	
18.2	Objet visible depuis le client/server	10	4	

Figure 56 : Rapport hebdomadaire de la 8^{ème} semaine (réalisé par l'auteur)

HES-SO Valais		Weekly Report							9
Title									
Name		Zufferey Jeff							
Week		25.06.18-01.07.18							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Recherche/Lecture									0.00
Analyse									0.00
Séance									0.00
Installation/Préparation/EcriturePV									0.00
Développement								4.00	4.00
Rédaction			8.00	8.00		8.00	8.00		32.00
Cours									0.00
Total		0.00	8.00	8.00	0.00	8.00	8.00	4.00	36.00
Working Report Details									
Date	Note								
26.06.2018	Ecriture de l'état de l'art sur les périphérique de la vr								
27.06.2018	Mise en forme des figures et de leur référence, continuation de l'état de l'art sur les périphériques de la VR								
29.06.2018	Ecriture de l'état de l'art sur les périphérique de la vr								
30.03.2018	Continuation de l'écriture de l'état de l'art sur les périphérique de la vr								
01.07.2018	Recherche et compréhension du problème réseau								
Comment									
28.06.2018	RDV pour un test à l'école de La source à lausanne pour parler de l'utilisation complémentaire aux cours d'une application en réalité virtuelle.								

Objectif Dev Correction réseau + Gamification + logs + test
Objectif TB Avancement des solutions existantes

N°US	Tâches	Temps estim	Temps réali	Statut
18.2	Recherche solution pour réseau (client/server)	10	4	
21	Création du timer	5	0	
22	Créer l'ambiance de l'urgence du moment	3	0	
23	Création du système de gamification	10	0	
25	Création de l'avatar du joueur et des mains	3	0	

Figure 57 : Rapport hebdomadaire de la 9^{ème} semaine (réalisé par l'auteur)

HES-SO Valais	Weekly Report							10
Title								
Name Zufferey Jeff								
Week 2.07.18-8.07.18								
	Mo	Tu	We	Th	Fr	Sa	So	Total
Recherche/Lecture								0.00
Analyse								0.00
Séance		2.00						2.00
Installation/Préparation/EcriturePV								0.00
Développement			1.00	6.00	4.00	9.00	11.00	31.00
Rédaction	4.00	6.00	4.00					14.00
Cours								0.00
Total	4.00	8.00	5.00	6.00	4.00	9.00	11.00	47.00
Working Report Details								
Date	Note							
02.07.2018	Finition des 6 sens et correction							
03.07.2018	RDV avec Christophe pour savoir ou est la pratique et voir les solutions restantes, RDV avec AnneDo, Ecriteur de la							
04.07.2018	Ecriture, Compréhension du problème réseau UNET sur le projet VR2C							
05.07.2018	Compréhension du problème réseau UNET sur le projet VR2C							
06.07.2018	Plusieur tentative de résolution du problème réseau sur le projet VR2C							
07.07.2018	Adaptation des solutions sur le projet PaintBall. Ajouter VRTK au STEAMVR et UNET + Switch sur Photon+VRTK,							
08.07.2018	Bref Conception d'un décors 3D, mise en place des objets 3D, Création des logs. US 19,21 rechercher pour le timer							
Comment								

Objectif Dev Correctifs du réseau
Objectif TB Mettre à jour l'état de l'art

N°US	Tâches	Temps estimé	Temps réalisé	Statut
18	Rendre indépendant les actions des players	8	2	
19	log de déplacement par joueur	5		
20	Création du chat vocale	10		
21	Stockage des voix par joueurs	5		
18.1	Avatar visible	5	5	
18.2	Objet visible depuis le client/serveur	10		
21	Création du timer			
22	Créer l'ambiance de l'urgence du moment			
23	Création du système de gamification			

Figure 58 : Rapport hebdomadaire de la 10^{ème} semaine (réalisé par l'auteur)

HES-SO Valais		Weekly Report							11
Title									
Name		Zufferey Jeff							
Week		10.07.18-16.07.18							
		Mo	Tu	We	Th	Fr	Sa	So	Total
Recherche/Lecture									0.00
Analyse									0.00
Séance		1.00							1.00
Installation/Préparation/EcriturePV									0.00
Développement		7.00							7.00
Rédaction			9.00	4.00	4.00	4.00	10.00	9.00	40.00
Cours									0.00
Total		8.00	9.00	4.00	4.00	4.00	10.00	9.00	48.00
Working Report Details									
Date	Note								
10.06.2015	Etablissement du timer, tentative de résoudre les problèmes de définir la fin de l'application ainsi que les logs de								
11.07.2018	Refonte du cahiers des charges, de la table des matières, nettoyage des parties à garder ou non								
12.07.2018	Nettoyage des parties à garder, réécriture de l'introduction des thèmes (collaboration + réalité virtuelle)								
13.07.2018	Finition de l'introduction des thèmes, écriture des solutions existantes								
14.07.2018	Ecritures des solutions existantes								
15.07.2018	Finition des solutions existantes et écriture de la comparaison des casques, logiciel de conceptions								
16.07.2018	Finition des comparaison des casques, écriture des logiciels de conceptions et des comparatifs des add-ons								
Comment									

Objectif Dev Système d'encadrement + création des logs
Objectif TB Refonte du document entier

N°US	Tâches	Temps estimé	Temps réalisé	Statut
19	log de déplacement par joueur	5	3	
21	Création du timer	5	3	
23	Création du système début fin de l'application	10	1	

Figure 59 : Rapport hebdomadaire de la 11^{ème} semaine (réalisé par l'auteur)

14.5. Procès-Verbaux

Procès-verbal du 30.04.2018

Lieu : Bureau d'Anne-Dominique Salamin

Personne : Anne-Dominique Salamin et, Jeff Zufferey

1. Présentation globale du travail de Bachelor.

Discussions de la forme du travail écrit, de la gestion du temps et de l'organisation de travail.

2. Points abordés :

- La nature de l'application basé sur l'observation ou suivant un cadre préfini.
- Définitions des critères d'application.

3. Question en suspens :

- Le scénario de l'application.

4. Objectifs à réaliser pour la prochaine fois :

- La table des matières complète allant jusqu'à trois niveaux dans l'optique d'avoir un fils rouge.
- Des recherches effectuées sur les différents types de collaboration la notion les séparant de l'aide ou de la communication.
- La méthodologie utilisée pour le projet.

5. Prochain Rendez-vous :

14.05.2018

Procès-verbal du 14.05.2018

La séance n'a pas eu lieu dû à empêchement de Jeff Zufferey

1. Prochain Rendez-vous :

21.05.2018

Procès-verbal du 21.05.2018

La séance n'a pas eu lieu dû à empêchement de Jeff Zufferey

1. Prochain Rendez-vous :

29.05.18

Procès-verbal du 29.05.18

Lieu : Salle 306 à Bellevue

Personne : Anne-Dominique Salamin et Jeff Zufferey

2. Points abordés :

- Finalisation du cahier des charges et signature.
- Scénario de l'application basé sur la collaboration.
- Structure de la table des matières.

3. Objectifs à réaliser pour la prochaine fois :

- Nouvelle table des matières avec les descriptions.
- Refonte du scénario basé sur la coopération et non sur la collaboration.
- Document pour la planification du projet.

4. Prochain Rendez-vous : 5.06.2018

Procès-verbal du 5.06.2018

Lieu : Bureau d'Anne-Dominique Salamin

Personne : Anne-Dominique Salamin et Jeff Zufferey

1. Points abordés :

- Validation de la table des matières.
- Validation du scénario d'application.

2. Objectifs à réaliser pour la prochaine fois :

- Avancer l'application
- Continuer l'écrit.
- Préparations de la journée test avec l'association des étudiants.
- Définir un protocole de test.

3. Prochain Rendez-vous : 15.06.2018

Procès-verbal du 15.06.2018

Lieu : Bureau d'Anne-Dominique Salamin

Personne : Anne-Dominique et Jeff

1. Points abordés :

- Les problèmes de l'application au niveau du réseau.
- Changement du public cible pour la journée test.

2. Objectifs à réaliser pour la prochaine fois :

- Mettre à jour le rapport écrit.
- L'aspect réseau de l'application doit être résolu.

3. Prochain Rendez-vous :

3.07.2018

Procès-verbal du 3.07.2018

Lieu : Bureau d'Anne-Dominique Salamin

Personne : Anne-Dominique Salamin et Jeff Zufferey

1. Points abordés :

- État de l'application non fonctionnelle.
- Recherche d'alternative pour résoudre le problème réseau.
- Recentrer les fonctionnalités de l'application.

2. Objectifs à réaliser pour la prochaine fois :

- Terminer l'application.
- Avancer le travail écrit.

3. Prochain Rendez-vous :

9.07.2018

Procès-verbal du

Lieu : Bureau d'Anne-Dominique Salamin

Personne : Anne-Dominique Salamin et Jeff Zufferey

1. Points abordés :

- Application terminée à 90% du cahiers de charges.
- Annulation de la phase de test.
- Adaptation et restructuration du travail écrit.

2. Objectifs à réaliser pour la prochaine fois :
 - Refonte du travail écrit selon les points abordés.
 - Terminer une version alpha du rapport écrit.

3. Prochain Rendez-vous :
16.07.2018

Procès-verbal du 16.07.2018

Lieu : Bureau d'Anne-Dominique Salamin

Personne : Anne-Dominique Salamin et Jeff Zufferey

4. Points abordés :
 - Conseils sur la structuration du document écrit.

5. Objectifs à réaliser pour la prochaine fois :

- Terminer le rapport

6. Prochain Rendez-vous :