



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEURE

ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA BOUMERDES

CENTRE DE DÉVELOPPEMENT DES

TECHNOLOGIES AVANCÉES

« CDTA »

Projet de fin d'études pour l'obtention du

Diplôme **Master 2** en : **Systèmes informatiques distribués**

THÈME

Un système de co-manipulation à distance
dans un environnement augmenté collaboratif

Réalisé par :

-Touel Sofiane

-Mekkadem Mounir

Encadreurs :

-Riahla Mohamed Amine (UMBB)

-Mouna Kenoui (CDTA)

-BenBelkacem Samir (CDTA)

2016/2017

REMERCIEMENT

Ce travail a été réalisé à l'université M'hamed Bougara de Boumerdès ainsi qu'au sein de l'équipe IRVA (Interaction hommes-machine & Réalité virtuelle/augmentée), au CDTA (Centre de Développement des Technologies avancées), sous la supervision de :

Monsieur Riahla Mohamed Amine (UMBB).

Madame Mouna Kenoui (CDTA).

Monsieur BenBelkacem Samir (CDTA).

Nous tenons à remercier nos trois encadreurs, pour l'aide qu'ils nous ont apportée dans le domaine de la 3D, le réseau, ainsi que pour les différentes orientations pour chaque phase de l'aboutissement de ce travail, nous les remercions aussi pour leurs commentaires et remarques pertinentes qui nous ont permis d'améliorer la rédaction de ce manuscrit.

Nous tenons également à remercier nos professeurs qui nous ont formés durant tout notre cursus et fourni les prérequis nécessaires à l'accomplissement de notre projet de fin d'études.

Nos sincères remerciements s'adressent également à tous les membres du jury pour l'immense honneur qu'ils nous accordent en acceptant d'évaluer ce modeste travail.

Nous remercions tous ceux qui ont participé de près ou de loin, de façon directe ou indirecte, à la réalisation de ce projet de fin d'études de master2.

Résumé

Dans les environnements collaboratifs en réalité virtuelle ou encore en réalité augmentée, des utilisateurs se trouvant parfois sur des lieux géographiquement éloignés peuvent partager des objets virtuels, mais surtout, réaliser une interaction 3D collaborative avec ces derniers. Ces utilisateurs vont donc pouvoir travailler ensemble dans un environnement virtuel partagé pour réaliser différentes activités comme, par exemple, de la formation à distance ou encore de l'expertise distante. Il est essentiel dans un tel environnement partagé, que les participants à la session collaborative puissent interagir conjointement d'une manière efficace, il est donc nécessaire pour chaque utilisateur de percevoir le même état des objets virtuels partagés et au même moment afin de garantir une bonne cohérence de l'environnement virtuel. En outre l'interaction de chaque utilisateur devra être la plus naturelle possible, il est de même pour la visualisation des actions des autres participants notamment pour des applications temps-réel.

L'objectif du travail de fin d'études consiste à réaliser un système collaboratif en réalité augmentée permettant à plusieurs utilisateurs de manipuler des objets 3D communs (Co-manipulation 3D) à distance, des dispositifs matériels très simples seront utilisés (clavier, souris, manette de jeux) pour la partie de l'interaction. Ce système devra garantir une meilleure cohérence entre les différents environnements 3D et une meilleure synchronisation des tâches des utilisateurs. Il devra également prendre en compte les contraintes technologiques par exemple la latence réseau.

Dans notre cas, nous avons opté pour une application E-Learning dans le domaine médical afin de résoudre le problème de compréhension des étudiants dans les cours d'anatomie.

Sommaire

Introduction & Problématique

Chapitre1: Coexistence du réel et du virtuel

I.Introduction	08
II. Réalité augmentée	08
II.1 Définition.....	08
III. Développement d'un système de réalité augmentée.....	08
III.1 Localisation en utilisant des capteurs.....	09
III.1.1 Capteurs magnétiques	09
III.2.1 Localisation basée sur des cibles (marqueurs ou tags) codées.....	09
III.2.2 Localisation basée sur des cibles naturelles.....	11
IV. Réalité augmentée collaboratif(EAC)	11
IV.1 Définition	11
V. Types de collaboration en réalité augmentée (RA)	11
V.1 Collaboration Co-localisée.....	11
V.2 Collaboration distante.....	12
VI. Projets développés dans le cadre des systèmes de RA collaboratif.....	12
VI.1. RA collaboratif en face à face.....	12
VI.2. Table ronde augmentée	13
VI.3. RA collaboratif dans un environnement extérieur (cas de la navigation).....	13
VI.4. RA collaboratif et mobile pour une application de peinture.....	15
VI.5 RA collaboratif pour les jeux d'échecs.....	15
VI.6 RA collaboratif pour ShareX3D	16
VI.7 RA collaboratif pour SPIN-3D	16
VI.8 Tableau comparatif	16
V.Conclusion	18

Chapitre2: Environnements Augmentés Distribués (EAD)

I. Introduction	19
II. Les environnements augmentés distribués.....	19
II.1 Architectures réseaux des environnements de réalité augmentée distribués.....	19
II.I.1 Architecture client/serveur.....	19
II.I.2 Architecture pair-à-pair.....	20
II.I.3 Architecture hybride	21
II.2. Protocoles de communication pour les EAD	21
II.2.1 Protocoles classiques.....	21
III. Mode de distribution de données pour les EAD.....	22
III.1 Mode homogènement répliqué	22
III.2 Mode centralisé.....	23
III.3 Mode partiellement répliqué (mode hybride)	24
IV. Maintien de la cohérence pour les EAD.....	24
IV.1 Synchronisation	25
IV.2 Gestion de la concurrence.....	25

IV. 3 Latence.....	25
V. Conclusion	26

Chapitre3: Conception d'un système EAD dans le domaine médical

I. Introduction	27
II. Conception du système	28
II.1 Architecture du système	28
II.2 Protocoles de communication	29
II.2.1 Protocole RPC	29
II.2.2 Protocole UDP.....	30
II.2.3 NAT (Network Address Translation).....	30
II.3 Gestion des données	30
II.3.1 Stockage de données	30
II.3.2 Traitement de données.....	31
III. Conception de l'interface.....	35
III.1 Identification des acteurs	35
III.2 Description du contexte	36
III.2.1. L'envoi des messages dans le système EAD	37
III.3 Identification des cas d'utilisation fonctionnels	37
III.4 Diagramme global des cas d'utilisation fonctionnels.....	39
IV- Conclusion.....	40

Chapitre4: Réalisation d'une application EAD médicale

I.Introduction.....	41
II-Outils	41
II.1 Plugin tiers (Artool kit, Vuforia).....	41
II.2 Unity3D & C#	42
II.3 Unet.	43
II.4 Photon	43
III.Description du problème	44
IV. Cas d'utilisation.....	44
IV.1. Scénario des opérations faites par les auteurs	46
IV.2.Scénario d'utilisation.....	47
IV.3.Scénario d'utilisation récapitulatif.....	52
V. Composant de notre système	54
V.1 Modèles 3D.....	55
V.2 Interactions.....	56
VI. Conclusion	57
Conclusion générale & perspective.....	58
Annexe	59
Bibliographie.....	69

Introduction & Problématique

Nous assistons durant cette dernière décennie à une évolution croissante du monde du numérique et ceci avec l'évolution des Smartphones, la télévision 3D. Cette évolution coïncide bien avec l'émergence de la réalité virtuelle et de la réalité augmentée où l'utilisateur peut visualiser et naviguer autour du monde qui l'entoure en étant émergé dans un monde 3D. La combinaison de ces outils software et hardware a permis l'évolution de plusieurs secteurs tel que l'industrie, la médecine et l'éducation. Ces secteurs s'intéressent à ces outils afin d'offrir de meilleures solutions pour la visualisation, la navigation et l'interaction avec un monde totalement ou partiellement numérisé.

Avec l'évolution des technologies de l'information et de communication, il est devenu possible que plusieurs utilisateurs partagent ce monde numérique et ainsi collaborer.

Dans notre cas, nous nous intéressons à la notion de multiutilisateurs qui partagent et interagissent dans un monde 3D. Cette discipline est d'actualité où beaucoup de chercheurs s'intéressent à développer des plateformes collaboratives utilisant les paradigmes de la réalité virtuelle et de la réalité augmentée pour satisfaire un besoin émanant du secteur socio-économique. La réalité augmentée collaborative (RAC) ou les environnements augmentés collaboratifs (environnements augmentés distribués) constituent un des objectifs de l'équipe IRVA (Interaction Homme-Système & Réalité Virtuelle/Augmentée) qui consiste à concevoir une plateforme collaborative en réalité augmentée pour l'assistance et la formation. Notre but dans ce mémoire, est de concevoir une mini-plateforme collaborative pour la formation dans le domaine médical et ceci en utilisant la réalité augmentée. Nous avons choisi le domaine médical, car la médecine est un domaine très difficile avec beaucoup de risques, car une mauvaise interprétation des symptômes d'une passion ou une faible connaissance en anatomie peut entraîner mort d'homme. Pour y remédié les médecins doivent posséder une formation très complète, mais avec les moyens mis à leurs dispositions bien souvent cela s'avère très difficile et ces derniers doivent avoir recours à un grand nombre de stages à l'hôpital (pas souvent évident) pour compléter leurs prés requis. Ce manque de pré requis est dû à un système d'enseignement classique, qui est maintenant obsolètes et aussi à cause du manque de moyens financiers pour fournir les équipements et modèles 3D à tous les étudiants en médecine. Durant la dispense des cours, souvent la projection d'un modèle 2D n'incite pas forcément la curiosité de l'étudiant et ne le pousse pas à s'intéresser à ce qu'il étudie. La complication principale dans les cours d'apprentissage est l'interaction sur les modèles de l'anatomie, une fois projetée dans la salle d'enseignement, leurs manipulations est impossible et l'enseignant à des difficultés à situer les questions des étudiants vis à vis des parties de l'anatomie qui sont Parfois dissimulés derrière d'autres organes plus grands et dont l'étudiant

n'a pas encore connaissance de leurs appellations. Pour cela, nous avons décomposé notre mémoire en deux parties :

Partie étude :

Le premier chapitre introduit la réalité augmentée en spécifiant son fonctionnement global ainsi que les différents domaines d'application.

Dans le deuxième chapitre, nous présentons des environnements augmentés collaboratifs et les différentes architectures réseaux pour le développement de ces environnements. Nous présentons aussi, les modes de distributions de données qui existent ainsi que les protocoles utilisés dans la littérature.

Partie contribution :

Dans le troisième chapitre, nous détaillerons la conception de notre système. Il s'agit de proposer une architecture réseau et un mode de stockage et de distribution de données. Nous utilisons comme cas d'étude la conception d'un système e-Learning par la réalité augmentée pour l'enseignement des cours d'anatomie.

Le quatrième chapitre décrit notre application suite la conception que nous avons effectuée. En fin, nous terminons notre mémoire par une conclusion et des perspectives.

Chapitre 1 : Coexistence du réel et du virtuel

I. Introduction

Nous présentons dans ce chapitre une étude bibliographique sur les systèmes de réalité augmentée et leur principe de fonctionnement. Par la suite nous introduisons les systèmes de réalité augmentée collaboratifs en s'intéressant aux différents projets développés. À la fin de ce chapitre, une étude comparative est effectuée entre les différents projets en considérant différents critères tels que l'architecture développée, la distribution des données et les protocoles de communication utilisés.

II. Réalité augmentée

II.1 Définition

Selon Ronald Azuma, chercheur à l'université de Caroline du Nord qui définit la réalité augmentée (RA) comme : << un système de réalité augmentée complète le monde réel avec des objets virtuels (générés par ordinateur) de telle sorte qu'ils semblent coexister dans le même espace que le monde réel >> [1].

Et attribue à la RA les propriétés suivantes :

1. Combiner des objets réels et virtuels dans un environnement réel,
2. Être en temps-réel et interactif,
3. Recalé (aligner) les objets réels et virtuels.

Ainsi la RA se définit par la création d'objets virtuels dans un environnement réel par rapport à un repère donné, de telle sorte qu'ils semblent coexister dans le même espace réel, ces objets peuvent interagir entre eux, ou avec d'autres objets réels.

III. Développement d'un système de réalité augmentée

Réaliser une application de réalité augmentée consiste à enrichir l'environnement de l'utilisateur par des modèles ou entités 3D. Afin de bien insérer ces objets et d'assurer leurs cohérences spatio-temporelles avec le monde réel, nous avons besoin de bien localiser l'utilisateur. Afin d'atteindre cet objectif, plusieurs approches de localisation ont été développées. Nous distinguons deux grandes approches : localisation en utilisant des capteurs et localisation par vision.

Chapitre 1 : Coexistence du réel et du virtuel

III.1 Localisation en utilisant des capteurs

III.1.1 Capteurs magnétiques

Dans ce cas, nous utilisons des caméras afin de localiser le point de vue de l'utilisateur. Il s'agit globalement d'élaborer un repéré au niveau de la scène permettant de localiser l'utilisateur et aussi insérer l'objet 3D. Dans le but de répondre à ce besoin, des algorithmes de traitement d'images ont été proposés et implémentés. Nous distinguons deux grandes approches pour la localisation par vision : localisation basée sur des cibles codées et localisation basée sur des cibles naturelles. [14]

III.2.1 Localisation basée sur des cibles (marqueurs ou tags) codées

Pour insérer l'objet virtuel dans un environnement réel nous avons besoin d'un repère (des coordonnées x,y,z référant) ce repère s'appelle marqueur, on utilise des algorithmes qui détectent et identifient le ou les marqueurs utilisés dans notre environnement réel.

Le principe de fonctionnement c'est que l'algorithme de détection capture chaque image du flux vidéo et essaye d'appliquer des opérations bien définies (qu'on détaillera plus tard) dans le but de récupérer un de nos marqueurs s'il est présent dans notre scène, cette procédure se déroule comme suit :

- La caméra capture l'image du flux vidéo .
- La binarisation de l'image par le programme en identifiant tous les cadres noirs.
- Si un cadre est trouvé, le programme récupère la position et l'orientation par rapport à la caméra en appliquant un ensemble de méthodes mathématiques.
- Une analyse du modèle qui se trouve à l'intérieur du cadre est alors effectuée afin de pouvoir comparer le modèle avec le pattern (marqueur) source qu'on a inclus dans le programme.
- Le programme associe une augmentation virtuelle à ce modèle.
- Ajuster cette augmentation par rapport aux marqueurs ainsi qu'à l'image capturée et mettre à jour la position l'orientation en temps réel.

La figure ci-après montre le schéma de principe permettant d'insérer un objet 3D en se basant sur l'utilisation du marqueur de la boîte à outil Artoolkit.

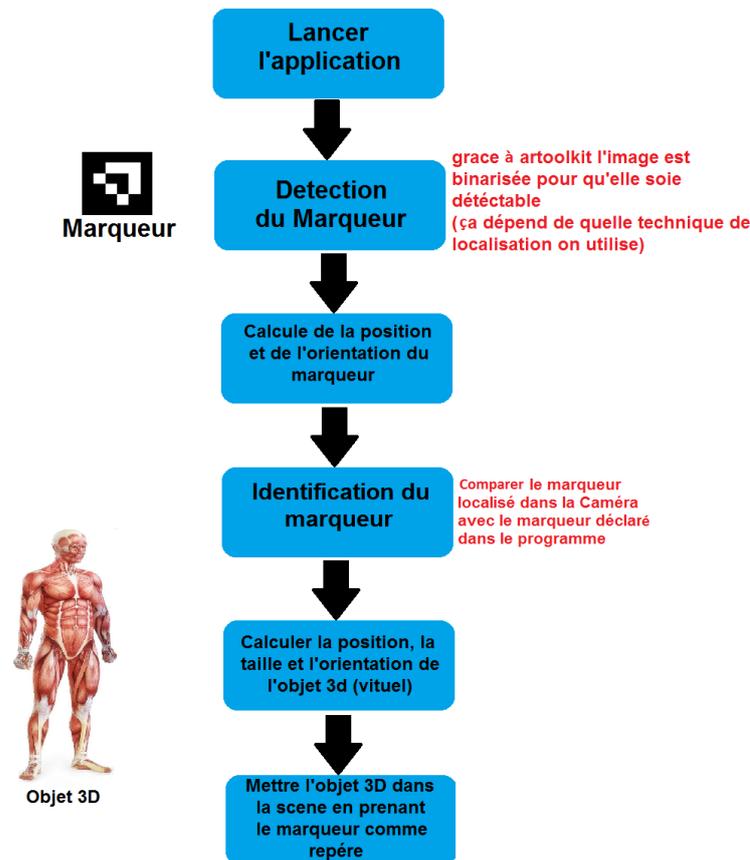


Figure III.2.1 Fonctionnement Artoolkit

a) Binarisation

L'opération initiale qui est effectuée l'identification c'est la binarisation, c'est la représentation par des pixels noirs et blancs dans l'en-tête du programme on trouve un threshold (un seuil ou une frontière) c'est lui qui s'occupe de la conversion du flux en gris, et tout pixel inférieur au threshold sera en noir et vice versa.

b) Normalisation

Le carré noir du marqueur va aider grâce à ses sommets de normaliser le motif et donc de l'identifier. Cette normalisation s'effectue en appliquant des fonctions mathématiques aux pixels.

c) Identification du marqueur

Une fois le motif redressé on compare avec le marqueur source, qui est sous forme d'un fichier texte ou chaque pixel prend une valeur entre 0 et 255 (en niveaux de gris)

Le programme compare alors les deux marqueurs dans 4 sens différents et prend le plus haut ensuite on applique l'augmentation choisie.

d) insertion de l'objet [Voir l'annexe, Partie 1]

Chapitre 1 : Coexistence du réel et du virtuel

III. 2.2 Localisation basée sur des cibles naturelles

Il s'agit de reconnaître des objets de la scène réelle afin de localiser le point de vue de l'utilisateur. Ici, deux approches se distinguent, la reconnaissance d'objets planaires (2D) et la reconnaissance d'objets volumiques (3D). Plusieurs algorithmes ont été développés dans cette direction.

La localisation basée sur des cibles naturelles se base sur trois aspects fondamentaux : la détection de points d'intérêt, la description des points d'intérêt et le calcul d'homographie.

IV. Réalité augmentée collaboratif

IV.1 Définition

“Ce qui donne naissance à la société, c'est l'impuissance où chaque homme se trouve de se suffire à lui-même, et le besoin qu'il éprouve de beaucoup de choses. La multiplicité de ses besoins a réuni dans une même habitation plusieurs hommes en vue de s'entraider : et nous avons donné à cette société le nom d'État.” Platon (Plâtôn).[15]

L'entre aide est selon Platon le fondement même de la société, la collaboration est donc vitale, l'homme a besoin de communiquer, d'échanger avec le reste du monde, idem en réalité augmentée l'utilisateur a souvent besoin d'interagir et de collaborer avec d'autres utilisateurs pour solutionner des problèmes précis, pour l'apprentissage ou simplement pour se divertir.

Cette collaboration peut se faire de plusieurs façons en étant face à face comme à l'autre bout du monde, l'interaction entre ces entités comprend des contraintes diverses, nous allons détailler ces caractéristiques et problématiques propres aux EAD.

V. Types de collaboration en réalité augmentée (RA) [16]

V.1 Collaboration co-localisée

Un groupe d'utilisateurs qui travaillent en collaboration dans une et même pièce, ils ont donc une scène réelle commune, mais avec des vues différentes, comme nous le montre la figure suivante :

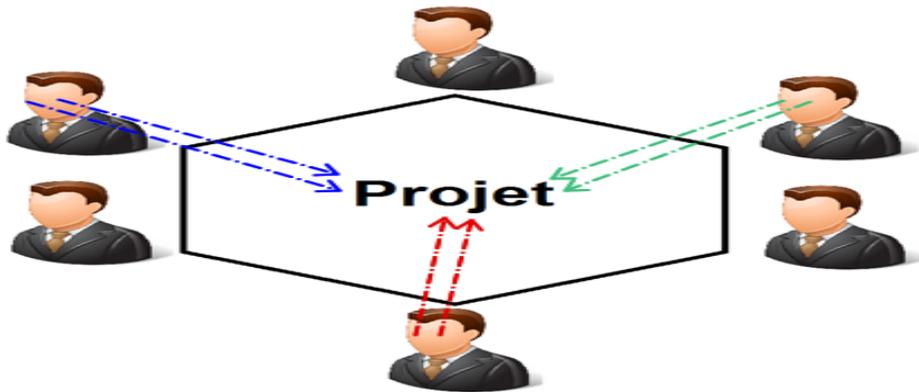


Figure V.1 Collaboration co-localisée.

V.2 Collaboration distante

Un groupe d'utilisateurs qui travaillent en collaboration dans un environnement réel différent c'est-à-dire dans des endroits séparés, on a recourt aux différents protocoles réseaux pour pouvoir communiquer entre eux, comme l'illustre l'image suivante :

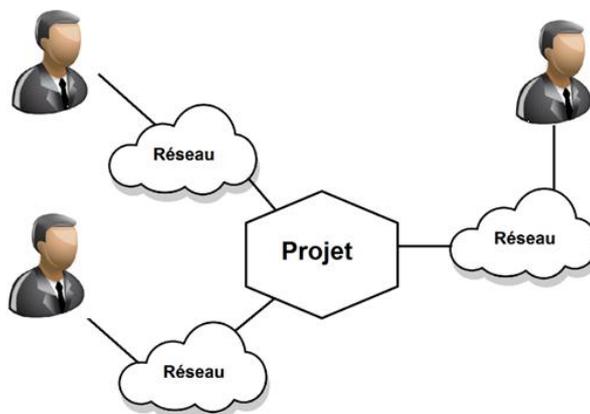


Figure V.2 Collaboration à distance

VI. Projets développés dans le cadre des systèmes de RA collaboratif

VI.1. RA collaboratif en face à face

Application de jeux mobile en Réalité augmenté ou deux joueurs se placent face à face et prennent chacun son téléphone et mettent trois marqueurs sur la table et un marqueur sur chaque téléphone comme nous le montre la Figure VI.1. Suivante [2].



Figure VI.1. Jeux dans un EAD[2]

Cette application implémentée sur un téléphone mobile sous Symbian utilise la bibliothèque Artoolkit pour la partie RA. Pour la collaboration, le système développé est basé sur l'architecture réseau pair-à-pair.

VI.2. Table ronde augmentée

Ce projet consiste à afficher une ville augmentée sur une table (voir Figure VI.2) [3]. Cet affichage permet à des spécialistes en Architecture de régler des problèmes liés à la planification des bâtiments urbains.



Figure VI.2. Projet table ronde augmentée [3].

VI.3. RA collaboratif dans un environnement extérieur (cas de la navigation)

Une application mobile et dédiée aussi pour ordinateur utilisant la Réalité augmentée pour la navigation collaborative et les tâches de navigation, et l'affichage d'information dans un environnement urbain [4].

Un ou plusieurs utilisateurs peuvent se déplacer dans une ville toutes en étant guidé par l'application à l'endroit choisi (voir Figure VI.3).

Chapitre 1 : Coexistence du réel et du virtuel



Figure VI.3.1 Un utilisateur entrain d'utiliser l'application de navigation [4]

Outils et architectures utilisés:

Un pc portable avec un processeur 2/GHz et un accélérateur graphique NVidia Quadro4Go fonctionnant sous Windows XP et un Un récepteur GPS différentiel Trimble Pathfinder Pocket et aussi un HMD Head mounted display, ces composants sont alors mis dans un sac a dos que l'utilisateur va porter, le dispositif d'affichage porter sur la tête(un écran de tête) et un capteur d'orientation InterSense InertiaCube2 et une caméra sont montés sur le casque .

L'utilisation d'un réseau wifi pour le canal de transmission, et une architecture a trois niveaux pour la gestion des données.

Le premier niveau est une base de données centrale stockant le modèle global la seconde couche correspond a une couche de conversion qui convertit les modèles de la base de données en modèle utilisable par l'application et enfin la troisième couche c'est l'application elle-même.

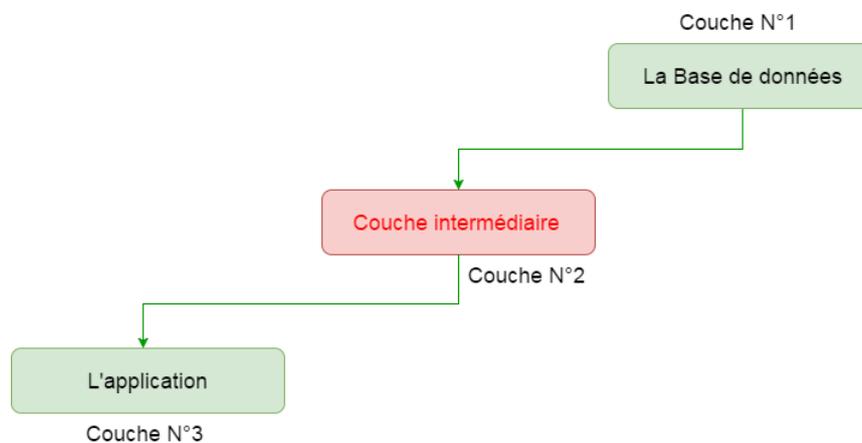


Figure VI.3.2 Schéma de l'architecture a trois niveaux.

Chapitre 1 : Coexistence du réel et du virtuel

VI.4. RA collaboratif et mobile pour une application de peinture

Cette application a été mise en place pour un travail collaboratif assisté par ordinateur (CSCW) qui signifie que l'ordinateur est utilisé comme support pour la communication.

Elle consiste à peindre des objets 3D tout en collaborant. L'interaction se fait avec un stylet.

Il sélectionne la couleur avec qui va peindre dans le pad (un tableau de couleur) et après il peint avec ce qu'il a choisi comme couleurs [5].



Figure VI.4.1 Application de peinture [5]

Outils et architectures utilisés:

L'utilisation d'une architecture Client/Serveur avec un serveur dédié qui communique toutes les données aux autres utilisateurs.

Et le canal de transmission est un réseau LAN pour les ordinateurs et un Réseau wifi pour les mobiles.

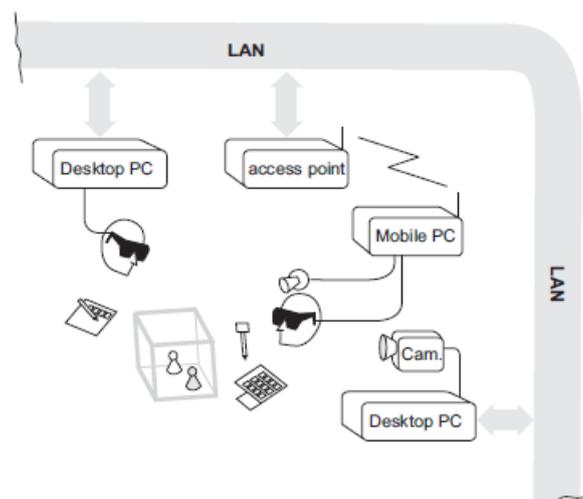


Figure VI.4.2 Schéma Application Peinture en EAD

VI.5 RA collaboratif pour les jeux d'échecs

Cette application a été mise en place sous ordinateur et sous mobile et dans un environnement augmenté collaboratif qui permet de jouer à une partie d'échecs sur table en utilisant un stylet comme outil de manipulation et d'interaction avec l'échiquier virtuel.



Figure VI.5 Application de jeux d'échec[5].

VI.6 RA collaboratif pour ShareX3D

Ce projet appelé ShareX3D [6] et développé en 2008 est une application collaborative qui permet à plusieurs utilisateurs connectés à un serveur de partager une même scène virtuelle.

architectures utilisés:

L'architecture réseau utilisée est Client/Serveur .

Protocole de connexion utilisé est le HTTP

VI.7 RA collaboratif pour SPIN-3D

Ce projet appelé **SPIN-3D** [7] montre une application qui était développée en 2002 par le laboratoire d'informatique fondamentale de Lille, elle est mise en œuvre dans un environnement virtuel collaboratif où plusieurs utilisateurs représentés par des personnes virtuelles dans une salle de réunion (pouvant s'échanger des documents et interagir entre eux).

architectures utilisés:

L'architecture réseau utilisée est pair-à-pair.

La distribution de donnée utilise un mode homogènement répliqué qu'on détaillera dans le chapitre suivant.

VI.8 Tableau comparatif

Afin de synthétiser les travaux présentés, nous donnons ici un tableau comparatif et récapitulatif, ce tableau nous permet également de nous concentrer sur des points importants qui nous serviront plus tard dans la conception et la réalisation de notre projet.

Chapitre 1 : Coexistence du réel et du virtuel

Nom de l'application	Auteur	Architecture Réseau	Protocole
Face to Face Collaborative AR On Mobile	Anders Henrysson, Mark Billinghurst, Mark Ollila [1]	pair-à-pair	NONE
Réalité augmenté collaborative mobile "Application Peinture"	Gerhard Reitmayr,Dieter Schmalstieg [5]	Client/Serveur	NONE
Réalité augmenté collaborative mobile "Application jeux d'echec "	Gerhard Reitmayr,Dieter Schmalstieg [5]	NONE	NONE
ShareX3D	Jourdain,2008 [6]	Client/Serveur	HTTP
SPIN-3D	Dit Picard,2002 [7]	pair-à-pair	NONE
The Augmented Round Table	Wolfgang Broll, Moritz Stoerring & Chiron Mottram[3]	NONE	NONE

Conclusion

Ce premier chapitre nous a permis de découvrir les notions liées au domaine de la réalité augmentée, un domaine qui s'installe fortement dans les milieux industriel, commercial et scientifique.

Les applications qui font recours à cette technologie sont nombreuses et véhiculent des défis majeurs que ça soit pour le domaine médical, industriel ou autres. Dans cette thématique, les applications collaboratives paraissent comme des solutions présentant une valeur ajoutée reconnue, ce qui les rendent parfois nécessaires pour la réalisation de tâches jusque là irréalisable par une seule personne. L'aspect multiutilisateur en réalité augmentée a été alors étudié dans ce chapitre et nous a permis de sortir avec un certain nombre de points importants pour la suite de notre travail :

D'abord nous remarquons qu'en terme d'architecture réseau, les deux architectures (pair à pair et client/serveur) sont souvent déployées dans les projets étudiés, l'autre point est que souvent les travaux ne détaillent pas les aspects liés aux protocoles de communication utilisés, les données 3D sont également très peu documentées dans les travaux que nous avons présentés dans ce chapitre. Afin d'aller plus loin dans notre étude, nous allons dans le chapitre suivant nous concentrer sur les mécanismes réseaux utilisés en RA collaboratif notamment dans un environnement distribué.

Chapitre 2 Environnements Augmentés Distribués (EAD)

I. Introduction

Nous présentons dans ce chapitre une étude bibliographique sur les environnements de réalité augmentée distribués.

II. Les environnements augmentés distribués

Les environnements augmentés distribués sont des systèmes de réalité augmentée multiutilisateur, ils sont donc utilisés dans le réseau et contribuent à résoudre des problèmes complexes. Pour les mettre en œuvre, l'utilisation des protocoles et architectures réseaux suivants sont requis.

II.I. Architectures réseaux des environnements de réalité augmentée distribués

La réalité augmentée collaborative est généralement composée de plusieurs nœuds interconnectés en local ou dispersés géographiquement. La transmission de données entre les nœuds dans un réseau peut s'effectuer selon trois configurations possibles :

- ➔ communication de type « *unicast* » : transmission uniquement d'un nœud à un autre (point-à-point),
- ➔ communication de type « *broadcast* » : transmission d'un nœud à tous les autres nœuds du réseau,
- ➔ communication de type « *multicast* » : transmission d'un nœud à un sous-ensemble de nœuds du réseau.

Dans [8], nous distinguons deux configurations possibles des nœuds sur le réseau pour un environnement distribué de réalité augmentée : les architectures pair-à-pair (voir figure II.I.2) et les architectures client-serveur (voir figure II.I.1). Lee [9] combine les deux types d'architecture pour aboutir à une architecture hybride (voir Figure II.I.3).

II.I.1 Architecture client/serveur

Malgré l'évolution des technologies dans les architectures des ordinateurs, leurs capacités demeurent limitées quand il s'agit d'accueillir des milliers de connexions entrantes qui peuvent envoyer des millions de requêtes au même temps cela peut causer un déni de service, alors l'utilisation d'un serveur **est de mise**, fournissant les ressources et la bande passante nécessaire, les autres ordinateurs du réseaux peuvent s'acquitter de leurs tâches locales seulement, et informer le serveur qui s'en chargera d'informer les destinataires. [10]

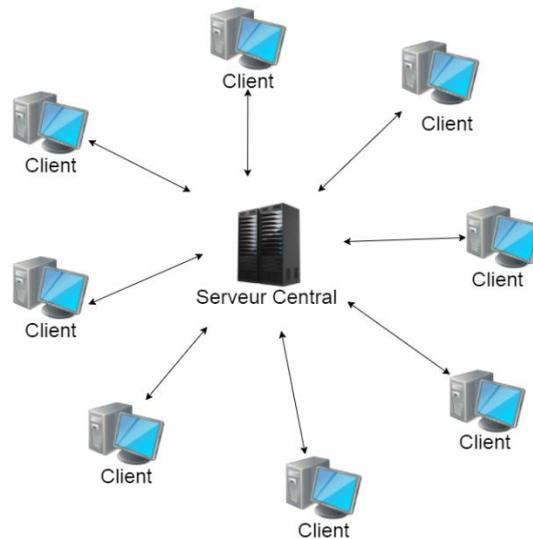


Figure II.I.1 Architecture client/serveur

L'avantage : toutes les ressources sont centralisées comme par exemple une base de données centralisée, une meilleure sécurité, une administration au niveau serveur, et enfin un réseau évolutif.

Les Inconvénients : un cout élevé pour le mettre en œuvre, et le serveur est un maillon faible du réseau.

II.I.2 Architecture pair-à-pair

Dans ce modèle (Peer to Peer) tous les ordinateurs du réseau sont égaux chacun d'eux est à la fois un client et un serveur, l'échange de données entre deux entités se fait de manière directe (sans avoir à passer par un intermédiaire central). [10]

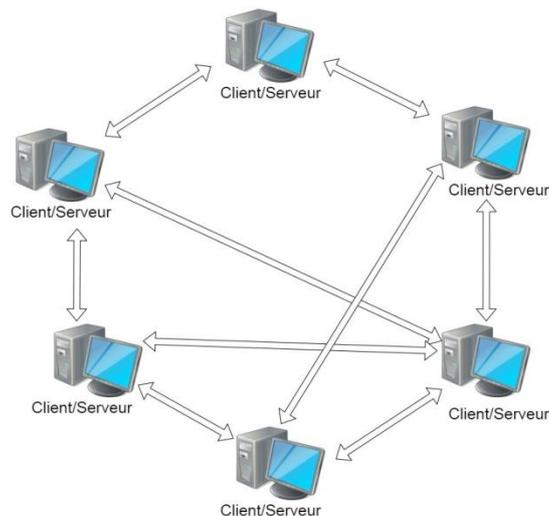


Figure II.I.2 Architecture pair-à-pair

Chapitre 2 Environnements Augmentés Distribués (EAD)

Avantages : Plus il y a d'utilisateur plus les performances du réseau grandissent, Un coût réduit, facilité et rapidité d'installation, si une source tombe en panne le réseau continue à fonctionner.

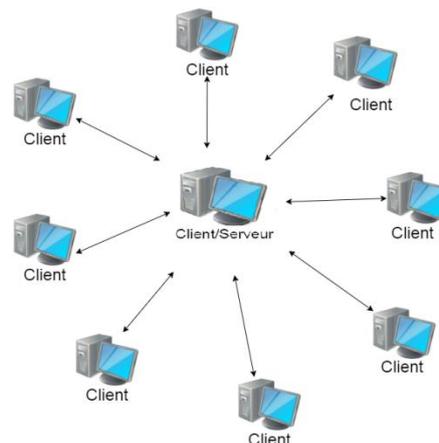
Inconvénients : Gros consommateur de Bande passante, Pas de sécurité, maintenance difficile, Et enfin il nécessite plusieurs machines pour que le réseau soit performant.

II.I.2 Architecture hybride

Le host combine les deux architectures, celle d'un Client/serveur et celle de Paire à Paire, car les ordinateurs démarrent tous égaux (tous client) puis l'un d'eux sera élu comme serveur, il sera alors un Client/serveur sachant que tout ordinateur du réseau a des chances de devenir un Serveur, une fois le Client/Serveur choisi, le réseaux reprendra le fonctionnement d'une architecture Client/Serveur.

L'avantage : c'est qu'en cas de panne du serveur, le reste du réseau doit juste élire un nouveau Client/Serveur.

L'inconvénient : dans le cas où les paquets circulants dans le réseau sont très lourds ou que la bande passante du Client/Serveur est limitée cela peut provoquer un déni de service, et le temps lors d'une élection d'un nouveau host. [10]



Figur II.I.3 Architecture hybride

II.2. Protocoles de communication pour les EAD [17]

II.2.1 Protocoles classiques

Transmission Control Protocol (TCP) est un protocole orienté connexion, fiable et la communication est bidirectionnelle c'est-à-dire que si l'émetteur envoie un paquet au récepteur, il l'informe d'abord par une ouverture de connexion, le récepteur accusera réception, ça permet ainsi de garantir la transmission des paquets et éventuellement les renvoyés s'ils ne sont pas arrivés à destination, mais l'inconvénient de ce protocole est lors de

Chapitre 2 Environnements Augmentés Distribués (EAD)

l'envoi de beaucoup de paquets qui ont besoin d'être traité rapidement, ainsi le retour de réception ralentit le processus de communication.

User Datagram Protocol (UDP) est un protocole qui n'est pas orienté connexion, et ce contrairement à TCP. Dans ce cas, le récepteur n'est donc pas prévenu de l'arrivée des paquets, et le récepteur n'accusera pas réception quand il reçoit les paquets, ce protocole est utile seulement lorsque la donnée envoyée n'est pas très importante comme un déplacement d'objet 3D, cela ne provoquera pas l'arrêt du système si un paquet contenant un pas (mouvement) de quelques millimètres disparaît, car il y'aura des millions d'autres paquets qui accompagneront ce déplacement, et cela n'apparaîtra pas dans l'affichage, car c'est de l'ordre des millisecondes (c'est ce que nous utiliserons pour notre application).

Remote Procedure Call (RPC) est un protocole réseau qui s'occupe d'appeler des méthodes (Fonctions) sur un autre ordinateur du réseau, ou bien sur l'ensemble des appareils du réseau. Son avantage c'est qu'il cache la complexité du réseau, on peut donc programmer avec RPC en ignorant tout les numéros de port, les Sockets...). L'inconvénient de ce protocole c'est la récupération d'une valeur de retour d'un registre ainsi que le passage de paramètre lors de l'appel de la fonction, car on passe les paramètres par la pile sans que la mémoire soit partagée, et cette dernière cause même des soucis pour l'envoi des pointeurs, car ils sont propres à une mémoire locale et n'auront aucun sens s'ils sont utilisés sur une autre mémoire, ils entraîneront alors des résultats erronés.

III. Mode de distribution de données pour les EAD

Lors de la conception d'un environnement augmenté collaboratif, il est important de définir comment les données seront stockées et où elles seront stockées, et quelle est l'entité ou les entités chargées d'exécuter les instructions et les commandes liées aux objets virtuels qui composent notre scène, ainsi l'instanciation des objets et leurs distributions peut se faire de plusieurs manières.

III.1 Mode homogènement répliqué

Les informations (Objets 3D, Position, Rotation, etc.) sont récupérées depuis la même base de données pour toutes les entités du réseau lors de la connexion initiale, une fois ces données répliquées chaque utilisateur peut les modifier, les manipuler en locale et pour assurer une bonne synchronisation seulement les évènements spéciaux et le changement d'état sont diffusés à tous les autres utilisateurs.

Ce mode est utilisé pour assurer une faible latence, car les modifications se font en locale avant d'être envoyé sur le réseau, car les mises à jour sont les seules à circuler dans le réseau,

Chapitre 2 Environnements Augmentés Distribués (EAD)

mais cela pose problème dans le cas ou des utilisateurs possédant de simples machines vu que les données stockées chez chaque utilisateur, si les données sont imposantes cela posera problème pour les entités disposant d'une machine à capacité moyenne, et le problème majeur si des utilisateurs manipulent le même objet en lui changeant la même propriété, après l'envoi de la mise a jour sur le réseau, la dernière mise à jour reçue seulement qui sera visible, car elle aura écrasé la précédente et cela provoquera des incohérences.

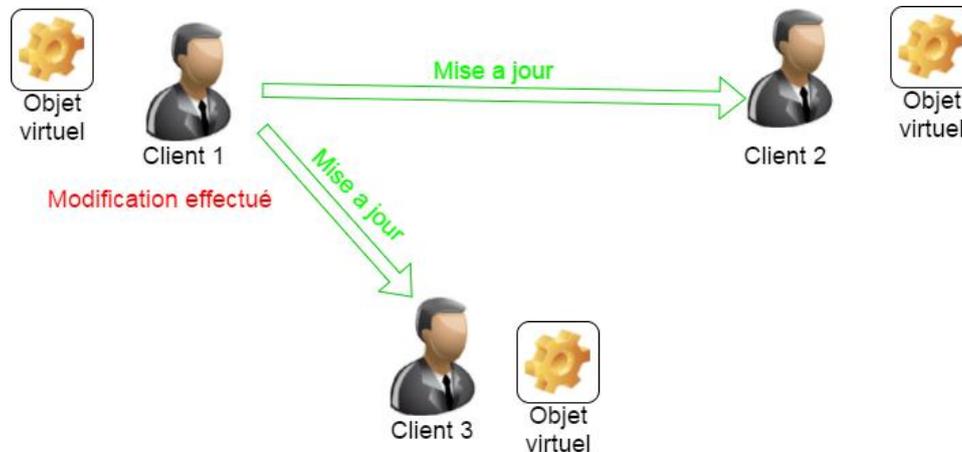


Figure III.1 Mode homogènement répliqué

III.2 Mode centralisé

Ce mode est basé sur l'architecture Client/Serveur, toutes les informations (Objets 3D, Position, Rotation, etc.) sont stockées dans un serveur central et toutes modifications ou manipulations d'un composant virtuel se fera sur le serveur,

Si un client souhaite faire une translation d'un objet par exemple, il envoie une requête au serveur, ce dernier traite la requête puis renvoie la mise à jour (la translation de l'objet) aux autres utilisateurs ou bien a tout le monde (Le client qui a envoyé la demande inclus).

Ce mode permet d'assurer une bonne synchronisation et nous libère de l'obligation de faire répliquée nos données sur chaque entité du réseau. Mais si la latence est élevée le temps de réaction entre le moment d'envoi de la requête et le moment de réponse sera également élevé, et si la qualité du serveur n'est pas bonne et que plusieurs utilisateurs constituent notre réseau, l'envoi d'une donnée assez lourde a tous ces utilisateurs au même moment peut provoquer un goulet d'étranglement.

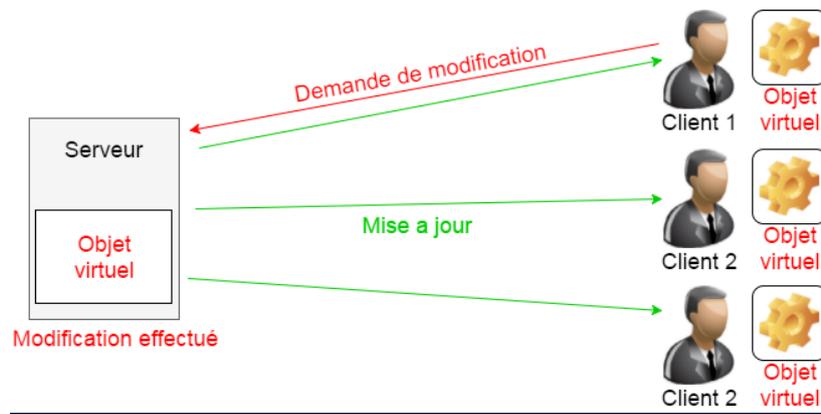


Figure III.2 Mode centralisé

III.3 Mode partiellement répliqué (mode hybride)

C'est un mode distribué qui consiste à avoir une mémoire partagée (une base de données partagée) et distribuée sur les différentes entités de notre réseau. Selon leur besoin, donc chaque entité du réseau possède seulement les données dont elle a besoin,

Ce mode est utilisé quand le nombre d'utilisateurs est très élevé ou les données sont de taille imposante, mais il représente un problème lors des mises à jour, car les utilisateurs ne savent pas quelle autre entité du réseau possède quel objet, donc la mise à jour est envoyée à tout le monde même s'ils n'ont pas l'objet qui doit être mis à jour.

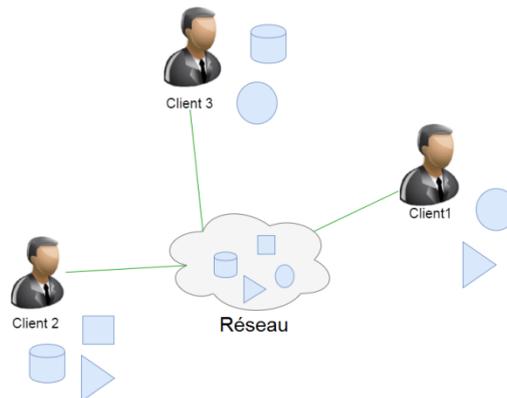


Figure III.3 mode partiellement répliqué

IV. Maintien de la cohérence pour les EAD

L'un des champs d'études des systèmes de RA collaboratifs est de maintenir la cohérence de ces systèmes caractérisés par le mixage du monde réel et du monde virtuel. L'étude de la problématique de maintien de cohérence passera essentiellement par la prise en compte de certains aspects, tels que la synchronisation et la gestion de concurrence et de latence.

IV.1 Synchronisation

Dans un environnement de RA collaboratif, les différents nœuds devront avoir le même état global au même instant. Pour atteindre cet objectif, les états associés à chaque utilisateur devront être synchronisés afin de garantir une bonne cohérence dans le temps. Ainsi, plusieurs méthodes de synchronisation ont été proposées pour répondre aux problèmes de cohérence des environnements virtuels collaboratifs.

Dans l'EAD tous les utilisateurs doivent voir la même chose au même moment, un système assure une bonne synchronisation s'il garantit la perception d'un même état de l'environnement (le même point de vue) aux mêmes instants.

Donc théoriquement avoir une bonne cohérence c'est avoir à un moment T le même état global pour tous les utilisateurs, mais dans la pratique cet état parfait n'existe pas à cause des problématiques liées à la latence.

IV.2 Gestion de la concurrence

À partir du moment où les informations sont réparties sur les nœuds de chacun des utilisateurs (comme dans le cas du mode de distribution homogènement répliqué), ces utilisateurs peuvent accéder et modifier localement les données avant que les modifications ne soient transmises aux autres utilisateurs. Cela peut entraîner des incohérences dans l'environnement virtuel, et il est nécessaire de gérer les accès concurrents des utilisateurs aux données. Pour gérer les accès concurrents aux données lorsque seul un utilisateur peut modifier un objet virtuel en même temps (interactions non collaboratives), [9]

IV. 3 Latence

Le maintien de la cohérence dans les EAD est primordial, mais entraîne parfois des effets néfastes sur la transmission de données, parfois l'envoi de la mise à jour des composants virtuels de notre environnement engendrera ce qu'on appelle une latence.

Lors des interactions il est important de réduire ce temps de réaction (latence) afin d'offrir un effet visuel fluide.

La latence est donc définie par le temps que met un paquet dans le réseau depuis la source vers la destination, On a de plus en plus de latence si on diminue de bande passante ou si cette dernière est saturée, la latence peut varier aussi selon la taille des paquets.

On conclut qu'on ne peut pas envoyer les mises à jour à chaque frame (top d'horloge) sur le réseau sous peine de voir notre bande passante saturée, on a recours à des méthodes d'extrapolation et d'interpolation [voir Annexe partie 3] qui nous garantissent un effet "smurf" aux actions et aux déplacements de nos objets 3D.

V. Conclusion

Dans les environnements augmentés collaboratifs, l'architecture réseau est très importante, en choisissant une architecture adaptée à nos besoins, nous pouvons concevoir une application réussie, la qualité du serveur est aussi une caractéristique importante, car une bande passante large permet d'éviter quelques problèmes de latences, un bon processeur est requis aussi pour les traitements d'images afin de nous éviter les contraintes de la cohérence spatio-temporelles et photométriques.

Le choix du protocole peut varier selon le besoin, parfois l'utilisation des deux protocoles est requise, dans le but d'avoir de meilleures performances et une bonne synchronisation.

Enfin le protocole RPC, est très utile dans le réseau pour véhiculer diverses informations liées à nos utilisateurs ou aux objets virtuels 3D de notre scène, mais encore faut-il faire attention au passage de paramètre et le retour de résultat lors d'appels de fonctions à cause de la mémoire privée de chaque utilisateur.

Dans le chapitre suivant nous allons parler de la conception de notre système de réalité augmenté distribué.

Chapitre 3 : Conception d'un système EAD dans le domaine médical

I. Introduction

Pour faire une bonne conception, des choix d'architectures et de protocoles réseaux ont été faits de telle sorte qu'on puisse avoir un système fonctionnel, mais surtout un système temps réel qui nous garantis une synchronisation complète pour tous les utilisateurs, ainsi qu'une réduction de latence. Nous avons aussi fait, une conception qui nous permet, de se connecter avec ou sans internet pour ne pas dépendre de ce dernier, et de synchronisé les utilisateurs qui rejoignent la session en cours en retard.

Dans notre cas, nous nous intéressons à concevoir un système de réalité augmentée collaboratif pour la formation en médecine que nous appelons « e-Learning par la réalité augmentée ». L'objectif est de permettre à un ou plusieurs enseignants de médecine de donner des cours à des étudiants qui se trouvent dans plusieurs sites distants. L'avantage de cette approche se résume en deux points : le premier permet à un nombre important d'étudiants de suivre le même cours sur plusieurs sites (amphis, salles, bibliothèque...), même si l'enseignant (professeur dans une discipline pointue) se trouve à l'étranger. Le deuxième avantage est de permettre aux enseignants et étudiants de visualiser le contenu du cours en 3D (cœur 3D, colonne vertébrale 3D, cerveau 3D...). Cette visualisation facilite la compréhension du cours, au lieu de le faire en utilisant la documentation papier et/ou une présentation PowerPoint classique. Encore plus, les participants que ce soit enseignants et/ou étudiants peuvent manipuler ces organes 3D et visualiser leurs composantes internes.

Pour des contraintes de temps, nous avons choisi un cas simple, mais qui peut être généralisé. Nous avons considéré un enseignant en anatomie qui peut enseigner deux groupes d'étudiants qui se trouvent dans deux salles différentes et ceci en utilisant le réseau internet. Nous offrons aux différents acteurs la possibilité de visualiser et manipuler des organes 3D du corps humain. Dans notre cas nous considérons deux organes : le cœur et le cerveau. L'enseignant, peut donc expliquer aux étudiants l'organe « cœur » et l'organe « cerveaux » en manipuler leurs représentations 3D. L'enseignant peut aussi décortiquer ces organes afin d'expliquer leur fonctionnement interne. L'enseignant peut discuter directement avec les étudiants s'ils sont dans la même salle ou via un chat vocal s'ils se trouvent dans une autre salle. D'autre part, les étudiants à la fois peuvent manipuler ces objets (après avoir été autorisé par l'enseignant) afin de poser des questions.

II. Conception du système

II.1 Architecture du système

Nous avons adopté une architecture de type client/serveur où le serveur est considéré comme un Host (un serveur et en même temps un client) (figure II.1). Ce dernier est attribué au professeur qui constitue le facteur central du système puisque c'est lui qui donne le cours aux étudiants qui se trouvent dans le même site ou dans des sites distants. Ainsi, les utilisateurs qui se trouvent en réseau local ou distant sont connectés au host du professeur qui contrôle le système, puisque toutes les commandes transitent par lui. Cette architecture permet une réduction importante de la latence, car il y a moins de messages qui circulent dans le réseau. Dans ce cas, le professeur sélectionne l'utilisateur qui peut manipuler les objets 3D.

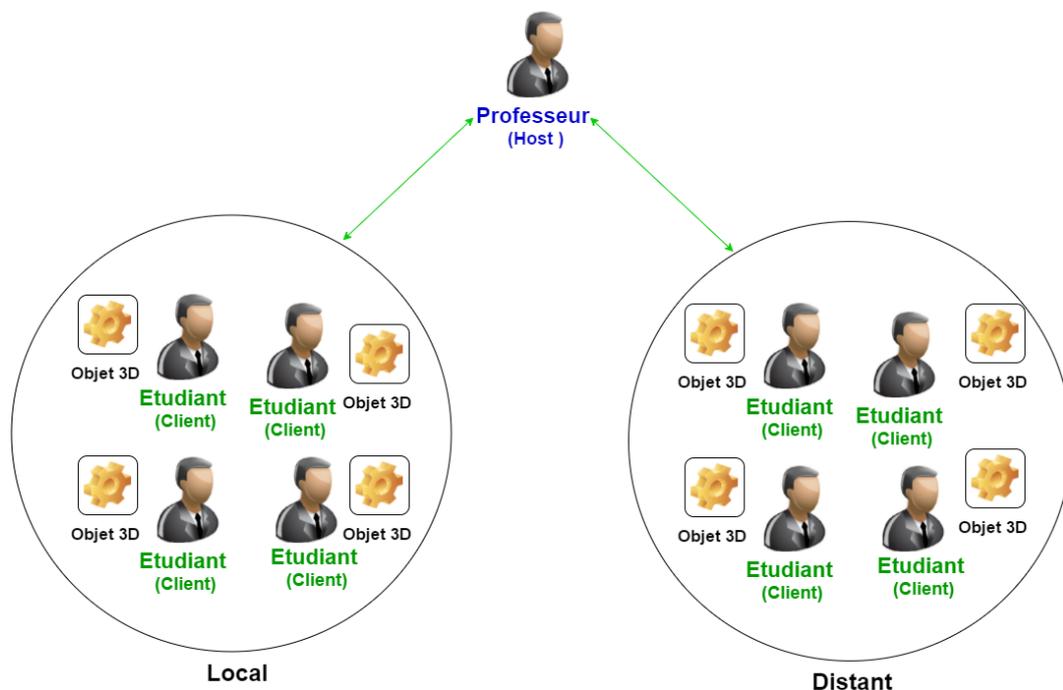


Figure II.1. Architecture de notre système.

Cette architecture permet, surtout, en cas de déconnexion du host, d'élire un nouveau host. Par exemple, le professeur peut changer de machine si la machine initiale est déconnectée ou en panne.

II.2. Protocoles de communication

Nous avons utilisé deux types de protocoles selon leur appartenance au modèle OSI : (1) le protocole RPC dans les couches « Présentation » et « Session » et (2) le protocole UDP dans la couche transport.

II.2.1 Protocoles RPC

Le protocole RPC (figure II.2.1) sert à appeler des fonctions sur des machines distantes, il est très utile dans les environnements collaboratifs, mais reste difficile à utiliser, car chaque machine possède une mémoire privée et donc les adresses mémoires dans chaque machine ont différentes significations voir aucune. Pour remédier à ce problème, nous avons dans notre cas, associé aux fonctions RPC des variables locales et des instructions prédéfinies. Cette approche permet aussi de garantir une bonne synchronisation de données.

Ils existent deux types de fonctions RPC dans notre système :

« **Commands** » : qui sont des fonctions (exemple : demande de translation) appelées à partir du client et exécutées sur le serveur.

« **ClientRpc** » : qui sont des fonctions (exemple : exécution de la translation au niveau de chaque client) appelées du serveur et s'exécutent sur les clients, nous donnons ci-dessous un schéma résumant leur fonctionnement.

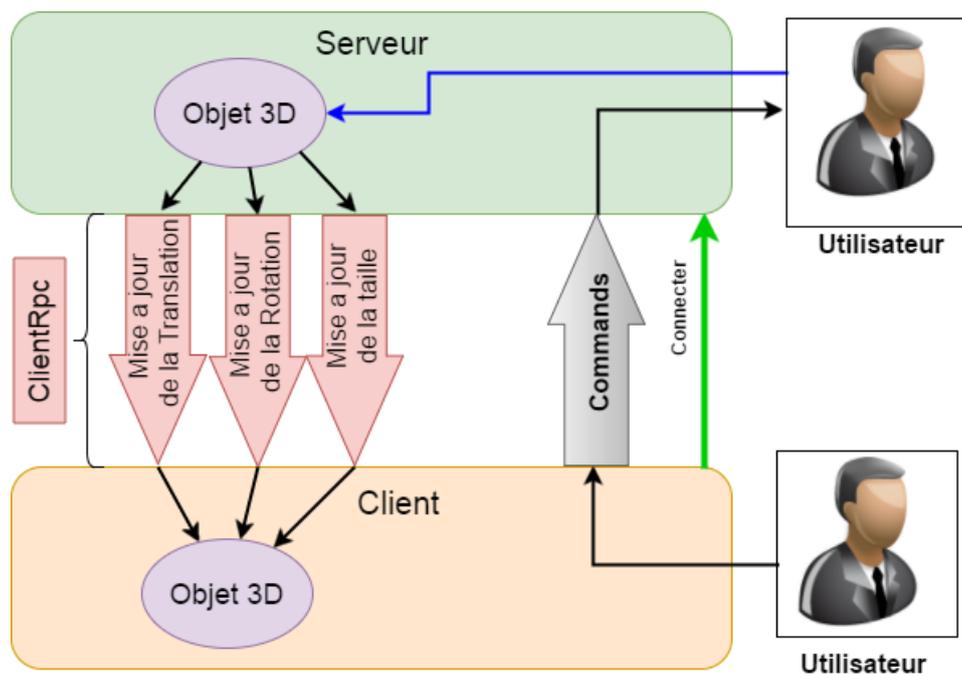


Figure II.2.1. Fonctionnement de RPC

II.2.2 Protocole UDP

Nous avons utilisé le protocole UDP afin de réduire la latence, car ce dernier n'attend pas et n'envoie pas d'accusés de réception lors de l'envoi/réception de paquets. Même si un paquet se perd l'impact ne sera pas vraiment visible, car pour garantir un déplacement plusieurs paquets sont envoyés donc un paquet perdu peut être négligeable.

II.2.3 NAT (Network Address Translation)

Il faut comprendre que les ordinateurs d'un réseau interne ne sont pas directement reliés à l'extérieur. Tout passe par un équipement appelé passerelle. Cette passerelle va faire le lien entre le réseau interne et le réseau externe (Internet). Pour cela, elle va traduire les adresses internes en adresses externes et vice-et versa lors du retour. Quand un paquet quitte le réseau interne, on remplace son adresse source par l'adresse publique, et quand un paquet arrive du net, on remplace son adresse destination par une adresse privée. [10]

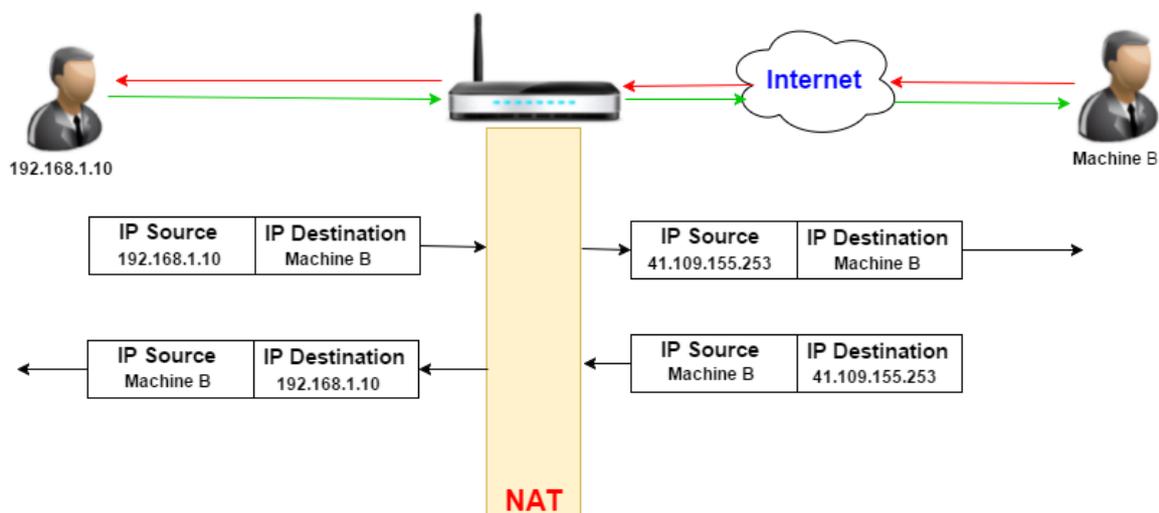


Figure II.2.3 Fonctionnement du NAT

II.3 Gestion de données [18]

II.3.1 Stockage de données

Les données du monde 3D sont stockées au niveau d'une base de données locale située dans le disque dur de chaque nœud (client et host). Ceci est justifié par le fait que ces données sont de taille importante et qui n'est pas intéressant de les envoyer via le réseau (l'inconvénient réside dans la saturation de la bande passante et l'augmentation de la latence/temps de réponse et l'effet du goulot d'étranglement). Ici, nous sommes dans le cas du mode homogènement répliqué.

Dans ce cas, c'est le serveur qui effectue l'instanciation des modèles 3D comme le montre la figure II.3.1 ci-dessous.

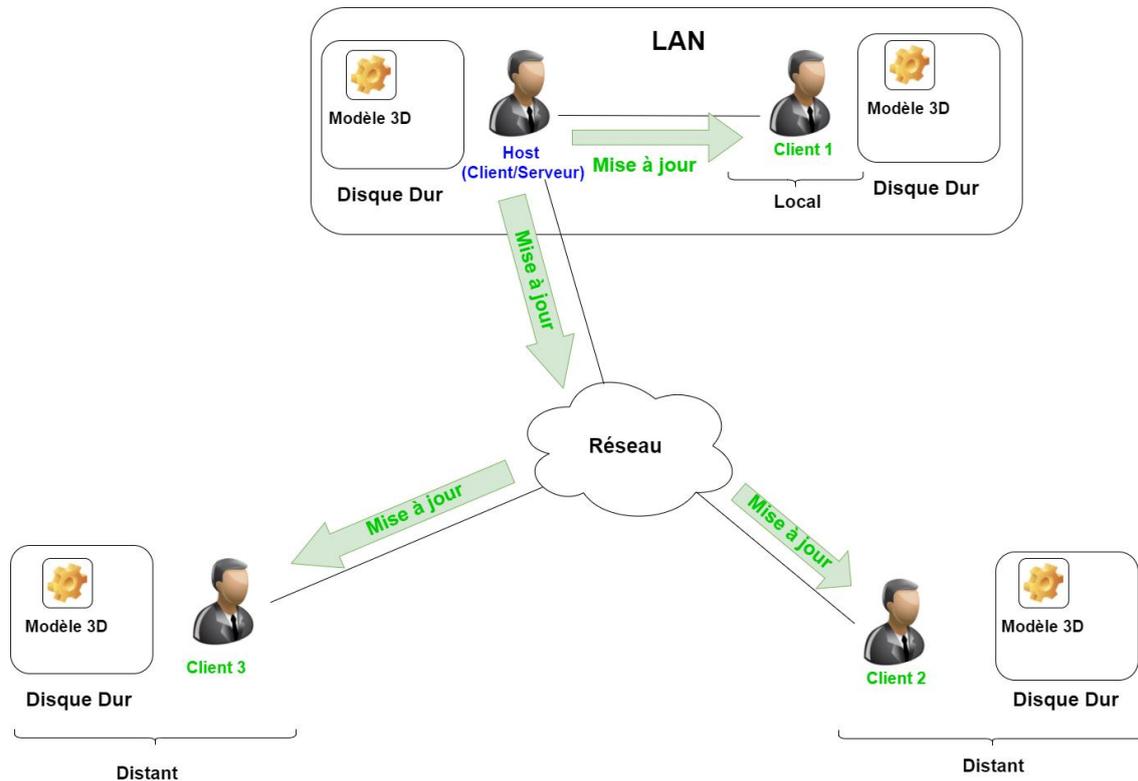


Figure II.3.1 Stockage des modèles 3D chez chaque utilisateur.

II.3.2 Traitement de données

Le traitement des données 3D sont des commandes de types translation, rotation et zoom. Ces commandes sont des informations légères comparées au modèle 3D. Pour cela, elles sont utilisées pour la communication entre les nœuds du réseau. Cependant, si le nombre de commandes est élevé, la bande passante sera encombrée, ce qui augmente la latence. Pour y remédier, nous avons proposé une approche pour réduire le nombre de commandes en favorisant le traitement au niveau du serveur.

De plus, le traitement de données s'effectue uniquement au niveau du serveur (figure II.3.2.1). Ce dernier envoie les mises à jour aux clients. Dans notre cas, nous avons opté pour le mode centralisé pour le traitement de données.

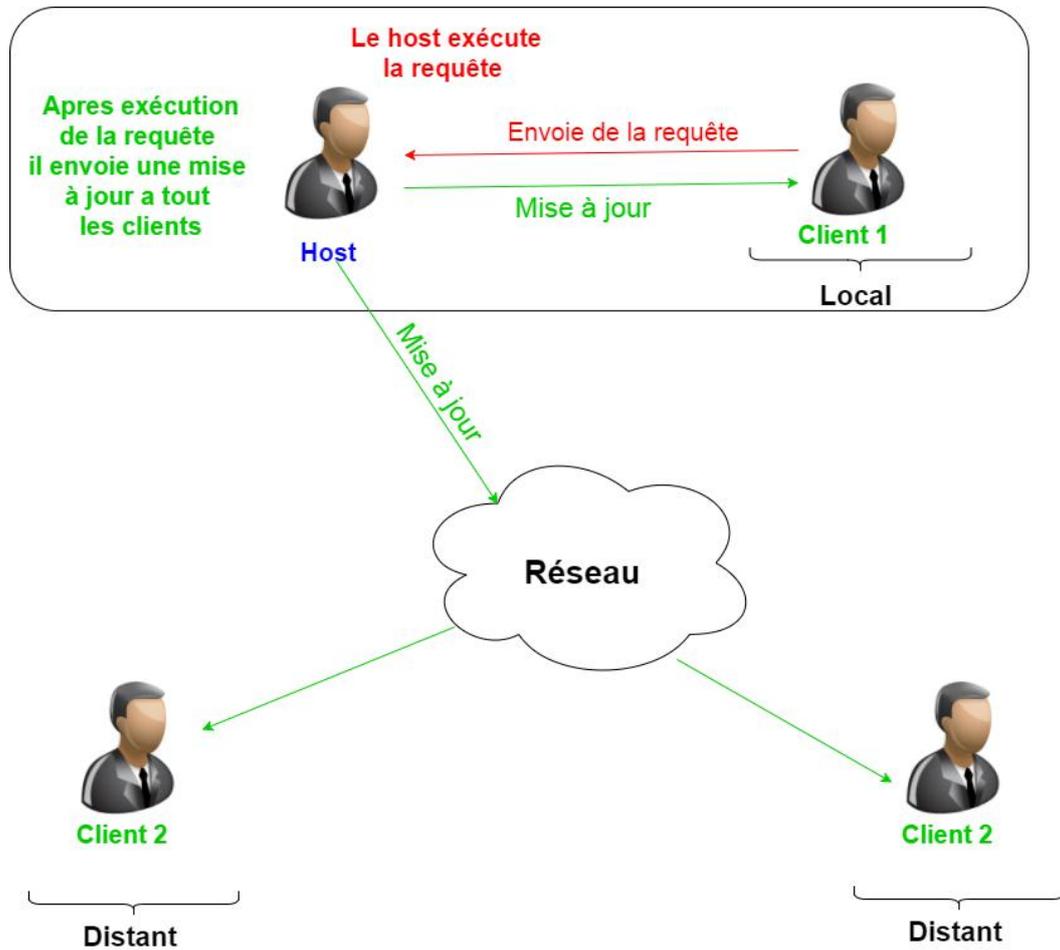


Figure II.3.2.1 Mode centralisé pour le traitement de données.

Nous présentons ci-après l'organigramme global (figure II.3.2.2) qui résume le fonctionnement de notre système du point de vue, protocole, données et architecture.

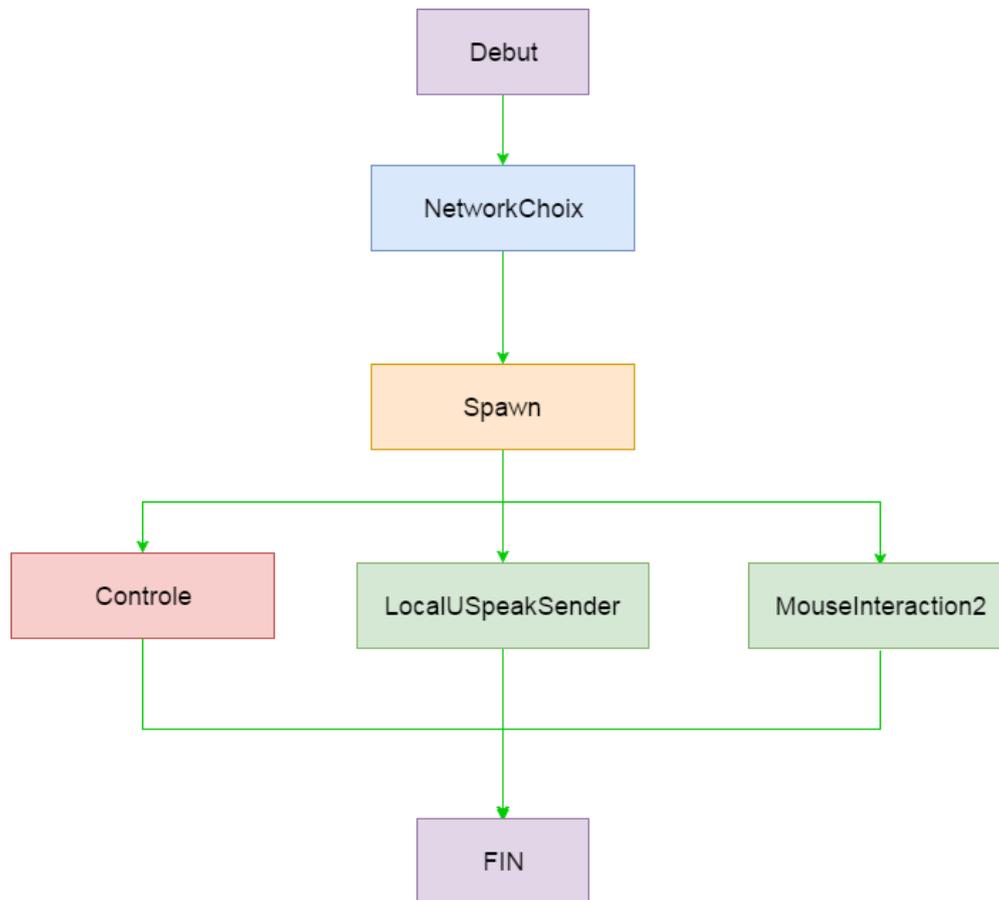


Figure II.3.2.2 Organigramme global de notre système.

Classe NetworkChoix [Voir l'annexe, Partie 2]

Cette classe s'occupe de la connexion autant que host ou client (professeur ou étudiant).

- 1- Si l'utilisateur est un professeur, il clique sur le bouton « Professeur ».
- 2- Si le bouton « professeur » est appuyé, il exécute la fonction **Host ()**.
- 3- Si l'utilisateur est un étudiant, il appuie sur le bouton « Etudiant ».
- 4- Si le bouton « Etudiant » est appuyé, il exécute la fonction **Client ()**.

Classe Spawn [Voir l'annexe, Partie 2]

Cette classe s'occupe de l'instanciation des objets 3D.

- 1- Si le marqueur est visible, le host instancie l'objet 3D.
- 2- Si l'objet 3D est présent dans la scène, il sera mis fils du marqueur.

Classe Controle [Voir l'annexe, Partie 3]

Cette classe s'occupe de la mise à jour et de la synchronisation entre les utilisateurs

- 1- Chaque client, lors de sa connexion, demande au host de lui envoyer la mise à jour de la session en cours.

- 2- Le host envoie à tous les clients connectés la mise à jour de la session.
- 3- Lorsque l'étudiant effectue une opération sur un objet 3D (exemple : translation du cœur), le client envoie une commande au host qui s'en charge de faire exécuter l'opération à tous les clients y compris lui-même.
- 4- Lorsque le professeur (propriétaire du host) effectue une opération sur un objet 3D, s'en charge de faire exécuter l'opération à tous les clients y compris lui-même.

Classe MouseInteraction2 [Voir l'annexe, Partie 2]

Cette classe est exécutée que par le host et non pas par le client.

- 1- Le host peut sélectionner une partie du modèle 3D en appuyant sur les numéros de 1 à 9
- 2- Le host peut reconstituer le sous-modèle 3D après décomposition en appuyant sur «A»
- 3- Le host peut reconstituer le modèle 3D en appuyant sur « P »

Cette classe traite aussi les interactions de la souris comme suite :

1. Si le host clique sur un composant du modèle 3D, il sera sélectionné et envoie à tous les clients une mise à jour de la situation
2. Si un utilisateur (Client ou Host) pointe la souris sur un objet, il voit une notification apparaître qui représente le nom de la partie pointée du modèle 3D.

Classe LocalUSpeakSender [Voir l'annexe, Partie 2]

Cette classe s'occupe de la transmission et de la réception de la voix.

- 1- Si un utilisateur appuie sur la touche « V », il commence à communiquer avec le professeur et les autres utilisateurs via un microphone.
- 2- L'utilisateur peut entendre aussi les autres quand ils parlent.

III. Conception de l'interface

III.1-Identification des acteurs :

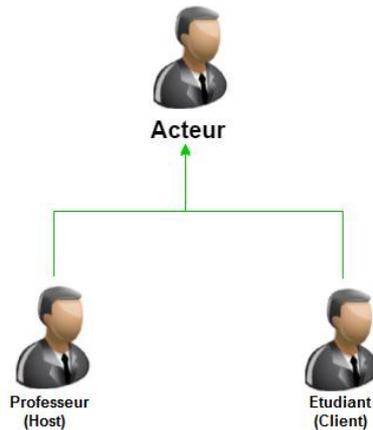


Figure III.1 Acteurs impliqués dans le système.

Acteur	Rôle
Professeur	<ul style="list-style-type: none">• Sélection des parties du modèle 3D• Agrandir les parties du modèle 3D• Translation du modèle 3D• Rotation du modèle 3D• Parler à travers le microphone
Étudiant	<ul style="list-style-type: none">• Translation du modèle 3D• Rotation du modèle 3D• Parler à travers le microphone

Tableau III.1 Rôle de chaque Acteur

III.2 Description du contexte :

Message	Description du message	Source	Destination
Message 1	Connexion autant que host	Professeur	
Message 2	Connexion autant que Client	Étudiant	professeur
Message 3	Instanciation du modèle 3D	Professeur	Étudiant
Message 4	Sélection d'une partie du modèle 3D	Professeur	Étudiant
Message 5	Sélection des sous-parties du modèle 3D	Professeur	Étudiant
Message 6	Translation du modèle 3D	Professeur Étudiant	Professeur Étudiant
Message 7	Rotation du modèle 3D	Professeur Étudiant	Professeur Étudiant
Message 8	Agrandir le modèle 3D	Professeur	Étudiant
Message 9	Envoie de la voix	Professeur Étudiant	Professeur Étudiant
Message 10	Réception de la voix	Professeur Étudiant	Professeur Étudiant

Tableau III.2 message Professeur / Étudiant

III.2.1. L'envoi des messages dans le système EAD

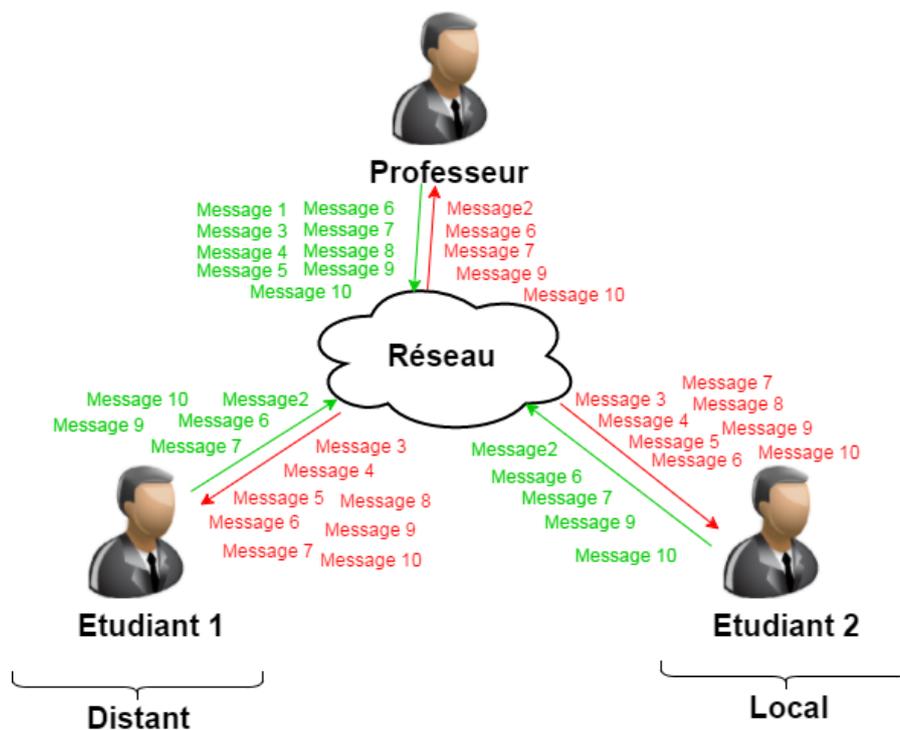


Figure III.2.1 L'envoi des messages dans le système

III.3 Identification des cas d'utilisation fonctionnels :

Ici nous donnons sous forme de tableau les différents cas d'utilisation de l'étudiant :

Cas d'utilisation	Messages (émets/reçus)	Interaction
Translater le modèle 3D	<p>Emet : Une commande au host pour faire la translation</p> <p>Reçois : La mise à jour de la rotation envoyée par le host</p>	<p>Objets réels : - Marqueur</p> <p>Objets virtuels : - Modèle 3D</p> <p>Taches : appuie sur les Touches directionnelles.</p>
Pivoter le modèle 3D	<p>Emet : Une commande au host pour faire la rotation</p> <p>Reçois : La mise à jour de la rotation envoyée par le host</p>	<p>Objets réels : - Marqueur</p> <p>Objets virtuels : - Modèle 3D</p> <p>Taches : appuie sur la touche « R »</p>

Parler	Emet : La parole de l'étudiant Reçois : Les paroles de celui qui parle	Objets réels : - Marqueur Objets virtuels : - Modèle 3D Taches : appuie sur la touche « V »
--------	---	--

Tableau III.3.1: Identification des cas d'utilisation fonctionnels pour l'étudiant

En ce qui concerne la partie liée au Professeur, nous résumons également dans le tableau ci-dessous les différents cas d'utilisation du le professeur :

Cas d'utilisation	Messages (émets/reçus)	Interaction
Sélection des parties du modèle 3D	Emet : La partie du modèle 3D sélectionné	Objets réels : - Marqueur Objets virtuels : - Modèle 3D Taches : Click gauche sur la partie choisie.
Translater le modèle 3D	Emet : Mise à jour de la translation du modèle 3D	Objets réels : - Marqueur Objets virtuels : - Modèle 3D Taches : appuie sur les Touches directionnelles.
Pivoter le modèle 3D	Emet : Mise à jour de la rotation du modèle 3D	Objets réels : - Marqueur Objets virtuels : - Modèle 3D Taches : appuie sur la touche « R »
Agrandir la partie sélectionnée du modèle 3D	Emet : Mise à jour de la taille du modèle 3D	Objets réels : - Marqueur Objets virtuels : - Modèle 3D

		Taches : Click gauche sur la partie du modèle 3D sélectionnée.
Parler	Emet : La parole du professeur Reçois : Les paroles de celui qui parle	Objets réels : - Marqueur Objets virtuels : - Modèle 3D Taches : appuie sur la touche « V »

Tableau III.3.2: Identification des cas d'utilisation fonctionnels pour le Professeur

III.4 Diagramme global des cas d'utilisation fonctionnels :

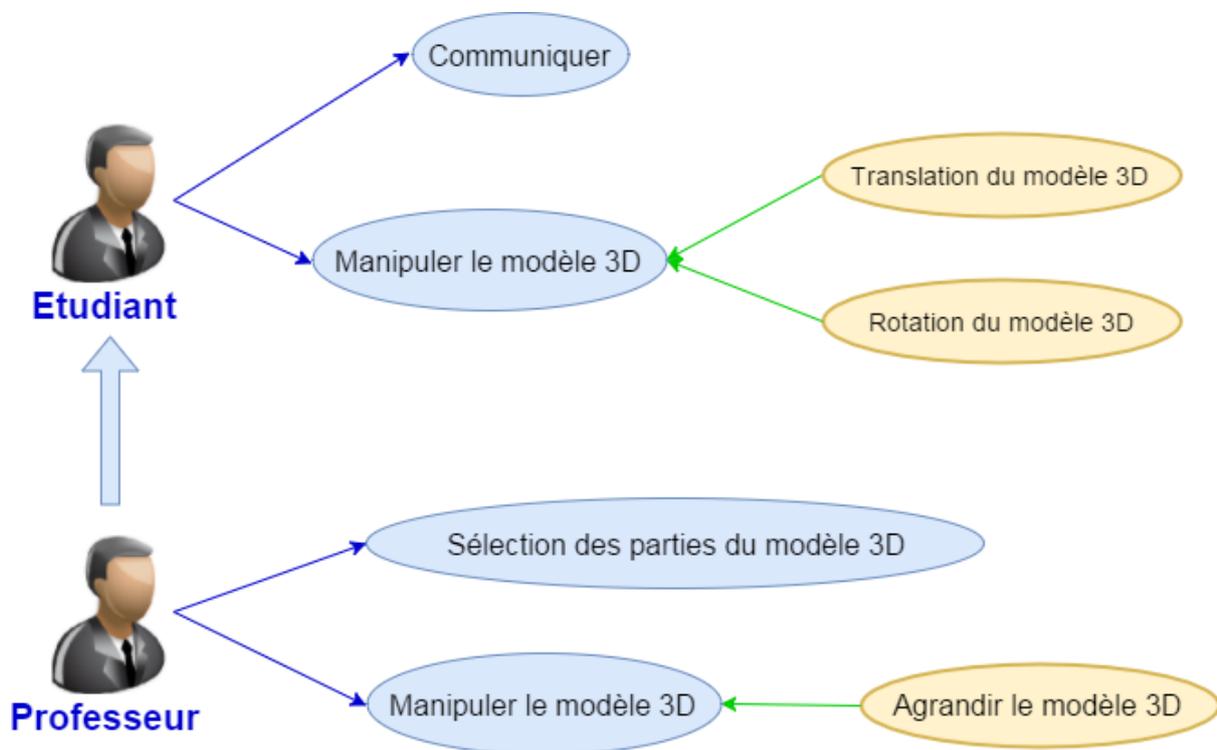


Figure III.4.1: schéma global du fonctionnement de chaque acteur.

IV -Conclusion

Notre conception d'un Système de co-manipulation à distance, dans un environnement augmenté collaboratif a été faite après une étude approfondie sur les applications existantes dans les environnements augmentés collaboratifs, ainsi que différents tests, nous avons opté au final pour une architecture Client/Serveur (Host), et les protocoles RPC et UDP, qui nous ont aidés à la résolution des problématiques liées à la latence et la synchronisation en plus de notre choix minutieux pour la distribution de données, et leurs traitements qui nous garantissent de meilleures performances et une bonne consistance.

Le domaine d'apprentissage médical utilise des moyens archaïques pour transmettre l'information aux étudiants, nous avons proposé alors de moderniser ce système en utilisant un environnement augmenté collaboratif ainsi l'enseignant ne devrait plus se soucier de la clarté du modèle d'anatomie (Cœur ou Cerveaux), vu que chaque étudiant pourra voir ce modèle en 3D et le faire pivoter en temps réel, ainsi l'enseignant peut faire un zoom sur des parties précises du corps ou de l'organe s'il le juge nécessaire.

Chapitre 4 : Réalisation d'une application EAD médicale

I.Introduction :

Dans ce dernier chapitre nous allons parler du système augmenté collaboratif qu'on a réalisé, ce type de système aide à rendre la vie moins difficile dans certains domaines tels que la mécanique ou bien la médecine.

II.Outils :

Les protocoles vus dans les chapitres précédents doivent avoir des outils permettant de les exploiter efficacement, et également des softwares qui supportent la combinaison de ces différentes technologies.

Le choix d'un langage de programmation orienté objet pertinent qui nous offre de diverses possibilités est primordial afin d'arriver à réaliser une application qui répond à nos problématiques. Dans ce qui suit, nous présentons les outils que nous avons utilisés pour l'implémentation de notre solution.

II.1-Plugin tiers (Artool kit, Vuforia) :

- **Artoolkit [11]** : ARToolkit est un outil open-source dédié à la réalité augmentée, il a été développé en 1999 dans l'Université d'Osaka par le docteur Hizokazu Kato..

C'est un système de marqueurs qui utilise des algorithmes qui détectent et identifient le ou les marqueurs utilisés dans notre environnement réel.

C'est un système qui utilise des algorithmes qui détectent et identifient le ou les marqueurs utilisés dans notre environnement réel.

Le principe de fonctionnement : l'algorithme de détection capture chaque image du flux vidéo et essaye d'appliquer des opérations bien définies (qu'on détaillera plus tard) dans le but de récupérer un des marqueurs définis, si ce dernier est présent dans la scène, cette procédure se déroule comme suit :

- La caméra capture l'image du flux vidéo.
- La binarisation de l'image par le programme en identifiant tout les cadres noirs.
- Si un cadre est trouvé, le programme récupère la position et l'orientation par rapport à la caméra en appliquant un ensemble de méthodes mathématiques.
- Une analyse du modèle qui se trouve à l'intérieur du cadre est alors effectuée afin de pouvoir comparer le modèle avec le pattern (marqueur) source qu'on a inclus dans le programme.
- Le programme associe une augmentation virtuelle à ce modèle.

Chapitre 4 : Réalisation d'une application EAD médicale

- Ajuster cette augmentation par rapport aux marqueurs ainsi qu'à l'image capturée et mettre à jour la position l'orientation en temps réel. [Annexe 2]

- **Vuforia** [14]: est un SDK (un kit de développement de logiciels) destiné à la réalité augmentée, il utilise la vision par ordinateur pour détecter les marqueurs et leurs associe différents objets 3D, Vuforia est un produit payant et nécessite une licence spécifique au besoin de l'utilisateur.

II.2- Unity3D & C# [12]:

Unity est un logiciel de développement créé par Unity Technologies, l'une de ses utilisation principale est orienté moteur de jeu, car il dispose de plusieurs plug-in près-configurés (évitant aux développeur de tout reprogrammer à zéro) comme le moteur physique (PhysX) en temps réel développé par Nvidia, la simulation des forces et des gravités devient alors un jeu d'enfant, Unity peut aussi être utilisé pour la programmation orientée agents.

Unity permet aussi la création de site web grâce à sa caractéristique principale de multiplateforme, ainsi nous pouvons avec un seul clique exporter vers la plateforme HTML5 et bien d'autres :



Figure II.2 Unity-Multi plateforme [13]

Donc on peut tout ignorer de l'architecture de ces plateformes, cela ne nous empêchera pas de leurs adapter des applications fonctionnel, créés avec l'un des deux langages qu'accepte Unity, le Javascript et le C#, l'utilisation de ce dernier est plus recommandé car le C# est de base un langage orienté objets et la communauté grandissante d'Unity utilise majoritairement ce dernier, il est donc plus facile de trouver des tutoriels ou aides sur les forums et sites web.

De plus le langage C# offre un type de variable particulier appelé decimal de 128 bits idéale pour les calculs financiers et les calculs imposants ainsi nous avons put aller au bout de notre application citée en haut, et elle nous permet d'avoir des résultats détaillés avec des données très grandes.

Les dernières mises à jour d'Unity ont facilité l'immersion des développeurs dans le monde de la réalité virtuelle et la réalité augmentée si on ajoute à cela les plugin réseau comme Photon ou Unet, cela crée un environnement de travail idéal et efficace pour le développement d'application dans les environnements collaboratifs.

II.3- Unet [12] :

Les développeurs d'Unity ont développé Unity networking qui succède à Raknet qui était un moteur de réseau présenté sous forme d'une bibliothèque C++ et C#.

Networking fonctionne qu'avec Unity, c'est un système basé sur l'architecture Client/Serveur, utilisable avec un serveur dédié ou Host.

L'avantage avec Unet c'est qu'il offre la possibilité de travailler hors ligne, ainsi qu'une programmation plus simple et avantageuse par des variables communes dans le réseau par exemple, qu'on appelle variables Synchronisées.

Durant les dernières mises à jour, Unet a acquis des outils adéquats pour tester les performances réseaux, et cela en simulant de la latence (Simulation d'une mauvaise connexion chez un utilisateur), et des systèmes adaptés au besoin d'utilisateurs, si par exemple des utilisateurs veulent développer une application multiutilisateur avec Unity ils peuvent utiliser le NetworkManager ou l'API de haut niveau tant dit que les utilisateurs qui veulent construire une infrastructure réseau ou des applications multiutilisateurs avancées doivent s'orienter vers l'API NetworkTransport.

II.4-Photon :

La société allemande Exit game a développé l'SDK photon destiné à offrir aux développeurs de différentes plateformes Android, Windows, iOS, Unity... une interface de développement conviviale et adaptée à leurs besoins, basée sur le principe de Room, les utilisateurs sont donc classés dans des chambres réseaux dans lesquelles ils pourront communiquer et interagir, cette notion de Room permet d'ordonner les utilisateurs ainsi si deux utilisateurs appartenant à des rooms différentes ne peuvent communiquer à moins que l'un d'eux rejoigne la room de l'autre, disposant d'un seul master client par room (généralement le premier qui se connecte et donc qui crée la room), mais ce dernier peut être remplacé par un autre utilisateur si il se déconnecte du réseau.

Chapitre 4 : Réalisation d'une application EAD médicale

Photon offre alors un serveur dédié avec une large bande passante, pouvant contenir jusqu'à 20 personnes pour la version gratuite, et le prix varie selon la capacité du serveur.

PUN, qui désigne PhotonUnityNetwork est spécialement destiné à Unity, sa programmation simple permet au plus novice dans PUN de se retrouver, à condition de maîtriser les bases de la programmation réseaux (UDP, TCP, HTTP, WebSocket, RPC).

III-Description du problème:

Nous avons interviewé des étudiants et professeurs en médecine dans une université algérienne, nous avons noté la difficulté flagrante que rencontrent ces derniers dans l'apprentissage de l'anatomie, particulièrement le cœur et le cerveau qui représentent pour les nouveaux étudiants en médecine un semestre d'apprentissage, ce problème de compréhension selon certains étudiants et professeurs est dû au manque d'imagination de l'étudiant, car ce dernier doit être capable de voir et comprendre toutes les parties du cœur et même les veines sur des supports de cours ou la représentation est en 2D est certaines parties sont situées dans d'autres parties plus grandes et qui sont assez dures à percevoir dans leurs positionnements au sein d'un cœur humain, certes il existe des modèles artificiels réels, mais dans un amphithéâtre fournir ce modèle à tous les étudiants deviens très couteux.

Nous avons alors proposé une solution, qui est celle de changer les modèles 2D vers des modèles 3D que les étudiants peuvent voir, inspecter, déplacer, tourner et manipuler à leurs convenance, les exigences matérielles sont simples, un PC portable ou un téléphone et un marqueur (une feuille à imprimé) .

Les étudiants pourront se connecter dans un réseau local avec le wifi de l'université par exemple avec le professeur qui sera dans la même salle, ou un cours à distance sera effectué ou le professeur serait dans un site distant avec certains étudiants avec lui, et d'autres dans une autre université, il pourra alors communiquer avec eux via un chat vocal et pourra se faire aider par notre application qui lui offre de nombreuses fonctionnalités de manipulation de ces modèles 3D.

IV-Cas d'utilisation :

Notre système augmenté collaboratif va être utilisé dans les universités pour des cours de médecines précisément en cours d'anatomie.

Il existe deux types d'utilisateurs : Un Host (Client/Serveur) et des clients.

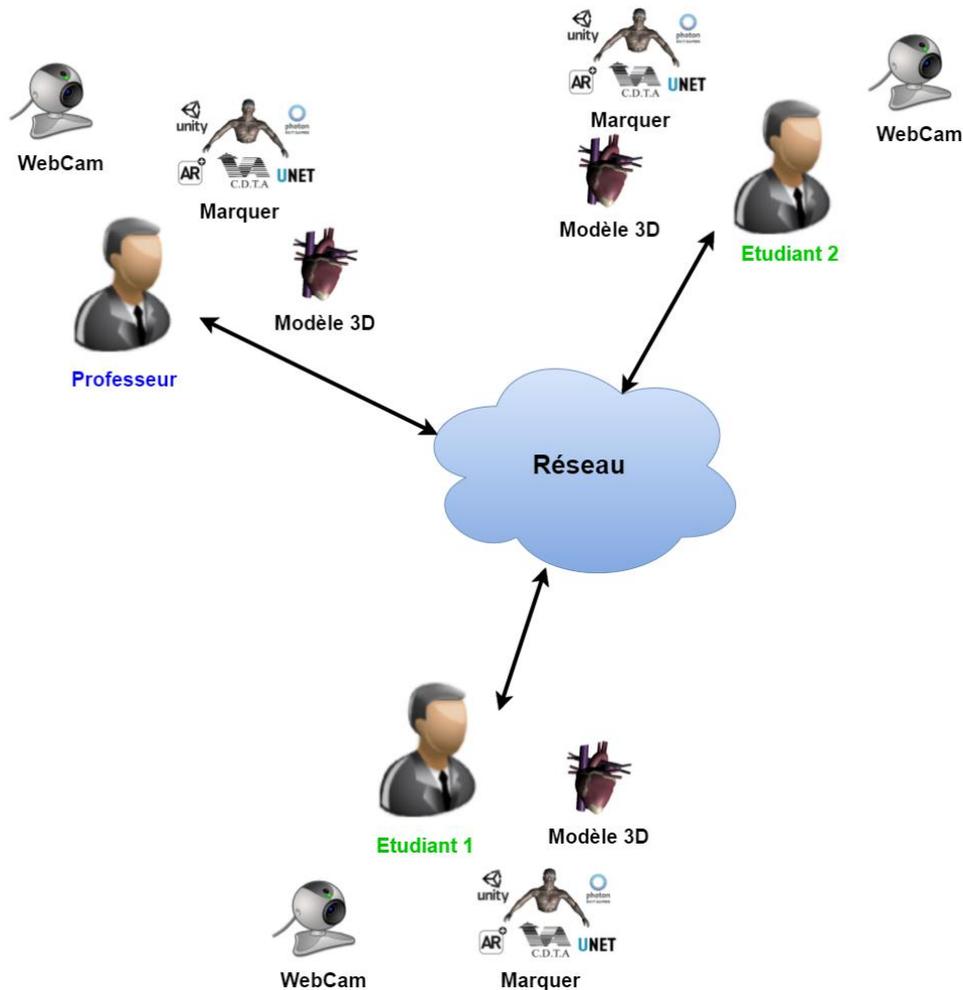


Figure IV: Architecture et configuration générale du Système

IV.1- Scénario des opérations faites par les auteurs:

Le professeur : C'est le Host, le point central du système, il ouvre une connexion initiale et attends que les étudiants se connectent.

Les étudiants : C'est des clients différenciés par leur nom et leur couleur (aléatoire), ils ne peuvent se connecter que si le professeur a déjà lancé son application.

Le Professeur :

- 1- Lance L'application.
- 2- Il saisit son Nom dans la case suivante : Nom
- 3- Clique sur le bouton **Professeur** Pour se connecter autant que host.
- 4- Il accède à la scène réelle, il voit le corps (plus exactement un torse humain est visualisé si le marquer est visible), il peut faire les actions suivantes :
 - a. Sélectionner le Cœur ou le cerveau en cliquant dessus (Click gauche de la souris), tous les utilisateurs le verront aussi.

- b. Il peut choisir les parties du Cœur en appuyant sur « 1, 2, 3, 4, 5, 6, 7 », tous les utilisateurs le verront aussi.
- c. S'il passe le curseur sur un organe ou ces parties une info Bull (notification) s'affiche (en local)
- d. Il peut faire une rotation de l'organe en appuyant sur la touche « R », tous les utilisateurs verront cette rotation tout en ayant le nom du professeur affiché pour indiquer que c'est ce dernier qui interagit avec l'objet 3D.
- e. Il peut faire la translation de l'organe en appuyant sur les touches directionnelles du clavier et tous les utilisateurs verront le résultat de la translation
- f. Il peut parler avec tous les utilisateurs en appuyant sur la touche "V".

L'Étudiant :

- 1- Lance L'application.
- 2- Il saisit son Nom dans la case suivante :  Nom
- 3- Il saisit aussi l'adresse IP du serveur sur la case suivante  L'adresse Ip du serveur
- 4- Clique sur le bouton  Pour se connecter tant que Client .
- 5- Il accède à la scène réelle, il voit le corps (si le marquer est visible) , il peut faire les actions suivantes :
 - a. S'il passe le curseur sur un organe ou ses parties une info Bull (notification) s'affiche (en local)
 - b. Il peut faire une rotation de l'organe en appuyant sur la touche « R », tous les utilisateurs verront cette rotation tout en ayant le nom de l'étudiant affiché pour indiquer que c'est ce dernier qui interagit avec l'objet 3D
 - c. Il peut faire la translation de l'organe en appuyant sur les touches directionnelles du clavier et tous les utilisateurs verront le résultat de la translation
 - d. Il peut parler avec tous les utilisateurs en appuyant sur la touche V.

L'Étudiant (en retard) :

- 1- Lance L'application.
- 2- Il saisit son Nom dans la case suivante :  Nom
- 3- Il saisit aussi l'adresse IP du serveur sur la case suivante  L'adresse Ip du serveur
- 4- Clique sur le bouton  Pour se connecter autant que Client.

Chapitre 4 : Réalisation d'une application EAD médicale

- 5- Dès qu'il accède à la scène, le système demande au serveur de lui transmettre la mise à jour de la session.
- 6- Après avoir fait la mise à jour, il pourra faire toutes les actions autorisées pour le client.
- 7- Il pourra continuer le cours avec ces camarades comme tout étudiant participant.

IV.2-Scénario d'utilisation :

Pour le cas d'utilisation de notre système les figures suivantes le résume :

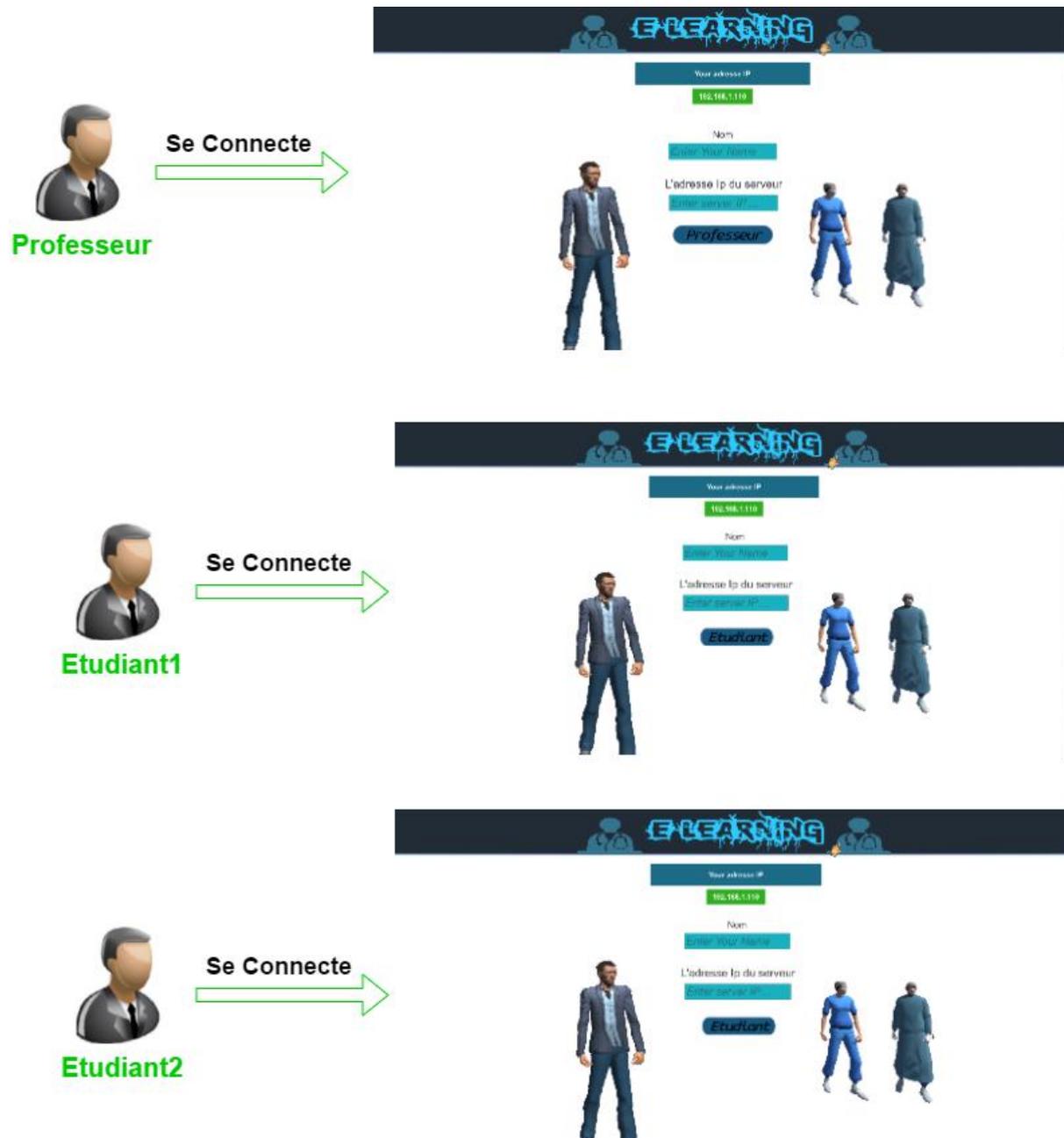


Figure IV.2 : Interface pour chaque utilisateur

Chapitre 4 : Réalisation d'une application EAD médicale

Ensuite après la connexion de chaque acteur, on montre ici des captures d'écrans tous en se basant sur le scénario suivant :

- 1- Tout d'abord le professeur se connecte et tout les étudiants se connecteront à lui
- 2- Le professeur sélectionne un organe et l'explique
- 3- Un étudiant a mal compris une partie, il pose une question au professeur
- 4- Le professeur lui demande de lui montrer cette partie en faisant tourner l'organe
- 5- L'étudiant fait tourner l'organe et lui montre la partie concernée via le chat vocal
- 6- Le professeur sélectionne la partie et puis il la réexplique.

On trouve ci-dessous les captures d'écrans que nous avons pris en suivant le scénario :

- 1- Le Professeur se connecte. En passant la souris sur le cœur, il y a une petite description qui apparaît en haut



Figure 1 : l'écran du professeur avec le tors.

- 2- Le professeur sélectionne le cœur en cliquant dessus et son nom apparaît en bas à gauche indiquant que le professeur a fait une interaction avec l'objet 3D, puis il présente le cours

Chapitre 4 : Réalisation d'une application EAD médicale

en exploitant le chat vocal mis à sa disposition directement dans l'application :



Figure 2 : Sélection du cœur par le professeur

3- Tous les étudiants voient cette sélection et le nom de celui qui la fait :



Figure 3 : La sélection du cœur chez tous les étudiants

4- Le professeur fait une translation au cœur vers la gauche comme suite :



Figure 4: Translation du cœur par le Professeur

5- Tous les autres voient la translation faite par le professeur :



Figure 5 : Translation faite chez tous les étudiants

6- Un étudiant veut poser une question, il parle via le chat vocal :



Chapitre 4 : Réalisation d'une application EAD médicale

Figure 6: Etudiant pose la question au professeur via le chat vocal

- 7- Le professeur lui demande de faire tourner le cœur pour voir la partie non compris par l'étudiant (toujours en exploitant le chat vocal)
- 8- L'étudiant fait tourner le cœur et son nom apparaît en bas à gauche :

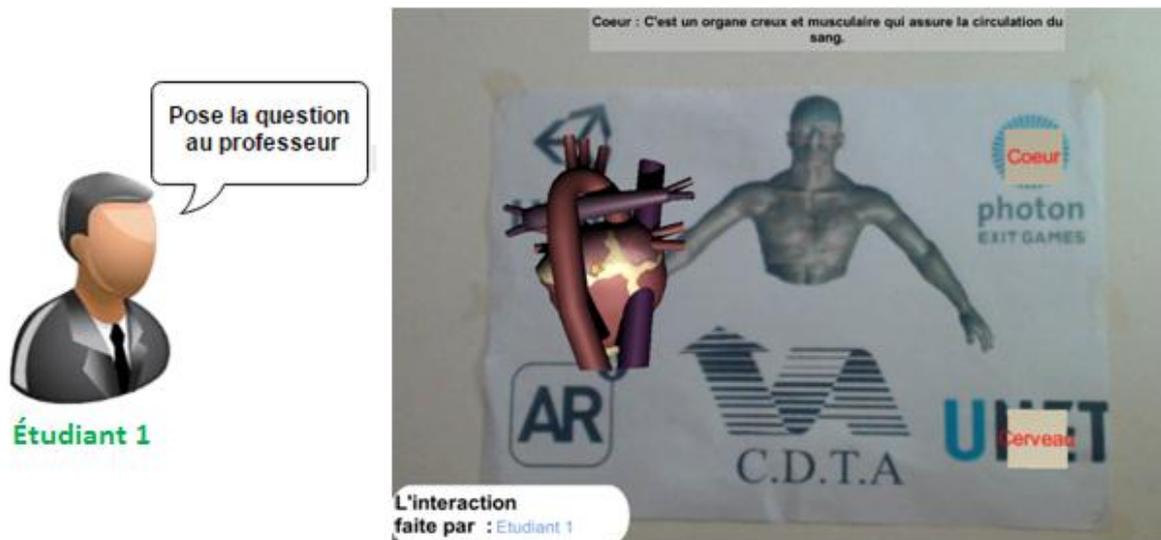


Figure 8 : Rotation faite par l'étudiant 1

- 9- Tous les utilisateurs y compris le professeur voient la rotation :



Figure 9 : Rotation faite chez tous les utilisateurs

10- Le professeur sélectionne la partie non compris par l'étudiant 1 :



Figure 10 : Le professeur sélectionne la partie du cœur concerné

1- Le professeur explique la partie non comprise et tous les étudiants écoutent et peuvent voir les annotations et la petite description en local (en passant la souris sur le modèle 3D)

IV.3-Scénario d'utilisation récapitulatif :

Le professeur commence à expliquer le cours sur la partie du cœur donc tous les étudiants voient la même chose qui est le cœur comme nous le montre la figure suivante :

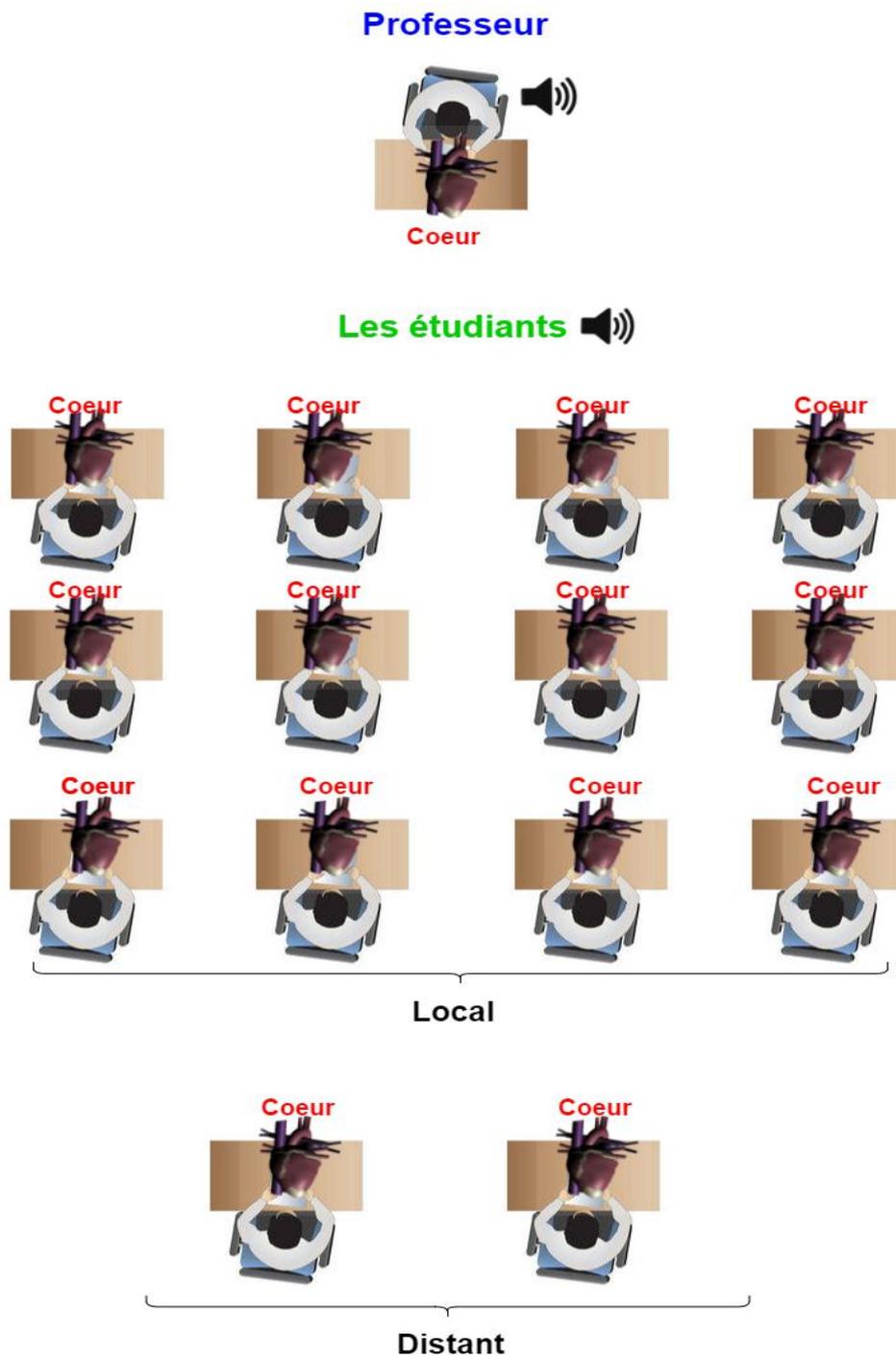


Figure IV.3.1 : Scénario début du cours

Après l'explication le Professeur décide de déplacer le Cœur à droite et donc tous les autres voient la translation effectuée comme nous le montre la figure suivante :

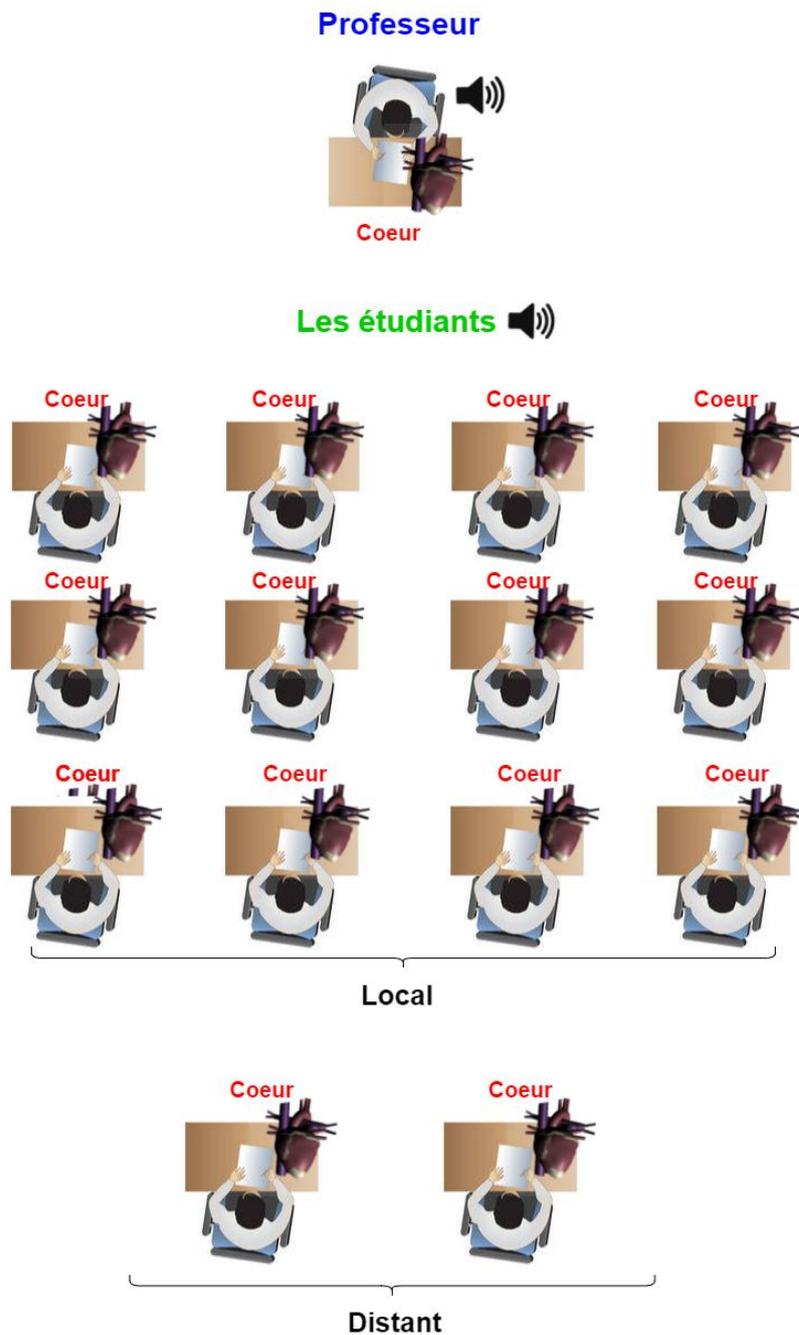


Figure IV.3.2 : Scénario Translation du Cœur

V -Composant de notre système :

On explique dans cette partie les composants de notre système qui sont les suivants :

-Model 3D

-Interaction Souris et Clavier

- et enfin une composante "Synchronisation" qui permet l'aspect collaboratif

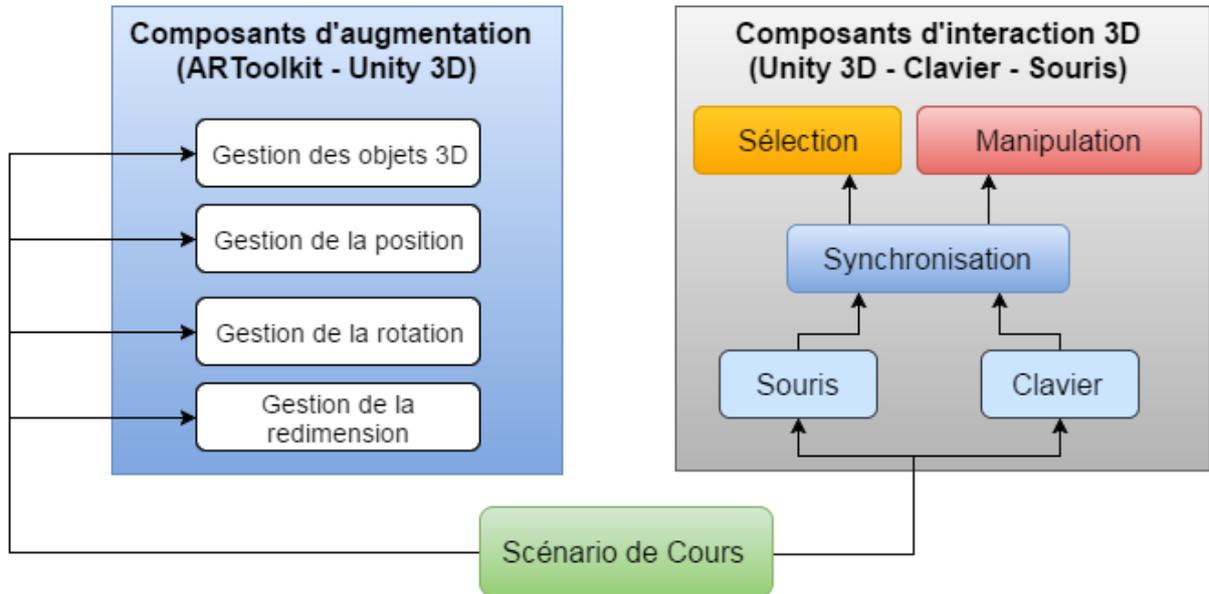


Figure V : Composants de notre système EAD

V.1-Modèles 3D :

Notre modèle 3D utilisé se compose comme suite :

Nom de famille	Ordre d'apparition	Composants
Corps (torse)	1	Cœur
		Cerveau
Cœur	2	Aorte et Artère Pulmonaire
		Ventricule gauche et droit
		Veine Cave supérieur et inférieur
		Veine Pulmonaire gauche et droite
Cerveau	3	Lobe frontal Lobe pariétal Lobe occipital Lobe temporal Cervelet Troc cérébral Stem Pitua

Tableau V.1 Organisation des composants du modèle 3D

Chapitre 4 : Réalisation d'une application EAD médicale

V.2- Interactions :

Pour les interactions Clavier et souris ils sont représentés dans le tableau suivant :

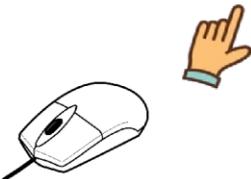
Touche clavier et souris	Description	L'intérêt de la touche
	Translation du Corps, Cœur, Cerveau	Interaction 3D manipulation de l'objet 3D
	Rotation du Corps, Cœur, Cerveau	Interaction 3D manipulation de l'objet 3D
	Revenir à l'état initial du modèle 3D	Interaction 3D manipulation de l'objet 3D
	Reconstitution du Cœur ou Cerveau après la décomposition	Interaction 3D, manipulation de l'objet 3D
	Sélectionner et agrandir le Cœur ou le Cerveau Et agrandir les composants du cœur et du cerveau	Interaction 3D manipulation de l'objet 3D
	l'affichage d'info bulle sur la partie du corps parcourus par le curseur de la souris	Augmentation par l'affichage d'une notification

Tableau V.2 Description des touches clavier et souris.

VI. Conclusion

Dans l'apprentissage médical, il est capital de bien connaître l'anatomie humaine pour aider les étudiants à bien assimiler les cours parfois très compliqués d'anatomie, nous avons conçu un système de réalité augmentée collaboratif qui leur est destiné (E-Learning). Ce système leur permet d'améliorer leur compréhension considérablement en leur offrant ainsi une formation complète dans le domaine médical, cela permet d'éradiquer les contraintes pouvant exister liées au manque du matériel médical pour l'apprentissage (figurine, etc.), ce matériel étant souvent bien trop cher ainsi que d'éviter des problèmes liés à une connexion internet vu que le système marche aussi bien en local qu'à travers internet, et permet de mixer des utilisateurs en local et via internet en même temps, ce qui permet à certains étudiants de suivre le cours à distance ou encore à l'enseignant de dispenser son cours sur un autre site sans se soucier d'avoir une quelconque incohérence, car notre système assure une bonne synchronisation et des interactions temps réel pour l'ensemble des utilisateurs distants et locaux.

Conclusion & perspective

Les environnements augmentés sont très utiles comme support à des sciences telles que la médecine, la mécanique, etc. Les applications collaboratives en réalité augmentée permettent d'appréhender la difficulté de certaines tâches difficiles à réaliser par une seule personne et apportent donc aux utilisateurs un moyen de partager certaines informations et d'interagir avec le monde 3D pour atteindre un même objectif.

Dans ce travail, nous nous sommes intéressés à cette thématique et nous avons pu détailler la question de la consistance du système dans les environnements augmentés collaboratifs, cette question est importante, car des utilisateurs qui ne voient pas le même état des objets 3D en temps réel, ne peuvent pas les manipuler d'une manière efficace et des ambiguïtés peuvent alors apparaître empêchant le bon déroulement des tâches collaboratives.

Notre système doit donc garantir une bonne synchronisation pour maintenir au maximum la consistance du système, mais l'amélioration de cette synchronisation peut parfois entraîner une augmentation de latence, alors nous avons de par notre conception créé un système qui nous garantis une bonne synchronisation en utilisant des mises à jour pertinentes quand c'est nécessaire et une latence réduite en limitant les paquets qui circulent dans le réseau ainsi que l'utilisation de protocoles adéquats permettant de gagner en latence. Nous avons appliqué notre système dans un domaine très intéressant qui est l'apprentissage médical, plus précisément dans une application de E-learning facilitant la compréhension des étudiants ainsi que la dispense du cours par le professeur. Notre conception a permis d'éradiquer le problème lié à la distance géographique grâce aux deux fonctions de communication que nous avons créée qui permettent d'accepter des utilisateurs qui sont en local et d'autres à travers internet et qui partagent toujours les mêmes événements au même moment avec une communication inter utilisateurs assurés par notre module de chat vocal. Notre proposition peut permettre alors d'économiser beaucoup d'argent aux facultés de médecine si le système est adopté, car nous avons simplifié le déploiement de l'application, qui ne nécessite pas de matériel coûteux. Elle est aussi adaptée à plusieurs plateformes, ce qui favorise son insertion en pratique.

Nous pouvons penser à certaines extensions pour ce travail, d'autres organes 3D peuvent être abordé ainsi l'application pourra couvrir toutes les spécialités en médecine. Nous pourrons aussi déployer cette solution sans l'usage des marqueurs (version de réalité augmentée MarkerLess) afin de rendre l'environnement plus naturel. Enfin, nous pouvons dire que ce projet nous a permis de renforcer nos connaissances théoriques et pratiques déjà acquises et également d'en développer d'autres connaissances nouvelles dans un domaine très exploré ces dernières années à savoir celui de la réalité augmentée.

Annexe

Cette annexe permet de détailler certaines parties importantes, mais qui ne figurent pas dans les chapitres présentés

1-Fonctionnement détaillé ArToolkit :

Voici les étapes détaillées suivies par l'algorithme, nous donnons également les points à suivre pour avoir un marker personnalisé.

- Binarisation :

L'opération initiale qui effectue l'identification c'est la binarisation c'est la représentation par des pixels noirs et blancs dans l'en-tête de notre programme on trouve un threshold (un seuil ou une frontière) c'est lui qui s'occupe de la conversion du flux en gris, et tout pixels inférieur au threshold sera en noir et vice versa.

- Normalisation :

Le carré noir de notre marqueur va nous aider grâce à ses sommets de normaliser le motif et donc de l'identifier, cette normalisation s'effectue en appliquant des fonctions mathématique aux pixels .

- Identification du marqueur :

Une fois le motif redressé en compare avec notre marker source ,qui es sous forme d'un fichier texte ou chaque pixel prend une valeurs entre 0 à 255 (en niveaux de gris) le programme compare alors nos deux marqueurs dans 4 sens différents et prend le plus haut ensuite en applique l'augmentation choisie.

- ❖ Création d'un pattern :

Il s'avère que la reconnaissance d'un texte est moins précise que celle d'un motif qui serait aligné au centre, nous avons donc incrusté le texte en bas afin que le programme n'ait aucune difficulté à repérer le sens du motif.

1. Crée un pattern grâce à un logiciel tel que Photoshop, paint ou dessin à la main (et le prendre en photo pour ce dernier).
2. L'imprimer.

Annexe

3. Lancer mk_patt et mettre le modèle choisi a la vu de la caméra pour faire la Binarisation et la normalisation puis l'identificateur du marqueur :

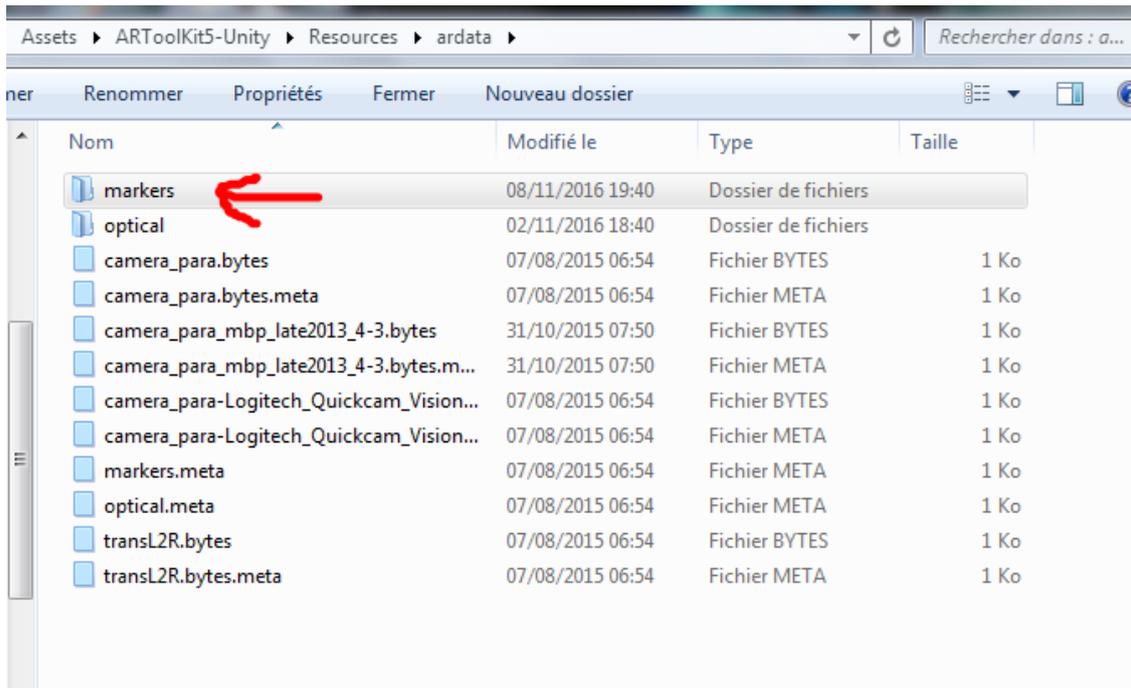


4. On donne un nom a notre fichier et puis en sauvegarde :

```
C:\Users\Sofianos\Desktop\AR\ARUnity5\bin\mk_patt.exe
Enter camera parameter filename (default: 'Data/camera_para.dat'):
Using default video config.
bmpBufferSize = 921600, width = 640, height = 480
Image size (x,y) = (640,480)
-----
SIZE = 640, 480
Distortion factor: k1=0.1147807688, k2=-0.5208189487, p1=-0.0002069871, p2=-0.00
40593124
fx=674.171631, fy=633.898087, x0=318.297791, y0=237.900467, s=
0.993923
678.29391 0.00000 318.29779 0.00000
0.00000 637.77411 237.90047 0.00000
0.00000 0.00000 1.00000 0.00000
-----
Enter filename:
```

Annexe

5. Puis mettre l'extension du fichier généré en .txt et copie le fichier dans le dossier par défaut d'Artoolkit: ARToolkit5-Unity\Resources\ardata\markers



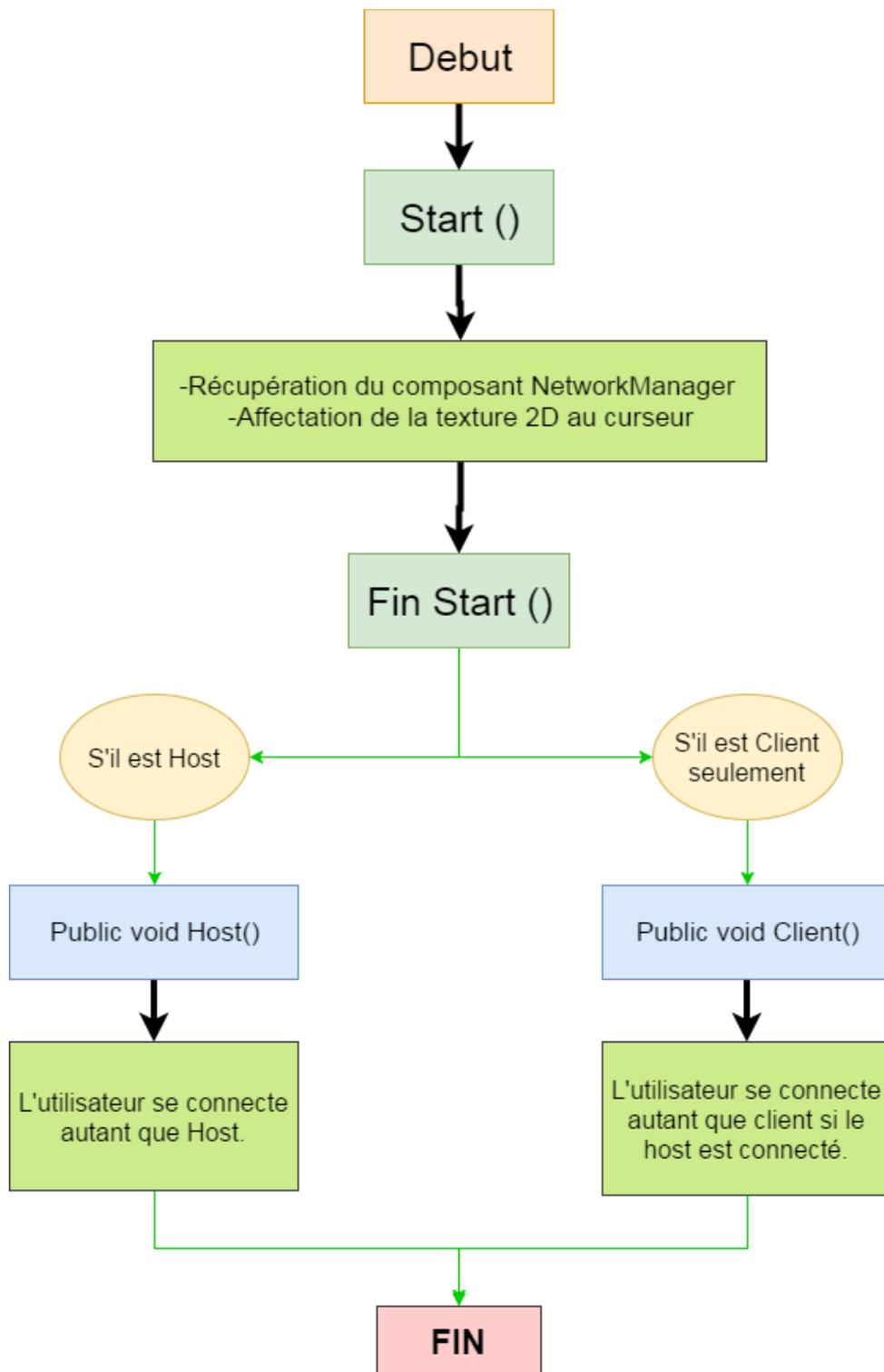
6. Au final l'intégrer a Unity 3D .

2- Les organigrammes :

Nous présentons ici les détails des organigrammes sur lesquels notre travail se base pour la partie réalisation

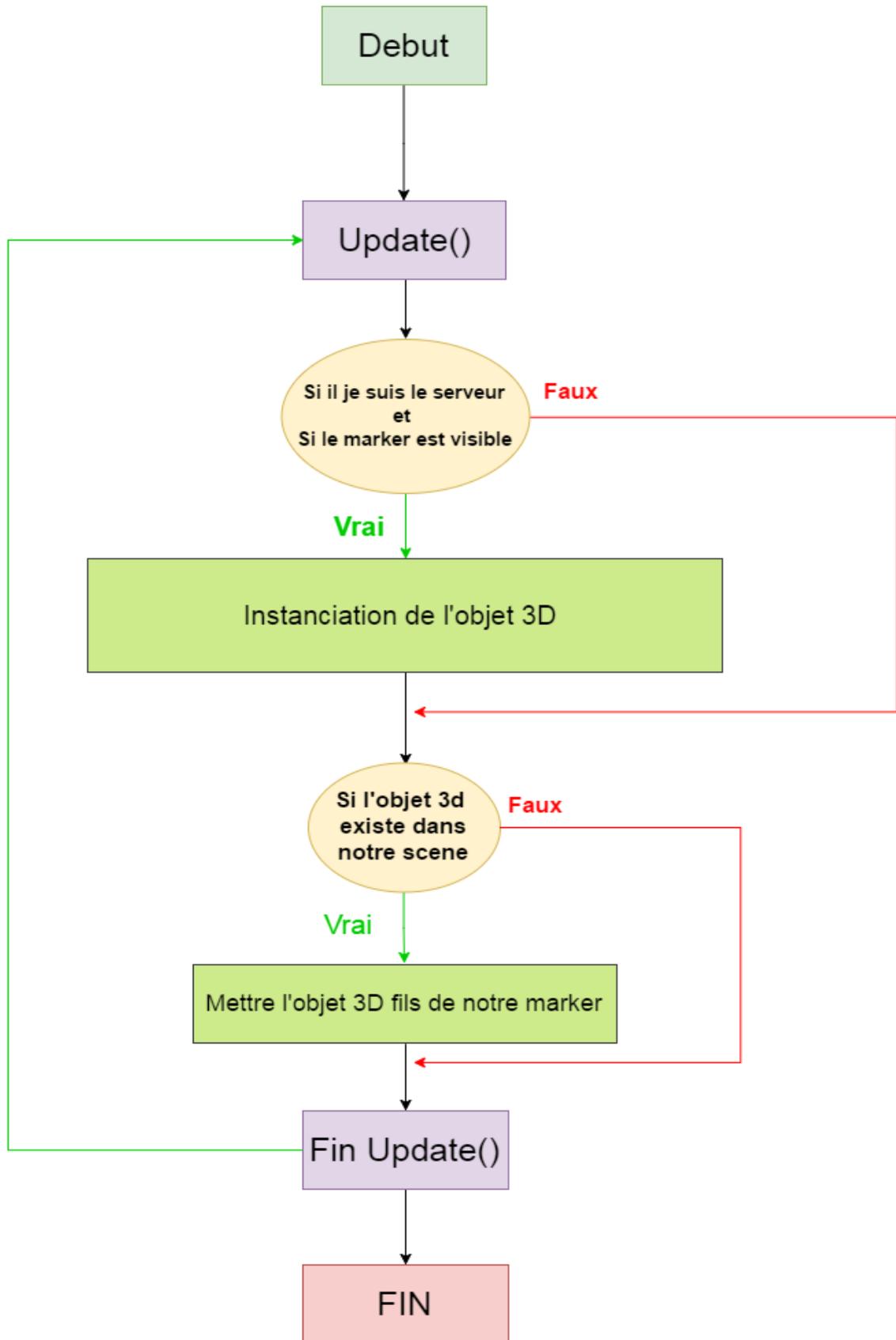
Annexe

Organigramme NetworkChoix : Cette classe s'occupe de la connexion autant que host ou client (professeur ou étudiant).



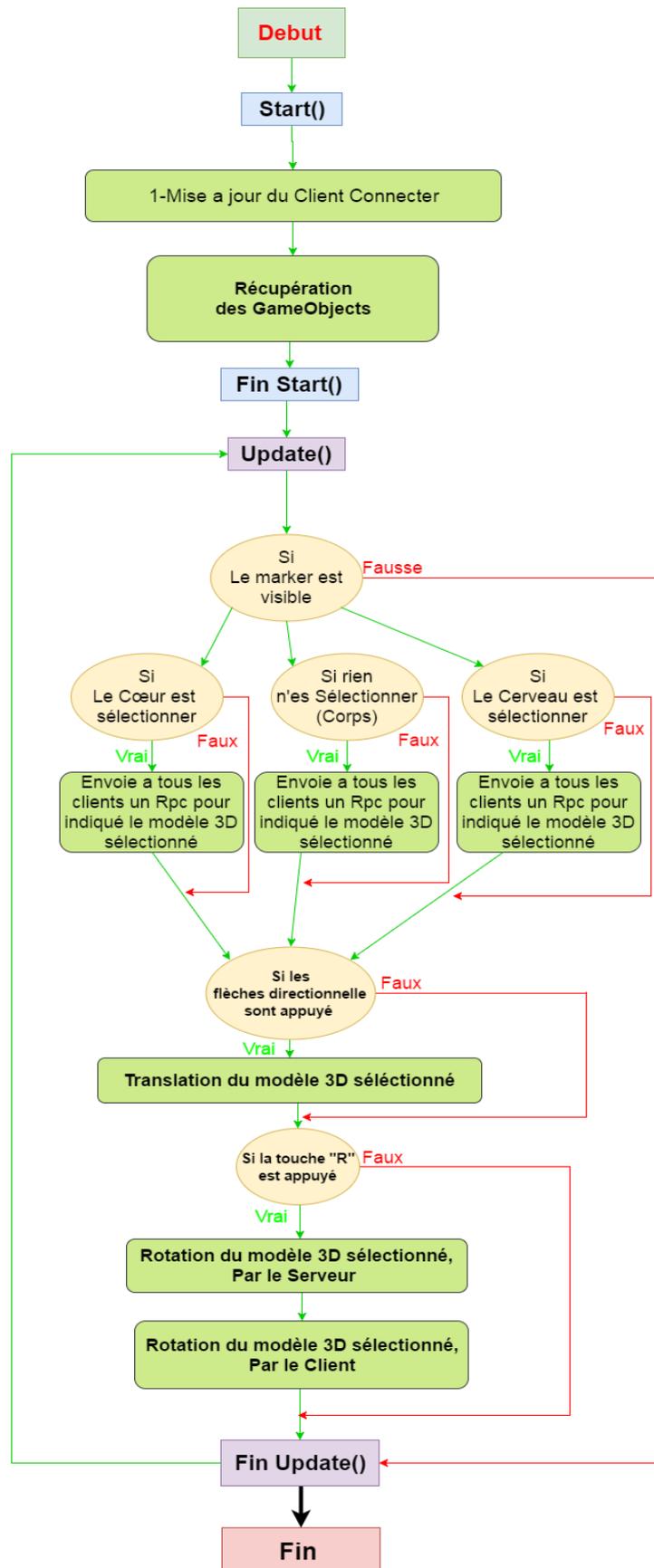
Organigramme NetwrokChoix

Organigramme Spawn : Cette classe s'occupe de l'instanciation des objets 3D.



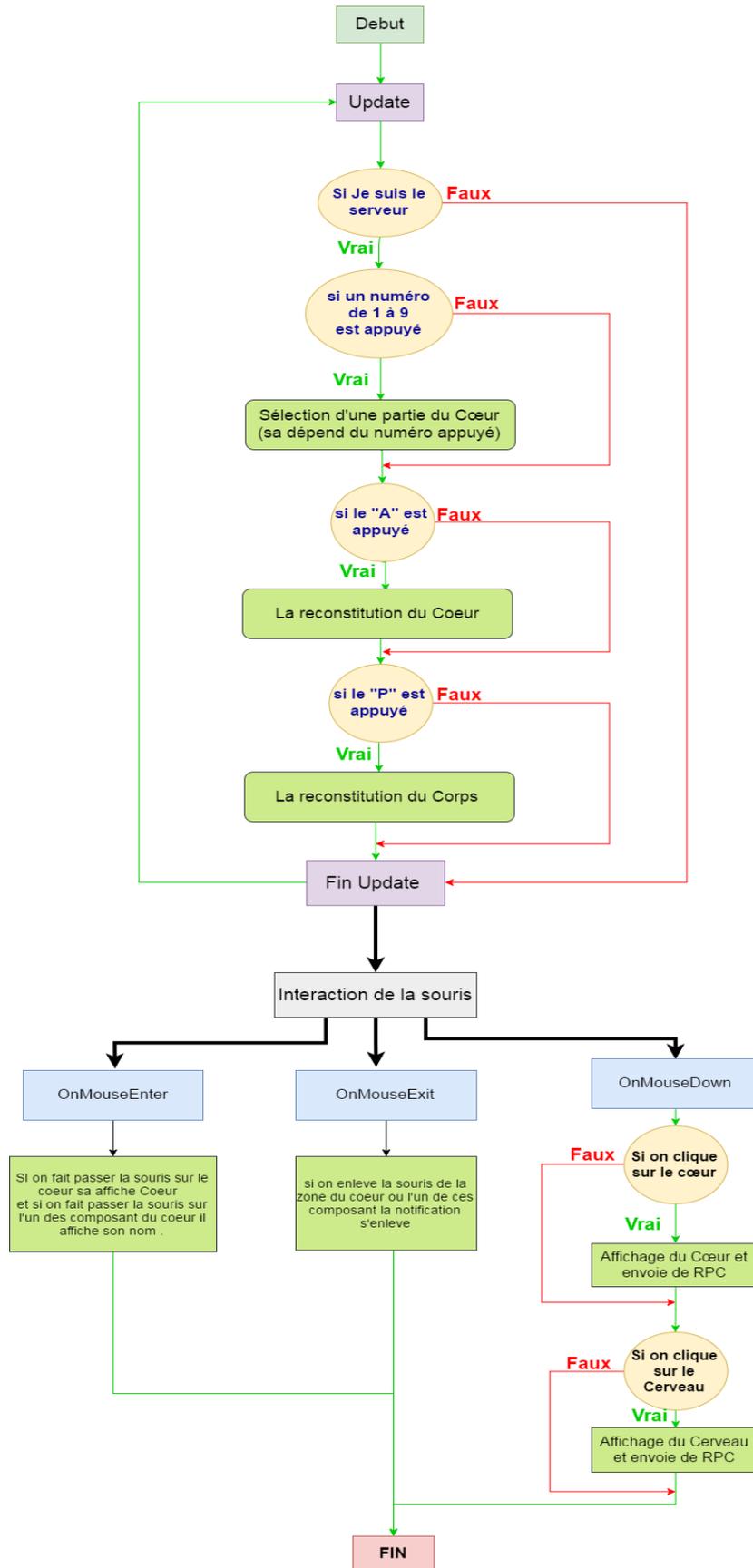
Organigramme spawn

Organigramme Controle : Cette classe s'occupe de la mise à jour et de la synchronisation entre les utilisateurs



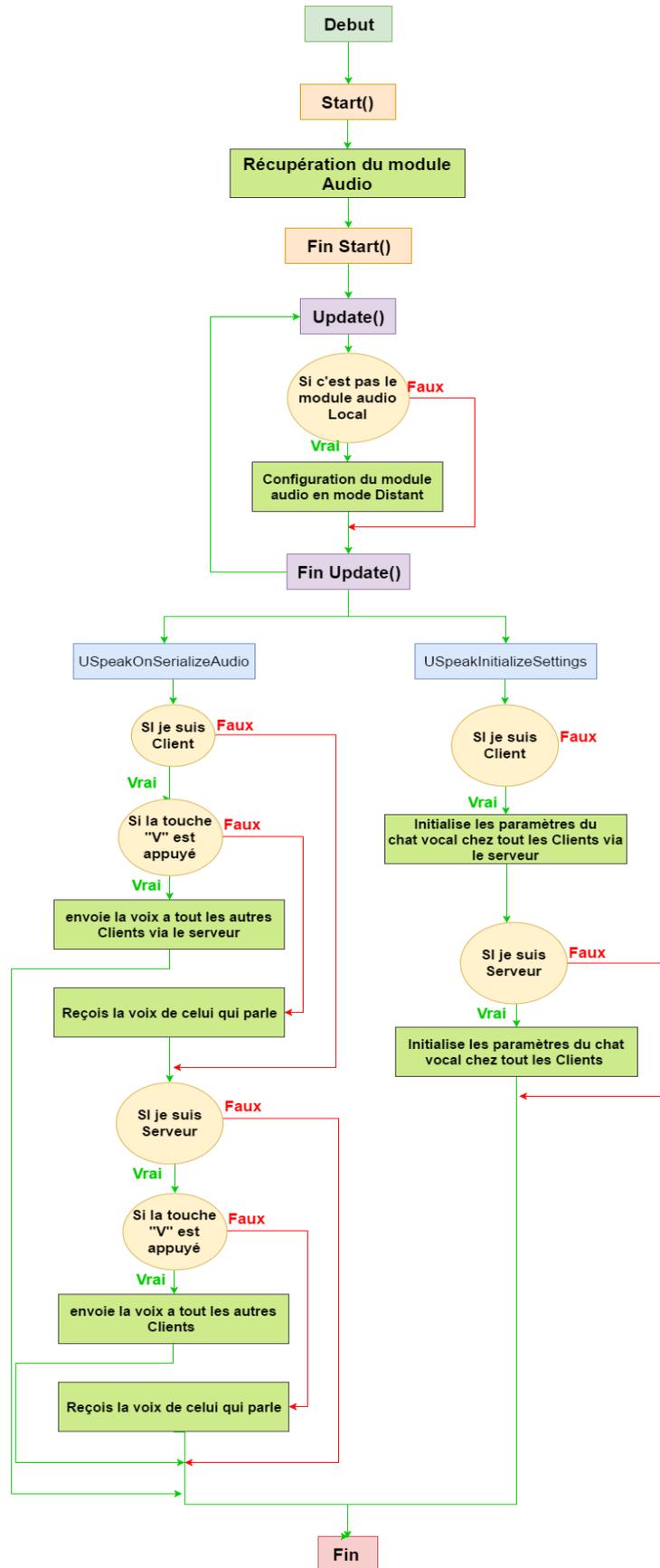
Organigramme Contrôle

Organigramme MouseInteraction2 : cette class s'occupe des interactions clavier et souris



Organigramme MouseInteraction2

Organigramme LocalUSpeakSender: cette classe s'occupe du chat vocal



Organigramme LocalUSpeakSender

3-Extrapolation et interpolation :

3.1-Extrapolation

L'extrapolation prédit la position d'un objet 3D qui appartient à un autre utilisateur en se basant sur la vitesse afin de nous donner un effet de déplacement lisse (sans bug, ni saut);

Par exemple la mise à jour de la position de l'objet se fait tous les 0.5 seconde, et un événement se produit à 1,3 sec .

Notre programme n'a pas encore calculé la position à 1.5sec , mais il connaît la position de l'objet à 1.0 seconde, ainsi que sa vitesse angulaire et linéaire, il peut extrapoler en supposant que ces vitesses restent constantes. Ainsi il prend la position à 1.0 sec et lui rajoute les informations dont il dispose pour prédire la position de l'objet à 1,3 sec .

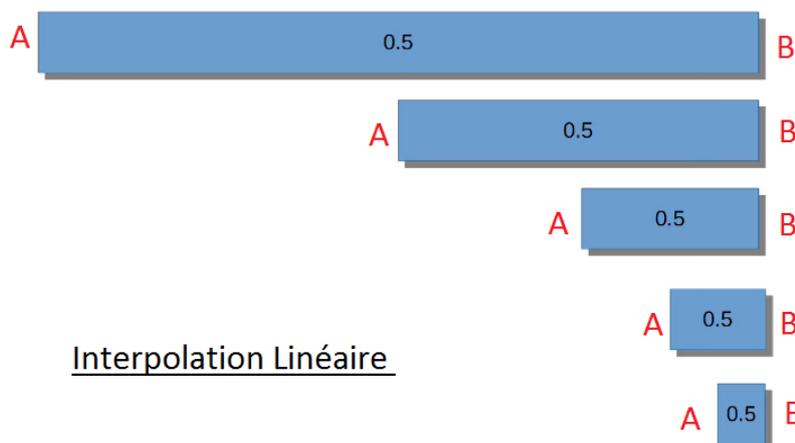
3.2 Interpolation :

L'interpolation permet d'avoir un affichage (Déplacement, Rotation...) fluide, son principe de fonctionnement est qu'en prenant deux anciennes positions (connues), et on pourra déplacer l'objet 3D d'un autre utilisateur entre eux en choisissant une valeur comprise entre ces deux positions, afin qu'elle devienne notre nouveau pas de déplacement et elle sera utilisée à chaque fois pour diminuer l'espace séparant ces deux positions.

Interpolation linéaire :

L'interpolation linéaire est utilisée pour déplacer un objet progressivement entre deux points connus, son principe de fonctionnement :

Si on souhaite déplacer un objet 3D d'un point A à un point B, au début notre objet 3D sera en position A. On choisit une valeur de déplacement par exemple 0.5 (50%), donc à chaque frame notre objet 3D se déplacera de la moitié (soit 50%) de la distance entre A et B, et à chaque frame on prend 50% de la nouvelle distance entre A et B ainsi de suite..., et on remarquera qu'on arrivera jamais vraiment à la destination B, mais on peut considérer qu'on est à une distance très proche de B quand on est arrivé à destination. En C# cette méthode s'appelle une fonction **Lerp** .



Interpolation sphérique :

L'Interpolation sphérique reprend la même définition qu'une interpolation linéaire, à la seule différence que la distance calculée entre les deux points ne se fait pas sur une distance linéaire, mais sur un cercle (si on travaille dans le 2D) ou une sphère (dans la 3D) .

En c# cette méthode s'appelle une fonction **Slerp** .

L'utilisation la plus fréquente de **Slerp** reste sur les rotations car ils se produisent sur des cercles, et non pas des lignes droites

Bibliographie

- [1] Azuma et al., «Tracking in Unprepared Environments»
Pour système de réalité augmentée, "Computers and Graphics,
Vol. 23, no. 6, décembre 1999, page. 787-793
- [2] Anders Henrysson, Mark Ollila et Mark Billinghurst
Réalité augmentée par téléphone portable
Technologies émergentes de la réalité augmentée: interfaces et conception, Idea Group Publishing, 2006, 90-109.
- [3] The Augmented Round Table Wolfgang Broll, Moritz Stoerring & Chiron Mottram Computer Vision and Media Technology, Aalborg University, Denmark, Collaborative Virtual and Augmented Environments Department, FIT, Sankt Augustin, Germany , Foster and Partners, London, UK. Published by IOS Press, (c) IFIP, 2003, pp. 1103-1104
- [4] Collaborative Augmented Reality for Outdoor Navigation and Information Browsing. Gerhard Reitmayr and Dieter Schmalstieg, Interactive Media Systems group, Vienna University of Technology. Publié en 2004
- [5] Mobile Collaborative Augmented Reality Gerhard Reitmayr and Dieter Schmalstieg Augmented Reality, 2001. Proceedings. IEEE and ACM International Symposium on.
- [6] Jourdain S., Forest J., Mouton C., Nouailhas B., Moniot G., Kolb F., Cha-bridon S., Simatic M., Abid Z. et Mallet L. (2008). « ShareX3D, a scientific 143 Bibliographie collaborative 3D viewer over HTTP ». Dans Proceedings of the ACM conference on 3D Web technology (Web3D), pages 35–41.
- [7] Dit Picard S. L., Degrande S., Gransart C. et Chaillou C. (2002). « VRML data sharing in the spin-3D CVE ». Dans Proceedings of the ACM conference on 3D Web technology (Web3D).
- [8] Delaney, J.R., Stoven, S., Uvell, H., Anderson, K.V., Engstrom, Y., Mlodzik, M. (2006). Cooperative control of Drosophila immune responses by the JNK and NF-kappaB signaling pathways. EMBO J. 25(13): 3068--3077. (Export to RIS)
- [9] Lee D., Lim M., Han S. et Lee K. (2007). « ATLAS : A Scalable Network Framework for Distributed Virtual Environments ». Presence : Teleoperators and Virtual Environments, 16(2):125–156.
- [10] Docteur Riaha Med Amine , cours réseau 2015 , Université m'hamed bougara Boumerdés .
- [11] www.artoolkit.org Consulté en 2017
- [12] www.unity3d.com Consulté en 2017
- [14] www.vuforia.com Consulté en 2017
- [13] <https://unity3d.com/fr/unity/multiplatform> Consulté en 2017
- [14] Cédric Fleury. Modèles de conception pour la collaboration distante en environnements virtuels distribués : de l'architecture aux métaphores. Synthèse d'image et réalité virtuelle [cs.GR]. INSA de Rennes, 2012. Français. <tel-00765338>.
- [15] Platon et Platonisme : [Greek title] : conférences de 1893 par Walter Pater, Jean-Baptiste Picy à Paris : J.Vrin 1998.

Bibliographie

- [16] C. Fleury, N. Ferey, J-M. Vézien, P. Bourdot, “Remote Collaboration across Heterogeneous Large Interactive Spaces”, IEEE Virtual Reality conference (VR 2015) 2015
- [17] V. Fernández, J-M. Orduña, P. Morillo, “Server implementations for improving the performance of CAR systems based on mobile phones”, Journal of Network and Computer Applications, Vol 44, pp. 72–82, 2014.
- [18] C. Villacís, W. Fuertes, A. Bustamante, D. Almachi, C. Procel, S. Fuertes, T. Toulkeridis “Multi-player Educational Video Game over Cloud to Stimulate Logical Reasoning of Children”, 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications