

Université Saad Dahlab Blida1



Faculté des Sciences

Département d'Informatique

Mémoire présenté par :

BOUTEFNA Mohamed

HALOUI Aissam

En vue d'obtenir le diplôme de MASTER

Domaine : Mathématique et Informatique

Filière : Mathématique et Informatique

Spécialité : Informatique

Option : Ingénierie du Logiciel

Sujet :

***Interaction 3D et travail Collaboratif en  
Réalité Virtuelle***

Mr. HAMMOUDA MOHAMED

président jury

Mr. ZAIR Mustapha

Promoteur

Mr. MESSACI Assia

Encadreur

Promotion : 2015/2016

# ***Remerciements***

Tout d'abord, louange à « Allah » qui nous a guidés sur le droit chemin tout au long du travail et nous a inspirés les bons pas et les justes reflexes. Sans sa miséricorde, ce travail n'aura pas abouti.

Nous tenons à remercier sincèrement nos encadreurs Mr. MESSACI Assia et Mr. BELLARBI Abdelkader ainsi que notre promoteur Mr. ZAIR Mustapha, pour leur aide précieuse et pour le temps qu'ils nous ont consacré.

Nous n'oublions pas nos parents pour leur patience illimitée, leur encouragement continu, leur aide, en témoignage de notre profond amour et respect pour leurs grands sacrifices.

Enfin, nous adressons nos remerciements aux membres du jury pour avoir accepté d'évaluer ce travail et pour le temps consacré à la lecture de ce mémoire.

Merci à vous.

## *Dédicaces*

*A mes chers parents*

*A mes tantes*

*A mes sœurs*

*A mes oncles Housyni, Abd-el-Kader, Boubaker,*

*Ahmadou*

*A mes cousins lamine et Ahmed*

*A mess cheers amiss Ahmed, Saddam, Mohamed*

*yak., Ismail, Djamel*

*A tous mes amis de l'USDB*

*Je vous dédie ce modeste travail.*

*Mohamed.*

## *Dédicaces*

*A mes chers parents*

*A mes tantes*

*A ma sœur*

*A mon frère*

*A mes oncles Said, Abd-el-Kader, Mohamed,  
A mes chers amis Yassin, Ossama, Maroin, Hadje*

*A tous mes amis de l'USDB*

*Je vous dédie ce modeste travail.*

*Aissam.*

## Résumé :

Les Environnements Virtuels Collaboratifs (EVCs) sont des environnements virtuels peuplés où les utilisateurs interagissent entre eux pour réaliser des actions communes (déplacement, sélection et manipulation d'objets en commun, communication, etc.). Dans les EVCs, la complexité se situe au niveau de l'interaction d'un groupe d'utilisateurs avec les objets. En effet, il y a plusieurs contraintes à respecter notamment celles liées à la coordination des actions des différents utilisateurs et la conscience de présence d'un utilisateur par rapport aux objets de l'environnement et par rapport aux autres utilisateurs.

La notion d'interaction est utilisée pour désigner un ensemble de règles et de techniques permettant à l'utilisateur d'accomplir des tâches d'interaction au sein d'un environnement virtuel. Le moyen le plus naturel pour effectuer une interaction étant : l'utilisation des gestes de la main, le domaine de l'interaction se retrouve de ce fait, lié à la reconnaissance de gestes.

Dans ce contexte, notre sujet de fin d'étude a pour objectif de réaliser une application d'interaction 3D et travail Collaboratif en Réalité Virtuelle et ceci en utilisant la Kinect et leap motion Comme dispositif de capture et un visiocasque comme dispositif d'affichage.

Afin de réaliser notre travail, nous avons utilisé de nombreux outils logiciel tel que Unity 3D et Blender 3D pour la modélisation 3D et Nimble SDK et leap motion control panel pour la reconnaissance de gestes Nous avons donc exploité ces outils afin de concevoir un environnement virtuel et d'y représenter des mains virtuelles qui reproduisent les gestes des l'utilisateurs.

Pour que les utilisateurs interagissent dans l'environnement virtuel nous avons implémenté la technique d'interaction 3D "Main virtuelle" et "La technique HOMER" pour les trois tâches principales de l'interaction : la navigation, la sélection et la manipulation.

Ce mémoire contient un état de l'art des notions nécessaires à la compréhension de notre travail, et une explication détaillée de la mise en œuvre conduisant aux résultats obtenus ainsi qu'une présentation de tous les outils utilisés.  
**Mots clé:** *Réalité virtuelle, interaction 3D, environnement virtuel, travail Collaboratif*

# *Abstract*

The Collaborative Virtual Environments (CVE) are populated virtual environments where users interact to perform common actions (moving, selecting and manipulating objects in common, communication, etc.). In CVE, the complexity lies in the interaction of a group of users with objects. Indeed, there are several constraints to be respected including those related to the coordination of actions of different users and presence of mind of a user relative to objects in the environment and in relation to other users.

The concept of interaction is used to designate a set of rules and techniques enabling the user to perform tasks interaction in a virtual environment. The most natural way to perform an interaction being: the use of hand gestures, the area of interaction is found thus related to gesture recognition.

In this context, our topic of end of study aims to achieve a 3D interaction application and Collaborative work in Virtual Reality and this using the Kinect motion and leap like capture device and a head-mounted display as a display device.

To do our work, we used many software tools such as Unity 3D and Blender 3D for 3D modeling and Nimble SDK and leap motion control panel for gesture recognition we've used these tools to design a virtual environment to represent them virtual hands that reproduce gestures of the users.

For users to interact in the virtual environment we have implemented the 3D interaction technology "Virtual Hand" and "HOMER technical" for the three main tasks of interaction: navigation, selection and manipulation.

This memory contains a state of the art concepts necessary for understanding of our work, and a detailed explanation of the implementation leading to results and a presentation of all the tools used.

**Key words:** Virtual Reality, 3D interaction, virtual environment,

<b>Introduction général.....</b>	<b>1</b>
----------------------------------	----------

## **Partie 1 : Etat de l'art**

### **Chapitre 1 : La réalité Virtuelle**

<b>1. Introduction.....</b>	<b>3</b>
<b>2. Historique.....</b>	<b>3</b>
<b>3. Quelques définition.....</b>	<b>4</b>
3.1. La Réalité Virtuelle.....	4
3.4. Environnement virtuelle.....	5
3.5. Les interfaces matérielles de la RV.....	5
3.6. Technique d'interaction. ....	5
<b>4. domaines d'application.....</b>	<b>6</b>
4.1. La Réalité Virtuelle et la Domaine médical .....	6
4.2. La Réalité Virtuelle et la simulation Militaire.....	6
4.3. La Réalité Virtuelle et le Handicap .....	7
4.4. La Réalité Virtuelle et les Jeux Vidéos .....	7
<b>5. Les composantes de la Réalité Virtuelle.....</b>	<b>8</b>
5.1. Interaction.....	8
5.2. Immersion .....	9
5.3. Autonomie .....	9
<b>6. Les Avantages de la Réalité Virtuelle .....</b>	<b>10</b>
<b>7. Les Inconvénients de la Réalité Virtuelle.....</b>	<b>10</b>
<b>8. Conclusion.....</b>	<b>10</b>

### **chapitre 2: Interaction 3D mono-utilisateur dans EV**

<b>1. Introduction.....</b>	<b>11</b>
<b>2. Les principales tâches de l'interaction 3D.....</b>	<b>11</b>
2.1. La navigation.....	12
2.2. La sélection.....	13
2.3. La manipulation.....	13
2.4. Le contrôle d'application.....	14
<b>3. Les techniques d'interaction mono-utilisateur.....</b>	<b>14</b>
3.1. Les techniques de navigation.....	14
3.2. Les techniques de sélection et de manipulation.....	15
3.2.1. Les techniques égocentriques.....	16
3.2.2. Les techniques exocentriques .....	18
3.2.3. Les techniques hybrides.....	18
<b>4. Conclusion.....</b>	<b>19</b>

## chapitre 3:Techniques d'interaction multi-utilisateurs en Réalité Virtuelle

1. Introduction.....	20
2. Environnements Virtuels Collaboratifs.....	20
3. Perception entre utilisateurs dans EVCs.....	21
4. Communication entre utilisateurs dans EVCs.....	23
5. Les techniques d'interaction multi-utilisateurs (Collaboratives ).....	23
5.1. Navigation collaborative. ....	23
5.2. Sélection/manipulation.....	24
5.2.1 Sélection/manipulation asynchrone.....	24
5.2.2 Sélection/manipulation synchrones.....	25
5.3. Co-manipulation . ....	26
8. Conclusion . ....	28

## Partie 2 : Analyse, conception, réalisation et tests

### Chapitre 4 : Analyse et conception

1. Introduction. ....	28
2. Etude des besoins. ....	28
2.1. Description du système.....	28
2.2. Objectif. ....	31
3. Spécifications de l'application. ....	31
3.1. Spécifications fonctionnelles.....	31
3.2. Spécifications techniques. ....	31
4-Diagramme des cas d'utilisation. ....	32
5. Architecture de l'application. ....	33
6. Diagramme de classe. ....	35
3.1. Les packages . ....	36
3.2. Les interfaces. ....	36
3.3. Les classes.....	37
3.4. Les fonctions.....	38
4. Description comportementale. ....	39
4.1. Diagrammes d'activité.. ....	40
4.2. Diagrammes de séquence. . ....	45
5. Conclusion. ....	49

### Chapitre 5 : Réalisation et tests

1. Introduction. ....	50
2. Objectifs de la réalisation.....	50
2.1. Choix de techniques d'interaction 3d.....	51
2.2. Les gestes à reconnaître.....	52



<b>3. Environnement logiciel de développement.....</b>	<b>54</b>
3.2. Unity 3D .....	54
3.3. Blender 3DS .....	55
3.4. Three Gear System (Nimble SDK) .....	56
3.5. Leap Motion Control panel.....	58
3.6. Exploitation des outils matériels.....	59
3.6.1. SDK de la Kinect.....	59
3.6.2. Vuzix VR Manager.....	60
3.6.3. SDK de leap Motion .....	60
<b>4. Mise en œuvre.....</b>	<b>60</b>
4.1. Mise en place des outils matériels.....	60
4.2. Réalisation de l'environnement virtuel .....	61
4.3. Représentation des mains virtuelles. ....	63
4.3.1 Représentation des mains virtuelles de la premier utilisateur .....	63
4.3.2 Représentation des mains virtuelles de la Deuxième utilisateur ....	65
4.4. Implémentation de l'interaction .....	67
4.4. 1 Sélection et manipulation des objets virtuels.....	67
4.4. 2 Navigation.....	68
4.4. 3 Direction .....	69
<b>5. Représentation de l'interaction 3D multiutilisateur.....</b>	<b>71</b>
5.1. Scénario1 .....	71
5.2. Scénario2 .....	72
<b>6. Conclusion .....</b>	<b>72</b>
<b>Conclusion générale.....</b>	<b>73</b>
<b>Perspectives.....</b>	<b>74</b>
<b>Références bibliographiques.....</b>	<b>75</b>

## Liste des figures

Figure 1 : Les trois I (Adapté de [Burdea and Coiffet,1993]) .....	04
Figure 2 : Les médiations du réel (Adapté de [Tisseau, 2001]) .....	05
Figure 3: Mains sur la formation avec la réalité virtuelle .....	06
Figure : des médecins de l'armée expérimentent les entraînements en réalité virtuelle pour.....	07
Figure 5 : La réalité virtuelle : de nouvelles perspectives pour les handicapés .....	07
Figure 6 : Présence et autonomie en réalité virtuelle (Adapté de [Tisseau,2001]) .....	08
Figure 7 : Le trèfle fonctionnel de l'interaction 3D. (Adapté de [Otmame et al., 2007]).....	09
Figure 8 : La technique "Ray-Casting" [Bolt, 1980].....	16
Figure 9 : Illustration de la technique Scaled-World Grab lors de la sélection d'un objet....	17
Figure 10 : La technique HOMER de la sélection d'un objet .....	18
Figure 11 : Illustration de la technique HOMER.....	18
Figure 12: Exemple de collaboration d'avatars utilisés dans une plateforme collaborative DIVE.....	21
Figure 13 : – (a) Vue intérieure et (b) vue extérieure du monde virtuel [Leigh et al., 1996]. .....	22
Figure 14 : Un EVC pour la visualisation de molécules (Extrait de [Arthur et al., 1998])..	24
Figure 15 : : La course se réalise avec deux participants. L'écran est partagé, les utilisateurs peuvent communiquer en direct mais agissent uniquement sur leur véhicule. Ils ne peuvent agir sur le véhicule adverse. (Extrait du jeu Gran Turismo 5).....	25
Figure 16 : Application de forces sur un objet virtuel grâce à un ressort entre chacune des mains virtuelles et le point d'application de leur force [Noma et Miyasato, 1997].....	27
Figure 17: Microsoft Kinect pour Windows version 1 [Melgar et al., 2012]. .....	29
Figure 18: casque vidéo Wrap 1200DXAR de Vuzix [Vuzix, 2014].....	30
Figure 19: The Leap Motion coordinate system . .....	30
Figure 20: Diagramme des cas d'utilisation.....	32
Figure 21: Description abstraite du système.....	33
Figure 22: Description de l'architecture du système.....	34
Figure 23 : Diagramme de classe. ....	35
Figure 24 : Diagramme d'activité décrivant le CU "Sélectionner Objet" .....	40

Figure 25 : Diagramme d'activité du CU "Co-Manipuler Objet".....	41
Figure 26 : Diagramme d'activité du CU "Avancer".....	42
Figure 27 : Diagramme d'activité du CU "Reculer".....	42
Figure 28 : Diagramme d'activité du CU "Rotation Droite ".....	43
Figure 29 : Diagramme d'activité du CU "Rotation Gauche ".....	43
Figure 30 :diagramme d'activité du CU "S'orienter par le regard".....	44
Figure 31 : Diagramme de séquence de la sélection et Co-manipulation d'objet. ....	45
Figure 32 : Diagramme de séquence du CU "Avancer" utilisateur kinect . ....	46
Figure 33 : Diagramme de séquence du CU "Avancer" utilisateur de leap motion.....	46
Figure 34 : Diagramme de séquence du CU "Reculer". utilisateur de la kinect.....	47
Figure 35 : Diagramme de séquence du CU "Reculer". utilisateur de leap motion.....	47
Figure 36 : Diagramme de séquence du CU "S'orienter par le regard".....	48
Figure 37 : Les gestes à reconnaître pour les deux utilisateurs.....	53
Figure 38 : interface graphique de Unity 3D.....	55
Figure 39 : Interface graphique de Blender 3D .....	56
Figure 40 : environnement conseillé par Three Gear Systems.....	57
Figure 41 : Leap Motion.....	58
Figure 42 : Capture d'écran du visualiseur Leap Motion.....	59
Figure 43: exemple d'utilisation de Kinect fusion [Microsoft, 2015].....	59
Figure 44: Mise en place des outils matériels.....	61
Figure 45: Application des textures.....	62
Figure 46 : Eclairage de l'environnement virtuel.....	63
Figure 47 : Hiérarchie de la main gauche sous Unity 3D.....	64
Figure 48: Représentation de la main en maillage de triangles. ....	65
Figure 49 : Représentation finale des mains en 3D.....	65
Figure 50 : Hiérarchie de la main de leap motin sous Unity 3D.....	66
Figure 51 : Représentation de la main de leap motion en maillage.....	66
Figure 52 : Représentation finale des mains en 3D.....	67

Figure 53 : Intégration de l'environnement virtuel dans Unity 3D.....	67
Figure 54 : Sélection et manipulation d'un objet pour les deux utilisateurs.....	68
Figure 55 : Index de la main droite pointé pour avancer de deux utilisateurs .....	69
Figure 56 : Geste en "L" de la main droite pour reculer de les deux utilisateur.....	69
Figure 57 : Changement de la direction droite et gauche pour la première Utilisateur .....	70
Figure 58 : Changement de la direction gauche et droite pour la désaime Utilisateur.....	71
Figure 59: Co-maipilation des Objet Avec les deux Utilisateur( Scénario1) .....	71
Figure 60 : Co-maipilation des Objet Avec les deux Utilisateur( Scénario2) .....	72

## **LISTE DES TABLEAUX**

Tableau 1: Historique de la RA en dates. ....	4
Tableau 2: Caractéristiques de la Kinect version 1 [Dal Mutto et al., 2012].....	29
Tableau 3 : caractéristiques du visiocasque Wrap 1200DXAR de Vuzix [Vuzix, 201.....	30
Tableau 4: Description des packages.....	36
Tableau 5 : Description des interfaces.....	36
Tableau 6 : Description des attributs et fonctions de chaque interface.....	37
Tableau 7: Description des classes.....	38
Tableau 8 : Description des attributs et fonctions de chaque classe.....	39
Tableau 9 : Tableau comparatif des techniques d'interaction gestuelles.....	52
Tableau 10 : Correspondance entre les gestes, les tâches et les actions.....	53

***Introduction***  
***générale***

# L'introduction générale

## 1. Contexte général

La réalité virtuelle (RV) est une technologie jeune qui se situe à la croisée de plusieurs domaines comme l'infographie, la conception assistée par ordinateur, la simulation, la télé opération de systèmes complexes, etc... Cette technologie regroupe un ensemble de techniques et d'outils qui permettent aux utilisateurs, d'interagir avec un Environnement Virtuel (EV). Elle utilise de nombreux périphériques matériels et des techniques logicielles pour chaque domaine applicatif. Les actions dans l'EV s'opèrent via l'utilisation de techniques d'interaction 3D qui sont basées sur les facultés humaines d'action et de perception des utilisateurs.

L'interaction 3D est donc la composante motrice de la RV, de nombreuses techniques d'interaction 3D existent et sont dédiées à la réalisation d'une des quatre tâches (navigation, sélection, manipulation et contrôle d'application).

Le travail collaboratif autorise plusieurs utilisateurs, ou outils d'interaction, à agir simultanément sur un même objet virtuel. Fait donc apparaître une situation de collaboration.

Notre sujet s'inscrit dans le cadre du projet IM@REV « Interaction 3D Multimodale et Collaborative dans un environnement de Réalité Virtuelle et Augmentée » initié au sein de l'équipe IRVA de la division productique et robotique du CDTA. L'objectif de notre travail est de réaliser une application d'interaction 3D dans un environnement virtuel Collaborative en utilisant les gestes de la main.

## Problématique

Les Environnements Virtuels Collaboratifs (EVC) sont des environnements virtuels Peuplés où les utilisateurs interagissent entre eux pour réaliser des actions communes (déplacement, sélection et manipulation d'objets en commun, etc.) nous pouvons définir la problématique comme suit :

- Comment développer un mécanisme de Co-manipulation d'objets par plusieurs utilisateurs ?
- Comment coordonner leurs actions de façon à ce que notre système reste cohérent?

## Objectif:

L'objectif de notre travail consiste à réaliser un système de réalité virtuelle collaborative permettant à deux utilisateurs de manipuler des objets 3D partagés.

Proposition d'un modèle de co-manipulation d'objets permettant d'organiser et de gérer les actions des différents utilisateurs afin de réaliser des tâches communes.

## **Organisation du mémoire**

Afin de bien présenter notre travail, nous avons divisé notre mémoire en deux parties :

- ❖ **Partie 1** : qui représente l'état de l'art de notre travail, cette partie est constituée de trois chapitres comme suit :
  - ◆ **Chapitre 1** : Dans ce premier chapitre nous définirons le concept de réalité virtuelle, ainsi que les domaines d'applications et les Avantages et Inconvénients de la réalité virtuelle.
  - ◆ **Chapitre 2** : A travers ce chapitre nous présenterons la notion Interaction 3D mono-utilisateur dans Environnements Virtuels, les tâches et les différentes techniques d'interaction 3D, ainsi qu'une classification classique de celles-ci.
  - ◆ **Chapitre 3** : A travers ce chapitre nous présenterons la notion Techniques d'interaction multi-utilisateurs (Collaboratifs) en Réalité Virtuelle, Communication entre les utilisateurs dans EVCs et la Navigation collaborative.
  
- ❖ **Partie 2** : cette seconde partie contient une présentation détaillée du travail réalisé, à travers les deux chapitres suivants :
  - **Chapitre 4 : Analyse et conception**

Avant d'aller à la conception de notre application nous passerons d'abord par une analyse des besoins à travers une description fonctionnelle de l'application, les spécifications techniques et fonctionnelles ainsi que les cas d'utilisation de cette dernière.

Cette analyse nous mènera de façon logique à la conception que nous présenterons en utilisant les diagrammes UML (Diagramme de composants, diagramme de Classes, diagrammes de séquences et d'activités).
  - **Chapitre 5 : Réalisation et teste**

Dans ce dernier chapitre du mémoire, nous présenterons les outils utilisés pour la réalisation de notre travail, une explication détaillée de la mise en œuvre et les résultats des tests réalisés.



# ***Partie 1 : Etat de l'art***

# *Chapitre 1 : La réalité Virtuelle*

## 1-Introduction :

Il est naturel pour l'homme de s'échapper de la réalité quotidienne pour différentes raisons (artistiques, culturelles ou professionnelles). L'évolution des techniques aidant, l'homme a pu satisfaire ce besoin par des représentations principalement visuelles ou sonores mais figées du monde. Figées dans le sens où l'utilisateur ne peut observer la représentation qu'en spectateur, que ce soit une peinture, une photographie, un film d'images réelles ou d'images de synthèse. La réalité virtuelle lui offre une dimension supplémentaire en lui procurant un environnement virtuel dans lequel il devient acteur.

La réalité virtuelle (RV) peut donc être considérée telle qu'une révolution des interfaces homme-machine, ces deux concepts, vivent de nos jours un développement technologique considérable, et connaissent un grand intérêt de la part du public ainsi que de la communauté scientifique.

Avant de poursuivre notre travail nous sommes dans l'obligation de définir ce concept RV

Dans ce chapitre nous définirons la RV, et les composants de RV ensuite, nous présenterons les nombreux domaines d'application de la RV et enfin nous exposerons les avantages et les inconvénients de RV

## 2-Historique

La réalité virtuelle a vu le jour en 1956, par la création d'un appareil permettant de visionner un film en trois dimensions nommé « Sensorama », par Morton Heilig.

Depuis les années soixante à nos jours, de nombreux travaux ont été réalisés. Nous présentons les événements les plus importants liés aux deux domaines dans le tableau suivant (tableau 1.1).

Date	Evènements
1956	Morton Heilig, producteur de films, a développé « Sensorama », un appareil permettant de visionner un film en trois dimensions [McLellan, 1996].
1961	Philco a développé probablement le premier casque virtuel, permettant de voir « à distance » via une caméra vidéo [Kalawsky, 1993].
1963	Ivan Sutherland a développé le premier système graphique interactif appelé « Sketchpad ». Ce système pouvait créer des dessins précis [McLellan, 1996].
1966	Ivan Sutherland invente un écran monté sur la tête et le positionne comme une fenêtre sur un monde virtuel [McLellan, 1996].
1968	Ivan Sutherland s'est joint à David Evans pour construire un casque virtuel. Daniel Vickers a équipé un casque d'un appareil de suivi des déplacements. Peu après, Thomas Furness a développé un casque virtuel appelé « visually coupled system » au Wright-Patterson Air Force Base.
1970	Des logiciels sont conçus pour la navigation et l'exploration de mondes virtuels visualisés sur un écran [Fuchs et al., 2003].
1975	Krueger a le « Videoplace » qui permettait à un individu de s'immerger dans un monde généré par ordinateur habité par d'autres humains et participants [McLellan, 1996].
1980	Le Defense Advanced Research Project Agency (DARPA) a fondé un simulateur d'un jeu de guerre appelé « Simulation Networking » ou SIMNET [McLellan, 1996].
1981	Furness a développé un simulateur de vol virtuel pour les pilotes [McLellan, 1996].
1982	Apparition d'une nouvelle interface comportementale : le gant de données. Ce dispositif mesure les déplacements des doigts de la main et permet donc de communiquer avec un ordinateur [Fuchs et al., 2003].
1985	Début de la commercialisation de matériels et logiciels spécifiquement dédiés à la réalité virtuelle [Fuchs et al., 2003].
1989	Jaron Lanier propose l'expression « réalité virtuelle » [Ullah, 2011].
1992	L.B. Rosenberg développe l'un des premiers systèmes fonctionnels de RA,

	appelé « Virtual Fixtures » [Ullah, 2011].
1993	SiliconGraphics a annoncé leur « Reality Engine », un moteur graphique capable de supporter des applications d'environnements virtuels.
1994	Apparition des premiers systèmes de télé-chirurgie.
1998	La plupart des interfaces de la réalité virtuelle sont désormais accessibles au grand public.
2008	Wikitude réalise une application guide de voyage basée sur la RV.

Tableau 1: Historique de la RV en dates.

### 3- Quelques définitions

#### 3.1- La Réalité Virtuelle

La RV est une technologie qui se situe à la croisée de plusieurs domaines comme l'informatique, la simulation, la télé-robotique. Elle repose sur l'utilisation de différents dispositifs matériels hétérogènes ainsi que des logiciels. La RV dispose de nombreuses définitions dues à l'ambiguïté du terme virtuel.

En 1992, [Aukstalnins and Blatner,1992] ont proposé une définition très générique de Réalité Virtuelle. Ils considèrent la RV comme un moyen, pour des utilisateurs de visualiser et d'interagir avec des ordinateurs et des données variées et complexes.

Dans la communauté française de la RV, nous pouvons citer la définition proposée par Burdea et associés en 1993 [Burdea and Coiffet, 1993]. Ils ont introduit trois composantes à prendre en compte pour permettre à un utilisateur d'interagir en temps réel dans l'environnement virtuel (voir Figure 1.1) ces composantes sont l'Immersion, l'Interaction et l'Imagination.

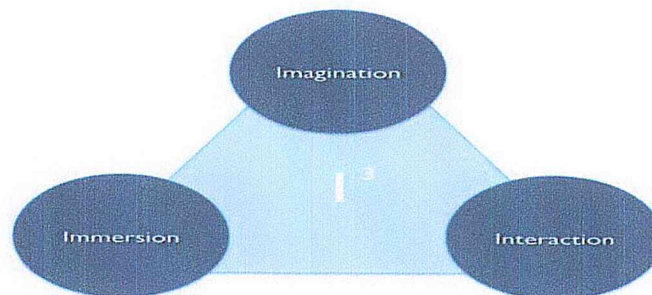


Figure 1 Les trois I (Adapté de [Burdea and Coiffet,1993]).

Tisseau [Tisseau,2001] définit la Réalité Virtuelle comme un univers de modèles qui propose la triple médiation des sens, de l'action et de l'esprit (voir Figure 1.2) cette médiation des sens permet la perception du réel, la médiation des actions permet de mener des expérimentations alors que médiation de l'esprit permet une représentation mentale de la réalité.

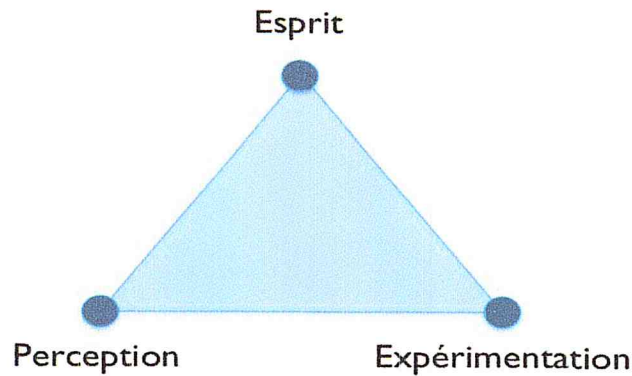


Figure 2: Les médiations du réel (Adapté de [Tisseau, 2001])

### 3.2- Environnement virtuel

Environnement virtuel est considéré comme le lieu suggéré par la RV pour accueillir un ou plusieurs utilisateurs et leur permettre d'accomplir certaines tâches avec l'impression d'être dans un cadre spécifique. L'environnement virtuel est représenté par un modèle 3D de données réelles ou imaginaires qu'on peut visualiser et avec lesquelles on peut interagir en temps réel [Hachet, 2003].

### 3.3- Les interfaces matérielles de la RV

Les interfaces matérielles utilisées dans un environnement virtuel constituent un élément majeur dans le développement des environnements virtuels ainsi que dans la mise en place des techniques d'interaction entre l'utilisateur et l'environnement virtuel. Ces interfaces matérielles sont utilisées comme des dispositifs d'entrée et/ou des dispositifs de sortie.

### 3.4- Technique d'interaction

Les interactions qui permettent aux utilisateurs d'agir sur le monde virtuel peuvent être très différentes en fonction des tâches que les utilisateurs doivent effectuer dans l'environnement virtuel. Dans certains cas, il faut trouver des techniques qui permettent aux utilisateurs d'effectuer des actions similaires à celles qu'ils effectueraient dans le monde réel.

Dans d'autres cas, il faut trouver des métaphores d'interaction qui permettent aux utilisateurs d'agir de manière plus simple et plus efficace afin de leur permettre de réaliser des tâches qu'ils ne pourraient pas faire ou, du moins, pas aussi facilement dans le monde réel. En conséquence, il existe un très grand nombre de techniques et de métaphores d'interaction qui varient en fonction des différentes applications, mais aussi en fonction des dispositifs immersifs utilisés. Dans cette partie, nous réalisons des différentes techniques couramment utilisées dans les environnements virtuels en se basant sur le livre de [Bowman et al., 2004] et sur le manuscrit de thèse de [Aguerreche, 2010].

Comme pour la navigation, les techniques d'interaction sont parfois classées en deux catégories : égocentrique et exocentrique. Les interactions égocentriques regroupent les techniques où les utilisateurs agissent depuis leur propre point de vue comme ils le feraient dans le monde réel. Les interactions exocentriques regroupent les

techniques où les utilisateurs prennent un point de vue extérieur au monde virtuel afin d'agir plus facilement. Les utilisateurs peuvent prendre ce point de vue extérieur en utilisant, par exemple, un monde en miniature [Stoakley et al., 1995] ou la technique du Scaled-world grab [Mine et al., 1997]. Néanmoins, lorsque les utilisateurs interagissent de manière égocentrique ou exocentrique, les techniques utilisées pour manipuler les objets virtuels sont souvent similaires. En effet, les tâches à effectuer sont généralement identiques que les utilisateurs agissent directement sur le monde virtuel ou sur une « maquette » du monde virtuel. Dans les deux cas, les interactions peuvent être décomposées en deux tâches : la sélection et la manipulation

## **4- domaines d'application**

La réalité virtuelle n'est pas l'apanage des joueurs de jeux vidéo. Elle fait également son chemin dans le monde Médical et Militaire et le Handicap..

### **4.1- La Réalité Virtuelle et la Domaine médical**

La réalité virtuelle est déjà un outil en médecine, notamment dans le traitement des phobies, de la douleur et dans la réalisation d'interventions chirurgicales. Une véritable révolution est en train de se produire dans le fait de faire passer cette technologie aux laboratoires, on cite comme exemple :

**La formation des chirurgiens :** Les chirurgiens commencent en général par s'exercer sur des cadavres avant de seconder des médecins expérimentés. Petit à petit, ils effectuent des tâches plus complexes et plus longues, jusqu'à pratiquer des opérations du début à la fin. La réalité virtuelle pourrait leur fournir une autre manière de s'exercer sans risquer de blesser un patient.



Figure 3 : Mains sur la formation avec la réalité virtuelle

### **4.2. La Réalité Virtuelle et la simulation Militaire**

Les simulateurs de réalité virtuelle contribuent à former les générations futures de soldats dans les tactiques et les plans de champ de bataille avant qu'ils aient jamais besoin de mettre les pieds dans une zone de guerre. La réalité virtuelle est une partie de plus en plus critique dans la préparation du soldat exemple :

- Simulation de combats
- Entraînement au tir
- Formation de pilotage pour les engins de guerre
- Lors de vrai combat la réalité virtuelle sert de technique stratégique pour prévoir les plans de bataille



Figure 4 : des soldats de l'armée expérimentent les entraînements en réalité virtuelle .

#### **4.3. La Réalité Virtuelle et le Handicap**

Les handicapés et les traumatismes physiques bénéficient eux aussi des technologies de la réalité virtuelle. Elle est en effet très développée en matière de rééducation physique ou la simulation va permettre aux patients de faire ses mouvements sans peur.

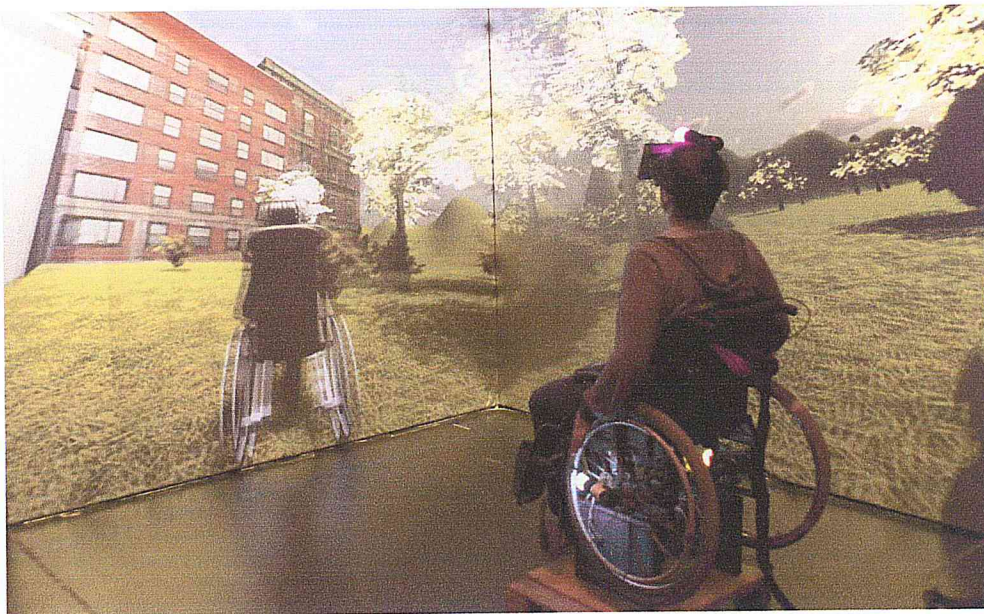


Figure 5 : La réalité virtuelle : de nouvelles perspectives pour les handicapés

#### **4.4. La Réalité Virtuelle et les Jeux Vidéos**

Le marché des jeux vidéo reste le plus gros client en termes d'utilisation de réalité virtuelle. Ces dernières années ce marché s'efforce d'augmenter la qualité des interfaces Hommes/Machines. Notamment une console vient de sortir dernièrement avec des capteurs de

mouvement. En effet le geste que l'on fait au réel se reproduit en temps réel sur l'écran. Etant donné le succès commercial de cette nouvelle console, en matière de jeu vidéo la réalité virtuelle a encore des marchés à conquérir.

## 5 Les composantes de la Réalité Virtuelle

A l'origine, les premiers travaux /recherche en RV se sont focalisés sur la conception et la réalisation d'interfaces favorisant l'immersion de l'utilisateur et ses possibilités d'interaction dans l'environnement virtuel (EV). Ces interfaces permettent la caractérisation de la présence de l'utilisateur et lui procurer une certaine autonomie. Zeltzer [Zeltzer,1992], a défini trois composantes de base pour la réalité virtuelle. L'autonomie, l'immersion et l'interaction pour définir une application de réalité virtuelle. Tisseau [Tisseau,2001] caractérise quant à lui, une application de RV en se basant sur les notions de présence et d'autonomie.

Cependant la présence intègre les notations d'interaction et d'immersion. Par conséquent, on distingue trois composantes principales qui définissent une application de RV. Tisseau représente graphiquement une application de RV par un point dans un espace à trois dimensions. Les axes de ce graphique correspondent aux différentes composantes avec une échelle allant de l'absence du composant à la composante (voir figure 6). Le graphique est illustré par des exemples d'application comme un virus informatique.

Ce dernier se caractérise par une autonomie forte mais une immersion et une interaction nulle.

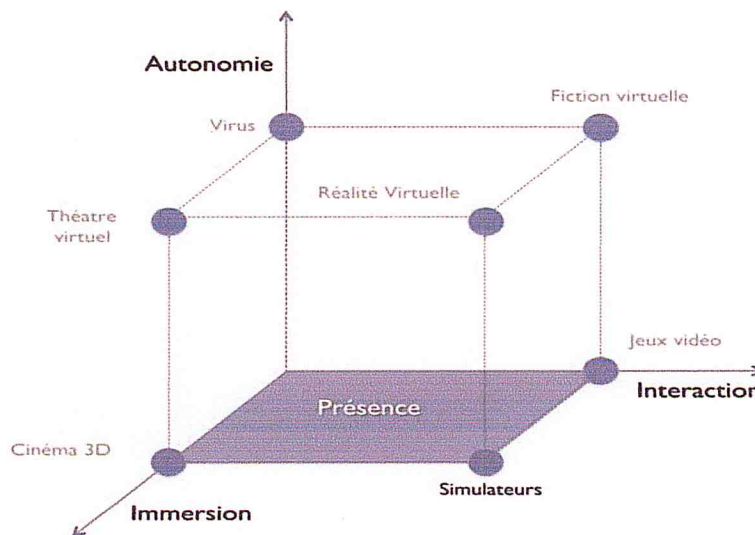


Figure 6 : Présence et autonomie en réalité virtuelle (Adapté de [Tisseau,2001])

Dans ce qui suit, nous présenterons les trois composantes de la Réalité Virtuelle.

### 5.1 Interaction

L'interaction est la composante qui permet aux utilisateurs d'interagir dans l'environnement virtuel (EV) donc c'est une composante dite motrice. Une technique d'interaction définit la méthode qui permet de réaliser une tâche dans l'environnement. C'est une sorte de protocole de communication avec le système qui représente les actions/réactions. On parle de métaphore d'interaction pour désigner un ensemble de techniques qui utilisent un même outil virtuel ou concept similaire. Dans le chapitre suivant, nous détaillerons les techniques utilisées pour interagir dans les environnements virtuels.

**Le trèfle fonctionnel de l'interaction 3D** Le modèle du trèfle fonctionnel de l'interaction 3D a été proposé par [Otmane et al., 2007]. Ce modèle s'inspire du modèle du trèfle fonctionnel des collecticiels



[Ellis et al, 1991]. Le trèfle est composé de trois feuilles (voir figure 7) représentant les trois tâches de base de l'interaction 3D (navigation, sélection et manipulation). La tâche de contrôle d'application est une tâche qui a un lien direct avec le système et qui intervient implicitement durant l'utilisation des trois autres tâches. Cela forme un cadre fonctionnel pour l'interaction 3D qui permet de représenter la fonction qu'un utilisateur est en train de réaliser à un moment donné dans l'environnement virtuel. Cette dernière peut-être simple telle que la navigation, la sélection et la manipulation ou la combinaison de deux tâches simples, par exemple, un utilisateur qui navigue et déplace un objet au même temps.

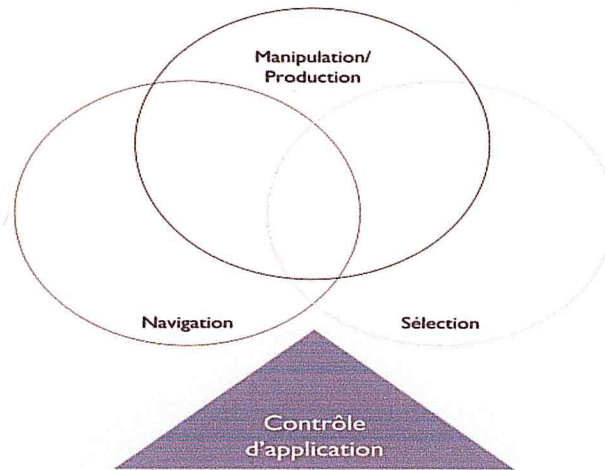


Figure 7: Le trèfle fonctionnel de l'interaction 3D. (Adapté de [Otmame et al., 2007])

## 5.2 Immersion

En RV, le terme « immersion » est souvent associé des formules du type « immersion dans un EV ou EVC » ou encore « l'utilisateur est immergé ». Par définition, l'immersion est la relation entre le sujet et l'environnement virtuel. Bowman a défini ce terme en 1999, comme la sensation d'être présent. Pour lui, un utilisateur est dit immergé lorsqu'il sent que le monde virtuel qui l'entoure a remplacé le monde physique. La présence donne à l'utilisateur l'impression d'être dans l'univers virtuel. Plusieurs définitions ont été proposées dans la littérature pour le terme de présence. Pour [Zeltzer, 1992], la présence, est la mesure du nombre de canaux sensoriels disponibles. Le terme de présence peut se découper en deux composantes, la présence sociale ou co-présence et la présence virtuelle.

La Co-présence ou présence sociale est le sentiment de présence avec d'autres participants partageant un même environnement virtuel. Pour [Biocca et al.,2001] la co-présence est le degré auquel un observateur présent dans environnement croit qu'il n'est pas seul ou isolé. Elle est liée au niveau de perception de la présence de l'autre personne présente dans le même environnement. Ce type de présence passé par l'utilisation d'un avatar représentant les utilisateurs dans l'environnement.

## 5.3 Autonomie

Dans un environnement virtuel, l'utilisateur est exposé dans un environnement autonome qui lui donne la liberté pour évoluer, modifier et interagir avant et pendant. Cela n'est pas le cas d'une application de calcul scientifique Car les paramètres sont initialement fixés. L'autonomie réside dans la capacité dont l'utilisateur dispose pour coordonner ses perceptions et ses actions au cours du processus d'interaction. L'autonomie par nécessité concerne la prise en instantanée des changements dans l'environnement, par les organismes comme par les mécanismes [Tisseau, 2001].

## 6- Les Avantages de la Réalité Virtuelle:

La Réalité Virtuelle permet à des utilisateurs d'être immergés dans des conditions très proche de la

réalité , ce qui fait que la RV permet de faire des projets très concret avant la réalisation , afin éventuellement d'avoir les bases pour améliorer un projet.

La Réalité Virtuelle va aussi permettre de s'immerger dans des endroits spécifiques tel le corps humain. C'est notamment dans le domaine médical que la RV va provoquer le plus de bouleversements.

Les loisirs sont bien entendus au centre des domaines d'application de la RV notamment avec les jeux vidéo. Cependant les scientifiques pensent et nous l'espérons que la RV dans les années à venir va faire progresser le monde scientifique à une allure considérable.

La Réalité Virtuelle est en fait un instrument guide, qui sert à résoudre la complexité du réel.

### **7- Les Inconvénient de la Réalité Virtuelle :**

Il y a 2 désagréments principaux :

- Le cout des matériels :

En effet toutes les interfaces de la réalité Virtuelle sont très onéreuses compte tenu de leur complexité. Le matériels de la RV est quasiment destiné à 80% aux laboratoires de recherches. La Réalité virtuelle sert un marché restreint et élitiste.

- La puissance de calcul :

Les simulations de RV sont en effet très gourmandes en calcul, notamment pour les immersions en temps réel. Aujourd'hui même les stations haut de gamme ne sont plus assez puissantes pour faire fonctionner certains programmes.

Cependant à la vitesse des avancées technologiques, on ne doute pas que dans les années à venir ses problèmes de calcul seront réglés.

## **8- Conclusion**

Dans ce chapitre, nous avons défini les concepts clés relatifs à la réalité virtuelle. Nous avons constaté qu'au fil du temps plusieurs définitions ont été proposées pour ce termes. et en présenté les différentes tâches et techniques de l'interaction 3D.

Par leur flexibilité applicative à travers beaucoup de domaines, la RV se sont imposées aussi bien à l'intérêt des chercheurs qu'à l'intérêt des grandes firmes informatiques, de nombreuses applications ont été réalisées et sont exploitées par le public.

Construire un univers virtuel n'est pas une finalité en soit, l'objectif principal est de pouvoir interagir avec les éléments de ce monde et pouvoir effectuer des actions. Ceci nous permet d'introduire la notion d'interaction qui sera l'objet du chapitre suivant.

## **synthèse...**

*chapitre 2: Interaction 3D mono-*  
*utilisateur dans Environnements*  
*Virtuels*

## 2.1 La navigation

Comme dans le monde réel, l'utilisateur a besoin de se déplacer dans la scène virtuelle pour réaliser certaines tâches. L'action « se déplacer » va de bouger sa tête Pour observer un objet de plus près ou d'un autre angle, à se promener dans un parc ou visiter un musée, voire se déplacer d'une planète à une autre. La navigation désigne l'ensemble des méthodes qui permettent de connaître la position d'un objet par rapport à un système de référence, ou par rapport à un point fixe déterminé. Certains auteurs considèrent la navigation comme la capacité de bouger à l'intérieur d'une scène générée par l'ordinateur [Rheingold, 1991].

D'autres encore la définissent comme l'ensemble des déplacements de l'utilisateur à l'intérieur de l'espace virtuel [Dumas et al., 1999]. Bowman et associés (Bowman et al., 2005) considèrent cette tâche comme la plus commune à tous les environnements virtuels. Elle permet aux utilisateurs d'explorer, de rechercher et/ou de manœuvrer dans l'espace virtuel. Bowman et associés définissent deux composantes principales pour la navigation : le déplacement et la recherche d'itinéraire [Bowman et al., 1997]. Le déplacement représente la composante motrice de la navigation.

Il se rapporte aux déplacements physiques de l'utilisateur d'un endroit à un autre. La recherche d'itinéraire correspond à la composante cognitive de la navigation. Elle permet aux utilisateurs de se repérer dans l'environnement et de choisir une trajectoire pour se déplacer [Fuchs et al., 2003].

Dans ce cas, l'utilisateur se pose des questions telles que : "où suis-je ?", "où dois-je aller ?", "comment arriver à cet endroit ?". Plusieurs facteurs influencent la qualité d'une technique de navigation. Ainsi Bowman et associés [Bowman et al., 1997] définissent les facteurs à prendre en considération dans la conception de toute technique de navigation :

- La vitesse de déplacement (le temps d'achèvement d'une tâche de navigation) ;
- La précision d'exécution;
- La conscience spatiale (connaissance de l'utilisateur de sa position et de son orientation dans l'environnement virtuel pendant et après la navigation) ;
  - La facilité d'apprentissage (la capacité d'un utilisateur débutant à s'appropriier la technique de navigation).
  - La facilité d'utilisation.
  - La capacité de l'utilisateur à collecter des informations sur l'environnement pendant le voyage.
- La présence virtuelle (le sentiment de l'utilisateur d'être immergé dans l'environnement virtuel).
- Le confort de l'utilisateur. Selon que l'on connaît ou non sa destination, la navigation peut être libre ou assistée. Dans le cas d'une navigation libre, l'utilisateur ne connaît pas précisément sa destination. Il se déplace dans l'espace dans le but de découvrir et d'explorer l'univers virtuel. Dans le cas d'une navigation assistée, l'utilisateur avance dans l'espace virtuel pour aller à un endroit bien précis qui

## 1 Introduction

Dans ce chapitre nous nous intéressons aux différentes formes d'interaction que l'on trouve en réalité virtuelle, pour les tâches les plus courantes. L'interaction 3D est le liant entre les périphériques d'interaction, qui fournissent des informations telles que la position et orientation d'un utilisateur. Les utilisateurs des environnements virtuels devront se déplacer, sélectionner et manipuler des objets. Ceci grâce aux techniques d'interaction 3D.

Ce chapitre va donc présenter tâches principales de l'I3D en RV et les techniques d'interaction 3D mono-utilisateur.

## 2 Les principales tâches de l'interaction 3D

La Réalité Virtuelle permet aux utilisateurs d'évoluer dans des environnements qui peuvent être soit imaginaires, soit des simulations du monde réel. L'interaction représente la composante motrice de la RV. Elle donne aux utilisateurs le pouvoir de réaliser différentes tâches et de changer les événements au cours du temps. Une tâche peut être définie comme la fonction que l'utilisateur exécute à un moment donné dans l'EV. Le résultat de l'exécution de cette fonction se traduira par les changements de l'état du système dans lequel l'utilisateur évolue. Nous pouvons définir l'interaction en RV comme la traduction des actions des utilisateurs dans le monde réel en des tâches spécifiques dans l'espace virtuel.

En 1999, Bowman [Bowman, 1999] propose une classification des différentes techniques d'interaction selon quatre tâches: la navigation, la sélection, la manipulation et le contrôle d'application. Sternberger et associés [Sternberger, 2006] ajoutent quant à eux, une nouvelle tâche qui est l'entrée de symboles. Cette dernière caractérise les techniques permettant d'entrer des symboles comme des chiffres et des lettres.

Une autre classification des techniques d'interaction a été proposée par Coquillart et associés ([Coquillart et al., 2003]). Les auteurs décomposent l'application de RV en des tâches élémentaires appelées primitives comportementales virtuelles (PCV). Ces PCV représentent les objectifs du niveau d'immersion et d'interaction (I2) fonctionnelles. Les tâches élémentaires vont être réalisées par l'utilisateur grâce aux techniques d'interaction. Les PCV sont regroupés en quatre catégories :

- Observer le monde virtuel;
- Se déplacer dans le monde virtuel;
- Agir sur le monde virtuel;
- Communiquer avec autrui ou avec le système.

Dans ce qui suit, nous présentons les quatre tâches d'interaction 3D qui constituent la classification moderne proposée par Bowman [Bowman, 1999]. La tâche d'entrée de symboles ne sera pas abordée car nous la considérons comme une composante de la tâche du contrôle d'application. Nous présenterons ensuite quelques techniques associées aux différentes tâches.

représente la cible du déplacement. Pour cela, l'utilisateur est aidé pendant le processus de navigation afin de se repérer et trouver la bonne trajectoire

### **2.2 La sélection**

L'action de sélection d'un objet est une tâche très courante que nous effectuons dans le monde réel. La préhension d'un objet se fait directement avec la main et tous les sens sont disponibles pour assister le mouvement. Dans les environnements virtuels, la sélection désigne la prise d'un ou de plusieurs objets pour permettre son utilisation ultérieure (manipulation) [Bowman et al. 2004]. La sélection d'un objet dans l'environnement virtuel peut être considérée comme deux sous tâches :

- ✓ La Désignation: c'est le choix par l'utilisateur d'un ou plusieurs objets dans le monde virtuel.
- ✓ La Validation: elle désigne la confirmation de la sélection. Elle peut se faire de plusieurs manières: presser un bouton, réaliser un geste ou encore utiliser une commande vocale.

### **2.3 La manipulation**

La manipulation est le processus par lequel l'utilisateur réalise une transformation de l'objet qu'il a sélectionné (rotation, translation, mise à l'échelle,...).

De manière plus générale, la manipulation peut être vue comme étant le processus d'édition/modification des propriétés d'un objet. Dans la littérature, sélection et manipulation sont souvent regroupées en une seule et même tâche.

Précisons que la tâche de manipulation peut être directe ou indirecte. La manipulation directe est celle où l'utilisateur opère directement sur l'objet affiché à l'écran. La manipulation indirecte est celle où l'utilisateur modifie les paramètres de l'objet par l'intermédiaire d'entités graphiques (boutons, menus,...), matérielles, etc.

Définition La manipulation regroupe l'ensemble des paradigmes d'interaction qui permettent de modifier un ou plusieurs attributs d'un objet virtuel, comme par exemple sa position, son orientation ou encore sa taille.

### **2.4 Le contrôle d'application**

[Bowman et al., 2001] définissent le contrôle du système comme étant les changements de l'état du système ou du mode d'interaction. En effet, toute modification qui intervient sur l'état et/ou le mode de l'application, ou de façon plus générale, du système, est considérée comme un contrôle du système. Ces modifications sont souvent réalisées à l'aide de menus ou de commandes en ligne, le contrôle d'application permet aux utilisateurs de :

- Exécuter une application particulière du système.
- Changer le mode d'interaction.
- Changer l'état du système.

### 3 Les techniques d'interaction mono-utilisateur

Cette section aborde la présentation de quelques techniques d'interaction 3D. Nous nous limiterons aux trois premières tâches de l'interaction: Navigation, Sélection et Manipulation. Le contrôle d'application ne sera pas abordé car nous ne l'utiliserons pas dans nos propositions. De plus, nous assimilerons la sélection et la manipulation lors de la présentation des techniques. Pour une présentation des techniques associées au contrôle d'application, nous vous proposons la lecture des thèses de [Ouramdane, 2009] et de [Sternberger, 2006].

#### 3.1 Les techniques de navigation

Dans un environnement, quelle que soit sa nature, l'interaction passe avant tout par la navigation, c'est-à-dire tout ce qui concerne le déplacement de l'être humain. Pour la navigation se compose de deux tâches fondamentales :

- le déplacement : c'est le contrôle du mouvement du point de vue de l'utilisateur dans l'environnement ;
- l'orientation, la découverte d'un chemin : il s'agit de la détermination d'un chemin à partir d'éléments visuels, et de la connaissance de l'environnement.

Nous allons voir un ensemble de métaphores de navigation utilisées en réalité virtuelle. Nous présentons les techniques de déplacement dans une scène virtuelle, et non celles qui servent à s'orienter. Sous cette restriction, la navigation va de tourner sa tête pour observer un objet sous un autre angle, à déplacer tout son corps dans un environnement, quelle que soit l'échelle de ce dernier. Nous avons choisi de scinder les métaphores de déplacement en 5 familles : celles qui se font par mouvements physiques (section 2.3.1), celles qui contrôlent le point de vue (section 2.3.2), celles qui indiquent la direction (section 2.3.3), celles qui permettent à l'utilisateur de s'accrocher à quelque chose pour se déplacer (section 2.3.4) et enfin, celles qui permettent de se déplacer selon un chemin planifié (section 2.3.5).

Selon que l'utilisateur connaisse sa destination ou non, sa navigation sera dite libre ou bien assistée. En effet, dans le cas de la navigation libre, l'utilisateur n'a pas un objectif précis. Il se déplace dans le monde virtuel pour explorer. Dans le cas de la navigation assistée, l'utilisateur se déplace pour se diriger vers un lieu identifié. Il sera alors aidé afin qu'il trouve le bon chemin. Il y a alors deux catégories de techniques dont :

- ✓ La direction du regard (Libre) [Mine, 1995] : le déplacement s'effectue comme dans le monde réel grâce à la direction du regard, donc par les rotations de la tête. Pour cela, la tête de l'utilisateur est suivie par un dispositif de trekking afin que l'image affichée dépende de la rotation de la tête.

- ✓ La direction de la main (Libre) [Mine, 1995]: on utilise la main pour indiquer dans quelle direction on souhaite aller. Dans ce cas, on procède par un trekking de la main pour en déduire la direction à prendre dans le monde virtuel;
- ✓ Le déplacement en utilisant les gestes de la main (Libre) [Mapes et Moshell, 1995]: cette technique est basée sur la métaphore de "tirer sur une corde". On réalise certaines combinaisons en pinçant ses doigts, pour que le système interprète ces combinaisons;
- ✓ Le déplacement en utilisant une carte (Assisté) [Bowman et al., 1998] : l'utilisateur est représenté par une icône dans une carte 2D. Le déplacement de l'icône par le stylet jusqu'au nouvel endroit de la carte implique le déplacement de l'utilisateur. Ce type d'interaction est repris dans les systèmes GPS pour explorer la carte géographique.

### **3.2 Les techniques de sélection et de manipulation**

Le toucher, la préhension et la manipulation sont des moyens utilisés pour communiquer et agir avec le monde réel. En réalité virtuelle, la sélection représente la désignation d'un objet ou d'un ensemble d'objets afin d'accomplir un objectif donné ; la manipulation quant à elle se rapporte à la spécification et la modification des propriétés des objets (en général la position et l'orientation). Plusieurs techniques existent et ont été classifiées en deux grandes familles par Poupyrev [Poupyrev et al., 1998] en fonction de la position et de la distance utilisateurs-objets virtuelles : les techniques exocentriques pour lesquelles le monde virtuel est manipulé depuis l'extérieur et les techniques égocentriques pour lesquelles l'utilisateur agit directement depuis l'intérieur de l'environnement virtuel. Cette dernière est composée de deux sous-familles : les techniques basées sur la métaphore de la main virtuelle et les techniques basées sur la métaphore du pointeur virtuel.

#### **3.2.1 Les techniques égocentriques**

La sous-famille contenant les techniques basées sur la métaphore de la main virtuelle utilise une représentation virtuelle de la main humaine, qu'on nomme avatar, afin de sélectionner et manipuler des objets. La sélection des objets est valable seulement quand la main "touche" les objets virtuels, ce qu'on appelle manipulation directe. On distingue différentes techniques :

##### **a) techniques basées sur la métaphore de la main virtuelle**

- ✓ **Main virtuelle** : la technique de sélection la plus naturelle consiste à représenter la main d'un utilisateur par une main humaine dans le monde virtuel [Jacoby et al., 1994].

Cette main permet de sélectionner les objets par le simple fait de les toucher. La représentation virtuelle de la main peut être Co-localisée avec la main réelle de l'utilisateur afin qu'il ait l'impression que cette main virtuelle est sa vraie main. En plus des changements de position, le mouvement des doigts peut être reproduit sur la main virtuelle si l'utilisateur porte un dispositif permettant de repérer la



## Interaction 3D mono-utilisateur dans EV

position de ses doigts comme un gant de données. Cette technique est particulièrement bien adaptée au HMD car l'utilisateur ne voit pas ses mains réelles avec un tel dispositif. Cependant, cette technique a un champ d'action limité qui correspond à la longueur de bras de l'utilisateur : il ne peut sélectionner que les objets virtuels qui sont apportés de sa main et il est obligé de se déplacer pour sélectionner des objets distants.

- ✓ **Métaphore du « Go-Go »** : cette technique proposée par [Poupyrev et al., 1996] est une extension de la main virtuelle. Elle permet à un utilisateur d'accéder à des objets éloignés sans avoir à se déplacer dans le monde virtuel. Lorsque la main réelle de l'utilisateur dépasse un seuil d'éloignement par rapport à son corps, la relation entre le déplacement de la main virtuelle et celui de la main réelle n'est plus linéaire, mais exponentielle. L'utilisateur peut ainsi sélectionner précisément les objets virtuels situés près de lui, mais il peut aussi aller sélectionner des objets lointains grâce à des déplacements plus importants de sa main virtuelle. Par contre, la co-localisation de la main de l'utilisateur n'est pas conservée avec cette technique.

### b) Les techniques basées sur la métaphore du pointeur virtuel :

Utilisent également une représentation virtuelle de la main. Dans ce cas, l'utilisateur sélectionne un objet à l'aide d'un pointeur laser et non pas en le touchant. Là encore plusieurs techniques existent :

- ✓ **La technique "Ray-Casting"** [Bolt, 1980] permet de pointer un objet avec un rayon laser issu de la main virtuelle. Pour valider la sélection, il faut donc détecter l'intersection entre le rayon et l'objet. C'est une technique simple car seulement deux degrés de liberté sont nécessaires. Bien qu'elle soit très pratique, elle pose des problèmes pour la sélection et la manipulation des objets petits ou lointains;



Figure 8:-La technique "Ray-Casting" [Bolt, 1980]

- ✓ **La technique "Flashlight"** : [Liang et al., 1994] permet de sélectionner des objets sans avoir besoin d'une grande précision: c'est une sélection "grossière". Cette technique est une extension du "Ray-Casting" pour laquelle le rayon laser est remplacé par un halo lumineux. Pour qu'un objet soit sélectionné, il faut qu'une

partie de celui-ci soit traversée par un halo de lumière entièrement ou partiellement. Cependant, d'après Bowman, la sélection d'objets devient très difficile quand ceux-ci sont groupés entre eux. Néanmoins, cela permet la sélection d'objets de petite tailles ou lointains. Pour pallier les problèmes liés à des objets groupés, la technique "Aperture" [Forsberg et al., 1996] permet de modifier dynamiquement le diamètre d'un halo de lumière afin d'obtenir une meilleure précision;

- ✓ **Les techniques plans-images**[Pierce et al., 1997] simplifient la sélection et la manipulation des objets en permettant de sélectionner et de manipuler les objets 3D en touchant et manipulant leurs projections sur un plan image 2D. On distingue quatre variantes qui utilisent des gants de données : dans le cas de la technique "StickyFinger", l'utilisateur pointe avec son doigt l'objet qu'il veut choisir ; pour "Head-crusher", la sélection s'effectue en entourant l'objet virtuel grâce à un geste du pouce et de l'index de la main réelle ; pour "Framing Hands" la sélection s'effectue en encadrant par les deux mains réelles le ou les objets ; finalement pour "Lifting Palm", pour sélectionner un objet virtuel on place sa main sous celui-ci.

### 3.2. Les techniques exocentriques :

Les techniques d'interaction 3D exocentriques permettent à l'utilisateur d'interagir avec le monde virtuel de l'extérieur. Une conséquence est que l'utilisateur n'aura pas le sentiment d'être totalement immergé dans le monde virtuel. Voici quelques techniques d'interaction 3D exocentriques :

- ✓ **La technique World In Miniature (WIM)** [Stoakley et al., 1995] utilise une représentation miniature réelle du monde virtuel afin que l'utilisateur puisse manipuler indirectement l'objet virtuel. La sélection des objets est réalisée par la technique de la main virtuelle simple. Cette technique ne permet pas un déplacement précis et pose des problèmes si l'environnement contient de nombreux objets, car cela engendre des miniatures trop petites et par conséquent difficilement sélectionnables et manipulables;
- ✓ **La technique Scaled-World grab**[Mine et al., 1997] permet la sélection d'objets éloignés. Lorsque l'utilisateur sélectionne un objet, l'échelle de l'utilisateur ou du monde virtuel est alors augmentée de manière à ce que la main virtuelle touche l'objet sélectionné (voir figure 11).

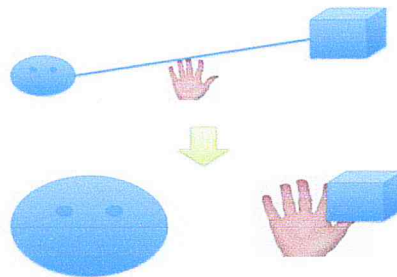


Figure 9 : Illustration de la technique Scaled-World Grab lors de la sélection d'un objet

### 3.3. Les techniques hybrides

Les techniques hybrides combinent au moins deux techniques ego et/ou exocentriques. En voici certaines :

- ✓ **La technique HOMER** : est une technique hybride (voir figure 12). Elle utilise le Ray- Casting pour la sélection d'un objet et la technique de la main virtuelle simple pour la partie manipulation. L'objet est alors attaché à la main et hérite de toutes les transformations que la main subie. Une fois que l'objet est relâché la main virtuelle reprend sa position initiale [Bowman et Wingrave, 2001];

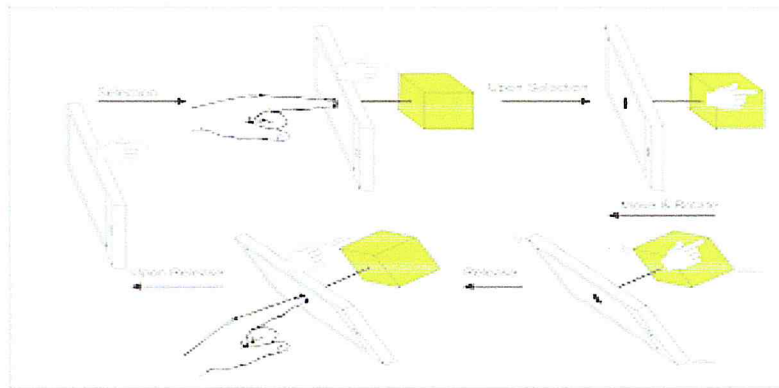


Figure 10 : La technique HOMER de la sélection d'un objet

- ✓ **La technique "Voodoo Dolls"** : est une technique qui combine la technique WIM et une des techniques plans-images. Cette technique nécessite l'utilisation de deux mains. La sélection des objets ou d'un groupe d'objets s'effectue par une technique d'occlusion comme "Framing Hands", puis le ou les objets sont alors dupliqués en miniature pour subir les manipulations qui seront transcrites à l'objet virtuel d'origine. Cette technique permet une manipulation aisée d'objets en mouvement ou animés [Bowman et Wingrave, 2001];

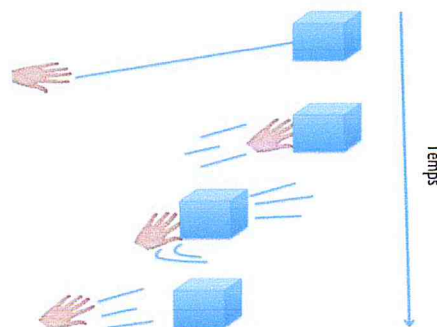


Figure 11: Illustration de la technique HOMER.

#### **4 Conclusion**

Nous venons d'étudier les techniques mono-utilisateurs qui sont à la base des actions des utilisateurs dans énervement virtuel collaboratif. Dans ce qui suit, nous étudierons quelques techniques d'interaction 3D collaborative, la coordination et communication entre les utilisateurs.

**chapitre 3:Techniques**  
**d'interaction multi-utilisateurs**  
**en Réalité Virtuelle**

## **1 Introduction**

Dans le dictionnaire Larousse, le verbe « collaborer » est défini comme « Travailler de concert avec quelqu'un d'autre, l'aider dans ses fonctions ; participer avec un ou plusieurs autres à une œuvre commune ». Le travail collaboratif est donc à la fois : partager, communiquer, coordonner, organiser et produire. Tous les participants sont engagés dans les mêmes tâches en se coordonnant pour résoudre le même problème ; cela impose une équipe, un but et un projet en commun. Un exemple classique pour la collaboration colocalisée et synchrone est donné par les tables rondes de réunion, d'où les participants échangent leurs idées et travaillent ensemble sur différents documents afin de réaliser un but défini (trouver la meilleure solution/décision pour un problème) dans un espace et un temps limités.

Dans Ce chapitre en présente les Environnements Virtuels Collaboratifs suivi d'une présentation des techniques multi-utilisateurs (collaboratives) où plusieurs utilisateurs vont agir ensemble sur un objet partagé.

## **2 Environnements Virtuels Collaboratifs**

Un environnement virtuel collaboratif est une application utilisant un environnement virtuel qui supporte des communications homme-machine-homme. A travers cet espace numérique, plusieurs utilisateurs peuvent communiquer, interagir, se toucher et collaborer même s'ils sont géographiquement éloignés. L'environnement virtuel dans un tel système peut varier du simple environnement basé sur du texte à un environnement graphique 3D très évolué. La représentation des utilisateurs peut aller d'une simple représentation en images 2D à des personnages 3D qui se déplacent et qui ont des expressions. Les EVCs se trouvent à l'intersection de plusieurs disciplines qui participent à leur développement, entre autres la réalité virtuelle, la psychologie cognitive, les IHM, l'ergonomie, la sociologie et le design [3]. Les EVCs ont de nombreuses applications dans la médecine, l'architecture, l'apprentissage ou simplement le divertissement. On peut notamment trouver sur le web de nombreux mondes 3D accessibles à tous et qui permettent des activités partagées entre les différents utilisateurs.

Selon Gibson « Les EVCs sont des systèmes distribués de RV qui offrent des environnements graphiquement, potentiellement infinis, numériques. Dans ces environnements, les individus peuvent partager l'information par l'interaction les uns avec les autres et par l'interaction collaborative et individuelle la représentation de données » [Gibson, 1989].

Dans ce contexte nous proposons la définition suivante :

## Techniques d'interaction multi-utilisateurs en Réalité Virtuelle

*«Un EVC est un espace virtuel de travail géré par un système informatique, permettant de regrouper plusieurs utilisateurs dans le même espace virtuel, les participants sont engagés dans les mêmes tâches en se coordonnant pour résoudre ensemble le même problème. Un EVC doit favoriser le partage des données, la communication et la production d'un résultat final »*

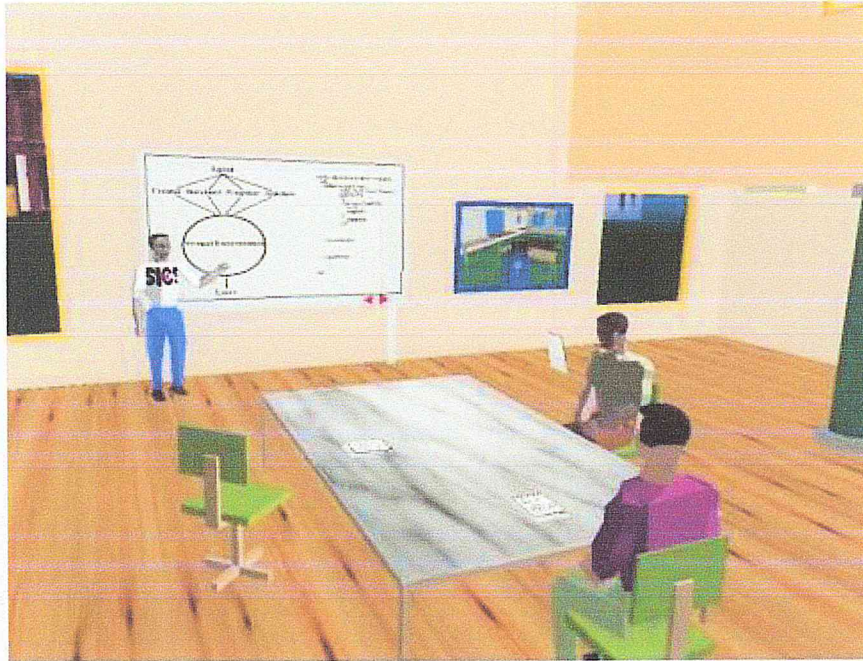


Figure 12 : Exemple de collaboration d'avatars utilisés dans une plateforme collaborative DIVE

### **3 Perception entre utilisateurs dans EVCs**

Le facteur préalable à toute collaboration est que chaque utilisateur soit capable de percevoir les autres dans l'environnement virtuel. Une mauvaise perception des autres peut entraîner des incompréhensions entre utilisateurs. En conséquence, les utilisateurs doivent pouvoir percevoir où se trouvent les autres, mais aussi ce qu'ils voient et ce qu'ils font. Il peut également être intéressant d'ajouter des informations supplémentaires sur leurs capacités de perception et d'interaction : ce qu'ils peuvent voir en fonction du dispositif de visualisation qu'ils utilisent, les objets virtuels avec lesquels ils peuvent interagir, etc.

## Techniques d'interaction multi-utilisateurs en Réalité Virtuelle

Cette perception entre utilisateurs peut se faire sur différents canaux sensoriels, dont principalement la vue, l'ouïe et le toucher. Au niveau visuel, la technique la plus utilisée est de représenter chaque utilisateur par un avatar dans le monde virtuel qui peut être plus ou moins réaliste comme le présentent [Benford et al., 1995]. Il est également possible d'utiliser un monde en miniature (comme le WIM de [Stoakley et al., 1995] pour représenter les utilisateurs sur cette sorte de « carte » du monde virtuel et ainsi permettre à chaque utilisateur de facilement localiser les autres. D'une façon un peu similaire), CALVIN [Leigh et al., 1996] propose aux utilisateurs de prendre une vue intérieure (mortal'sview) ou extérieure (deity'sview) du monde virtuel (figure 14). Les utilisateurs avec la vue extérieure peuvent facilement voir où se trouvent les autres et ce qu'ils font. Néanmoins, dans un environnement virtuel multi-échelle, lorsque les utilisateurs n'ont pas la même échelle, ils n'ont pas forcément la même perception du monde virtuel. Cela peut être une source d'incompréhension entre les utilisateurs comme l'expliquent [Zhang et Furnas, 2002]. Par exemple, un utilisateur qui est grand peut avoir l'impression d'être loin

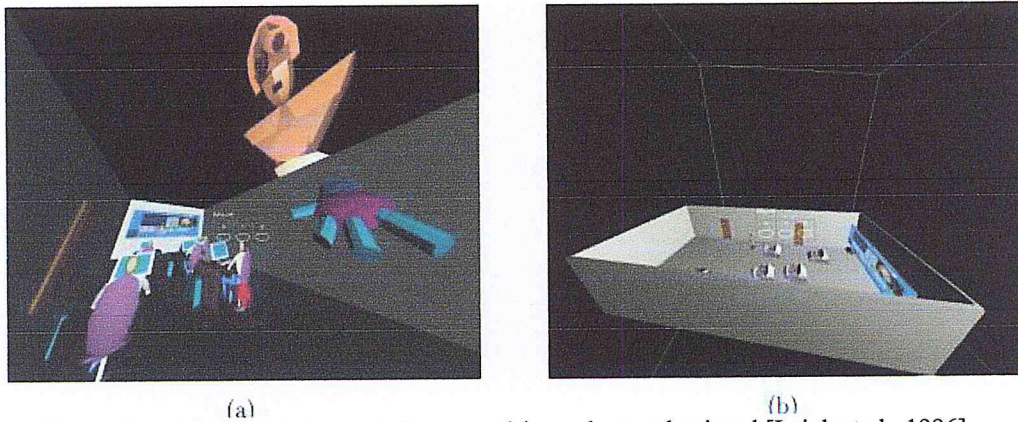


Figure 13. – (a) Vue intérieure et (b) vue extérieure du monde virtuel [Leigh et al., 1996].

d'un utilisateur qui est petit, et inversement. Au niveau sonore, il peut être intéressant de faire percevoir aux utilisateurs de façon spatialisée la parole, mais aussi les bruits des actions provenant des autres. Ces retours sonores sont essentiels à la communication, mais fournissent aussi des informations sur la position et les actions des utilisateurs. Au niveau haptique, lorsque plusieurs utilisateurs interagissent sur un même objet virtuel avec des dispositifs haptiques, ils peuvent percevoir les forces que les autres appliquent à l'objet. Ces informations kinesthésiques permettent de ressentir la présence et les actions des autres.



## **4 Communication entre utilisateurs dans EVCs**

Comme dans le monde réel, la communication entre les utilisateurs est essentielle pour permettre une collaboration efficace. La parole est le moyen de communication le plus simple à mettre en œuvre et le plus naturel pour les utilisateurs. En conséquence, de nombreuses plateformes de réalité virtuelle intègrent la voix des utilisateurs. Si la plateforme en elle-même n'intègre pas la voix, il est toujours possible d'utiliser un logiciel de voix sur IP pour remédier à ce manque. Comme le précisent [Hindmarsh et al., 1998], la communication orale apporte aux utilisateurs beaucoup d'informations complémentaires pour la collaboration et permet éventuellement de compenser une mauvaise perception de l'environnement virtuel ou des autres utilisateurs. Par exemple, elle permet aux utilisateurs de compléter leur expertise du monde virtuel lorsqu'ils ont des points de vue différents ou de se coordonner lorsqu'ils réalisent une co-manipulation. Même si la communication orale est souvent pertinente lorsque deux utilisateurs sont en train de réaliser une tâche ensemble, elle peut entraîner des discontinuités lors des interactions comme l'expliquent [Bowers et al., 1996]. De plus, ce mode de communication est difficile à utiliser lorsque les utilisateurs sont nombreux ou lorsque l'environnement est bruyant.

## **5 Les techniques d'interaction multi-utilisateurs (Collaboratives)**

### **5.1 Navigation collaborative**

Comme l'expliquent [Snowdon et al., 1995], les environnements virtuels collaboratifs sont du type WYSINWIS (WhatYourSee Is Not What I see) dans le sens où les utilisateurs peuvent prendre des points de vue différents sur le monde virtuel afin de compléter leur perception mutuelle. Néanmoins, il est important pour cela que ces utilisateurs perçoivent en même temps le même état de l'environnement virtuel. Pour aider à la navigation, un utilisateur peut prendre un point de vue extérieur au monde virtuel comme dans CALVIN [Leigh et al., 1996] afin de diriger les autres utilisateurs qui naviguent ou même de les placer directement comme un objet virtuel pour leur offrir un point de vue intéressant.

Même si les utilisateurs peuvent naviguer indépendamment, il peut être intéressant de leur permettre de synchroniser leur déplacement afin qu'ils puissent se suivre dans le monde virtuel ou se placer au même endroit pour avoir le même point de vue (si un utilisateur veut montrer quelque chose à un autre). [Duval et al., 2008] proposent de permettre aux utilisateurs de définir un ensemble de points de vue intéressants que les autres pourront ensuite parcourir pour explorer des données scientifiques. Par ailleurs, [Yang et Olson, 2002] proposent trois modes pour permettre à deux utilisateurs de naviguer ensemble :

- les deux utilisateurs ont la même vue mais un seul dirige le déplacement,
  - l'un des utilisateurs suit avec un offset celui qui dirige le déplacement,
  - l'un des utilisateurs à une vue extérieure au monde virtuel et seul l'autre se déplace.
- Enfin, [Dodds et Ruddle, 2008] proposent une navigation en groupe où les utilisateurs peuvent se déplacer indépendamment, mais où le système offre des fonctionnalités pour les aider à rester en groupe. Il leur permet, par exemple, de suivre automatiquement un membre du groupe ou d'être placé au centre du groupe (moyenne des positions des membres). Lors de ces déplacements automatiques, ils continuent à contrôler indépendamment leur orientation afin de pouvoir observer le monde virtuel comme ils le souhaitent.

### 5.2 Sélection/manipulation

#### 5.2.1 Sélection/manipulation asynchrone

La méthode asynchrone est apparue dès les débuts de la recherche dans ce domaine. Elle exploite des techniques mono-utilisateur que plusieurs utilisateurs vont appliquer à un objet à tour de rôle. Une première approche est apparue dans le projet PIT (Protein Interactive Theater) [Arthur et al., 1998]. Ce dernier est un projet qui permet à deux utilisateurs Co-localisés et physiquement proches de partager et d'agir sur un environnement virtuel commun pour l'étude des molécules. Chacun des deux utilisateurs possède sa propre vue de la scène donc des données car sa tête est "tracée" par des capteurs positionnés sur les lunettes stéréoscopiques (voir figure 16). Cependant, les deux protagonistes se partagent un même dispositif d'interaction pour la manipulation des objets avec une main virtuelle simple. Les actions sur l'objet sont réalisées par les utilisateurs, chacun leur tour, mais ils peuvent communiquer en direct pour se coordonner pour l'attribution des tâches.

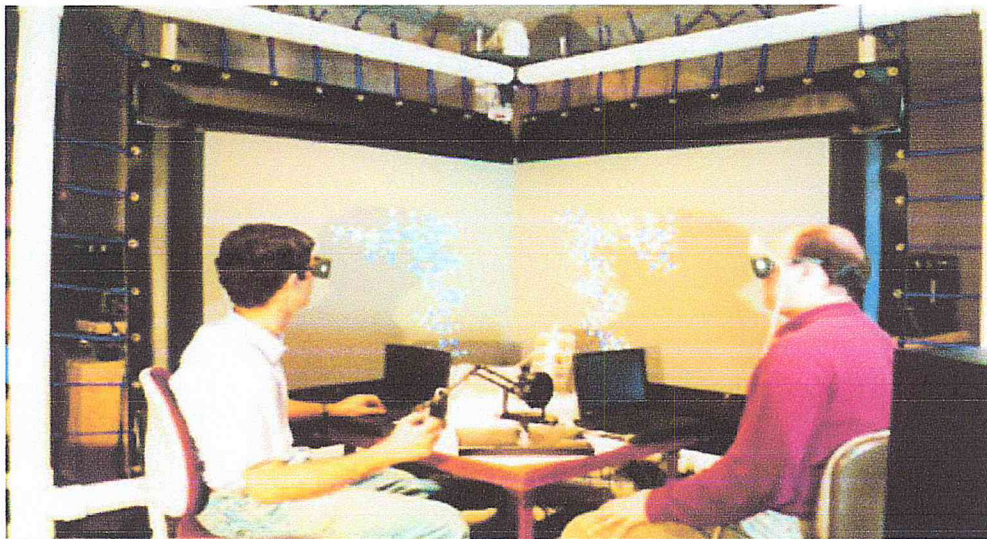


Figure 14: Un EVC pour la visualisation de molécules (Extrait de [Arthur et al., 1998])

## Techniques d'interaction multi-utilisateurs en Réalité Virtuelle

Des travaux similaires ont été proposés par Stewart et associés [Stewart et al., 1999]. La différence est que les deux utilisateurs disposent chacun d'un dispositif d'interaction. Cependant, ils disposent du même dispositif de sortie (écran). Ce principe se retrouve très souvent dans les jeux vidéos multi-joueurs ou l'écran est divisé en deux ou bien les utilisateurs peuvent voir sur un même écran non divisé les autres participants (voir figure 16). Les utilisateurs agissent uniquement sur leurs données mais peuvent visualiser les données des autres participants.

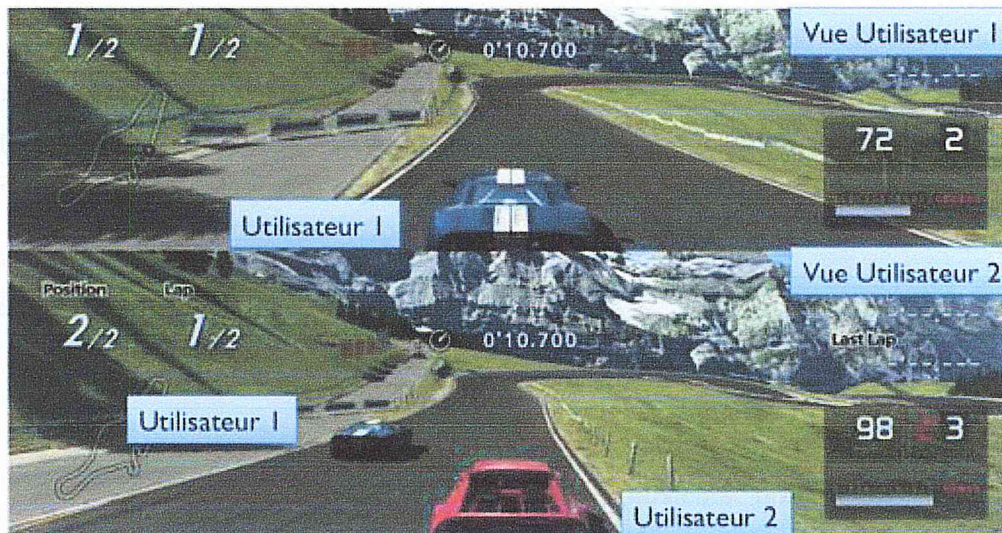


Figure 15: La course se réalise avec deux participants. L'écran est partagé, les utilisateurs peuvent communiquer en direct mais agissent uniquement sur leur véhicule. Ils ne peuvent agir sur le véhicule adverse. (Extrait du jeu Gran Turismo 5)

Dans [Widestrom et al., 2000], les utilisateurs doivent reconstituer un puzzle en 3D. Les utilisateurs prennent un objet à tour de rôle. Ils manipulent les pièces 3D du puzzle via la technique mono-utilisateur Main Virtuelle Simple.

L'utilisation successive de techniques mono-utilisateur ne permet pas une manipulation simultanée d'un objet par plusieurs utilisateurs. Les techniques d'interaction mono-utilisateur ne peuvent pas être employées telles quelles dans un EVC [Aguerreche et al., 2009a]. En effet, ces techniques ne gèrent pas les accès concurrents à un objet. Cependant, certaines techniques d'interaction peuvent être modifiées pour permettre l'interaction à plusieurs.

### **5.2.2 Sélection/manipulation synchrones**

Les techniques synchrones permettent aux utilisateurs d'agir sur des données partagées de l'objet, ensemble et en même temps. On distingue deux catégories : une première catégorie consiste à répartir des données partagées (degrés de liberté, couleur, taille). Par exemple, un utilisateur va agir sur les translations et l'autre sur les rotations de l'objet via des techniques mono-utilisateur. Elle combine les techniques d'interaction et non pas le mouvement des utilisateurs.

des forces appliquées par les utilisateurs. Afin d'éviter les mouvements brusques et les instabilités, un système de ressort est utilisé entre la main virtuelle et le point d'application de la force sur l'objet. Si deux utilisateurs seulement veulent réaliser ensemble cette manipulation physique (basée sur des forces), il est difficile de contrôler les six degrés de liberté de l'objet virtuel (en particulier la rotation) parce qu'il tourne de façon non contrôlée autour de l'axe formé par les deux points saisis.

[Aguerreche et al., 2009] proposent une technique de manipulation à trois mains : un utilisateur possède deux points de contrôle, alors que l'autre n'en possède qu'un seul. Ces trois points définissent un plan qui sert de support pour manipuler l'objet. Cette technique permet de contrôler plus facilement et d'une façon plus réaliste les six degrés de liberté de l'objet virtuel. Si les utilisateurs effectuent une collaboration locale, [Aguerreche et al., 2010] proposent d'utiliser en plus une interface tangible reconfigurable pour matérialiser les trois points de contrôle.

Dans ce contexte nous proposons la définition suivante :

*"LaCo-manipulation est un processus par lequel deux ou plusieurs personnes ou organisations s'associent pour effectuer un travail intellectuel suivant des objectifs communs."*

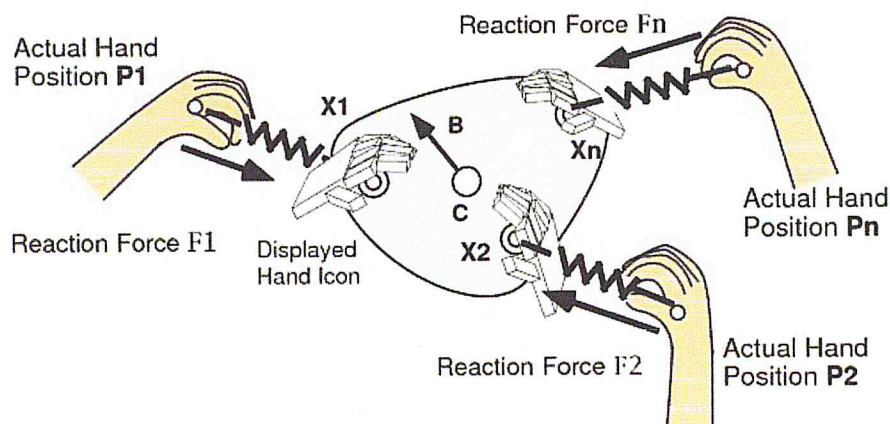


Figure 16– Application de forces sur un objet virtuel grâce à un ressort entre chacune des mains virtuelles et le point d'application de leur force [Noma et Miyasato, 1997].

## 6. Synthèse des techniques d'interaction 3D

Nous avons présenté et classifié les différentes techniques d'interaction 3D, celle-ci étant nombreuse, nous proposons donc une synthèse des techniques d'interaction 3D à travers un schéma que nous présentons dans la figure suivante (figure 17).

La seconde catégorie concerne "l'accès concurrent aux mêmes données". Cela permet de combiner les mouvements des différents utilisateurs pour déterminer le mouvement de l'objet. On retrouve dans cette catégorie des techniques qui exploitent des techniques mono-utilisateur (ray-casting ou main virtuelle simple) ou des extensions de techniques qui ont pour base les techniques mono-utilisateur.

### 5.3 Co-manipulation

Contrairement aux techniques d'interaction présentées, la Co-manipulation consiste à manipuler un objet virtuel à plusieurs utilisateurs en même temps. Interagir à plusieurs avec le même objet virtuel permet soit de réaliser des tâches compliquées, soit de reproduire la manière réelle de manipuler cet objet. Pour mettre en œuvre une Co-manipulation, il faut être capable de combiner les actions provenant de plusieurs utilisateurs en même temps. [Aguerreche, 2010] propose de classer les solutions existantes en deux catégories : la séparation des degrés de liberté et les accès concurrents à un même degré de liberté :

#### a) Séparation des degrés de liberté

La séparation des degrés de liberté permet à plusieurs utilisateurs d'accéder indépendamment à des degrés de liberté différents. Cela permet d'éviter la modification concurrente sur les mêmes paramètres d'un objet. Classiquement, six degrés de liberté sont associés à un objet virtuel : trois pour les translations et trois pour les rotations. Cependant, il est possible de modifier de cette façon de nombreux autres paramètres de l'objet virtuel comme sa taille, sa couleur, etc. [Pinho et al., 2002] proposent deux méthodes pour réaliser la séparation des degrés de liberté :

- **Les techniques coopératives homogènes** : les utilisateurs utilisent chacun un même outil d'interaction mono-utilisateur .
- **Les techniques coopératives hétérogènes** : les utilisateurs utilisent des outils d'interaction différents (par exemple : un rayon virtuel pour qu'un utilisateur applique les translations et une main virtuelle pour qu'un autre applique les rotations).

#### b) Accès concurrents à un même degré de liberté

Pour permettre à des utilisateurs d'accéder en même temps aux mêmes degrés de liberté d'un objet virtuel, il faut mettre en œuvre des techniques pour gérer les modifications concurrentes. Une première solution consiste à faire la moyenne des valeurs données par les outils d'interaction de chaque utilisateur afin de calculer la valeur du paramètre contrôlé comme la position ou l'orientation d'un objet virtuel.

[Noma et Miyasato, 1997] proposent une deuxième solution qui consiste à appliquer des forces sur les points de l'objet saisi par les utilisateur grâce à une main virtuelle. La position et l'orientation de l'objet virtuel sont le résultat de l'équilibre

## Techniques d'interaction multi-utilisateurs en Réalité Virtuelle

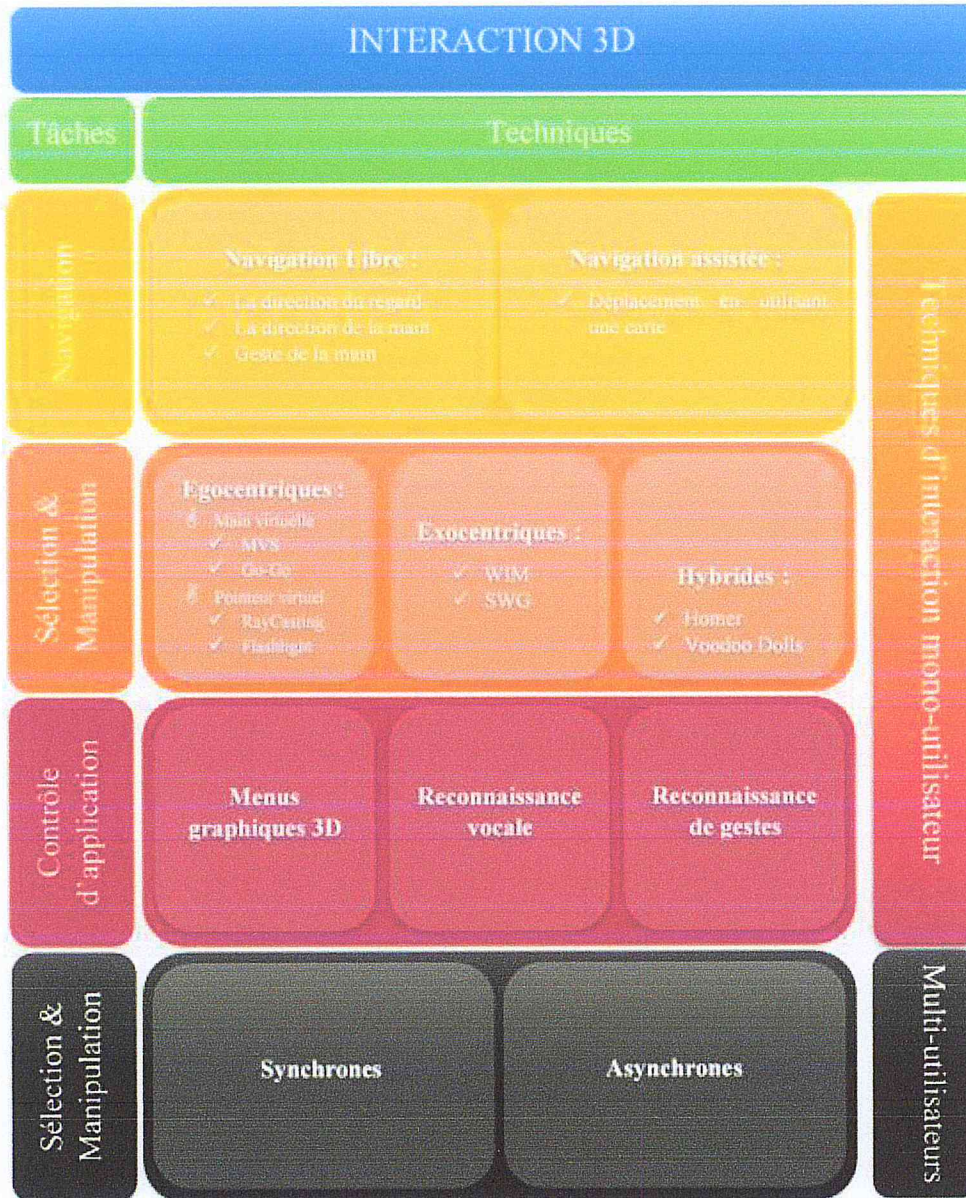


Figure 17: Synthèses des techniques d'interaction 3D

### 7. Conclusion

Dans ce chapitre, nous avons défini les concepts clés relatifs à l'environnement virtuel collaboratif. Nous avons constaté qu'au fil du temps plusieurs définitions ont été proposées pour ces termes et plusieurs travaux réalisés dans ce domaine.

Cependant, de nombreux travaux ont été réalisés dans ce domaine, mais celui-ci reste toujours un sujet de recherche d'actualité et les chercheurs espèrent toujours des résultats meilleurs.

**Partie 2 : Analyse,**  
**conception,**  
**réalisation et tests**

**Chapitre 4 : Analyse et**  
**conception**



### **1. Introduction**

Avant d'implémenter toute application informatique, il est utile, voir nécessaire, d'avoir recourt au préalable à une modélisation de cette dernière, et ce dans le but d'apporter une perspicacité et une clair voyance à la lisibilité et à la réalisation du travail effectuer ou à effectuer.

Dans ce chapitre nous présenterons l'analyse que nous avons effectuée à travers une description du système, les spécifications techniques et fonctionnelles de ce dernier, ainsi que les cas d'utilisation possibles. Ensuite, nous détaillerons la conception en utilisant le langage de modélisation UML (Unified Modeling Language), ainsi que des schémas fonctionnels que nous avons conçus.

#### **Description de l'UML**

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise et sont porteurs de sens au même titre que les mots d'un langage

### **2. Etude des besoins**

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système.

#### **2.1. Description du système**

Nous disposons d'un système composé de quatre outils matériels ; La Kinect et leap motion comme dispositif de capture et visiocasque et un Ordinateur comme dispositif d'affichage.

##### **2.1.1. Kinect**

La Kinect est un ensemble de technologies conçues par Microsoft, permettant à un utilisateur d'interagir naturellement avec une machine par les gestes ou par la voix [Microsoft, 2012]. La Kinect est un capteur contenant un projecteur infrarouge, une caméra infrarouge, une caméra RGB, ainsi qu'un réseau de microphones [Melgar et al., 2012].

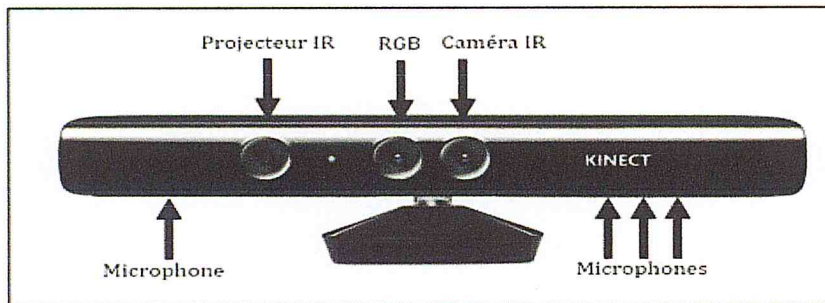


Figure 17: Microsoft Kinect pour Windows version 1 [Melgar et al., 2012].

Grace à cette composition matérielle, La Kinect fournit un flux de données de profondeur en plus de l'image RGB, ce qui permet d'aller vers l'interaction 3D. Le tableau suivant présente les caractéristiques de la Kinect pour Windows version 1.

Caractéristiques	Informations
Capteur	<ul style="list-style-type: none"> <li>✓ Lentilles détectant la couleur et la profondeur.</li> <li>✓ Micro à reconnaissance vocale.</li> <li>✓ Capteur motorisé pour suivre les déplacements</li> </ul>
Champ de vision	<ul style="list-style-type: none"> <li>✓ Horizontal: 57 degrés.</li> <li>✓ Vertical: 43 degrés.</li> <li>✓ Marge de déplacement : +/- 27 degrés.</li> <li>✓ Portée 50cm à 3,5m.</li> </ul>

Tableau 2: Caractéristiques de la Kinect version 1 [Dal Mutto et al., 2012].

### 2.1.2. Casque vidéo

Un visiocasque ou appelé casque vidéo ou encore casque de réalité virtuelle est un dispositif porté sur la tête qui dispose de caméra à la même position que les yeux et d'un ou deux écrans d'affichage en face des yeux.

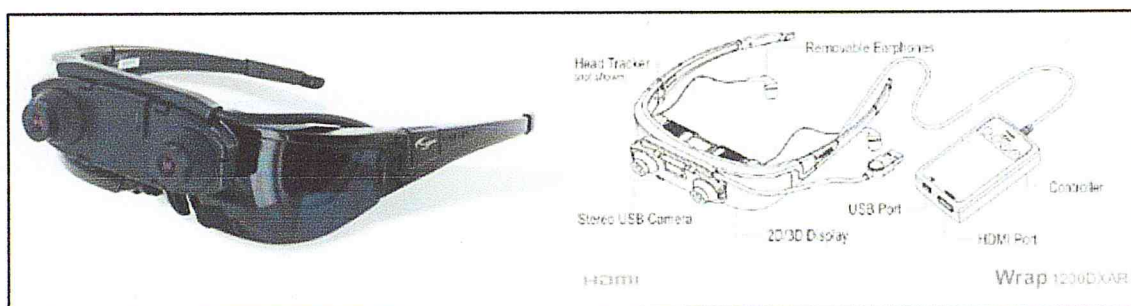


Figure 18: casque vidéo Wrap 1200DXAR de Vuzix [Vuzix, 2014].

Prenons l'exemple du Wrap 1200DXAR de Vuzix [Vuzix, 2014] qui dispose de deux caméras qui sont considérés par un ordinateur Windows comme de simple webcams USB et de deux écrans d'affichage à connecter par HDMI, ainsi que des magnétomètres, accéléromètres et gyroscopes qui permettent un suivi de la tête. De plus aucune carte graphique spéciale n'est nécessaire le casque s'adapte à n'importe quel ordinateur disposant de ports USB et d'entrée HDMI. Le tableau suivant présente les caractéristiques de ce casque.

Caméras stéréo	✓ Deux caméras VGA (640x480). ✓ Capture à 30 images/seconde. ✓ Connection USB 2.0.
Affichage	✓ Deux écrans LCD couleur 24 bits. ✓ Plusieurs résolutions supportées. ✓ Connexion HDMI.
Capteurs	✓ 3 magnétomètres. ✓ 3 accéléromètres. ✓ 3 gyroscopes.

Tableau 3 : caractéristiques du visiocasque Wrap 1200DXAR de Vuzix [Vuzix, 2014].

### 2.1.3. leap motion

Le leap motion est un petit appareil qui utilise la technologie infrarouge pour détecter les mains et les doigts d'un utilisateur dans un faible rayon d'action (moins de 60 cm)

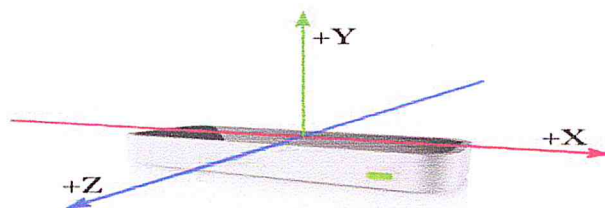


Figure 19 : The Leap Motion coordinate system.

## 2.2. Objectif

L'objectif de ce système est de permettre à deux utilisateurs de visualiser une scène virtuelle le premier à travers l'écran et le deuxième à travers le casque. Les utilisateurs auront alors la possibilité de se déplacer (avancer et reculer) dans l'environnement virtuel, ainsi que de sélectionner et manipuler collaborativement les objets virtuels contenus dans la scène, et ceci grâce à la reconnaissance de gestes de leurs mains.

## 3. Spécifications de l'application

Dans le but d'identifier toute les fonctionnalités que notre application doit offrir et les contraintes qu'elle doit respecter, nous allons lister les spécifications fonctionnelles et techniques de cette dernière.

### **3.1. Spécifications fonctionnelles**

- 1) Le système doit afficher une représentation 3D des mains de deux utilisateurs.
- 2) Le système doit suivre les déplacements et déformations des mains.
- 3) Le système doit reconnaître quelques gestes réalisés par les deux utilisateurs.
- 4) Le système doit détecter les collisions entre la main virtuelle et les objets virtuels.
- 5) Le système doit permettre à l'utilisateur de sélectionner des objets virtuels.
- 6) Le système doit permettre aux les deux utilisateurs de manipuler collaboratif des objets virtuels.
- 7) Le système doit permettre à l'utilisateur de se déplacer dans l'environnement virtuel collaboratif.
- 8) Le système doit permettre à première utilisateur de changer de direction en utilisant la direction de sa tête avec Casque vidéo.
- 9) système doit permettre à deuxième utilisateur de leap motion de changer de direction en utilisant les gestes réalisés par les mains.

### **3.2. Spécifications techniques**

- 1) Le système doit utiliser la Kinect Windows pour l'acquisition des données nécessaires à la reconnaissance de gestes de premier utilisateur.
- 2) Le système doit utiliser la Leap motion pour l'acquisition des données nécessaires à la reconnaissance de gestes de deuxième utilisateur.
- 3) Le système doit afficher l'environnement virtuel Collaborative à travers un visiocasque Vuzix Wrap 1200DXAR pour le premier utilisateur.
- 4) Le système doit afficher l'environnement virtuel à travers un ordinateur pour le deuxième utilisateur.
- 5) Le système doit être développé en C# sous Unity 3D conjointement avec Microsoft Visual Studio.
- 6) Le système doit se baser sur 3Gear Système pour la reconnaissance de gestes pour premier l'utilisateur .

7) Le système doit se baser sur leapSvc Système pour la reconnaissance de gestes pour le deuxième utilisateur.

#### 4-Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation présenté dans la figure suivante donne une vue globale du comportement fonctionnel de notre application, ainsi les utilisateurs est en mesure de se déplacer librement dans l'environnement virtuel colobarative et de manipuler les objets présents dans ce dernier.

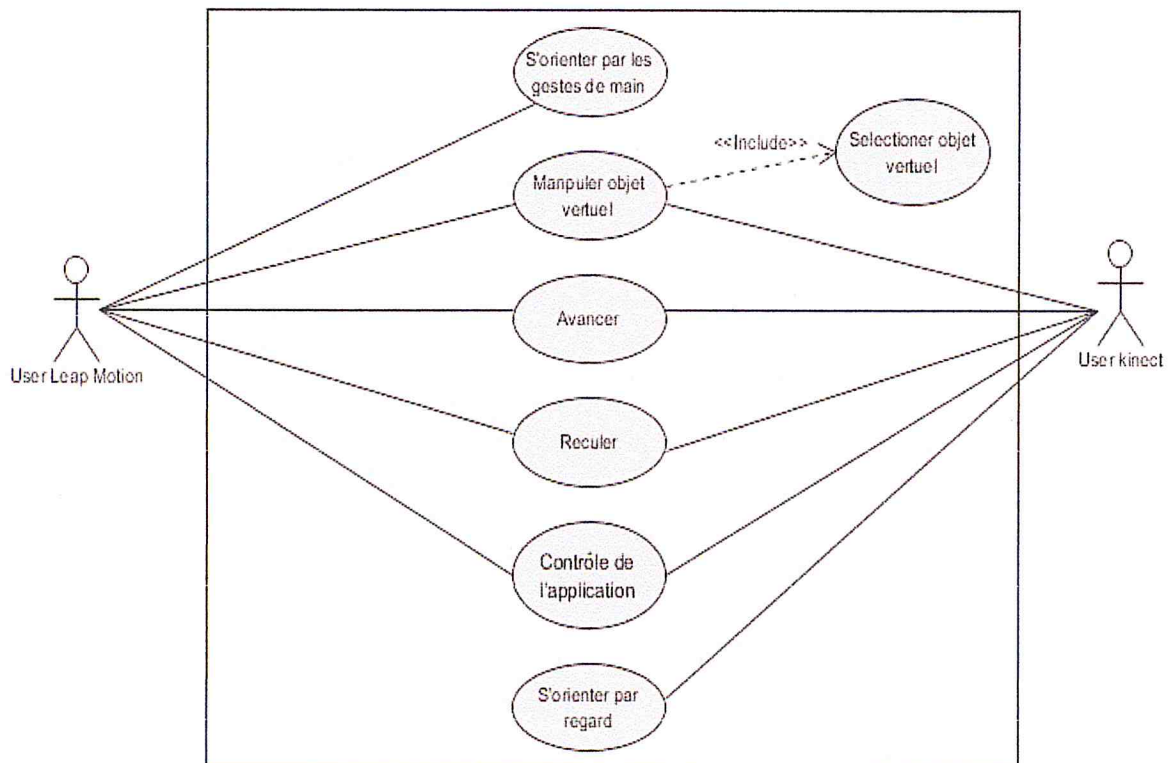


Figure 20 : Diagramme des cas d'utilisation

Notre diagramme est constitué de cinq cas d'utilisation principaux et un cas secondaire. Nous pouvons remarquer que la sélection d'objet virtuel est inclut dans la manipulation. En effet avant de pouvoir manipuler un objet, celui-ci doit être préalablement sélectionné.

## 5. Architecture de l'application

De façon abstraite nous pouvons commencer par décrire notre application comme étant un module de reconnaissance de gestes qui interagit avec une interface virtuelle 3D

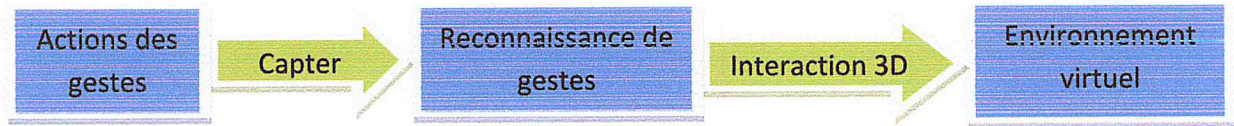


Figure 21: Description abstraite du système

A partir des spécifications du système décrite ci-avant nous avons réalisé un schéma décrivant l'architecture générale de notre application.

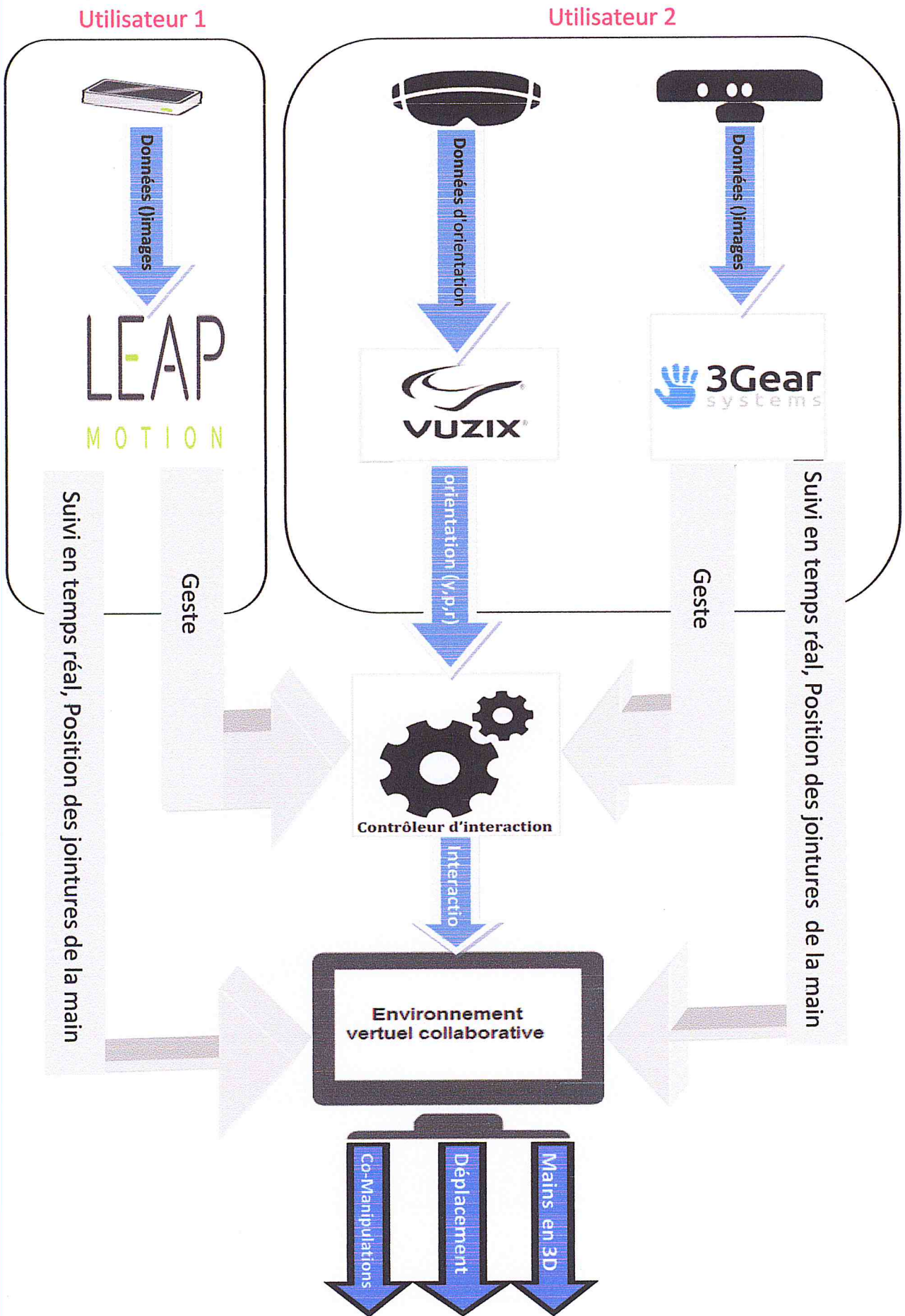


Figure 22 : Description de l'architecture du système.

## 6. Diagramme de classe

Après avoir détaillé l'architecture du système, nous allons présenter dans la figure suivante le diagramme de classe permettant ainsi d'éclaircir le développement de l'architecture présentée.

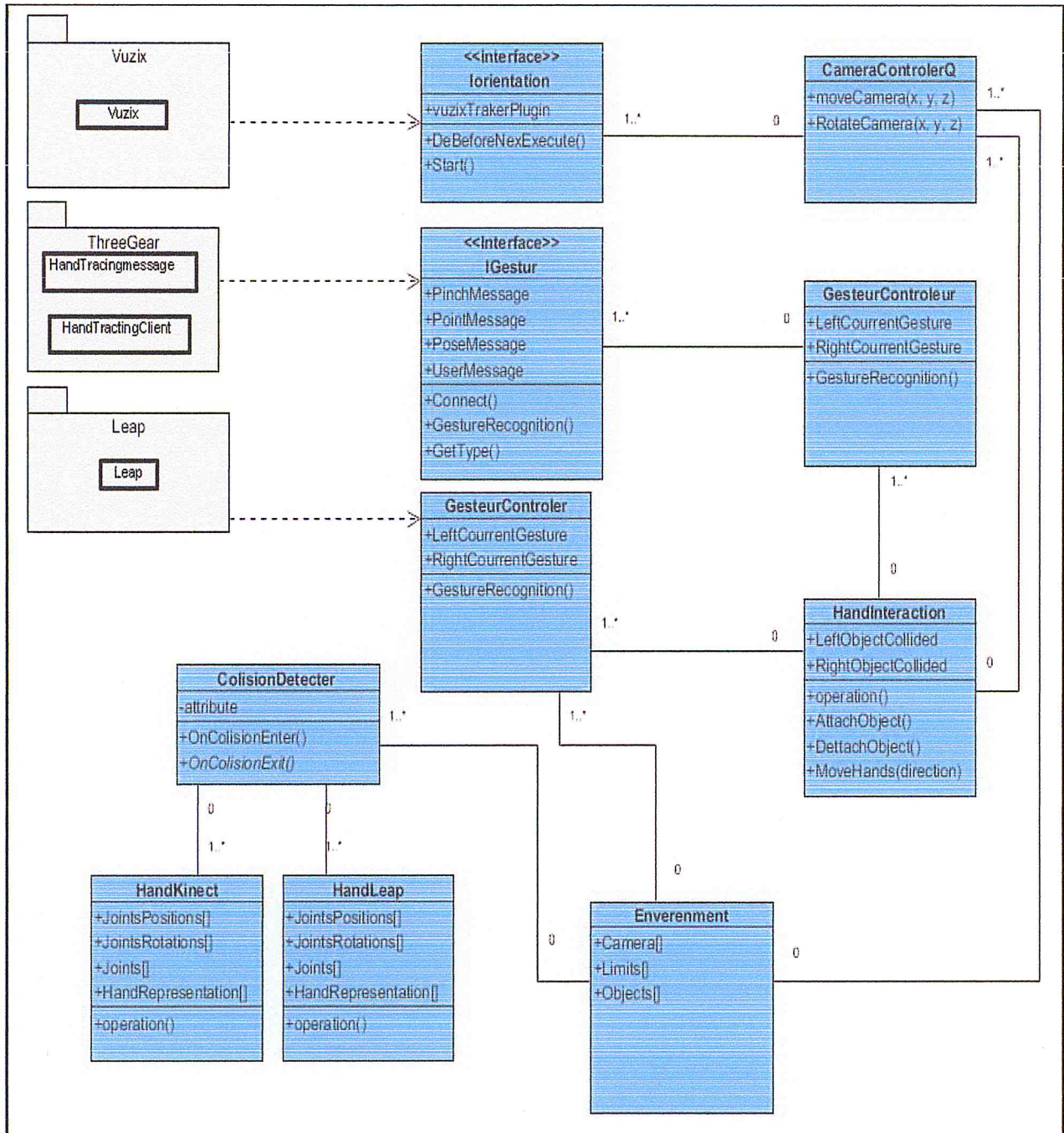


Figure 24 : Diagramme de classe.



### 3.1. Les packages

Le diagramme de classes présenté ci-avant est constitué de trois packages : « ThreeGear », « Vuzix » et « Leap ». La description de ces packages est donnée dans le tableau suivant.

Package	Description
ThreeGear	Package offrant la reconnaissance de gestes ainsi que le suivi des mains. ThreeGear fonctionne comme un serveur qui émet des messages à une application cliente contenant des informations relative à la main de l'utilisateur, à chaque fois qu'un utilisateur se connecte ou qu'un geste est reconnu.
Leap	Package offrant la reconnaissance de gestes ainsi que le suivi des mains. Leap fonctionne comme un serveur qui émet ls position des jointeure à une application cliente contenant des informations relative à la main de l'utilisateur, à chaque fois qu'un utilisateur se connecte ou qu'un geste est reconnu.
Vuzix	Package offrant les informations relatives au visiocasque Vuzix ainsi que ses données d'orientation.

Tableau 4 : Description des packages.

### 6.2. Les interfaces

Les packages ThreeGear » et « Vuzix » fournissent des interfaces qui sont exploitées tel que décrit dans le diagramme de classe, « IGesture » qui est fourni par « ThreeGear », ensuite « IOrientation » qui elle est fourni par « Vuzix ».

Le tableau suivant donne une description fonctionnelle de ces interfaces.

Interface	Description
IGesture	Offre toute les informations relatives à la position des mains de l'utilisateur, cette interface permet aussi la reconnaissance des gestes réalisés par l'utilisateur.
IOrientation	Fournit toutes les données qui concernent l'orientation du regard de l'utilisateur.

Tableau 5 : Description des interfaces.

Afin de mieux présenter les interfaces fournies par les Packages, nous donnons ci-après un tableau qui détaille ces dernières à travers la description de leurs attributs et fonctions.

Interface	Attributs/Fonctions	Description
<b>IGesture</b>	PinchMessage	Message indiquant si l'utilisateur a réalisé un « Pinch » (pince), ce message est envoyé par le serveur à chaque fois que l'utilisateur presse ou relâche un « Pinch ».
	PointMessage	Message envoyé à chaque fois que l'utilisateur pointe son index, cet attribut contient des données telles que la position de début et de fin de l'index ainsi que la probabilité du geste reconnu.
	PoseMessage	Message qui contient les informations relatives aux squelettes des deux mains.
	UserMessage	Message qui contient des informations concernant l'utilisateur, ce message est envoyé à chaque fois qu'un nouvel utilisateur se connecte.
	Connect( )	Cette fonction est utilisée pour se connecter au serveur « ThreeGear ».
	GestureRecognition( )	Fonction de reconnaissance de gestes.
	GetType( )	Fonction qui permet de récupérer le type de geste reconnu à partir d'un message du serveur de reconnaissance de gestes.
<b>IOrientation</b>	VuzixTrackerPlugin	Cet attribut importe toutes les DLLs nécessaires pour la récupération des données à partir du visiocasque.
	Start( )	Cette méthode est appelée pour lancer la récupération de donnée à partir du visiocasque.
	DoBeforNextExecute( )	Méthode qui permet l'exécution continue du script de récupération des données du visiocasque.

Tableau 6 : Description des attributs et fonctions de chaque interface.

### 6.3. Les classes

Les classes qui constituent notre système ont pour rôle de représenter les mains virtuelles et de gérer les interactions ainsi que le positionnement de les caméras.

Le tableau suivant donne une description générale des classes constituant notre système.

Classe	Description
HandLeap HandKinect	Contient les données nécessaires à la représentation de la main telle que la position des jointures. Exploite l'interface « IPosition » afin de représenter la main de l'utilisateur en 3D.
CollisionDetector	Détecte les collisions des mains avec les objets virtuels et identifie l'objet en collision afin de permettre l'interaction.
CameraControler	Contient des méthodes qui permettent de déplacer ou d'appliquer des rotations à la caméra de l'environnement virtuel.
GestureControler GestureControler1	Cette classe contient principalement la fonction de reconnaissance de gestes, fournissant ainsi le geste reconnu en temps réel.
HandInteraction	Gère l'interaction 3D par les gestes de la main en associant à chaque geste reconnu une fonction spécifique.
EnvironementData	Classe statique qui contient toute les données nécessaire à l'interaction telles que les limites de l'environnement virtuel et les identifiant des objets 3D contenus par celui-ci.

Tableau 7: Description des classes.

### 3.4. Les fonctions :

Le tableau suivant donne une description les fonctions et les attributs des classes constituant notre système.

Classe	Attributs/Fonctions	Description
HandKinect HandLeap	HandRepresentation[]	Tableau contenant les positions des jointures de la main de l'utilisateur correspondantes à la représentation sous Unity 3D, cet attribut est mis à jour en temps réel avec un mouvement de la main.
	JointsRotations [ ]	Tableau contenant les rotations des jointures de la main de l'utilisateur correspondantes à la représentation sous Unity 3D, cet attribut est mis à jour en temps réel avec un mouvement de la main.
	Joints [ ]	Tableau définissant les jointures des mains de l'utilisateur aux quelles vont être appliquées les transformations.
CollisionDetector	OnCollisionEnter( )	Evènement qui se déclenche lorsque la main de l'utilisateur entre en collision avec un objet virtuel.

	OnCollisionExit()	Evènement qui se déclenche lorsque la main de l'utilisateur n'est plus en collision avec un objet virtuel.
CameraControler	MoveCamera(x, y, z)	Fonction permettant de déplacer la caméra de l'environnement virtuel selon les paramètres en entrés
	RotateCamera(x, y, z)	Fonction permettant d'appliquer une rotation à la caméra de l'environnement virtuel selon les paramètres en entrés
GestureControle	LeftCurrentGesture	Attribut contenant le geste courant réalisé par la main gauche.
	RightCurrentGesture	Attribut contenant le geste courant réalisé par la main droite.
	GestureRecognition()	Fonction de reconnaissance de gestes.
HandInteraction	LeftObjectCollided	Attribut contenant l'identifiant de l'objet en collision avec la main gauche.
	RightObjectCollided	Attribut contenant l'identifiant de l'objet en collision avec la main droite
	AttachObject()	Fonction qui attache l'objet sélectionné à la main virtuelle pour permettre sa manipulation.
	DettachObject()	Fonction qui détache un objet de la main virtuelle à la fin de sa manipulation.
	MoveHands(direction)	Fonction qui fait avancer ou reculer les mains virtuelles selon le paramètre en entré « direction ».
EnvironementData	Camera[]	Attribut statique contenant l'identifiant de la caméra virtuelle.
	Limits[ ]	Tableau contenant les identifiants des objets représentant les limites de l'environnement virtuel.
	Objects[ ]	Tableau contenant les identifiants de tous les objets virtuels présents dans l'environnement virtuel.

Tableau 8 : Description des attributs et fonctions de chaque classe.

#### 4. Description comportementale

La description comportementale du système permet de décrire le flux d'action menant à la réalisation d'une fonction métier, ainsi que les interactions entre les différents composants de celui-ci, qui permettent d'effectuer un certain cas d'utilisation.

Nous allons présenter cette description à travers les diagrammes d'activité et les diagrammes de séquence correspondants aux cas d'utilisation décrits précédemment.

#### 4.1-Diagrammes d'activité

Tel qu'initié auparavant le cas d'utilisation « Sélectionner Objet », débute par la réalisation d'un geste de la main de l'utilisateur (Pinch), en suite on vérifie si la main virtuelle ou (Ray casting) est en collision avec un objet virtuel ou pas. Dans le premier cas l'objet est sélectionné. Le diagramme suivant décrit le déroulement de ces actions.

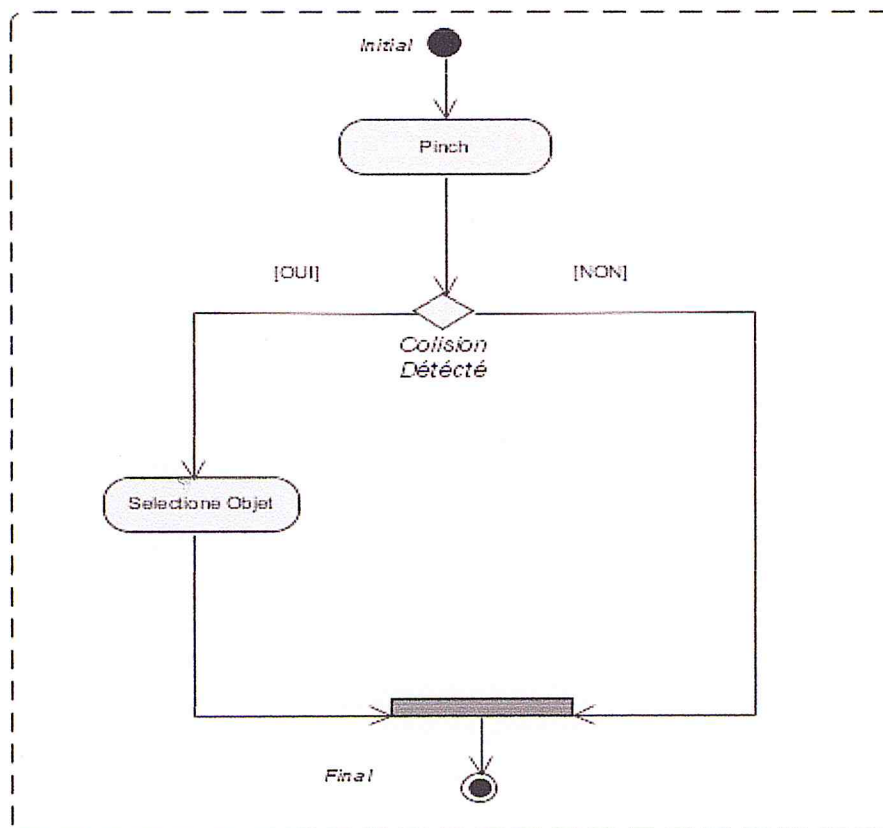


Figure 26 : Diagramme d'activité décrivant le CU "Sélectionner Objet".

A la suite de la sélection, la manipulation se fait par des transformations réalisées par la main de l'utilisateur. Deux conditions sont vérifiées pour permettre la manipulation de l'objet :

- Si un objet est sélectionné.
- Si déclencher la manipulation.

A défaut de l'une de ces conditions, la manipulation n'est pas possible.

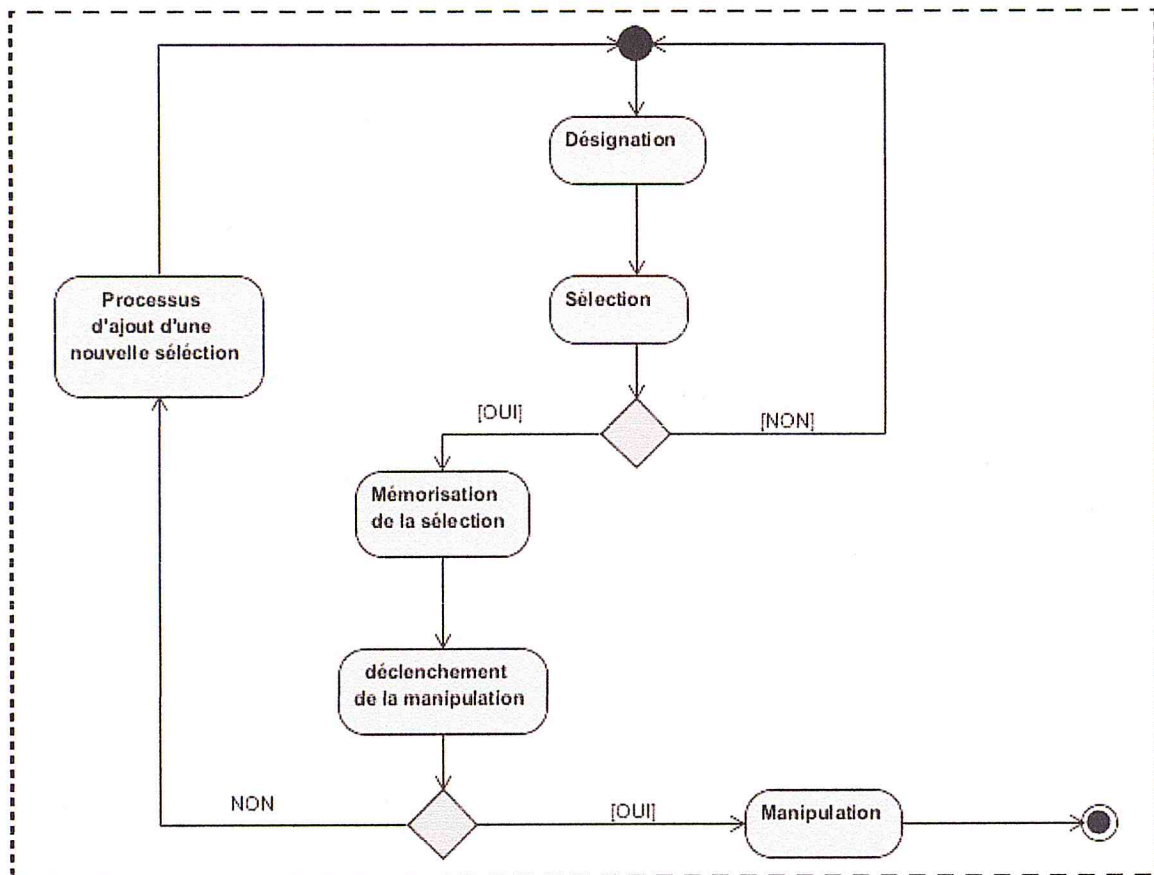


Figure 27 : Diagramme d'activité du CU "Co-Manipuler Objet".

Nous avons vu à travers la description de l'application qu'en plus de la manipulation d'objet 3D, les utilisateurs est aussi en mesure de se déplacer dans l'environnement virtuel Collaborative. Cette action apparait dans deux cas d'utilisation: « Avancer » et « reculer ».

Tant que les limites de l'environnement dans lequel évolue l'utilisateur ne sont pas atteintes, celui-ci peut avancer en pointant son index ou reculer en réalisant un geste en « L ».

Les diagrammes suivant décrivent le flux d'action des deux cas d'utilisation en question.

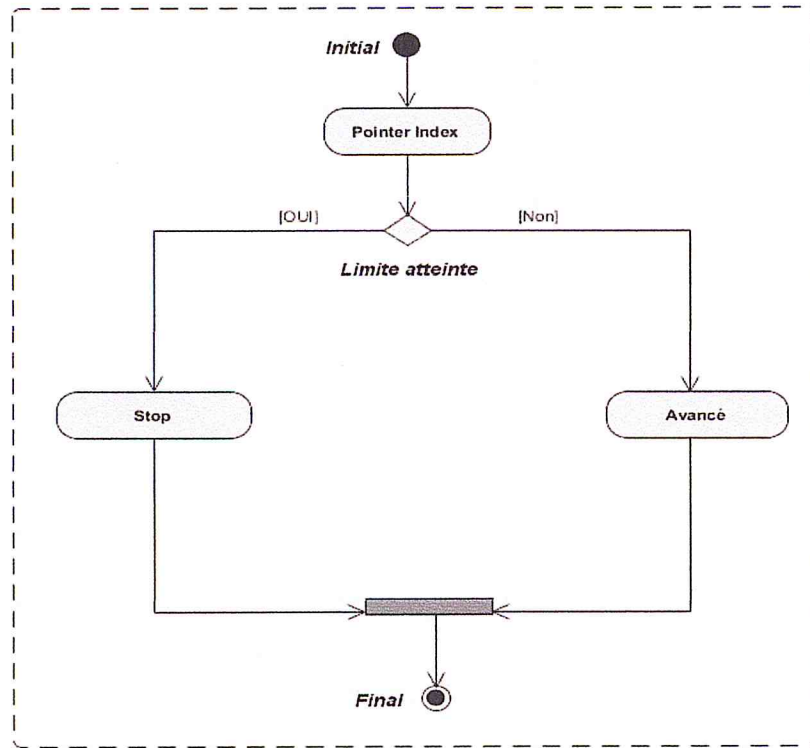


Figure 28 : Diagramme d'activité du CU "Avancer".

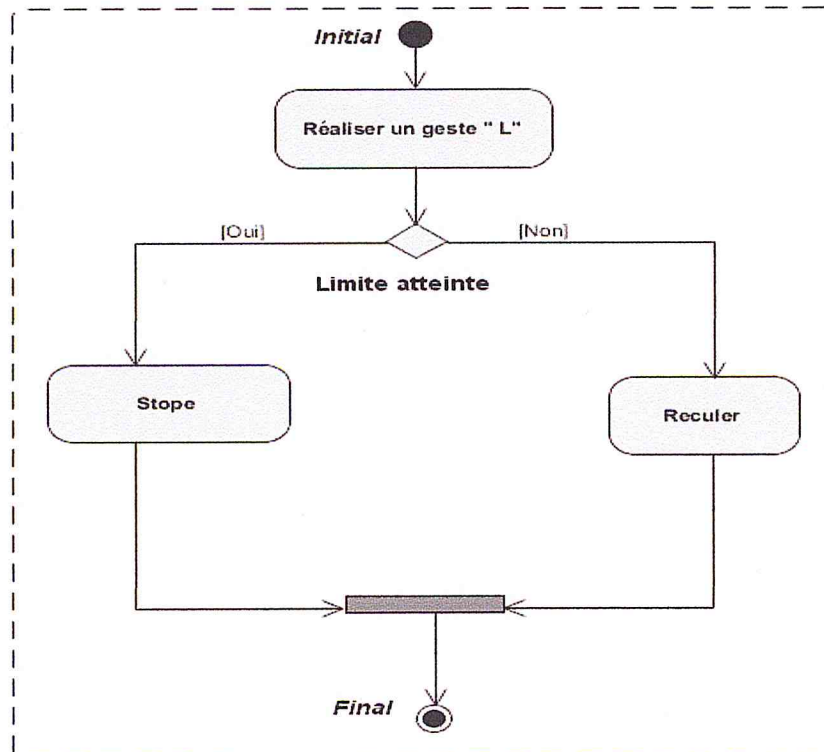


Figure 29 : Diagramme d'activité du CU "Reculer".

## Analyse et conception

Le Deux cas d'utilisation à décrire est « Rotation droite » et « Rotation gauche » celui-ci s'accomplit par une simple action pour l'utilisateur des leap motion. Les diagrammes suivant sa réalisation.

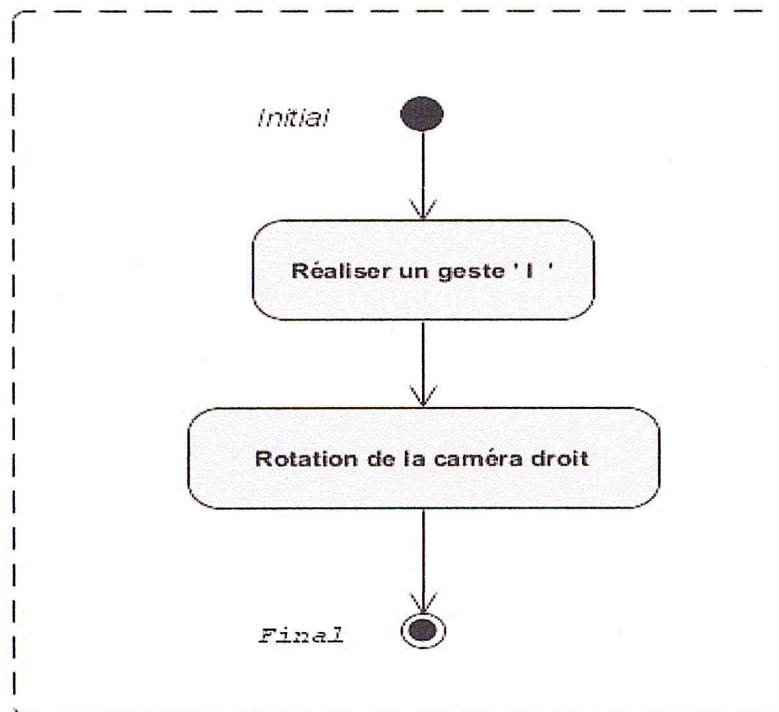


Figure 30 : Diagramme d'activité du CU "Rotation Droite".

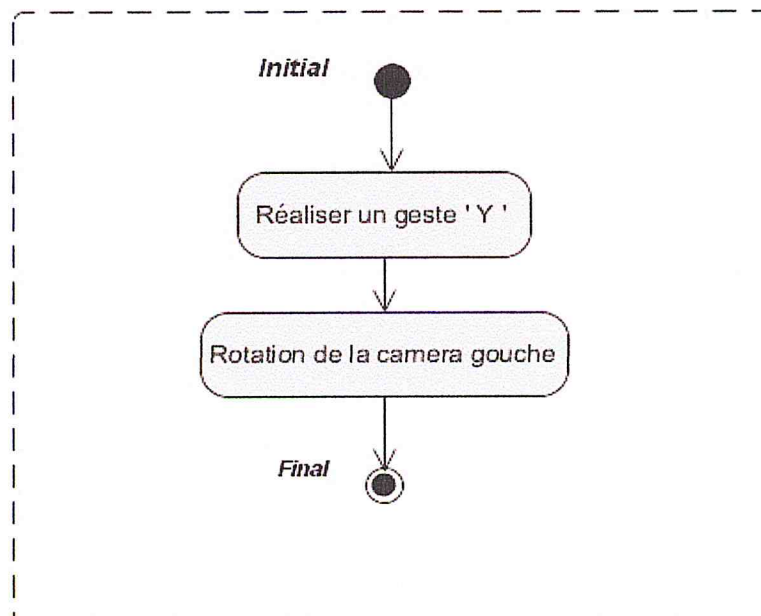


Figure 31 : Diagramme d'activité du CU "Rotation Gauche".



## Analyse et conception

Le dernier cas d'utilisation à décrire est « s'orienter par le regard » pour l'utilisateur des la kinect, celui-ci s'accomplit par une simple action. Le diagramme suivant illustre sa réalisation.

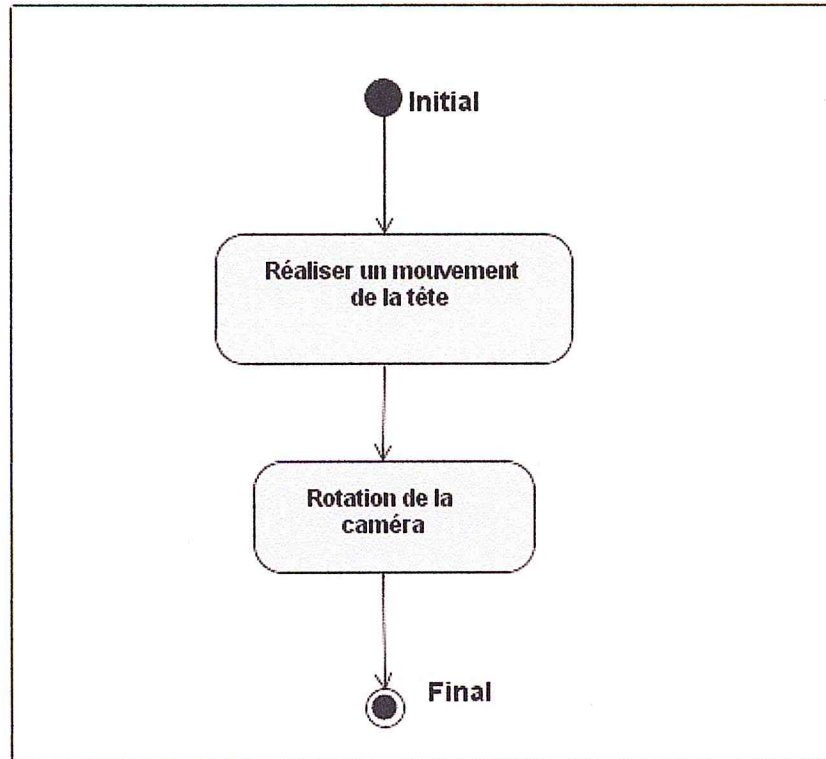


Figure 32 : diagramme d'activité du CU "S'orienter par le regard".

## Diagrammes de séquence

Après avoir présenté les flux d'action nécessaires à la réalisation de chaque cas d'utilisation, proposé par notre système, par les diagrammes d'activité, nous allons détailler les interactions et les messages circulants entre les différents composants du système à travers les diagrammes de séquence.

Dans le diagramme suivant, nous associons la sélection et la manipulation d'objet car pour manipuler un objet virtuel, il suffit de l'attacher à la main virtuelle.

Afin de sélectionner/Manipuler un objet,

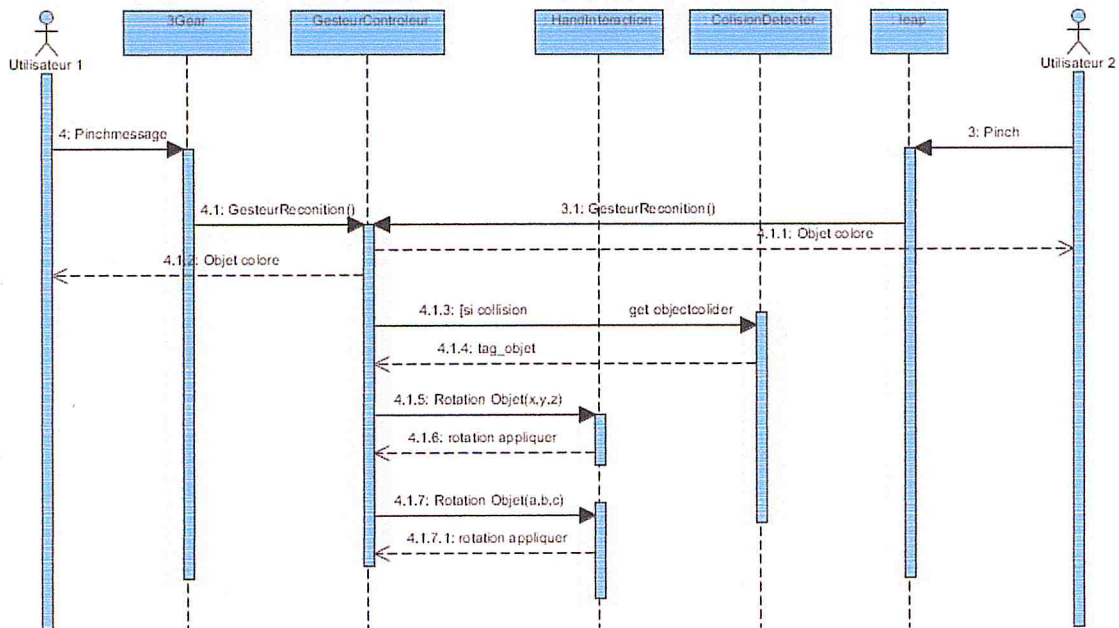


Figure : Diagramme de séquence de la sélection et manipulation d'objet.

Pour avancer dans l'environnement virtuel collaborative, l'utilisateur des la kinect initie l'interaction en pointant son index. De la même manière que la séquence précédente, 3Gear reconnait le geste (leap pour l'utilisateur des leap motion ), ce qui permet au contrôleur d'interaction de lancer la fonction « MoveHands » avec le paramètre « avancer » et d'envoyer un message « MoveCamera » au contrôleur de la caméra (CameraControler) afin de faire avancer le point de vue dans l'environnement virtuel proportionnellement à l'avancement des mains virtuelles et ceci tant qu'aucune limite de l'environnement n'est atteinte.

Le diagramme de séquence suivant illustre cette interaction.

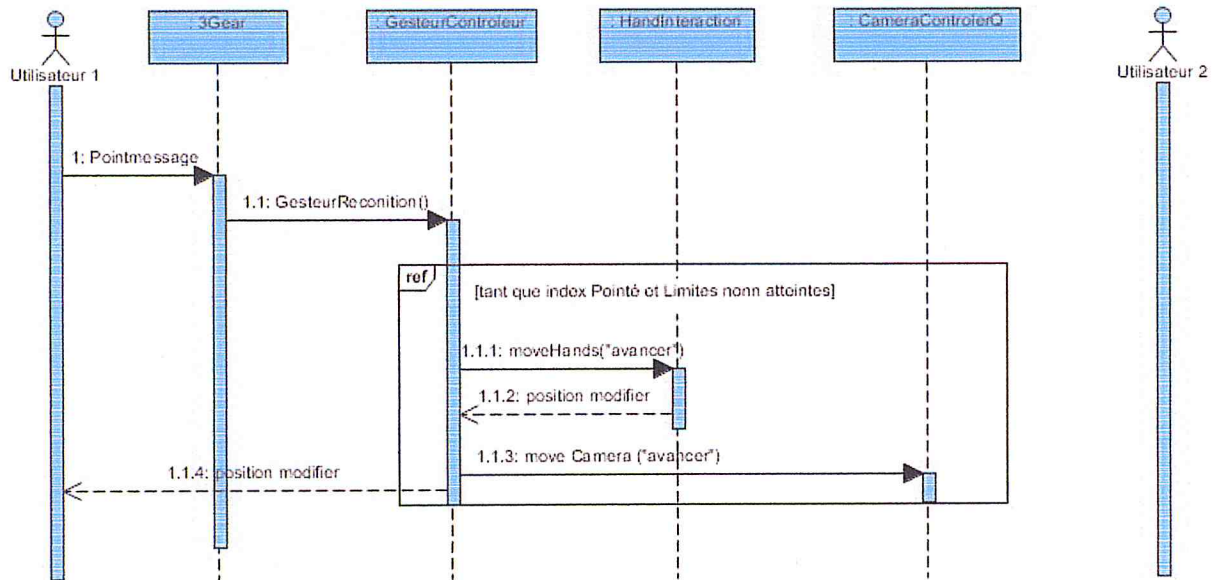


Figure : Diagramme de séquence du CU "Avancer" utilisateur kinect .

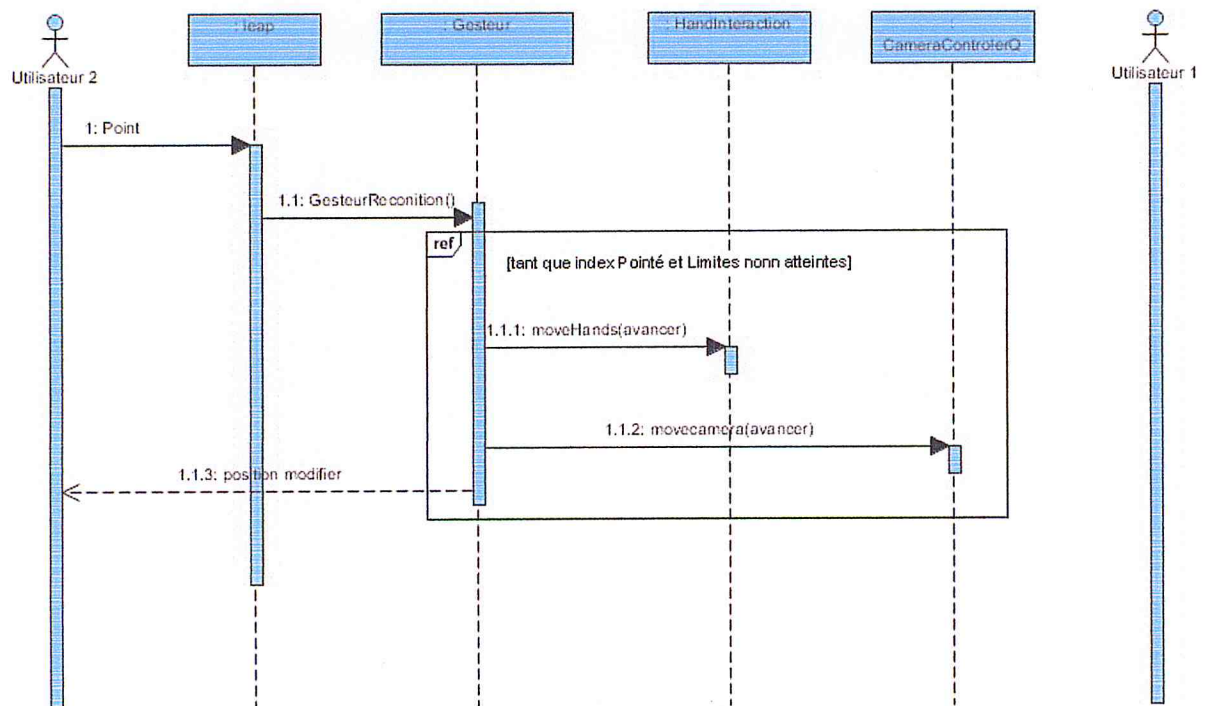


Figure :: Diagramme de séquence du CU "Avancer" utilisateur de leap motion.

Le diagramme de séquence suivant décrit les interactions et les messages circulant entre les différents composants du système pour reculer dans l'environnement virtuel. Cette séquence

d'interaction est similaire à la précédente, à la différence que l'utilisateur initie cette action par un geste en « L », ce qui correspond à l'appel de la fonction « moveHands » du contrôleur de l'interaction et « moveCamera » du contrôleur de la caméra avec le paramètre « reculer » sous la même condition qu'aucune limite de l'environnement n'est atteinte.

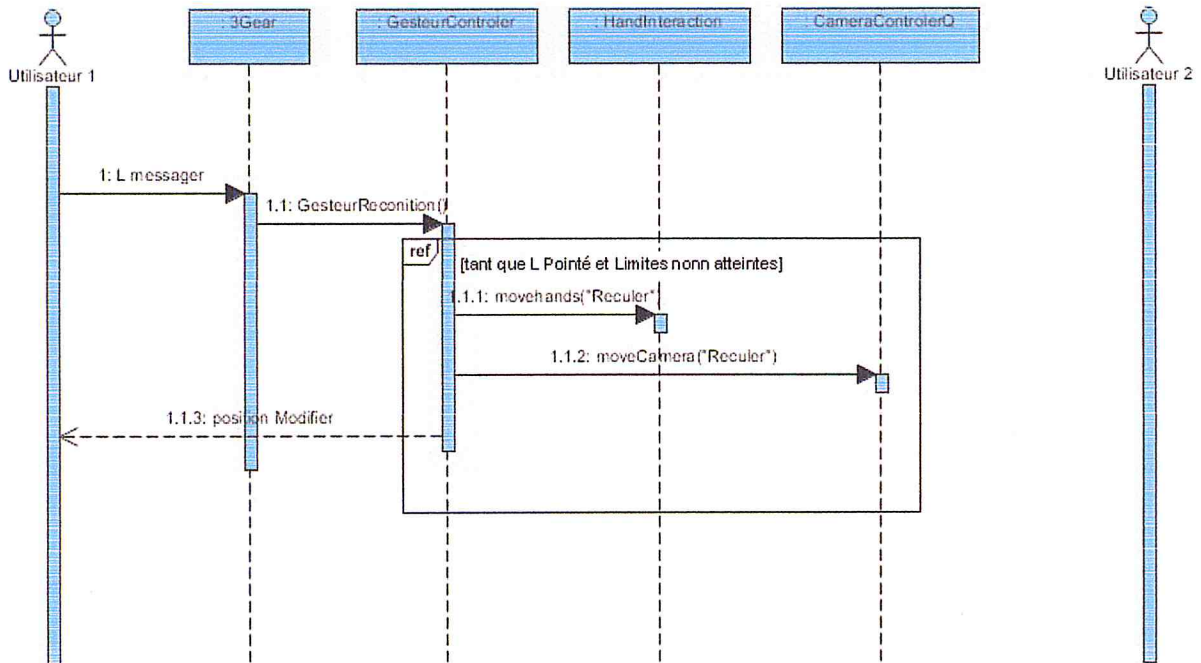


Figure : Diagramme de séquence du CU "Reculer". utilisateur de la kinect.

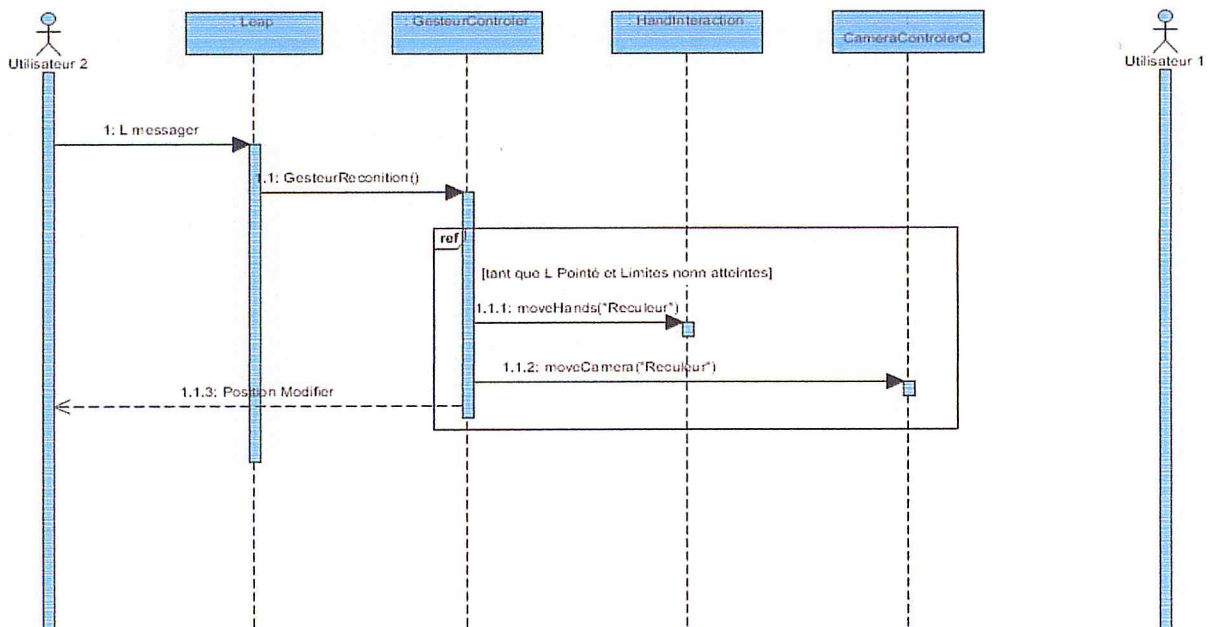


Figure : Diagramme de séquence du CU "Reculer". utilisateur de leap motion

## Analyse et conception

Le dernier cas d'utilisation à décrire est l'orientation par la direction du regard. Dès que l'utilisateur des la kinect fait un mouvement de la tête pour changer l'angle de vue, le composant « Vuzix » calcule l'intensité du mouvement effectué et appelle la fonction « rotateCamera » du contrôleur de la caméra afin d'orienter cette dernière selon la direction du regard de l'utilisateur.

Le diagramme de séquence suivant décrit ce processus.

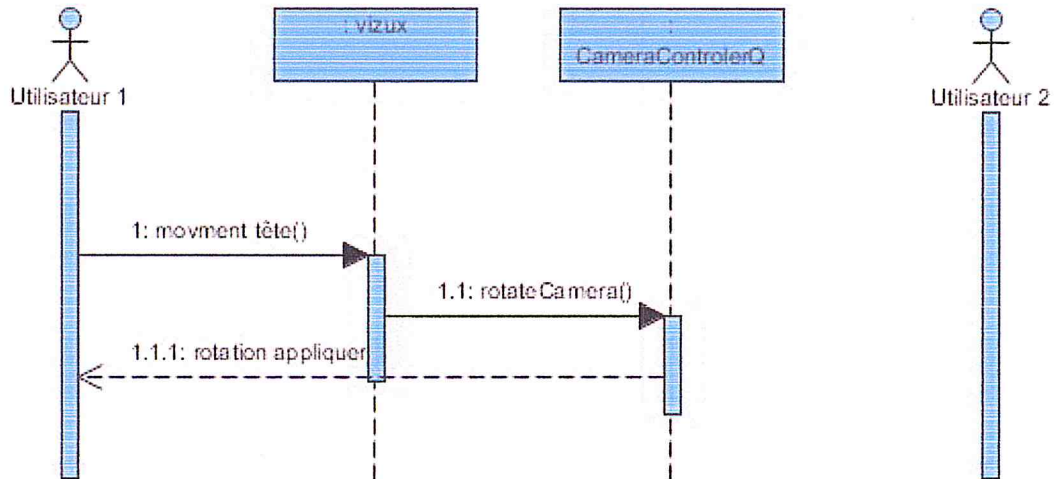


Figure : Diagramme de séquence du CU "S'orienter par le regard".

## 8. Conclusion

L'analyse est une étape nécessaire avant la conception d'une application. En exploitant ce concept nous avons pu décrire notre système sous différents aspects : technique, fonctionnel et comportemental, définissant ainsi les contraintes et les objectifs de l'application à réaliser.

Nous avons ensuite présenté à travers ce chapitre la conception de notre système qui est primordiale lors du développement, ce qui nous permet de passer à la mise en œuvre qui sera l'objet du chapitre suivant.

A l'issue de ce chapitre, nous avons présenté les différents processus nécessaires à la réalisation de notre application et cela, en passant par deux phases ; la première a été analytique du point de vue utilisateurs et les opérations qu'il est apte à réaliser ainsi que les besoins d'une telle application. La seconde nous a permis de détailler le processus de conception, objet d'une progressive, à travers des diagrammes UML de cas d'utilisation, de séquence et de classe.

Le chapitre suivant sera consacré à la réalisation de notre système, en présentant les outils de développement utilisés et les différentes fonctionnalités de notre application à travers ses différentes interfaces.

## **Chapitre 5 : Réalisation et tests**

## 1. Introduction

L'objectif de notre travail de master consiste à concevoir puis réaliser un système capable de reconnaître des commandes gestuelles dans le but d'interagir avec des objets 3D dans un environnement de RV. Dans le chapitre précédent, nous avons présenté la conception de notre système, dans le présent chapitre, nous allons aborder la partie réalisation ainsi que les scénarios sur lesquels nous nous sommes basés pour la mise en œuvre et les tests de notre application. En effet, le système implémenté, devra reconnaître des gestes simples et préalablement définies et les affectera par la suite à des actions au niveau de l'environnement 3D, ces actions reflètent des tâches d'interaction 3D.

### Objectif:

L'objectif de notre travail consiste à réaliser un système de réalité virtuelle collaborative permettant à deux utilisateurs de manipuler des objets 3D partagés.

Proposition d'un modèle de co-manipulation d'objets permettant d'organiser et de gérer les actions des différents utilisateurs afin de réaliser des tâches communes.



## 2.1. Choix de techniques d'interaction 3d

Cependant, de nombreuses techniques de sélection et de manipulation existent, il faudra donc en choisir deux technique adéquate au cadre applicatif défini, pour ce faire nous avons réalisé une comparaison entre ces dernières en se basant sur les avantages et inconvénients de chacune, ainsi que certains critères tel que la modalité d'interaction, un critère qui dépend des spécifications fonctionnelles de l'application.

Le tableau suivant résume la comparaison effectuée entre les différentes techniques d'interaction 3D gestuelles.

	Avantages	Inconvénients
Direction de la main	+ Simplicité. + Permet de se déplacer et de regarder dans différentes directions. + Bonne pour les mouvements relatifs.	- complexe cognitivement. - moins précis. - moins simple à apprendre.
Gestes de la main	+ Interaction très naturelle et libre. + Facile à comprendre.	- Espace de sélection limitée. - Difficulté avec les objets petits ou lointains.
MVS	+ Permet la sélection et la manipulation des objets proches et éloignés.	- lorsque la distance est importante, les mouvements de faible amplitude de la main réelle se traduisent en mouvements de forte amplitude. - La manipulation de la main Virtuelle à grande distance devient imprécise voir difficile.
Go-Go	+ Permet la sélection et la manipulation des objets proches et éloignés.	- lorsque la distance est importante, les mouvements de faible amplitude de la main réelle se traduisent en mouvements de forte amplitude. - La manipulation de la main Virtuelle à grande distance devient imprécise voir difficile.
RayCasting	+ Approche très rapide de la cible. + Geste naturel de pointage.	- Difficultés avec objets petits ou lointains. - Difficulté à être précis.
FlashLight	+ Approche très rapide de la cible. + Geste naturel de pointage	- La sélection d'objets devient très difficile quand ceux-ci

	+ permet la sélection. d'objets de petite tailles ou lointains.	sont groupés entre eux.
SWG	+ Permet la sélection d'objets éloignés.	- Difficulté à être précis lors de la sélection. - Difficultés avec la sélection objets petits ou lointains. - Pas très naturelle.
Homer	+ Approche très rapide de la cible. + Geste naturel de pointage. + Espace de sélection non limité.	- Difficulté à être précis lors de la sélection. - Difficultés avec la sélection objets petits ou lointains. - Apprentissage utilisateur complexe.
Voodoo Dolls	+ Permet de manipuler les objets articulés en mouvement libre dans l'environnement.	- Manipulation complexe. - Moins naturelle. - Nécessite des gants numériques pour la reconnaissance de gestes.

Tableau 9 : Tableau comparatif des techniques d'interaction gestuelles.

Les techniques choisies permettent la sélection et la manipulation d'objets mais celle-ci est limitée lorsque les objets virtuels sont lointains. Nous avons donc ajouté à celle-ci, une technique de navigation qui permet à les utilisateurs de se déplacer (d'avancer et de reculer) dans l'environnement virtuel c en utilisant des gestes de la main afin d'atteindre les objets lointains, ainsi que la possibilité de changer l'angle de vue dans l'environnement virtuel selon la direction du regard de le premier utilisateur (en utilisant des gestes de la main de la deuxième utilisateur pour changer l'angle de vue, Rotation gaucher, Rotation Droite) dans l'environnement virtuel.

## 2.2. Les gestes à reconnaître

Les tâches d'interaction souhaitées qui sont présentés dans le point précédent, nécessitent la reconnaissance de quelques gestes. Ainsi à chaque geste nous associons une certaine action.

Le tableau suivant présente la correspondance entre les gestes et les actions de utilisateur 1 et 2.

Geste	Tâche d'interaction	Action
Index pointé	Navigation	Avancer dans l'environnement virtuel
Geste en « L »	Navigation	Reculer dans l'environnement virtuel
Pincement « Pinch »	Sélection/Manipulation	Sélectionner un objet virtuel en l'attachant à la main pour sa manipulation
Geste en « I »	Navigation	Orientation gauche dans l'environnement virtuel
Geste en « Y »	Navigation	Orientation droite dans l'environnement virtuel

Tableau 10 : Correspondance entre les gestes, les tâches et les actions

Nous avons donc utilisé Cinq postures qui sont les suivantes :

- Index pointé : ce geste est reconnu lorsque l'utilisateur pointe son index en fermant le reste de ses doigts.
- Geste en « L » : ce geste est reconnu lorsque l'utilisateur réalise un angle droit entre son pouce et son index en fermant le reste de ces doigts.
- Pincement : « Pinch » en anglais, ce geste est reconnu lorsque l'utilisateur joint son pouce à son index en mettant le reste de ses doigts au même niveau que son index.
- Geste en « I » : , ce geste est reconnu lorsque l'utilisateur ouvrir son pouce en fermant le reste de ses doigts.
- Geste en « Y » : , ce geste est reconnu lorsque l'utilisateur ouvrir son pouce et son auriculaire en fermant le reste de ses doigts.

La figure suivante illustre les gestes en images :

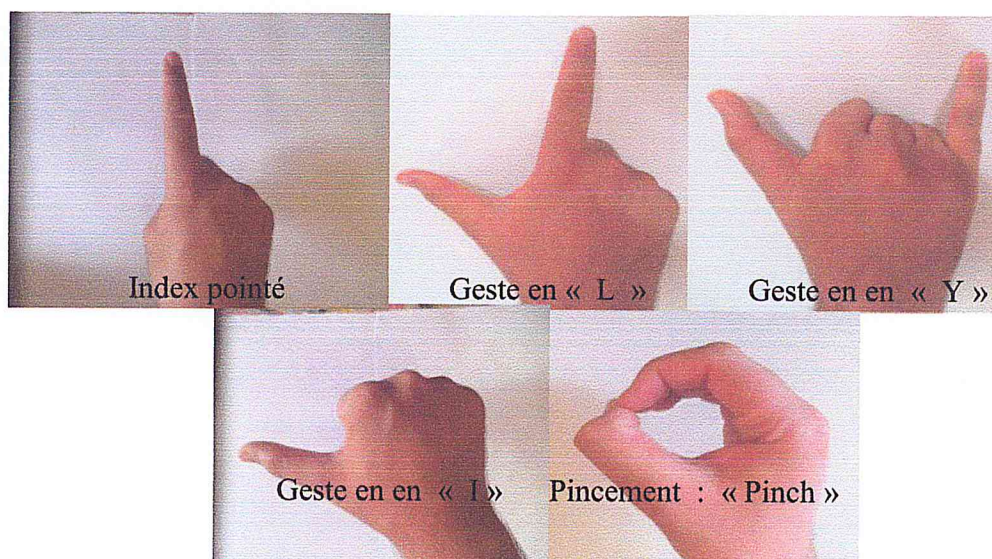


Figure 37 :Les gestes à reconnaître pour les deux utilisateurs.

## Etude des besoins

### 3. Environnement logiciel de développement

Afin de réaliser l'application souhaitée en se basant sur l'architecture et la conception présentées, nous avons exploité Cinq outils logiciels principaux que nous présentons dans ce qui suit.

#### 3.2.1. Unity 3D

Unity est un logiciel 3D temps réel et multimédia ainsi qu'un moteur 3D et physique utilisé pour la création de jeux en réseaux, d'animation en temps réel, de contenu interactif comportant de l'audio, de la vidéo et des objets 3D unity offre la possibilité d'écrire des scripts qui permettent de définir le comportement des objets de la scène .deux langages de programmation sont disponibles : javascript et c#. Etant donné que de nombreux scripts, comme la récupération de la représentation 3D des mains Il est capable d'importer de nombreux formats 3D (Maya, Cinema 4D, Cheetah3D, FBX), des ressources variées : des textures Photoshop, PNG, TIFF, audio, vidéo)

Unity 3D dispose également d'une interface graphique riche et facilement personnalisable. Elle permet de travailler sur le jeu de manière interactive.

L'interface graphique d'Unity 3D se décompose en cinq panneaux (figure 37) :

- 1-Scène :interface pour éditer les éléments de notre scène (ajout des objets, réglages des positions, orientations...)
- 2-Hierarchy : graphe de la scène en cours (ensemble des objets et leurs relations)
- 3-Inspector : détails des composants des objets et modification possible des valeurs
- 4-Project : contenu (scènes, packages, prefabs, matériaux, scripts...) disponible pour intégration dans l'application
- 5-Game : exécution de l'application lancement et arrêt par bouton Play (éviter Pause) ; l'arrêt revient aux conditions initiales

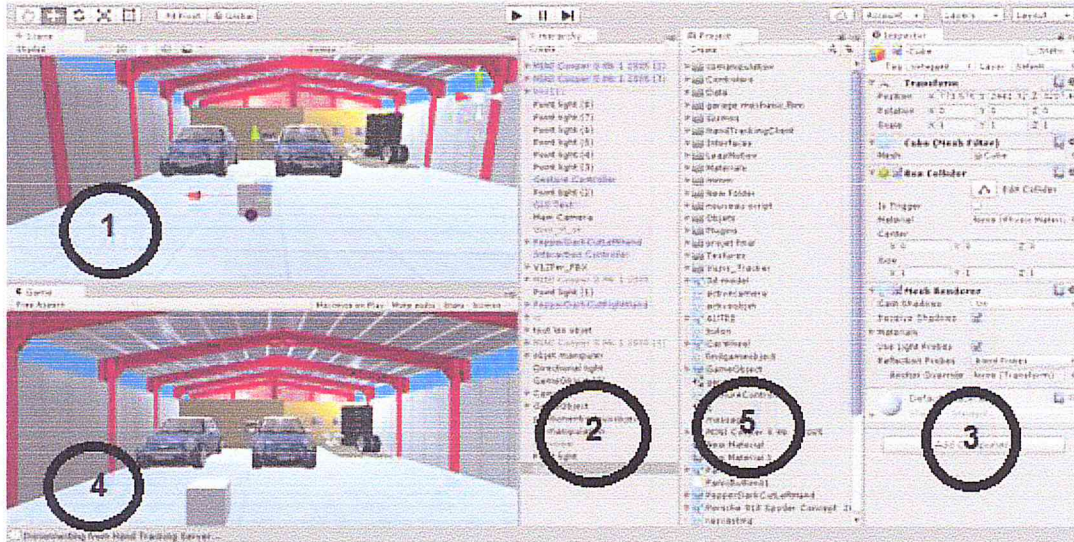


Figure 38: interface graphique de Unity 3D.

### Unity et c# :

la partie développée sous unity se décompose en plusieurs étapes. la particularité de la programmation sous unity réside dans le fait que nous programmons des scripts décrivant le comportement des objets de la scène (les GameObjects) et non des classes pouvant s'appeler entre elles .chaque script est une classe dérivant d'une classe virtuelle propre à unity nommée MonoBehaviour, qui implémente une méthode Start(), appelé au lancement de l'application, et une méthode Update() appelée a chaque frame. d'autres méthodes implémentées dans MonoBehaviour sont également disponibles, elles sont toutes adaptées au développement d'une application graphique

### 3.3. Blender 3D

3D [Blender, 2016], est un logiciel de modélisation et d'animation 3D, développé par la société Blender. C'est l'un des géants dans le domaine 3D, ses projets peuvent être exportés afin de les réutiliser par d'autres logiciels qui supportent les modèles 3D comme Unity 3D.

Blender 3D offre une interface assez complexe, mais propose de nombreux outils de modélisation professionnels. Nous présentons dans ce qui suit les composants principaux de cette interface graphique.

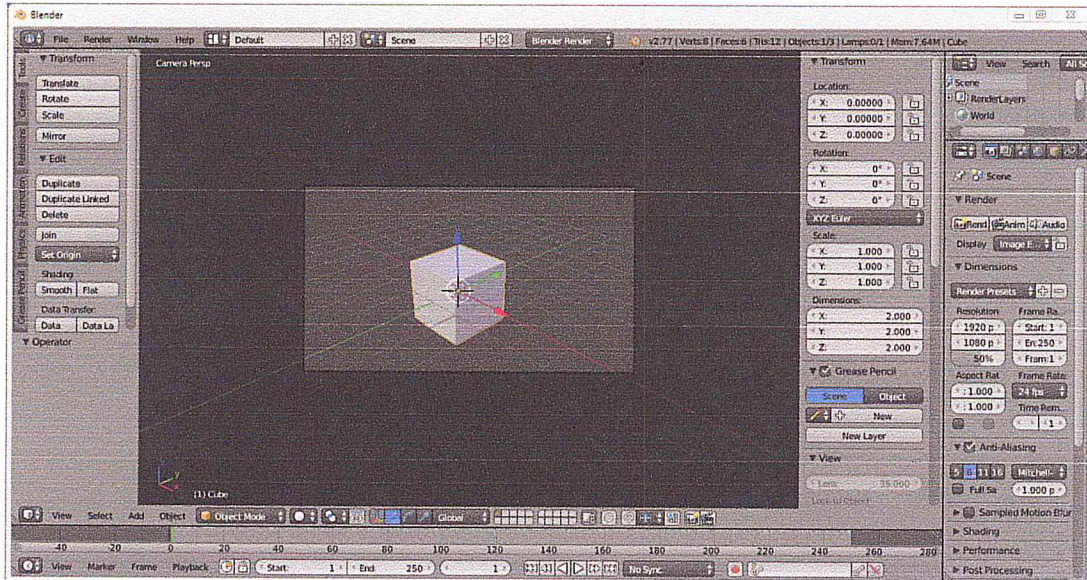


Figure 39 : Interface graphique de Blender 3D.

### 3.4. Three Gear System (Nimble SDK)

#### 3.4.1. Présentation

Three Gear [NimbleVR, 2015], est un système de suivi des mains, conçu pour suivre la position et l'orientation des mains d'un utilisateur ainsi que les postures des dix doigts. Nimble SDK supporte de nombreuses postures parmi elles le pincement et le pointage de l'index.

Ce système fonctionne mieux lorsque les caméras sont superposées au-dessus des mains de l'utilisateur de façon à ce qu'elles aient un champ de vision vers le bas (Voir figure 39). Aussi un environnement de bureau est également conseillé pour faciliter l'interaction avec des objets virtuels. Cette configuration permet à l'utilisateur de reposer ses mains sur le bureau à chaque fois qu'il le souhaite.



Figure 40 : environnement conseillé par Three Gear Systems

La version de Nimble SDK que nous avons utilisé se focalise sur un suivi robuste de la main et sur la reconnaissance de quelques gestes, en particulier le pincement (Pinch). Ce geste est un bon moyen pour les utilisateurs de sélectionner ou de "saisir" des objets virtuels. Il est analogue à un clic de souris.

Three Gear Systems reconnaît et reproduit certains gestes mieux que d'autres, car celui-ci repose sur une base de données de postures, ainsi une posture qui ne fait pas partie de la base de données produit des résultats moins satisfaisants en matière de suivi.

### 3.4.3. Implémentation

Nimble SDK exploite le principe de l'architecture client/serveur et joue le rôle de serveur dans celle-ci. C'est donc un serveur de suivi des mains et de reconnaissance de gestes qui communique avec les applications clientes par un protocole réseau.

Une fois une application cliente connectée au serveur, la bibliothèque reçoit des messages de cette dernière à chaque événement (gestes ou mouvement de la main), et émet à son tour des messages contenant des informations relatives à la main (Position, gestes, etc).

Nimble SDK a été développé afin d'être exploité par une large communauté de développeurs. En effet, il utilise un simple protocole de communication réseau et communique avec les applications clientes à travers les sockets, ce qui signifie que celles-ci peuvent être développées sous n'importe quel langage de programmation.

Three Gear Systems fourni des exemples de librairie sous Java, C++ et C# qui peuvent être directement exploités afin de communiquer avec le serveur. Cependant si l'application cliente est écrite dans un autre langage que ceux-là, il est nécessaire d'implémenter le protocole de communication de Nimble SDK dans le langage en question.

### 3.5 Leap Motion controller panel

Le contrôleur Leap Motion est un petit boîtier de quelques centimètres qui se branche à l'ordinateur par USB, créé par la société du même nom fondée en 2010 et localisée à San Francisco.



Figure 41 : Leap Motion

Il est composé de 2 caméras et de 3 LEDs infrarouges qui délivrent des images brutes traitées ensuite par le CPU. Il est capable de suivre les mains et les doigts ou les outils similaires de l'utilisateur à une précision de 0,01 millimètre dans l'espace. Son champ de vision est une pyramide inversée d'environ 1 mètre, bien qu'au-delà de 60 centimètres au-dessus de l'appareil, on perde beaucoup de précision.

La version développeur est livrée avec un kit de développement (SDK) disponible pour plusieurs langages de programmation (Java, C#, C++, Python, JavaScript, ObjectiveC) qui permet de développer des applications compatibles. Grâce au SDK, le Leap fournit toutes sortes d'informations sur les mains et doigts/outils de l'utilisateur, telles que la position 3D de la paume de la main, la position 3D des doigts, la vitesse de ces points, la longueur des doigts, la rotation/translation de la main d'une frame à une autre, etc...

Ces informations sont délivrées si disponibles à chaque frame. Le frame rate du Leap varie au cours du temps selon la complexité des calculs à effectuer sur les frames, et oscille entre 200 et 300 Hertz (frames par seconde).

Le périphérique est également livré avec un visualiseur dont voici une capture d'écran





Figure 42 : Capture d'écran du visualiseur Leap Motion

### 3.6 Exploitation des outils matériels

#### 3.6.1. SDK de la Kinect

Fournit les outils et les API, nécessaires au développement d'applications compatibles Kinect pour Microsoft Windows [Microsoft, 2015], principalement « DeveloperToolkit » qui contient des démos, des exemples et des codes sources, ainsi que des outils pratiques tels que Kinect Fusion, Kinect Interactions, Kinect Studio, et d'autres ressources pour simplifier le développement d'application pour la Kinect Windows. Il inclut :

- Des exemples et codes sources écrits en C++ et C#.
- Kinect Fusion, qui fournit la numérisation 3D des objets et la création de modèles en utilisant le capteur Kinect pour Windows. L'utilisateur peut peindre une scène avec la caméra de la Kinect et simultanément voir, et interagir avec un modèle 3D détaillé de la scène.

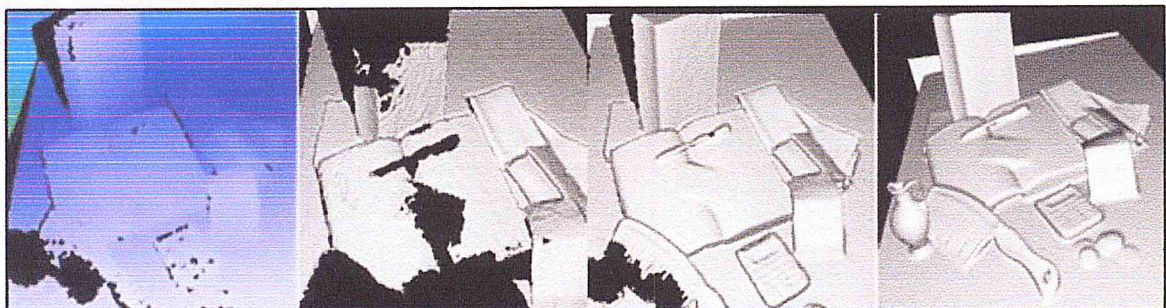


Figure 43: exemple d'utilisation de Kinect fusion [Microsoft, 2015].

- Kinect Controls pour la réutilisation des interactions gestuelles conçus pour le contrôle libre d'applications et d'autres formes d'interactivité.
- Kinect Studio, qui est un outil qui permet d'enregistrer ou lire la profondeur ou les couleurs des flux de données à partir de la Kinect. Cet outil aide à déboguer le code source, créer des scénarios répétitifs pour tester et enfin pour analyser les performances.
- The Face Tracking SDK, un outil qui analyse l'entrée d'une caméra Kinect, en déduit la position de la tête et les expressions faciales, et fait que les informations soient disponibles à une application en temps réel.

### **3.6.2. Vuzix VR Manager**

Vuzix VR Manager [Vuzix-Corporation, 2014], est une application logicielle qui fournit le suivi de la tête et le support de la vidéo 3D stéréoscopique pour les lunettes Vuzix connectées à un ordinateur.

L'interface du Gestionnaire de VR est divisée en trois sections, calibreur, moniteur, et divers.

### **3.6.3. SDK de leap Motion**

SDK Leap motion version 2.3.1 contient deux bibliothèques de base qui définissent l'API pour les données de suivi Leap Motion. Une bibliothèque est écrit en C ++, le second est écrit dans les classes C.

A partir de là, on peut commencer à intégrer le Leap Motion au sein du logiciel Unity. Pour cela, on va se servir d'un asset Leap Motion présent sur l'Asset Store [assetstore,2016]. A partir de ce dernier, on récupère les scripts, scènes nécessaire à l'utilisation du projet.

## **4. Mise en œuvre**

### **4.1. Mise en place des outils matériels**

#### **4.1.1. Installation**

Afin d'avoir un environnement de développement physique qui permet une interaction pratique avec l'application que nous avons développé tout en respectant les conditions idéales décrites par Three Gear Système, nous avons placé la Kinect à la hauteur d'un bras métallique fixé à une table en verre assez haute, de façon à permettre à un utilisateur d'utiliser l'application dans une position assis sur une chaise ou debout face à la table.

L'image suivante représente l'environnement que nous avons utilisé au centre de développement des technologies avancées (CDTA).

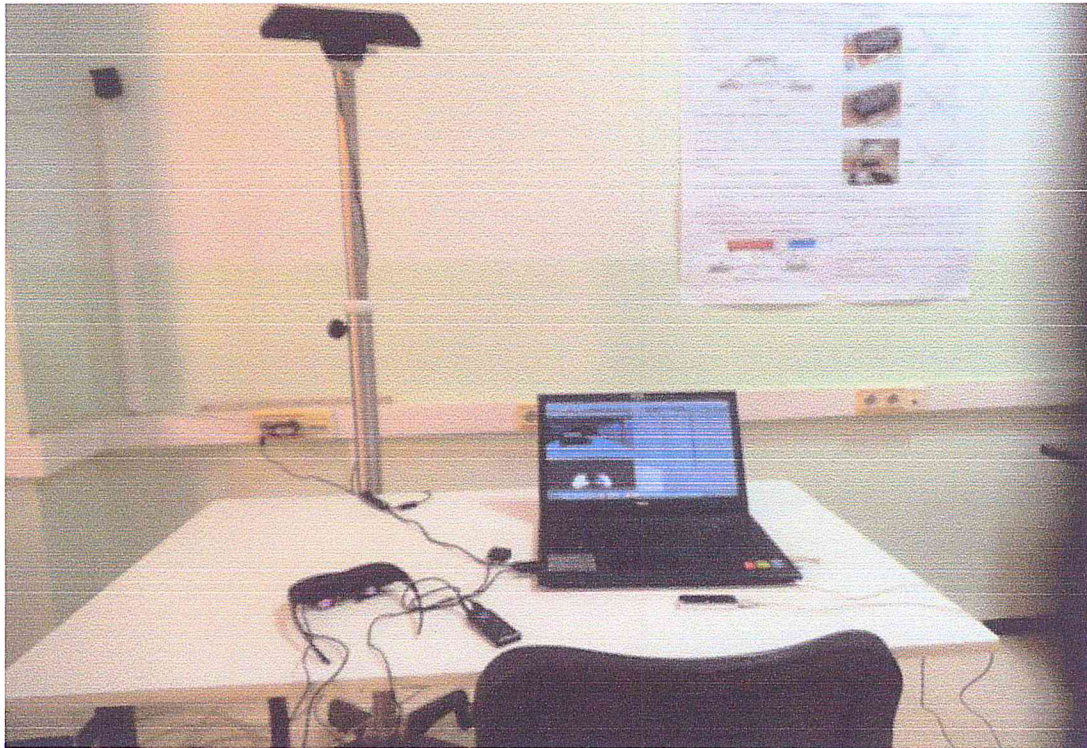


Figure 44: Mise en place des outils matériels.

### **4.2. Réalisation de l'environnement virtuel**

Nous allons créer un garage de maintenance afin de simuler un mécanicien automobile en train de travailler dans son garage. L'utilisateur (mécanicien) doit naviguer dans son garage librement et manipuler différents objets en les soulevant et les déplaçant.

La réalisation d'un environnement virtuel passe par plusieurs étapes : la modélisation, l'éclairage et les textures .

#### **4.2.1. La modélisation**

Dans cette étape, nous avons réalisé le plan et les dimensions de notre scène, ainsi que la création de quelques modèles en 3D en partant de formes géométriques simples que l'on élabore au fur et à mesure pour obtenir les formes appropriées. Pour le cas des objets complexes (moteur, véhicule..) nous avons pu les télécharger de plusieurs sites qui offrent des objets 3D gratuitement.

#### **4.2.3. Les texture :**

Sans les textures, la scène reste opaque et non réaliste. Il est cependant nécessaire de définir des couleurs, effets et des transparences. Le texturing est l'application d'une image ou de plusieurs images 2D sur un objet 3D.

Il existe plusieurs techniques pour donner une texture aux objets. Une des techniques que propose 3Ds Max sont les matériaux. La différence entre les matériaux et les textures, est que les matériaux représentent les objets sur lesquels on pose l'image 2D (ou texture).

La figure suivante représente le résultat obtenu après application des textures dans notre environnement, nous pouvons remarquer que le sol est maintenant couvert par de la dalle de sol donnant ainsi une apparence réaliste au sol.



Figure 45: Application des textures

### 4.2.2. L'éclairage

La phase d'éclairage joue un rôle important dans la création de l'environnement et améliore son rendu et le rend plus réaliste. Afin de donner une atmosphère à la scène, nous avons rajouté des sources de lumières dans notre environnement tel qu'illustré dans la figure suivante.



Figure 46 : Eclairage de l'environnement virtuel.

### **4.3. Représentation des mains virtuelles.**

Afin de représenter les mains virtuelles en 3D nous nous sommes basés sur la structure osseuse de la main.

#### **4.3.1 Représentation des mains virtuelles de la premier utilisateur**

Pour arriver à cette représentation squelettique, nous avons représenté la main sous Unity 3D par un « Game Object » pour chaque main, qui contient les 17 jonctions de la main que nous avons représentée par des sphères en exploitant la notion de hiérarchie des « Game Objects ». La figure suivante montre la hiérarchie de de la main gauche comme exemple.

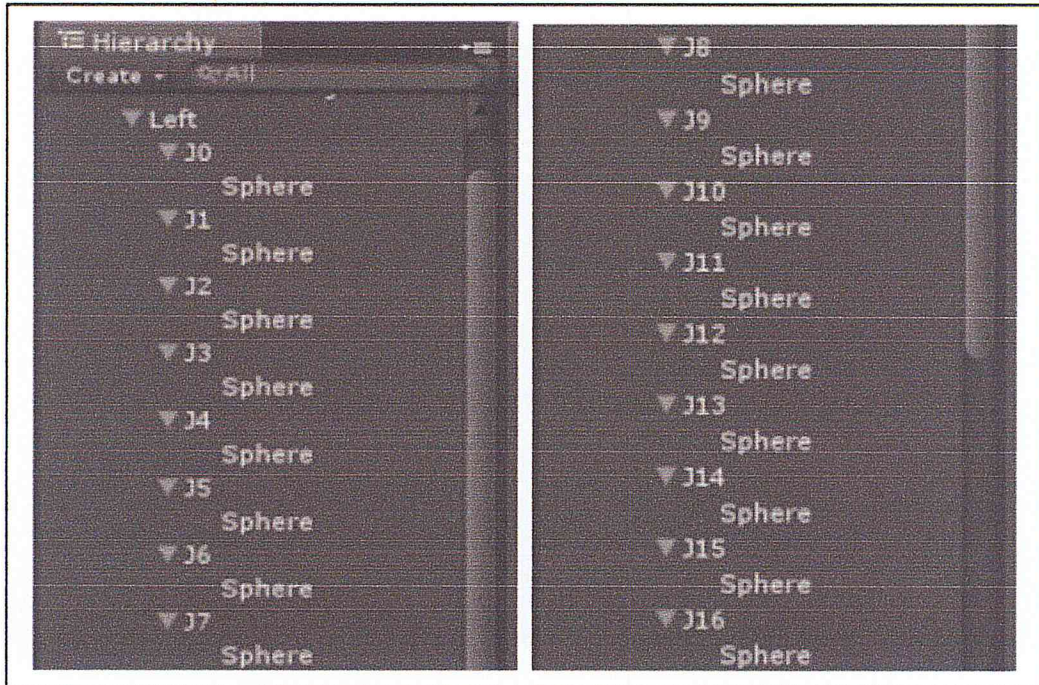


Figure 47 : Hiérarchie de la main gauche sous Unity 3D.

Nous allons maintenant procéder à la modélisation de la forme de la main et son habillage. Unity 3D n'offre pas d'outils de modélisation, mais supporte les maillages de triangles ou de quadrangles, ce qui permet de réaliser n'importe quelle forme. Nous allons utiliser les maillages de triangles puisque le serveur Nimble donne des triangles constituant un maillage modélisant la main de l'utilisateur.

Nous allons donc créer un maillage pour chaque main en utilisant les triangles fournis par le serveur Nimble, ce maillage sera en mise à jour continue de la même manière que la position des sphères représentant les jonctions des mains. Ce sera le « Game Object » représentant la main qui héritera de ce maillage comme composant. Nous obtenons ainsi une modélisation de la main en maillage de triangles tel qu'illustré dans la figure suivante.

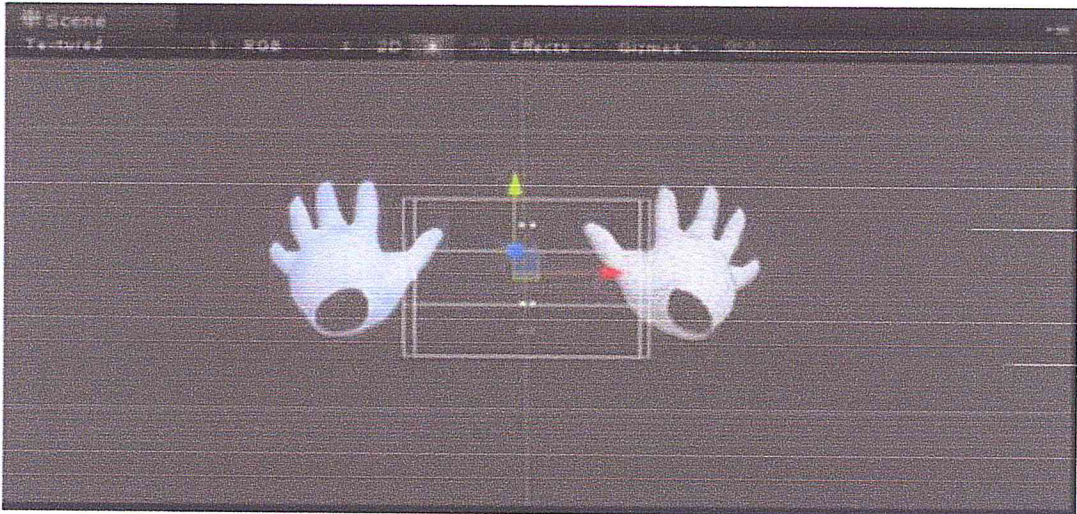


Figure 48: Représentation de la main en maillage de triangles.

Il suffit maintenant d'habiller les mains par une texture pour le remplissage des triangles, la (figure 49) suivante montre le résultat obtenu après l'habillage des deux mains.

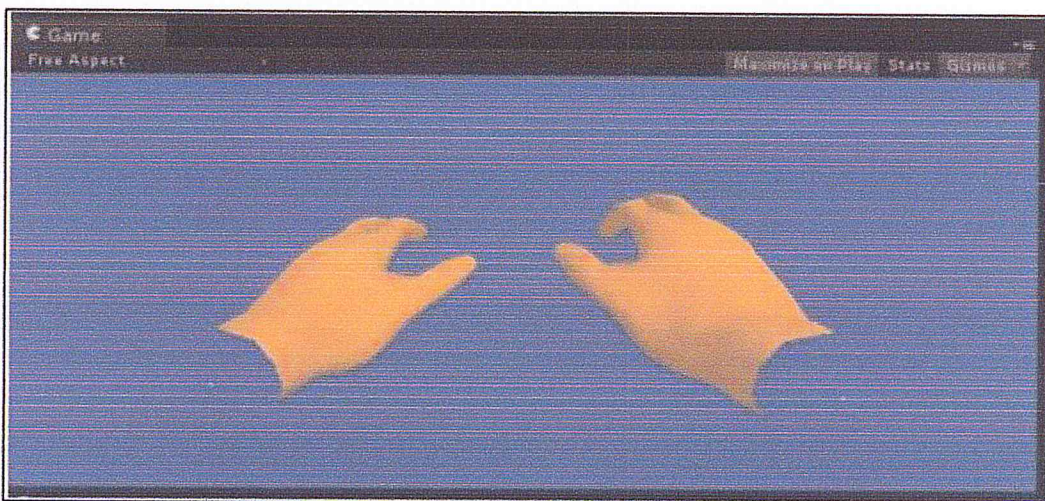


Figure 49 : Représentation finale des mains en 3D.

Grace à la mise à jour continue des données par le serveur Nimble, les mains virtuelles reproduisent les gestes de l'utilisateur en temps réel.

#### 4.3.2 Représentation des mains virtuelles de la Deuxième utilisateur

Pour arriver à cette représentation squelettique, nous avons représenté la main sous Unity 3D par un « Game Object » pour chaque main, qui contient les 17 jonctions de la main que nous avons représentée par des sphères en exploitant la notion de hiérarchie des « Game Objects ». La (figure 50) suivante montre la hiérarchie de la main Droite de leap motion comme exemple.

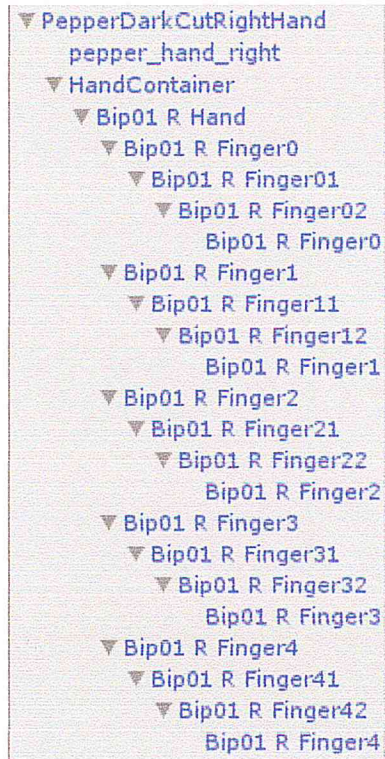


Figure 50 : Hiérarchie de la main de leap motion sous Unity 3D.

Nous allons donc créer un maillage pour chaque main en utilisant les triangles fournis par le serveur leapSvc,

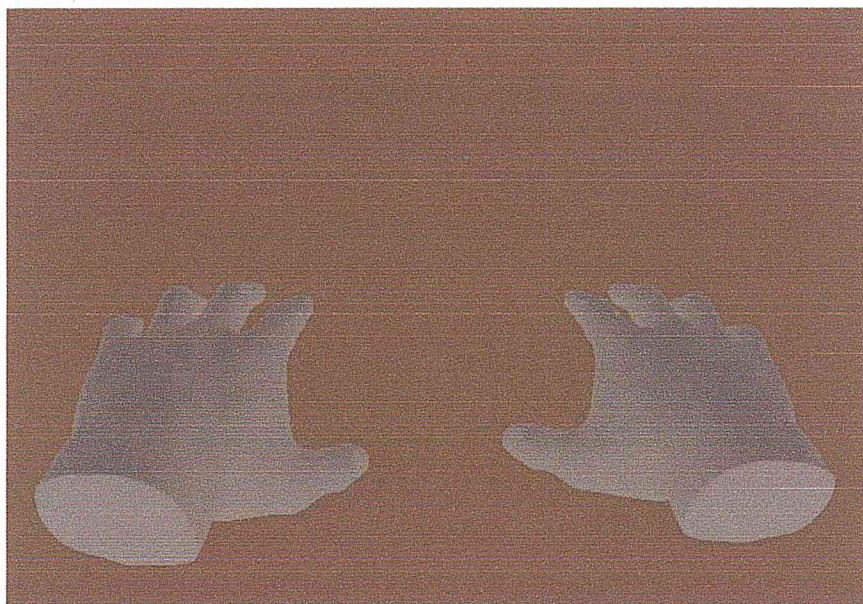


Figure 51 : Représentation de la main de leap motion en maillage

Il suffit maintenant d'habiller les mains par une texture pour le remplissage des triangles, la (figure 52) suivante montre le résultat obtenu après l'habillage des deux mains.



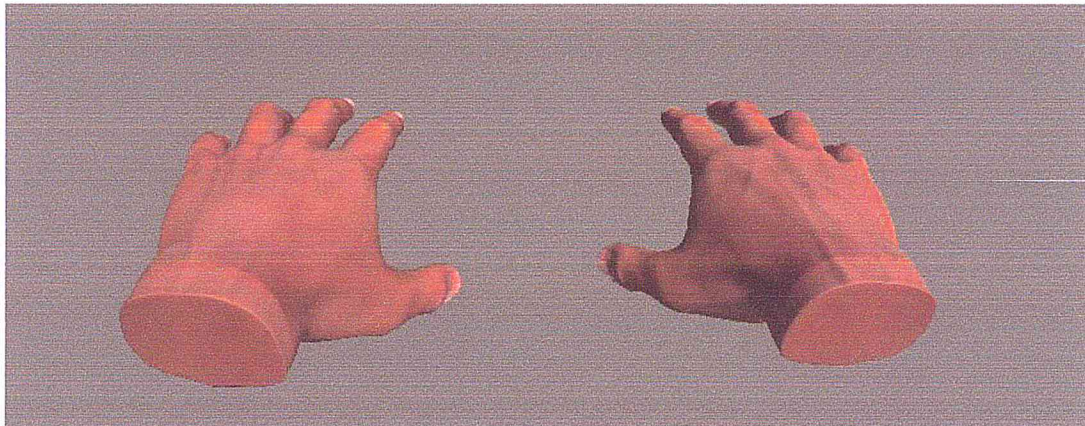


Figure 52 : Représentation finale des mains en 3D.

#### 4.4. Implémentation de l'interaction

Nous avons à présent créé notre environnement virtuel en utilisant blender 3D et représenté les mains virtuelles grâce à Unity 3D. Cependant, notre application finale tournera sous Unity 3D, nous allons donc importer l'environnement que nous avons conçu dans Unity 3D afin d'obtenir une représentation des mains et notre environnement dans une même scène la (figure 53) suivante illustre le résultat obtenu sous Unity 3D.

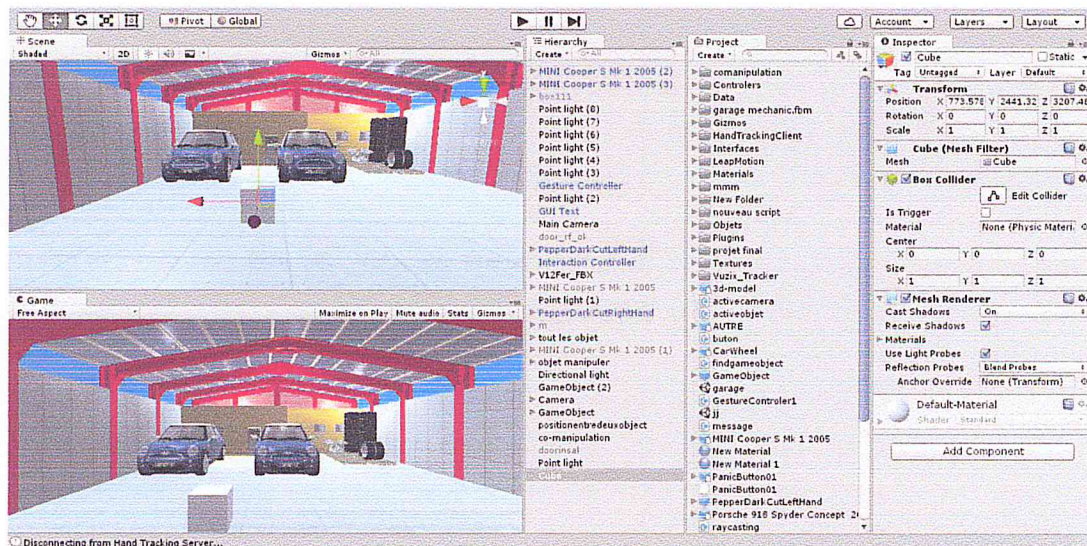


Figure 53 : Intégration de l'environnement virtuel dans Unity 3D.

Afin qu'on puisse exécuter des actions d'assemblage de voiture, il nous fallait un modèle 3D de voiture, pour cela nous avons choisi un modèle 3D de voiture MINI Cooper, notre choix a été basé sur le modèle qui contient les pièces essentielles pour la tâche d'assemblage selon les scénarios que nous avons parler plus tard, aussi parce

que ce modèle nous permet d'interagir avec chaque pièce individuellement, ce qui est très important pour pouvoir sélectionner chaque pièce et la manipuler .

Les objets virtuels correspondants que nous mettons à la disposition de l'utilisateur sont (capot, moteur, port1, port2, roue1, roue2, roue3, roue4).

### 4.4. 1 Sélection et manipulation des objets virtuels

Dans l'interaction 3D, on associe souvent la sélection à la manipulation car pour pouvoir manipuler un objet celui-ci doit être préalablement sélectionné. Pour sélectionner un objet, il suffit que la main virtuelle entre en collision avec l'objet en question de utilisateur 1 ( la raycasting en collision avec l'objet de utilisateur 2).

Afin de mettre en œuvre ce mécanisme, nous avons exploité un des composants d'un « Game Object » qui est le « Collider », celui-ci a pour rôle de détecter les collisions entre les objets. Nous avons donc ajouté ce composant à tous les objets portatifs, ainsi qu'aux sphères définissons les articulations de la main. De cette manière le « Collider » s'occupe d'identifier l'objet en collision avec la main et génère des évènements à chaque entrée ou sortie de collision.

Le « Script » est aussi un composant nécessaire afin d'exploiter les évènements générés par le détecteur de collisions. En effet, nous avons changé la couleur des objets en collision avec la main au bleu de utilisateur 1 et verte de utilisateur 2 grâce à ce deuxième composant cité afin de donner un effet visuel à la sélection.

A ce stade il suffit que l'utilisateur pince l'objet (réalise un geste « Pinch ») pour le sélectionner et le manipuler en toute liberté. Là encore nous avons utilisé une des propriétés du « Game Object » de Unity qui est la relation parent/enfant, afin d'attacher l'objet sélectionné à la main. En d'autres termes, dès que l'utilisateur fait un « pinch », l'objet est affecté à la main comme enfant et héritera donc de toutes les transformations que subira la main et sera donc manipulé en tout confort.

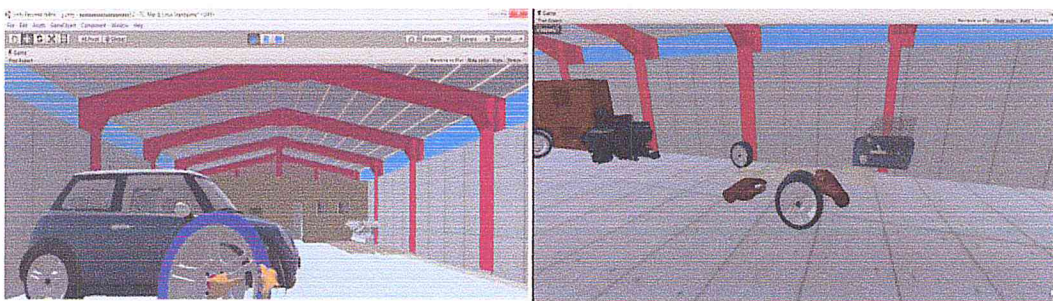


Figure 54 : Sélection et manipulation d'un objet pour les deux utilisateurs.

#### 4.4.2 Navigation

La navigation permet à les utilisateurs immergés dans l'environnement virtuel d'avoir la sensation de se déplacer à l'intérieur de celui-ci. Un autre objectif qui est ceux d'atteindre les objets lointains. En effet la technique de main virtuelle simple ne permet pas la sélection et la manipulation d'objets lointains d'utilisateur 1 en contraire d'utilisateur 2 avec la technique de raycasting.

Nous avons à cet effet implémenté une technique de navigation par les gestes de la main, permettant ainsi à l'utilisateur d'avancer et de reculer dans l'environnement virtuel, et d'atteindre ainsi les objets lointains pour utilisateur 1.

Avancer ou reculer dans l'environnement virtuel signifie le déplacement des mains virtuelles et de la caméra virtuelle tout en gardant le même point de vue afin de donner la sensation de déplacement à l'utilisateur.

Une fonction de déplacement est appelée par script lorsque l'utilisateur pointe son index pour avancer ou fait un geste en « L » pour reculer. L'utilisateur peut aussi aller plus vite si le geste est fait par les deux mains simultanément.

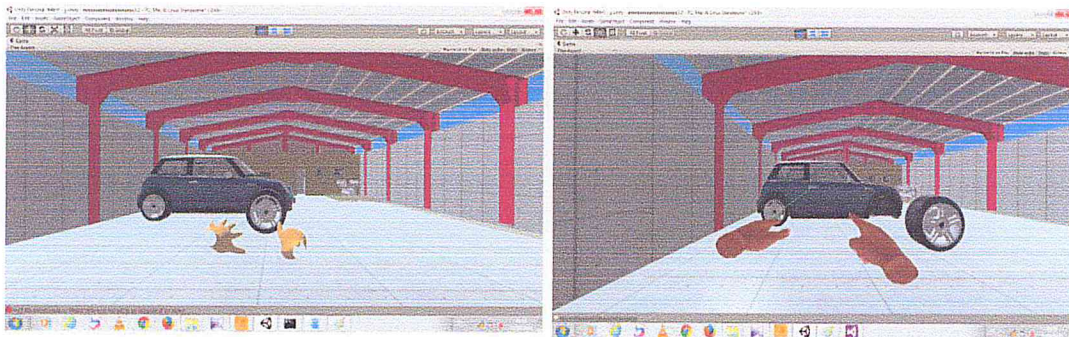


Figure 55 : Index de la main droite pointé pour avancer de deux utilisateurs.

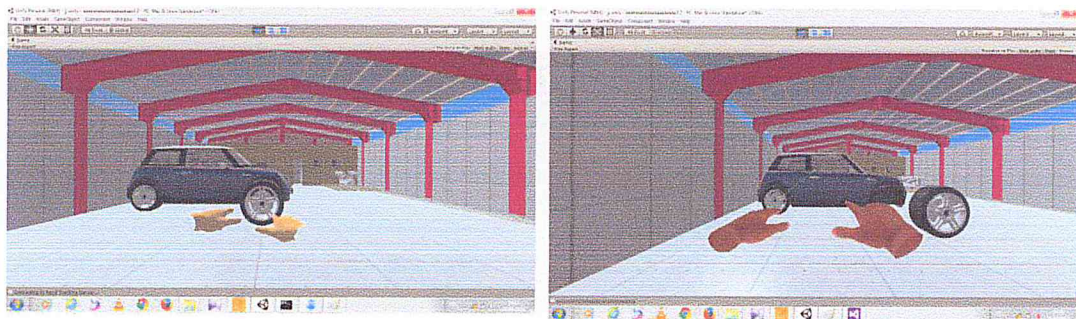


Figure 56 : Geste en "L" de la main droite pour reculer les deux utilisateurs.

A cet effet, un problème se pose : comment définir des limites à notre environnement virtuel et empêcher les mains virtuelles de les franchir ?

Pour résoudre ce problème nous avons affecté des « tags » (un tag est une étiquette qu'on attache à un Game Object) à toutes les limites de notre environnement (mur, sol, etc) que nous avons stocké dans une classe statique : « EnvironnementData », ce qui nous permettra de vérifier l'atteinte de limite à chaque collision de la main avec un objet. Ainsi si la main virtuelle entre en collision avec un objet défini parmi les limites de l'environnement virtuel, celle-ci est alors empêchée d'avancer.

### 4.4.3 Direction

#### Direction par regard du premier utilisateur :

Afin de rendre l'interaction encore plus naturelle, nous avons utilisé les données fournies par le visiocasque pour permettre à l'utilisateur de changer de direction et d'explorer tous les coins de l'environnement virtuel en faisant des rotations de la tête.

Dès que l'utilisateur fait un mouvement de la tête le visiocasque fournit aussi tôt l'intensité du mouvement par rapport aux trois axes de rotation de la tête, qui sera utilisé pour faire une rotation de la caméra permettant ainsi de changer d'angle de vue dans l'environnement virtuel.

Nous donnons en images quelques exemples de changement de la direction du regard dans l'environnement virtuel.



Figure 57 : Changement de la direction droite et gauche pour la première Utilisateur.

#### 4.4.3.1 Direction par les gestes du deuxième utilisateur :

Nous avons utilisé les gestes 'Y' et 'I' pour permettre à l'utilisateur de changer de direction et d'explorer tous les coins de l'environnement virtuel.

Nous donnons en images quelques exemples de changement de la direction dans l'environnement virtuel.

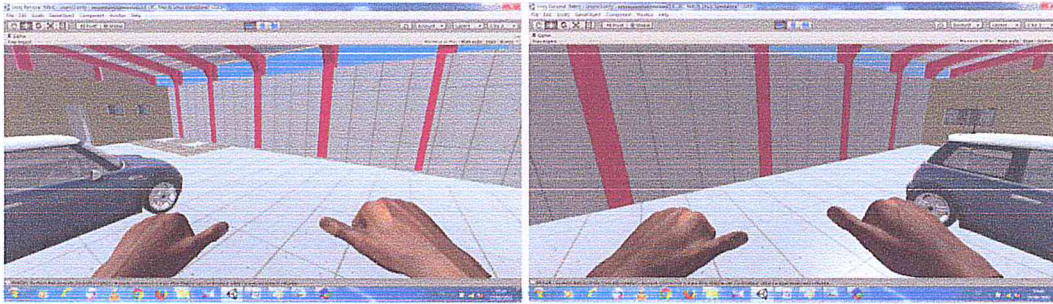


Figure 58 : Changement de la direction gauche et droite pour la désaïme Utilisateur.

### 5. Représentation de l'interaction 3D multiutilisateur

Manipulation collaborative de objets semble en effet nécessaire dans de nombreuses applications différentes de VR, comme pour le prototypage virtuel, simulations de formation, simulations d'assemblage et d'entretien, etc...

Dans de telles tâches collaboratives, tous les utilisateurs de participer à la mouvement appliqué à l'objet manipulé dans le VE la motions distinctes des différents utilisateurs sont utilisés pour définir la mouvement final de l'objet virtuel qui permettent à plusieurs utilisateurs d'interagir avec un objet partagé simultanément séparation des DOF pourrait être utile lorsque les utilisateurs ne disposent pas des même vue de la scène virtuelle

#### 5.1. Scénario1 :

Dans ce scénario nous avons utilisé la co-manipulation synchrone avec séparation des degrés de liberté donc le premier utilisateur modifie trois degrés de libertés concernant la position et le deuxième modifie les trois degrés de libertés concernant l'orientation.



Figure 59: Co-manipulation des Objet Avec les deux Utilisateur(Scénario1).

Dans ce scénario les deux utilisateurs en train déplacé l'objet (capot) à son position initiale dans la voiture. Le premier utilisateur faire la rotation seulement et le deuxième utilisateur faire la translation.

### 5.2. Scénario2 :

Dans ce scénario nous avons utilisé co-manipulation synchrone sur les mêmes degrés de libertés avec accès concurrent aux mêmes degrés de liberté cette technique est utilisé pour gérer les modifications concurrentes nous avons choisie de faire la moyenne d'interaction (translation et orientation) appliqué par chaque utilisateur.

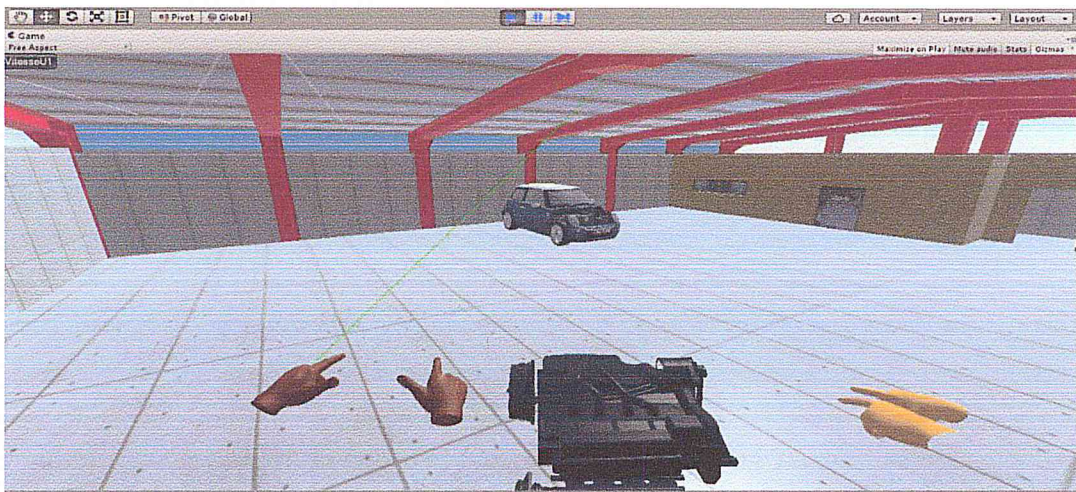


Figure 60: Co-manipulation des Objet Avec les deux Utilisateur(Scénario2).

Dans ce scénario les deux utilisateurs en train de déplacer l'objet (moteur) à son position initiale dans la voiture les deux utilisateurs peuvent Controller la translation et la rotation fait par premier utilisateur

## 6. Conclusion

Nous avons présenté à travers ce dernier chapitre les outils que nous avons utilisé pour la réalisation de notre travail, ainsi que la mise en œuvre détaillée que nous avons suivi afin d'arriver à une application fonctionnelle.

Cependant, en réalisant cette application, nous avons constaté que l'interaction 3D est un domaine où la perfection n'existe pas, c'est-à-dire que le rendu de l'interaction peut toujours être amélioré selon le domaine d'application afin d'être encore plus réaliste.

**Conclusion**  
**générale et perspectives**

## Conclusion générale

Dans ce projet de fin d'étude, notre mission a été de mettre en place un système capable de reconnaître des gestes du main, afin de contrôler des objets 3D dans un environnement collaboratif de RV.

Au fil de ce mémoire, nous avons vu plusieurs aspects, nous avons commencé par une étude bibliographique sur la RV. Ensuite, nous avons présenté les différentes techniques d'interaction existantes dans la RV. Une comparaison de ces techniques a été fait. Enfin, nous nous sommes recentrés sur l'aspect conceptuel et implémentation.

L'objectif de notre travail été d'implémenter une technique d'interaction 3D par les gestes en utilisant la Kinect et leap motion comme dispositif de capture, permettant ainsi de manipuler (Co-manipuler) des objets 3D dans un environnement virtuel.

Le résultat de nos efforts est une représentation des mains en 3D reproduisant les gestes de les utilisateurs en temps réel, les tâches d'interaction principalement implémentées sont la sélection et la manipulation(Co-manipulation).

Par ailleurs, nous avons aussi implémenté la navigation en ce qui concerne la RV afin de donner à l'utilisateur une sensation d'immersion et un contrôle total dans l'environnement dans lequel il évolue, et ceci en exploitant les données de la reconnaissance de gestes pour le déplacement.

Ce projet a donné lieu à un système permettant d'abord représenter une scène par un ensemble d'objets 3D, dans notre cas, il s'agit des différents composants d'un véhicule. Ce système permet également à plusieurs utilisateurs travailler ensemble en partageant l'espace de travail.

Il s'agit de collaborer afin de faire un assemblage de pièces automobile en 3D.

Pour conclure, nous tenons à souligner que ce travail a été une agréable expérience qui nous a permis de mettre en pratique les connaissances que nous avons acquises tout au long de notre cursus à l'USDB. En espérant que ce mémoire sera bénéfique pour ceux qu'ils le consulteront.



Lors de la réalisation de notre projet, nous avons été confrontés à plusieurs problèmes, parmi ceux, l'implémentation de la reconnaissance des geste sous Unity 3D.

Ce travail nous a, non seulement, donné l'opportunité de manipuler des technologies comme Unity3D, Interaction et de connaitre des nouveaux concepts comme la RV, mais aussi permis d'approfondir nos connaissances, en matière d'analyse et de conception des systèmes.

## **Perspectives**

Dans le but d'enrichir notre système, certaines améliorations peuvent être ajoutées par la suite, elles sont définies comme suit :

- Utilisation d'une interaction multimodale, par exemple, la fusion de l'interaction gestuelle et l'interaction vocale à base de reconnaissance des vocal.
- Implémentation de notre solution dans un réseau pour partager la même scène à plusieurs utilisateurs.

# *Bibliographie*

## References bibliographies

- [Azuma, 1997] Azuma, R. T. (1997). "A survey of augmented reality." *Presence* 6(4): 355-385.
- [Aukstakalnis and Blatner, 1992] Aukstakalnis, S. and Blatner, D. (1992). *Silicon mirage : The art and science of virtual reality*. Peachpit Press.
- [Azuma et al., 1997] R.T. Azuma et al. (1997). A survey of augmented reality. *Presence Teleoperators and Virtual Environments*, 6(4) :355–385.
- [Burdea and Coiffet, 1993] Burdea, G. and Coiffet, P. (1993). *La réalité virtuelle*. Edition Hermes Sciences Publication.
- [Biocca et al., 2001] Biocca, F., Harms, C., and Gregg, J. (2001). The networked minds measure of social presence : Pilot test of the factor structure and concurrent validity.
- [Caudell et Mizell, 1992] Caudell, T. et Mizell, D. (1992). Augmented reality: An application of heads-up display technology to manual manufacturing processes. *proc. Hawaii International Conf. on System Science*, 2:659-669.
- [Fuchs et al., 2003] Fuchs, P. and G. Moreau (2003). "Le traité de la réalité virtuelle—Volume 1: fondements et interfaces comportementales." Les presses de l'École des Mines.
- [Bowman et al., 1997] Doug A. Bowman, David Koller, et Larry F. Hodges (1997). Travel in immersive virtual environments: An evaluation of viewpoint motion control techniques. Dans *VRAIS '97 : Proceedings of the 1997 Virtual Reality Annual International Symposium (VRAIS '97)*, page 45, Washington, DC, USA. IEEE Computer Society.
- [Bowman, 1999] D. A. Bowman (1999). *Interaction Techniques for Common Tasks in Immersive Virtual Environments : Design, Evaluation, and Application*. thèse de doctorat, Georgia Institute of Technology.
- [Bolt, 1980] R. A. Bolt (1980). Put-that-there : voice and gesture at the graphics interface. Dans *7th annual conference on Computer graphics and interactive techniques*, pages 262–270.
- [Bowman et Wingrave, 2001] Doug A. Bowman et Chadwick A. Wingrave (2001). Design and evaluation of menu systems for immersive virtual environments. Dans *VR '01 : Proceedings of the Virtual Reality 2001 Conference (VR'01)*, page 149, Washington, DC, USA. IEEE Computer Society.
- [Bowman et al., 2001] D.A. Bowman, D.B. Johnson, et L.F. Hodges (2001). Testbed evaluation of virtual environment interaction techniques. *Presence : Teleoperators & Virtual Environments*, 10(1) :75–95.
- [Bowman, 1999] D. A. Bowman (1999). *Interaction Techniques for Common Tasks in Immersive Virtual Environments : Design, Evaluation, and Application*. thèse de doctorat, Georgia Institute of Technology.
- [Fuchs et al., 2006] Fuchs, P., Moreau, G., Berthoz, A., Vercher, J., and Auvray, M. (2006). *Le traité de la réalité virtuelle : Volume 1, l'homme et l'environnement virtuel*. Presses de l'École des mines.
- [Fuchs et al., 2003] Fuchs, P. and G. Moreau (2003). "Le traité de la réalité virtuelle— Volume 1: fondements et interfaces comportementales." Les presses de l'École des Mines.

## References bibliographies

- [Kalawsky, 1993] Kalawsky, R. (1993). *The Science of Virtual Reality and Virtual Environments*. Addison-Wesley Ltd, UK.
- [Leigh et al., 1996] Leigh, and Thomas A, De Fanti. "Multiperspective Collaborative Design in Persistent Networked virtual Environments," *Proceedings of the IEEE 1996 Virtual reality Annual International Symposium (VRAIS)*, 1996, pp, 253-260
- [Rheingold, 1991] Howard Rheingold (1991). *Virtual Reality*. Summit Books, London.
- [Sternberger, 2006] Ludovic Sternberger (2006). *Interaction en réalité virtuelle*. thèse de doctorat, Université Louis Pasteur de Strasbourg.
- [Stokley et al., 1995] Stokley, Richard, Matthew J, Conway, and Randy Pausch. "Virtual Reality on a WIM: Interaction Worlds in Miniature," *Proceedings of the ACM SIGCHI Human Factors in Computer Systems Conference*, 1985, pp. 265-272.
- [Tisseau, 2001] Tisseau, J. (2001). *Réalité virtuelle : autonomie in virtuo*. Habilitation à Diriger des Recherches– Université de Rennes 1.
- [Mine et al., 1997] Mark R. Mine, Jr. Frederick P. Brooks, et Carlo H. Sequin (1997). Moving objects in space : exploiting proprioception in virtual-environment interaction. Dans *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, volume 31, pages 19–26. ACM Press/Addison-Wesley Publishing Co.
- [McLellan, 1996] McLellan, H. (1996). "Virtual realities." *Handbook of research for educational communications and technology*: 457-487.
- [Milgram et al, 1994] Augmented reality: A class of displays on the reality-virtuality continuum, *SPIE Telemanipulator and telepresence Technologies*, 2351, 282-291.
- [Mapes et Moshell, 1995] D. Mapes et J. Moshell (1995). A two-handed interface for object manipulation in virtual environment. *Presence, Teleoperation and Virtual Environment*, 4(4) :403–416.
- [Otmame et al., 2007] Otmame, S., Ouramdane-Djerah, N., and Mallem, M. (2007). Towards a collaborative 3d interaction model for cooperative design in virtual environments. 11th International Conference on Computer Supported Cooperative Work in Design, 2007., pages 198–203.
- [Poupyrev et al., 1996] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, et Tadao Ichikawa (1996). The go-go interaction technique : non-linear mapping for direct manipulation in vr. Dans *UIST '96 : Proceedings of the 9th annual ACM symposium on User interface software and technology*, pages 79–80, New York, NY, USA. ACM.
- [Pierce et al., 1997] J. S. Pierce, A. Forsberg, M. J. Conway, S. Hong, R. Zeleznik, et M. R. Mine (1997). Image plane interaction techniques in 3d immersive environments. Dans *Proceedings of 1997 Symposium on Interactive 3D Graphics*, pages 39–43.
- [Poupyrev et al., 1998] I. Poupyrev, S. Weghorst, M. Billinghurst, et T. Ichikawa (1998). Egocentric object manipulation in virtual environments : empirical evaluation of interaction techniques. Dans *Proceedings of EUROGRAPHICS, Computer Graphics Forum*, volume 17, pages 41–52.

## References bibliographies

- [Ullah, 2011] Ullah, S. (2011). Multi-modal Assistance for Collaborative 3D Interaction: Study and analysis of performance in collaborative work, Université d'Angers.
- [Zeltzer, 1992] Zeltzer, D. (1992). Autonomy, interaction, and presence. Presence : Teleoperators and Virtual Environments, 1(1) :127-132.