



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie

Département d'informatique

N° d'ordre : IVA 09 / M2/2022

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : Image et Vie Artificielle (IVA)

Titre :

Simulation de l'évitement de collision entre un humain réel et des humains virtuels en utilisant la réalité virtuelle

(Simulation of collision avoidance between a real human and Virtual humans using Virtual reality)

Par : Louetri Taha

Soutenu le 27/06/2022 devant le jury composé de :

Chighoub Rabiaa

MCB

Président

Cherif Foudil

Pr

Encadrant

Fekraoui Farah

MCB

Examineur

Année universitaire 2021-2022

Acknowledgements

All praises to Allah, the Most Gracious, the Most Merciful, the Lord of the 'Alamin.

I would like to thank Almighty Allah for giving me the determination and strength during the whole process of preparation, compiling and writing of this thesis

Professor Cherif Foudil, my supervisor, deserves special thanks for his encouragement, incisive remarks, vast knowledge, patience, and, most significantly, his research vision. When the going became difficult, his office was always open with a cheerful mood. It was a wonderful honor to work on this thesis with his supervision. I will always be grateful for this wonderful and life-changing event as I continue on my path.

I extend my sincere thanks to all members of the jury for agreeing to review my work and to participate in and enrich the defense of this thesis.

My heartfelt thanks goes out to everyone who helped me with my thesis and supported me in whatever manner.

Dedication

بسم الله الرحمن الرحيم والصلاة والسلام على سيد الخلق سيدنا
الكريم محمد صلى الله عليه وسلم وعلى آله وصحبه أجمعين

I dedicate my dissertation work to my family and many friends. A special feeling of gratitude to my loving parents, Djamel and Chabane Kamir whose words of encouragement and push for tenacity ring in my ears. My 3 brothers Aziz and Hani and Nafaa have never left my side and are very special.

I also dedicate this dissertation to my many friends who have supported me throughout the process. I will always appreciate all they have done, especially Tiar Ayoub, Haba Abdelhai, Debabech Islam, Segni Aymen, Zendagui Rafik, Djalloul Baderdine, Taouririt Moncef, Abdesselam Taki and all the other friends.

Abstract

Virtual Reality technology is a simulated experience that can be similar to or completely different from the real world. Applications of virtual reality include entertainment (particularly video games), education (such as medical or military training) and business (such as virtual meetings). Two of the most interesting topics in this field are the interaction and navigation in a virtual environment, but the problem is most simulations and Experiments of virtual reality don't take into account the realistic side especially in collision avoidance when the users navigate and interact with objects and other users or avatars in the virtual environment.

Therefore, and in order to provide an efficient tool for collision avoidance we created a serious game application based on virtual reality that makes the player navigate and interact with a virtual human or avatar who can simulate the collision avoidance of real humans in an interactive and fully immersive 3D environment using the Oculus Quest 2 headset. Our application gives the users complete freedom to navigate in the virtual environment and responses to different scenarios with a different type of navigation in the same virtual world. A realistic experience with a high sense of immersion and smooth interactions was achieved taking the advantage of the new VR technologies.

ملخص

تقنية الواقع الافتراضي هي تجربة محاكاة يمكن أن تشبه أو تختلف تمامًا عن العالم الحقيقي. تشمل تطبيقات الواقع الافتراضي الترفيه (وخاصة ألعاب الفيديو) والتعليم (مثل التدريب الطبي أو العسكري) والأعمال (مثل الاجتماعات الافتراضية). هناك اثنان من أكثر المواضيع إثارة للاهتمام في هذا المجال هما التفاعل والملاحة في بيئة افتراضية، لكن المشكلة هي أن معظم عمليات المحاكاة والتجارب في الواقع الافتراضي لا تأخذ في الاعتبار الجانب الواقعي وخاصة في تجنب التصادم عندما ينتقل المستخدمون ويتفاعلون مع الكائنات والمستخدمين الآخرين أو اللاعبين الافتراضيين في البيئة الافتراضية.

لذلك، ومن أجل توفير أداة فعالة لتجنب الاصطدام، أنشأنا تطبيقًا جادًا للعبة يعتمد على الواقع الافتراضي الذي يجعل اللاعب ينتقل ويتفاعل مع إنسان افتراضي يمكنه محاكاة تجنب الاصطدام مثل البشر في بيئة تفاعلية وكاملة ثلاثية الأبعاد باستخدام سماعة "Oculus Quest 2". يمنح تطبيقنا للمستخدمين الحرية الكاملة للتنقل في البيئة الافتراضية والاستجابات لسيناريوهات مختلفة مع نوع مختلف من التنقل في نفس العالم الافتراضي. تم تحقيق تجربة واقعية مع شعور كبير بالغمر والتفاعلات السلسة مع الاستفادة من تقنيات VR الجديدة.

CONTENTS

Acknowledgements	1
Dedication	2
Abstract	3
المخلص	4
Contents	5
List of Figure	8
List of Tables	11
General Introduction	12
Overview about Virtual Reality	15
1.1 Introduction	16
1.2 Definition of Virtual Reality	16
1.3 A brief history of virtual reality	17
1.4 Type of virtual reality	21
1.5 Augmented Reality	23
1.5.1 Understanding Augmented Reality	24
1.5.2 Augmented Reality Vs Virtual Reality	25
1.6 Equipments for virtual reality	25
1.7 Application of virtual reality	28
1.7.1 Education	28
1.7.2 Scientific visualization	28

1.7.3	Medicine	29
1.7.4	Industrial design and architecture	29
1.7.5	Game and entertainment	29
1.8	Conclusion	29
Navigation and Interaction in Virtual Reality		31
2.1	Introduction	32
2.2	Definition of Navigation in Virtual Reality	32
2.3	The development of the 3D navigation system for Virtual Reality	33
2.3.1	Indoor Navigation	33
2.3.2	Outdoor Navigation	36
2.4	Virtual Reality Navigation Techniques	38
2.5	Factors influencing Navigation in Virtual Environment	43
2.6	Interaction in Virtual Environments	46
2.6.1	Introduction	46
2.6.2	Definition of Interaction	47
2.6.3	Collision Avoidance and Interaction	53
2.7	Conclusion	55
System and Architecture Design		56
3.1	Introduction	57
3.2	Presentation of system for collision avoidance	57
3.2.1	System objectives	57
3.2.2	System Architecture Design	58
3.3	Methodology Setting	61

3.3.1 First Step: Identify	63
3.3.2 Second Step: Research	63
3.3.3 Third Step: Create	65
3.3.4 Fourth Step: Test	65
3.3.5 Fifth Step: Evaluation	65
3.3.6 Six Step: Implementation	66
3.4 Conclusion	66
Implementation, Results and Evaluation	67
4.1 Introduction	68
4.2 Technical Description	68
4.3 Hardware and Software for Development Environment	69
4.3.1 Hardware	69
4.3.2 Software	71
4.4 Development Process	73
4.4.1 Creation of the Virtual Environment	73
4.4.2 Creation of the Real Human	78
4.4.3 Creation of the human virtual	82
4.5 Result	86
4.6 Conclusion	92
Conclusion and Perspectives	93
5.1 Summary	93
5.2 Future Work	94
Bibliography	95

List of Figure

Figure 1.1: Picture of the Kinetoscope [3]	18
Figure 1.2: Picture of the Cinematography camera [3].....	18
Figure 1.3: Link Trainer, Edwin Link [3]	19
Figure 1.4: The side view of the Sensorama [3].....	20
Figure 1.5: The first wired dataglove [3].....	21
Figure 1.6: Types of virtual reality systems [4]	22
Figure 1.7: Examples of immersion levels: (A) a non-immersive VR system, (B) a semi-immersive VR system, and (C) an immersive VR system [5]	23
Figure 1.8: Augmented reality: A heads-up display, like this one used by the US Air Force, superimposes useful, computer-based information on top of the things you see with your own eyes. Picture by Major Chad E. Gibson courtesy of US Air Force [6].....	25
Figure 1.9: A typical HMD has two tiny screens that show different pictures to each of your eyes, so your brain produces a combined 3D (stereoscopic) image.	26
Figure 1.10: EXOS data gloves produced by NASA in the 1990s had very intricate external sensors to detect finger movements with high precision.	26
Figure 1.11: This more elaborate EXOS glove had separate sensors on each finger segment. Picture by Wade Sisler courtesy of NASA Ames Research Center.	27
Figure 1.12: A typical handheld virtual reality controller (complete with elastic bands), looking not so different from a video game controller. Photo courtesy of NASA Ames Research Center.	27
Figure 2.1: Generated 3D CAD network (Left) and some selected nodes in the network (Right)[11]	34
Figure 2.2: 3D point differential for node A (x1, y1, z1) and node B (x2, y2, z2) in 3D axis[11].	35
Figure 2.3: The Shortest Route Calculation Windows [11]	35
Figure 2.4: 3D buildings with real textures in the outdoor environment [11].....	36
Figure 2.5: The system navigates users along the shortest path [11]	37
Figure 2.6: The four different VR navigation techniques:(a) flight, (b) teleport, (c) world in-miniature, (d) 3D cone drag. [12]	38
Figure 2.7: The World in Miniature (WIM) viewed against the background of a life-size VR [12]	39

Figure 2.8:The 2.5m play area and 1m diameter mini-map configuration used in our test environment [12]..... 40

Figure 2.9:Overview of the entire virtual world used during the study tasks. Insets show beacons (a) used for navigation tasks and coins (b) used for search tasks [12]. 41

Figure 2.10:A picture showing the way of teleport navigation in virtual reality [12]..... 42

Figure 2.11:A picture showing selection objects using the controllers in a Virtual Environment [12]..... 48

Figure 2.12:A picture showing Laser pointer selection in Virtual Environment [12]..... 49

Figure 2.13: A picture showing Gesture based selection in Virtual Environment [12] 49

Figure 2.14: A picture showing Gaze base selection pointer with cursor in Virtual Environment [12]..... 49

Figure 2.15: A picture showing Teleportation based locomotion in virtual environment [12] 52

Figure 2.16: A picture showing Quest’s guardian protection system [12]..... 52

Figure 2.17: Image show the shortest path using Lozano-Perez algorithm..... 54

Figure 2.18: Image show an explication of the QuadTree algorithm 55

Figure3.1:Image show the System Architecture 58

Figure3.2:Image show the steps of the methodology 62

Figure3.3:Image show the Activity diagram of the user/VR system interactions..... 64

Figure 4.1:A diagram of Technical description of the application requirements 68

Figure 4.2:Image show the Oculus Quest 2 device with the controllers 69

Figure 4. 3:Image show the oculus quest 2 controllers 70

Figure 4.4:Image show the visual studio 2019 Logo 71

Figure 4.5:Image show the unity logo..... 72

Figure 4.6:Image show the Rigidbody component 74

Figure 4.7:Image show a part of the city model 77

Figure 4.8:Image show a part of the city model 77

Figure 4.9:Image show a part of the city model 78

Figure 4.10:Image show the configuration of the user (the player) 78

Figure 4.11:Image show the script who controller the continuous movement 80

Figure 4.12:Image show the capsule collider propriety..... 80

Figure 4. 13:Image show the configuration for the teleportation movement 81

Figure 4.14: Image show the virtual human..... 83

Figure 4.15: Image show the animation of the avatar..... 84

Figure 4.16: Image show the script who control the movement of our avatar 84

Figure 4.17: Image show the NavMeshAgent component..... 85

Figure 4.18: Image illustrate the difference between the teleportation movement and continuous movement in virtual world.....88

Figure 4.19: Image illustrate the difference between the continuous movement and the teleportation movement in the interact with objects (obstacle).....90

Figure 4.20: Image illustrate the collision avoidance between a real human with teleportation movement and the virtual human.....91

Figure 4.21: Image illustrate the collision avoidance between a real human with continuous movement and the virtual human.....91

Figure 4.22: Image show the effect of the pitch of ladder and the gravity in the climbing of ladder (Continuous Movement navigation with physics law).....92

Figure 4.23: Image illustrate the pitch of the ladder.....93

Figure 4.24: Image shows the effect of the pitch of the ladder and the gravity in the climbing of the ladder(teleportation navigation with no physics law).....93

Figure 4.25: Image illustrate the Exit method of the game application.....94

List of tables

Table 1.1 Factors influencing navigation and their example characteristics.....	45
Table 4.1: a table show the Properties of the rigid body component.....	75
Table 4.2: a table show the propriety of the capsule collider.....	82
Table 4.3: A table show the properties of NavMeshAgent.....	85

General Introduction

You'll probably never go to Mars, swim with dolphins, run an Olympic 100 meters, or sing onstage with the Rolling Stones. But if virtual reality ever lives up to its promise, you might be able to do all these things—and many more—without even leaving your home. Unlike real reality (the actual world in which we live), virtual reality means simulating bits of our world (or completely imaginary worlds) using high-performance computers and sensory equipment, like headsets and gloves. Apart from games and entertainment, it's long been used for training airline pilots and surgeons and for helping scientists to figure out complex problems such as the structure of protein molecules.

Today, virtual reality is considered one of the most important research topics in computer graphics, and it has become widely used as a functional tool in many fields. In the past, VR has typically been restricted to research universities, large corporations, and the military for training. However, fast progress in both hardware and software technology has helped overcome these limitations and will allow VR to be available and useful to a much wider population.

Way to the development of CAD software program, graphics hardware acceleration, head set up presentation and database gloves, VR in contemporary decade becomes overwhelmingly well-known and used to describe a massive form of applications typically associated with immersive rather seen, 3D environments, consisting of the gaming care sector in which the VR business enterprise marketplaces is growing every day because the immoderate-performance images going to be greater price-powerful.

The gaming industry has brought new technologies to enhance the gadget in terms of each software and hardware in video games. The VR generation is taking into consideration one of the freshest topics in gaming tendencies. VR offers a new feature that gives the potential for gamers to enter the sport in real time while they're gambling.

The approaches in which VR change the arena of gaming is by way of growing new content material in the game to be greater practical and exiting, unique from gaming in

the beyond except, VR video games also offer new methods to play for users by means of the usage of the innovation of devices to connect with video games together with the VR headset, VR controller, haptic VR gloves, mechanized VR chair and leap movement VR.

All the VR technology which can be used in the gaming industry also provide and improve a pleasant person revel in the virtual environment (VE) of 3D creatures that are generated by way of computer systems. The VR will make users feel like they're in a reel situation both mentally and physically through the motion of the body that syncs with VR games. So the VR can assist gaming industries by increasing the popularity and value of the industry, in addition to attracting extra purchasers to the market.

To sum up, the new era has turned out to be vital nowadays and the destiny to change the way we live in lots of industries and also improvement. The gaming industry has used VR technologies to improve the gaming gadget and to attract more customers.

The proposed solution, and the objective of this work is the development of a virtual reality experience game that simulates the interaction in a virtual environment between a real human and a virtual human or objects and find solution to the collision during the interaction.

Our project thesis is organized into several chapters. In the 1 chapter we take an overview about the virtual reality where we give an introduction about the virtual reality from definition of virtual reality to a brief history of it to give some details about VR types and some talk about the Augmented reality and then we talk about the Hardware we need for VR and some application to we reach finally to the end of chapter 1 with a conclusion.

In the 2 chapter we will talk about what is the navigation in VR, there systems and techniques and the influencing factor of navigation and before and we end our chapter with a conclusion we give a little information about interaction which is our next topic in chapter 3.

In the 3 chapter we extend our last topic in chapter 2 which is the interaction and we talk about it deeply and understand the definition of interaction and the relation between it and the VR and talk about the technique provided by it and then we talk about the

main topic in this thesis which is the collision avoidance and their relation with interaction. All of this research are our resources to the real deal which is the creation of a game application can provide a realistic interaction in Virtual Environment and to do that we need first to design an architecture for our system and then we prepare and setting a methodology of how the work should have done, after all these steps we go the chapter 4 which is the implementation of our job with a results and some evaluation.

In the 4 chapter we start by give some details about our hardware and software requirement that we need for the development process and show our results and make some evaluation to write some note and Remarque that will help us in the future work.

Chapter 01

Overview about Virtual Reality

1.1 Introduction

According [1], one way to view virtual reality (VR) is as a “magical window into other worlds, from molecules to minds”, and since science fiction has always played the role in not only foretelling but also advancing our scientific and technological directions, virtual reality has been reaching lately the unlimited point of artificial experience.

Abstract worlds and phenomena that were previously only describable through words and crude diagrams, become as immersive as life itself with the virtual reality art of simulating a physical presence for the users in places both real and imaginary allowing them to feel they are part of the action. Today, with the fast progress in both hardware and software technology virtual reality is no longer restricted to research universities, large corporations, and the military for training, but rather it is becoming open and useful to a much wider population. Very few people, however, really know what virtual reality is, what its basic principles, its application areas and its challenges are. Therefore, an insightful overview of VR is done in this chapter.

1.2 Definition of Virtual Reality

Virtual reality (VR) means experiencing things through our computers that do not really exist. From that simple definition, the idea does not sound especially new. When you look at an amazing Canaletto painting, for example, you're experiencing the sights and sounds of Italy as it was about 250 years ago—so that's a kind of virtual reality. In the same way, if you listen to ambient instrumental or classical music with your eyes closed, and start dreaming about things, isn't that an example of virtual reality—an experience of a world that doesn't really exist? What about losing yourself in a book or a movie? Surely that's a kind of virtual reality?

If we are going to understand why books, movies, paintings, and pieces of music are not the same thing as virtual reality, we need to define VR fairly clearly. For the purposes of this simple, introductory article, I am going to define it as:

A believable, interactive 3D computer-created world that you can explore so you feel you really are there, both mentally and physically. Putting it another way, virtual reality is essentially:

- **Believable:** You really need to feel like you're in your virtual world (on Mars, or wherever) and to keep believing that, or the illusion of virtual reality will disappear.
- **Interactive:** As you move around, the VR world needs to move with you. You can watch a 3D movie and be transported up to the Moon or down to the seabed—but it is not interactive in any sense.
- **Computer-generated:** Why is that important? Because only powerful machines, with realistic 3D computer graphics, are fast enough to make believable, interactive, alternative worlds that change in real-time as we move around them.
- **Explorable:** A VR world needs to be big and detailed enough for you to explore. However realistic a painting is, it shows only one scene, from one perspective. A book can describe a vast and complex "virtual world," but you can only really explore it in a linear way, exactly as the author describes it.
- **Immersive:** To be both believable and interactive, VR needs to engage both your body and your mind. Paintings by war artists can give us glimpses of conflict, but they can never fully convey the sight, sound, smell, taste, and feel of battle. You can play a flight simulator game on your home PC and be lost in a very realistic, interactive experience for hours (the landscape will constantly change as your plane flies through it), but it's not like using a real flight simulator (where you sit in a hydraulically operated mockup of a real cockpit and feel actual forces as it tips and tilts), and even less like flying a plane. [2]

1.3 A brief history of virtual reality

So much for the future, but what of the past. Virtual reality has a long and very rich history. Here are a few of the more interesting highlights:

- 1890s: **Thomas Edison** and his assistant **William Dickson** pioneer the Kinetograph (a camera for recording pictures) and Kinetoscope (a projector for playing them back)—in effect, the first one-person "movie experience"[3].

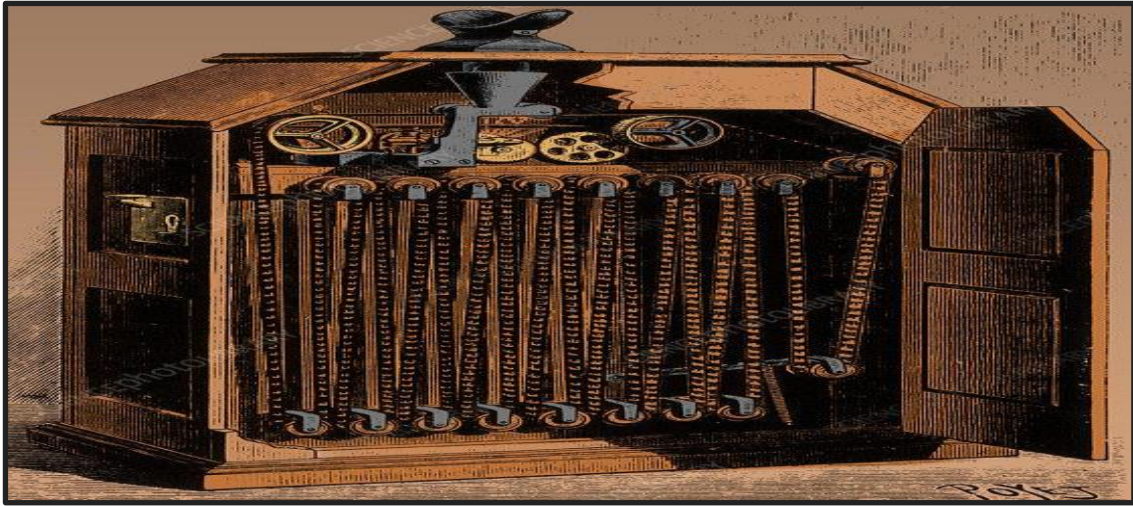


Figure 1.1: Picture of the Kinetoscope [3]

- 1895: French brothers **Auguste and Louis Lumière** open the first movie theater in Paris, France. Legend has it that one of their movie shorts, *Arrival of a Train at La Ciotat*, is such a convincing (they did use the Cinematography camera) depiction of reality that people in the audience scream and run to the back of the room [3].

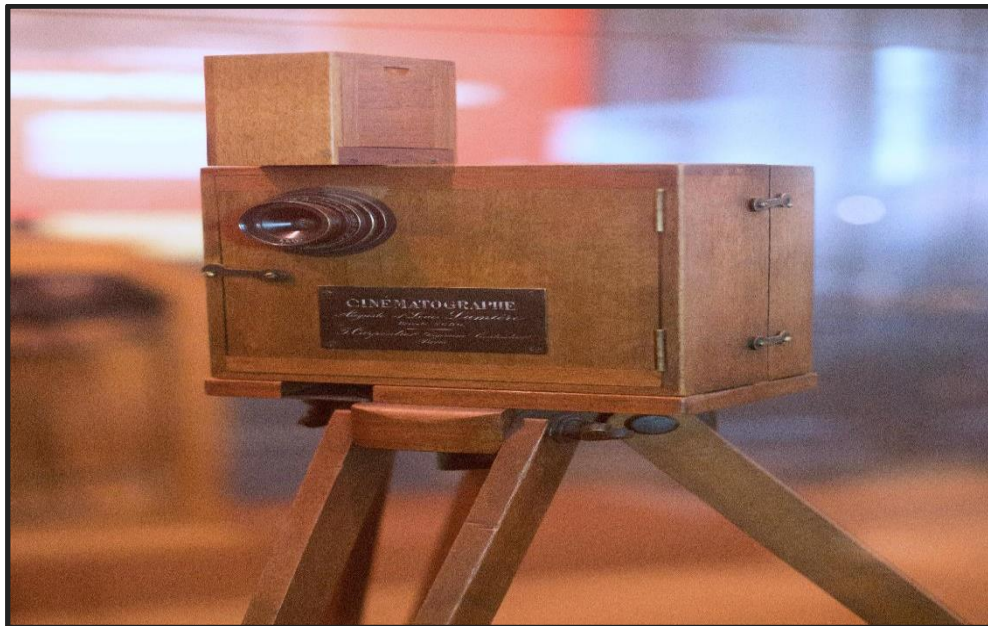


Figure 1.2: Picture of the Cinematography camera [3]

- 1929: **Edwin Link** develops the Link Trainer (also called the Pilot Maker), a mechanical airplane simulator. The original Link Trainer was created in 1929 out of the need for a safe way to teach new pilots how to fly by instruments. Ed Link used his knowledge of pumps, valves and bellows gained at his father's Link Piano and Organ Company to create a flight simulator that responded to the pilot's controls and gave an accurate reading on the included instruments. More than 500,000 US pilots were trained on Link simulators, as were pilots of nations as diverse as Australia, Canada, Germany, New Zealand, United Kingdom, Japan, Pakistan, and the USSR. Following WWII, Air Marshal Robert Lecky (wartime RAF Chief of Staff) said, "The Luftwaffe met its Waterloo on all the training fields of the free world where there was a battery of Link Trainers"[3].

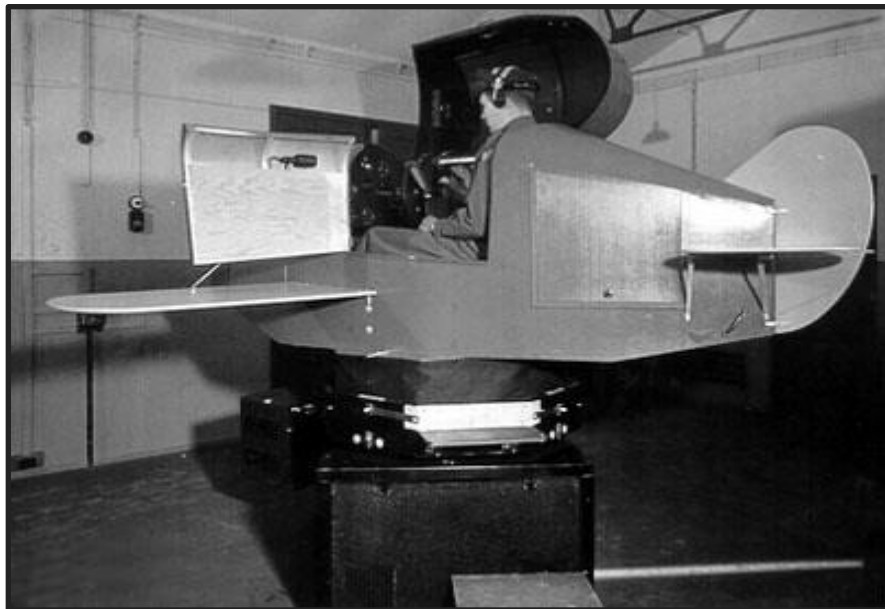


Figure 1.3: Link Trainer, Edwin Link [3]

- 1950: US Air Force psychologist **James J. Gibson** publishes an influential book, *The Perception of the Visual World*, describing how people see and experience things as an "optic flow" as they move through the world. These ideas, and those of contemporaries such as Adelbert Ames, help to form the foundations of the 20th century psychology of visual perception, which feeds into academic studies of computer vision and virtual reality [3].

- 1956: Cinematographer **Morton Heilig** begins developing machines that can produce artificial sensory experiences. In 1957, he developed a pioneering 3D head-mounted display. In 1962, he has granted a patent for a machine called the Sensorama (a kind of updated, highly sophisticated Kinetoscope) that can immerse its user in artificial vision, sound, smell, and vibration. Many people regard Heilig as the true father of virtual reality, though he has seldom been acknowledged as such [3].
- 1962: **Ivan Sutherland**, pioneer of human-computer interaction, developed the light pen and a program called Sketchpad that allows people to draw on a computer screen—opening the way for the kind of computer graphics later used in virtual reality [3].

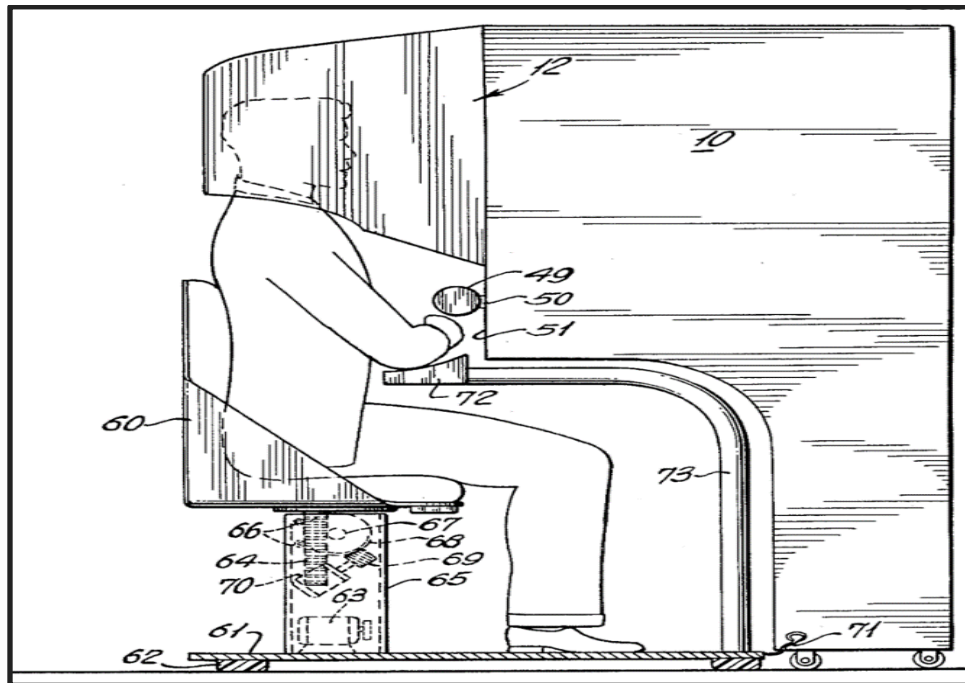


Figure 1.4: The side view of the Sensorama [3].

- 1977: **Dan Sandin**, **Richard Sayre**, and **Thomas Defanti** produce the first data glove based on an idea by Richard Sayre as a project for the National Endowment for the Arts. It used light based sensors with flexible tubes with a light source at one end and a photocell at the other. As the fingers were bent, the amount of light that hit the photocells varied, thus providing a

measure of finger flexion. It was a lightweight, inexpensive effective method for multidimensional control mainly used to manipulate sliders [3].



Figure 1.5: The first wired dataglove [3].

- 1989: Computer scientist and musician Jaron Lanier: coins the now preferred term "virtual reality." His company, VPL Research, garners huge media attention and develops pioneering VR peripherals, including a HMD and dataglove. Lanier has been popularly referred to as the "father of virtual reality" ever since, though, as this timeline demonstrates, he was quite a late entrant to the field [3].

1.4 Types of virtual reality

There is clearly more than one approach to building virtual worlds—and more than one flavor of virtual reality. Here are a few of the bigger variations:

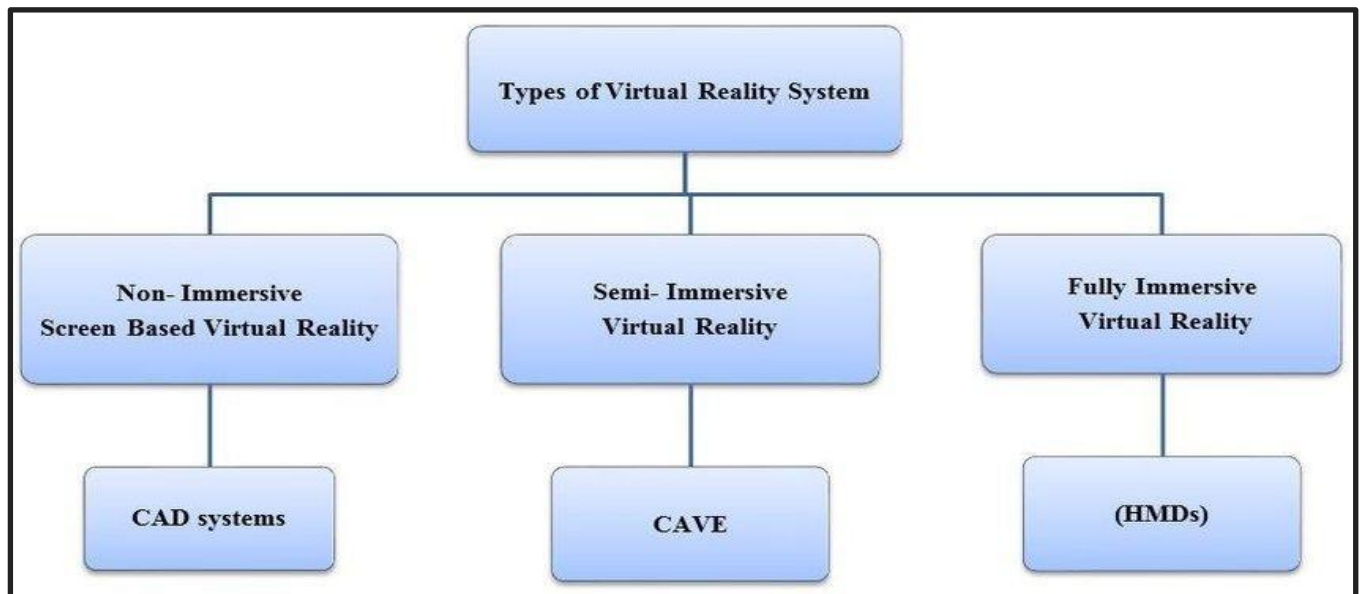


Figure 1.6:Types of virtual reality systems [4]

- **Fully immersive:**

For the complete VR experience, we need three things. First, a plausible, and richly detailed virtual world to explore; a computer model or simulation, in other words. Second, a powerful computer that can detect what we're going and adjust our experience accordingly, in real time (so what we see or hear changes as fast as we move—just like in real reality). Third, hardware linked to the computer that fully immerses us in the virtual world as we roam around. Usually, we would need to put on what is called a head-mounted display (HMD) with two screens and stereo sound, and wear one or more sensory gloves.

- **Not-immersive:**

A highly realistic flight simulator on a home PC might qualify as non-immersive virtual reality, especially if it uses a very wide screen, with headphones or surround sound, and a realistic joystick and other controls.

- **Collaborative:**

The idea of sharing an *experience* in a virtual world with other people, often in real time or something very close to it. Collaboration and sharing are likely to become increasingly important features of VR in future.

- **Web-based:**

Computer scientists developed a way of building virtual worlds on the Web (using a technology analogous to HTML called Virtual Reality Markup Language, VRML). With Facebook's growing interest in the technology, the future of VR seems likely to be both Web-based and collaborative.

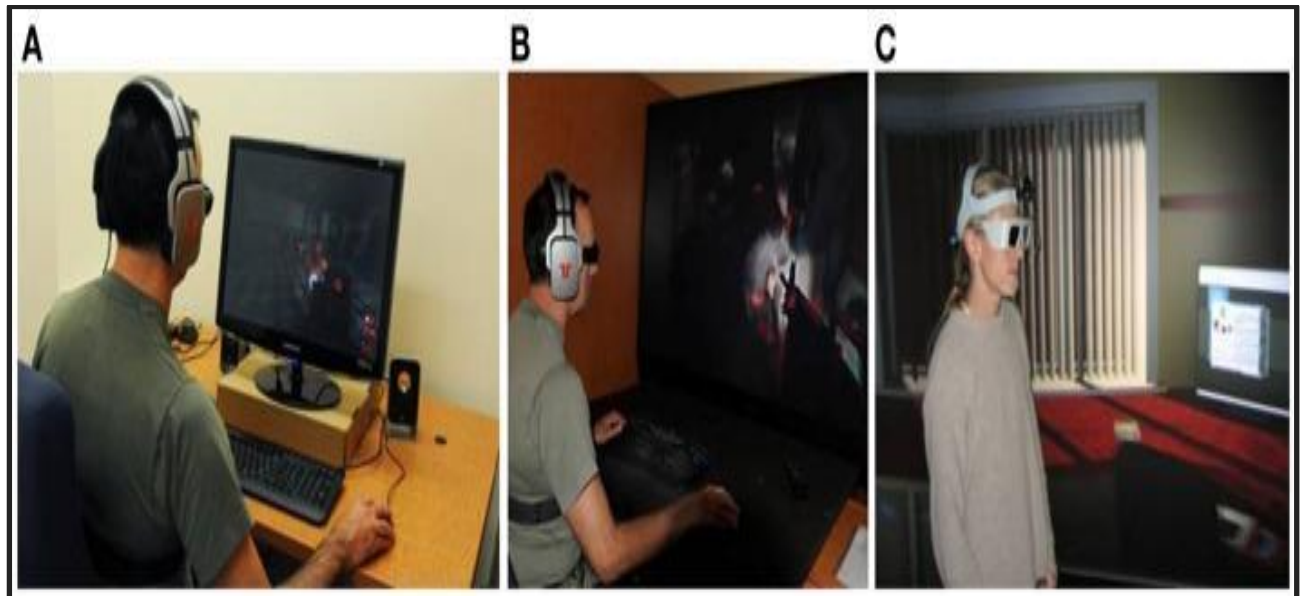


Figure 1.7: Examples of immersion levels: (A) a non-immersive VR system, (B) a semi-immersive VR system, and (C) an immersive VR system [5]

1.5 Augmented Reality:

Augmented reality (AR) is an enhanced version of the real physical world that is achieved through the use of digital visual elements, sound, or other sensory stimuli delivered via technology. It is a growing trend among companies involved in mobile computing and business applications in particular.

Amid the rise of data collection and analysis, one of augmented reality's primary goals is to highlight specific features of the physical world, increase understanding of those features, and derive smart and accessible insight that can be applied to real-world applications. Such big data can help inform companies' decision-making and gain insight into consumer spending habits, among others[5].

1.5.1 Understanding Augmented Reality:

Augmented reality continues to develop and become more pervasive among a wide range of applications. Since its conception, marketers and technology firms have had to battle the perception that augmented reality is little more than a marketing tool. However, there is evidence that consumers are beginning to derive tangible benefits from this functionality and expect it as part of their purchasing process.

For example, some early adopters in the retail sector have developed technologies that are designed to enhance the consumer shopping experience. By incorporating augmented reality into catalog apps, stores let consumers visualize how different products would look like in different environments. For furniture, shoppers point the camera at the appropriate room and the product appears in the foreground. Elsewhere, augmented reality's benefits could extend to the healthcare sector, where it could play a much bigger role. One way would be through apps that enable users to see highly detailed, 3D images of different body systems when they hover their mobile device over a target image. For example, augmented reality could be a powerful learning tool for medical professionals throughout their training.

Some experts have long, may provide a more complete link between real and virtual realms if it develops enough speculated that wearable devices could be a breakthrough for augmented reality. Whereas smartphones and tablets show a tiny portion of the user's landscape, smart eyewear, for example to become mainstream.

1.5.2 Augmented Reality Vs Virtual Reality:

Augmented reality uses the existing real-world environment and puts virtual information on top of it to enhance the experience.

In contrast, virtual reality immerses users, allowing them to "inhabit" an entirely different environment altogether, notably a virtual one created and rendered by computers. Users may be immersed in an animated scene or an actual location that has been photographed and embedded in a virtual reality app. Through a virtual reality viewer, users can look up, down, or any which way, as if they were actually there. [6]



Figure 1.8: Augmented reality: A heads-up display, like this one used by the US Air Force, superimposes useful, computer-based information on top of the things you see with your own eyes. Picture by Major Chad E. Gibson courtesy of US Air Force [6].

1.6 Equipment for virtual reality

A virtual reality headset is a head-mounted device that provides virtual reality for the wearer. Virtual reality (VR) headsets are widely used with video games but they are also used in other applications, including simulators and trainers. ... Some VR headsets also have eye-tracking sensors and gaming controllers:

- **Head-mounted displays (HMDs):**

Head-mounted displays (HMDs) are small displays or projection technology integrated into eyeglasses or mounted on a helmet or hat. Heads-up displays are a type of HMD that does not block the user's vision, but superimposes the image on the user's view of the real world.



Figure 1.9: A typical HMD has two tiny screens that show different pictures to each of your eyes, so your brain produces a combined 3D (stereoscopic) image.

- **Data gloves:**

Glove connected to a computer and equipped with sensors allowing the actual movements of a person's hand to manipulate virtual objects.



Figure 1.10: EXOS data gloves produced by NASA in the 1990s had very intricate external sensors to detect finger movements with high precision.

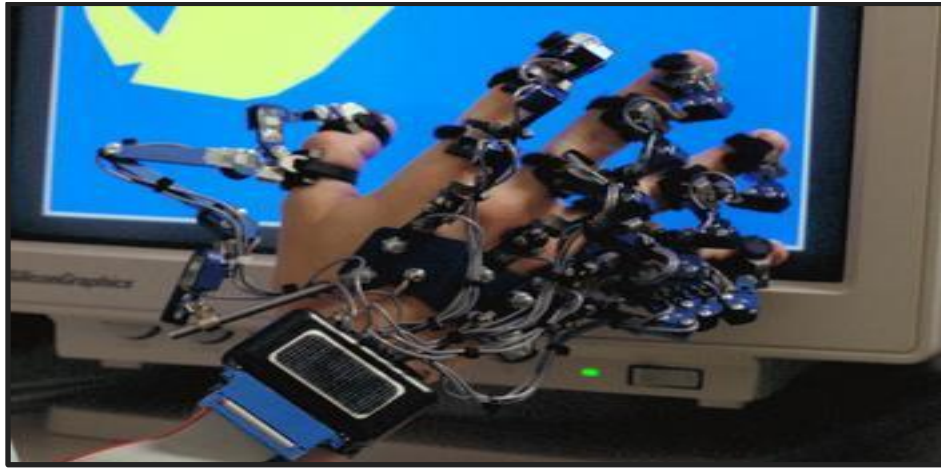


Figure 1.11: This more elaborate EXOS glove had separate sensors on each finger segment. Picture by Wade Sisler courtesy of NASA Ames Research Center.

- **Wands:**

Even simpler than a dataglove, a wand is a stick you can use to touch, point to, or otherwise interact with a virtual world. It has position or motion sensors (such as accelerometers) built in, along with mouse-like buttons or scroll wheels. Originally, wands were clumsily wired into the main VR computer; increasingly, they're wireless.



Figure 1.12: A typical handheld virtual reality controller (complete with elastic bands), looking not so different from a video game controller. Photo courtesy of NASA Ames Research Center.

1.7 Applications of virtual Reality

What sorts of things can we do with Virtual Reality?

Virtual reality technology allows the creation of virtual experience with the intention of replicating real-life scenarios or developing a virtual world consisting of non-realistic elements such as in games providing participation in various situations with risk-free choices.

The expense of executing this situation in real life is not advantageous when compared with the anticipated ROI yet with the aid of Virtual Reality (VR) and Augmented Reality (AR) technologies, all the likely applications can cut down the expense of employing these situations for varied purposes such as training of employees, products trial, examining of locations, projects trial and various other purposes, which can be accomplished through the personalized use of these technologies.[7]

1.7.1 Education

VR technology is so well embraced by educators and trainers because of the rich and interactive experience that it provides. The fully Immersive and engaging learning and training environment helps them develop their skills without the real world consequences of failing through a wide variety of terrains, situations and scenarios. Therefore, various simulators have been developed for all types of tasks such as plane operation, tanks, submarines, helicopters, car driving and others[7].

1.7.2 Scientific visualization

Anything that happens at the atomic or molecular scale is effectively invisible unless you are prepared to sit with your eyes glued to an electron microscope. However, suppose you want to design new materials or drugs and you want to experiment with the molecular equivalent of LEGO. That is another obvious application for virtual reality. Instead of wrestling with numbers, equations, or two-dimensional drawings of molecular structures, you can snap complex molecules together right before your eyes. This kind of work began in the 1960s at the University

of North Carolina at Chapel Hill, where Frederick Brooks launched GROPE, a project to develop a VR system for exploring the interactions between protein molecules and drugs[7].

1.7.3 Medicine

Apart from its use in things like surgical training and drug design, virtual reality also makes possible telemedicine (monitoring, examining, or operating on patients remotely). A logical extension of this has a surgeon in one location hooked up to a virtual reality control panel and a robot in another location (maybe an entire continent away) wielding the knife. The best-known example of this is the DaVinci surgical robot, released in 2009, of which several thousand have now been installed in hospitals worldwide. Introduce collaboration and there is the possibility of a whole group of the world's best surgeons working together on a particularly difficult operation—a kind of WikiSurgery, if you like [7].

1.7.4 Industrial design and architecture

Architects used to build models out of card and paper; now they are much more likely to build virtual reality computer models you can walk through and explore. By the same token, it's generally much cheaper to design cars, airplanes, and other complex, expensive vehicles on a computer screen than to model them in wood, plastic, or other real-world materials. This is an area where virtual reality overlaps with computer modeling: instead of simply making an immersive 3D visual model for people to inspect and explore, you are creating a mathematical model that can be tested for its aerodynamic, safety, or other qualities [7].

1.7.5 Games and entertainment

From flight simulators to race-car games, VR has long hovered on the edges of the gaming world—never quite good enough to revolutionize the experience of gamers, largely due to computers being too slow, displays lacking full 3D, and the lack of decent HMDs and data gloves. All that may be about to change with the development of affordable new peripherals like the Oculus Rift.

1.8 Conclusion

Virtual reality has become one of the most important technologies to be discussed regarding its applications, usage, different types and potential to change our lives and the real world. This technology has cut across all facets of human activities, and it is used widely today in almost all sorts of fields including Science, Engineering, Technology, Medicine and much more. As the cost

of virtual reality goes down and it becomes more mainstream, we can expect more serious uses with high degrees of immersion, trustworthy, and interaction, to come to the floor. This chapter introduced the concepts of VR systems, challenges, applications and other essential elements to better understand this technology.

Chapter02

**Navigation and Interaction in Virtual
Reality**

2.1 Introduction

3D environments account for a large portion of user interfaces in e.g. World Wide Web, game industry, architecture, tourism industry as well as within the scientific area with visualization and simulation. However, in many cases existing virtual 3D environments indicate a lack of usability, which can be attributed to non-intuitive and misleading navigational techniques. Non-intuitive and/or misleading navigation techniques do not only cause frustration but also provoke unrecoverable errors of the user. Therefore, the development of efficient navigation concepts for virtual 3D environments becomes more important.

2.2 Definition of Navigation in Virtual Reality

Darken and Sibert [8] define the term navigation e.g. as “the process by which people control their movement using environmental cues and artificial aids such as maps so that they can achieve their goal without getting lost” Elvins [9] defines the term more generally: “Navigating is most often defined as the process of following a course by making directed movements, . . .” Closely connected to the process of navigation is the process of wayfinding. Elvins [9] defines this process as “. . . determining the strategy, direction and course needed to reach a desired destination.” The difference between these two procedures explains Elvins [9] as follows: “Without wayfinding, a navigator won’t know in which direction to steer and without navigating, a wayfinder will not have the means to move toward their destination.” So we understand navigation in virtual environments as: “the process of determining a strategy, direction and course (wayfinding) with controlling the movement using some aids to achieve a desired goal”. The third important term associated with navigation and wayfinding is ‘spatial orientation’. Maier [10] defines this term as “the ability to physically and mentally find one’s way in two- and three-dimensional space”. This ability is important for an effective navigation and wayfinding, because an effective navigation involves good orientation, that means e.g. to know where to be, to know where to go and to know where to come from.

2.3 The development of the 3D navigation system for Virtual Reality

3D navigation is one of the components in the 3D geographic information system (GIS). Many navigation systems use 2D maps to represent the real world objects and are unable to provide users with enough information on real world objects such as their colors, shapes and materials. Features and navigational landmarks were not easily available even though they were very useful and helpful for navigating in the real world[11].

2.3.1 Indoor Navigation

2.3.1.1 Network Analysis for Indoor Navigation: Implementation of network analysis (i.e. shortest path analysis) mostly based on application objectives and situation. For instance, disaster based or emergency applications in urban areas need the shortest path routine to navigate in a complex surrounding architecture. System that could provide a route network within a building helps to locate safety routes for example during fire and rescue missions and other emergency operations. Shortest path routines are useful in assisting human navigation. There are several types of navigations, for instance navigation for pedestrians and cars (on roads or in racing circuits).[11]listed some of the applications that need those routines e.g. searching the nearest restaurant, a petrol station, jungle trekking on mountains/hills and locating the emergency exit doors during fire in a building. Shortest path network for multilevel buildings thus the connectivity between levels or floors are certainly very useful and interesting to be investigated. The Dijkstra algorithm could be used for this sort of application by extending to a three-dimensional (3D) environment. Some research works on the Dijkstra algorithm for 3D environment have been investigated. Other related works such as Meijers and Pu and Zlatanova focused on indoor navigation routing. This research describes the experiment on Dijkstra algorithm based on 3D CAD network data (i.e. inside of a building). The basic structure for the 3D network is a group of connecting nodes that consists of x, y and z (height) values. The system will extract the values for the network analysis calculation. [11]

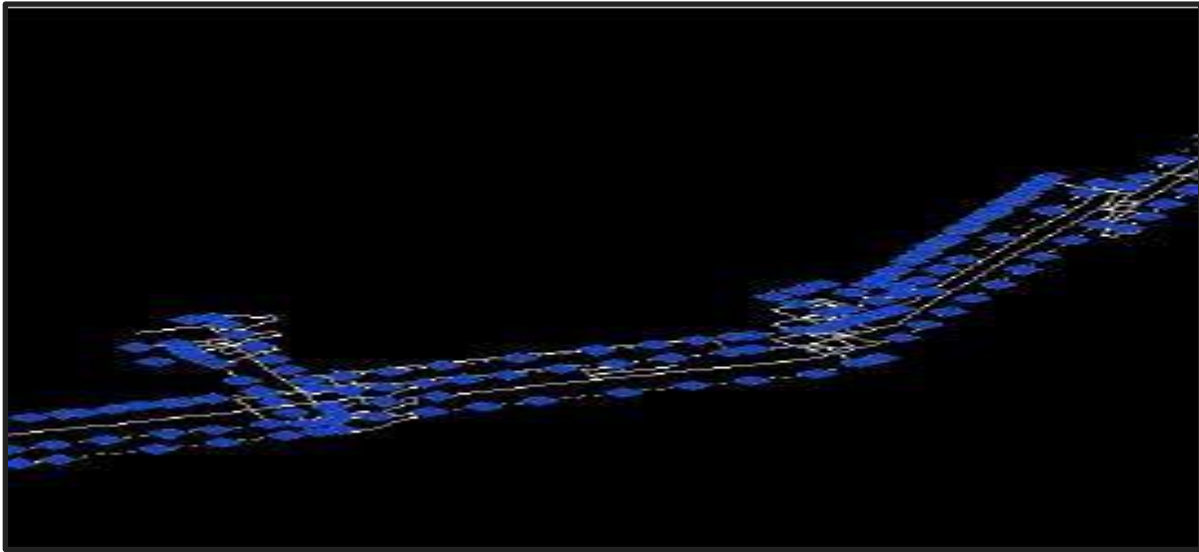


Figure 2.1:Generated 3D CAD network (Left) and some selected nodes in the network (Right)[11]

2.3.1.2 Dijkstra's Algorithm for 3D Network: After we have established a 3D network for the indoor area, important information like IDs, distances, and potential candidates of nodes need to be retrieved, they are very important for the algorithm and can be extracted and assigned with proper IDs. By putting an ID to each node, then the network is now applicable for Dijkstra's algorithm. The next step is to calculate the distance between the connected nodes (i.e. successor and predecessor nodes). It is not necessary to calculate the distance between all the nodes in the network as this will consume a lot of time. If node #1 is connected to node #2 and node #3, then the calculation will be based on those three nodes. To accomplish that, information on predecessor and successor nodes is important. To make computers understand which nodes are connected or not, therefore the network data file must be examined thoroughly. From here the data file will be examined in order to find the useful data and to determine the connected/reachable node. The distance between nodes for a 3D network can easily be calculated by using a simple mathematical formula. Example: distance between node A (x_1, y_1, z_1) to node B (x_2, y_2, z_2)[11]:

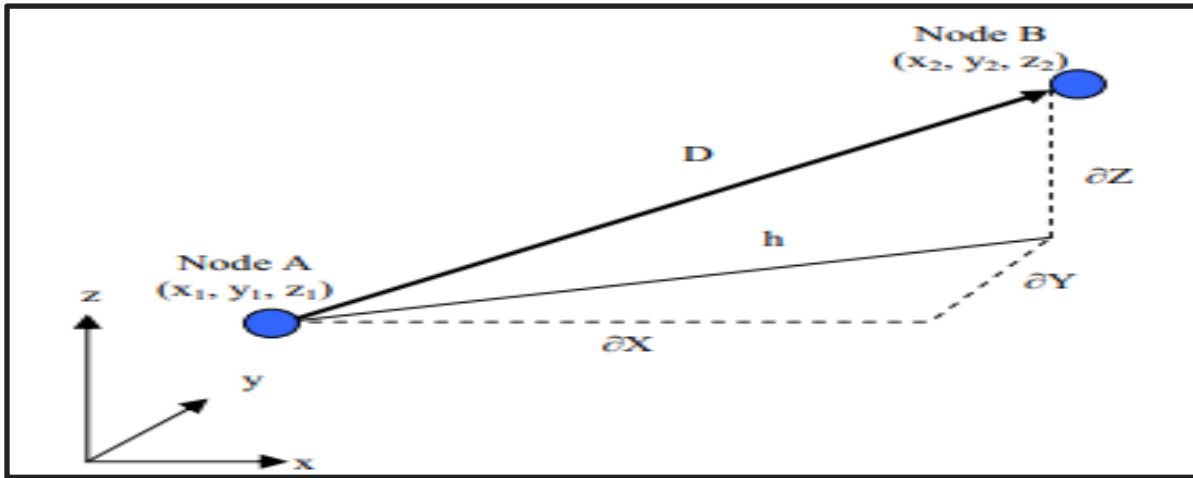


Figure 2.2:3D point differential for node A (x_1, y_1, z_1) and node B (x_2, y_2, z_2) in 3D axis[11].

2.3.1.3 Indoor Shortest Path Result: Figure 2.3 shows the result of the Dijkstra's calculation statistics. The shortest path calculation windows display the information about the Status, Date, Calculation time, Nodes Calculated, Nodes Used, Shortest Distance, and both information about current and destination location (including the x, y and z coordinates). The Status of the calculation is the information whether the executable calculation is successful or the destination location is not reachable for the user to navigate to. Shortest Distance shows the distance (shortest) in meters from current location to destination location. The system is also able to show the shortest route graphically in the 3D network as illustrated. The shortest route of the 3D navigation is highlighted. [11]

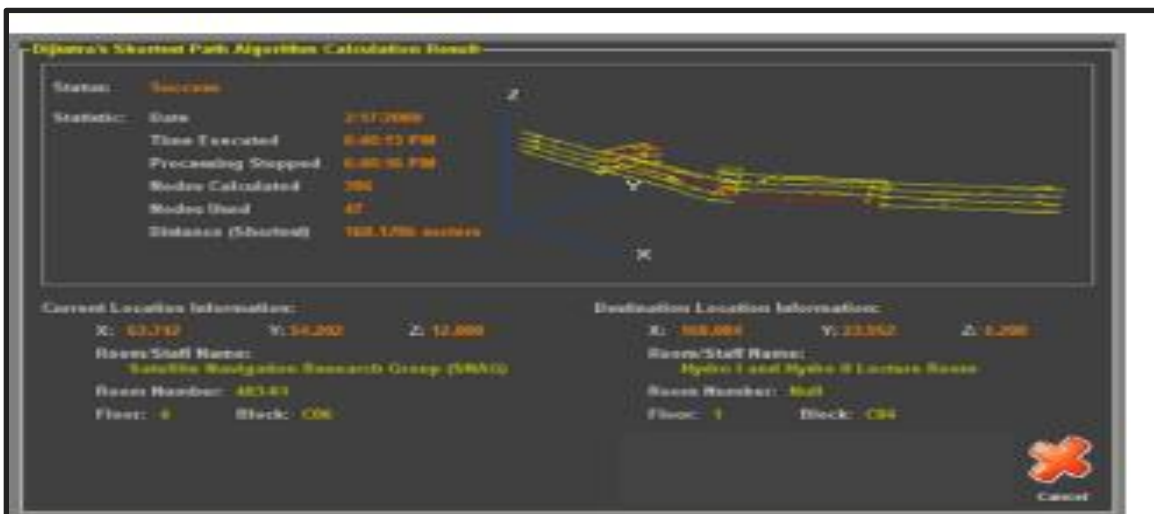


Figure 2.3: The Shortest Route Calculation Windows [11]

2.3.2 Outdoor Navigation

The second component of the system is the outdoor navigation. The 3D outdoor navigation is developed to navigate users from location A to location B and at the same time provides them with the real spatial information of the area e.g. shapes of buildings, structures, and other objects. As mentioned in the previous sections, 2D navigation systems are proven to be useful in locating the desired locations, but with 3D navigation system, users can see how the real environment looks like, marks out the possible navigational landmarks (i.e. distinct permanent objects or buildings) and familiarize themselves with the routes. Figure 2.4 shows the outdoor environment in the navigation system. This navigation system, like in most 3D games, allows users to walk around and explore the outdoor environment. In detail, the system provides users with the following functionality e.g. users can set a desired destination and navigate to it, and retrieve information of the surrounding buildings.[11]



Figure 2.4:3D buildings with real textures in the outdoor environment [11]

Spatial analysis is an extremely important aspect in GIS as it allows users to conduct analyses on available data and assists them in making better decisions based on the findings. This functionality is a huge advantage as it makes the GIS applications more valuable and useful. The engine offers some great functionality that is commonly used in computer games and can be utilized as part of the 3D spatial analysis tools in this navigation system. [11]

2.3.2.1 3D Network Analysis: As mentioned in previous sections, the network analysis in this research is used to determine the shortest path from point A to point B based on the available paths. Users can select their destination and the system will determine the shortest path from the network. The path network is a combination of static and dynamic nodes. Static nodes represent the permanent paths that exist in the real world and are unlikely to change in the future unless there are no other ways. Dynamic nodes mark the temporary paths that are only available for a short period of time. With the dynamic nodes, users can add or delete nodes to create or update new paths in case changes occur in the real world. If the determined shortest path is somehow blocked or inaccessible, the algorithm will find another route to reach the destination.

The shortest path will be marked with navigational markers (arrows, lines) to guide users to their selected destination as shown in Figure 2.5.[11]



Figure 2.5: The system navigates users along the shortest path [11]

2.3.2.2 Database Integration: Since data is a very important element in GIS-related applications, databases play a crucial role in storing and managing those data. The attributes database is then linked with its spatial objects (3D buildings). All stored data can be edited, added or deleted during the runtime. The database is developed using proper DBMS software so that users can manage (add, edit or delete) the stored data externally (using the DBMS itself) and easier to convert the

database from or to another database format. This database plays an important role for 3D spatial query that will be discussed in the next section. [11]

2.3.2.3 3D Spatial Query: Spatial query on 2D environment is a common function in most GIS software. Since databases can be incorporated with this program, it is possible for users to generate spatial queries on 3D objects in the 3D world. The engine's ability to select 3D objects via Point-and-shoot (clicking on the object) technique allows users to retrieve information related to the 3D buildings by clicking on the buildings. The information displayed is retrieved directly from the attributes database as mentioned in the previous section. Users can add, edit or delete the information during runtime or using the DBMS itself.[11]

2.4 Virtual Reality Navigation Techniques

The four different VR navigation techniques are: **(a)Flight, (b) teleport, (c) world-in-miniature, (d)3D cone drag.**[12]

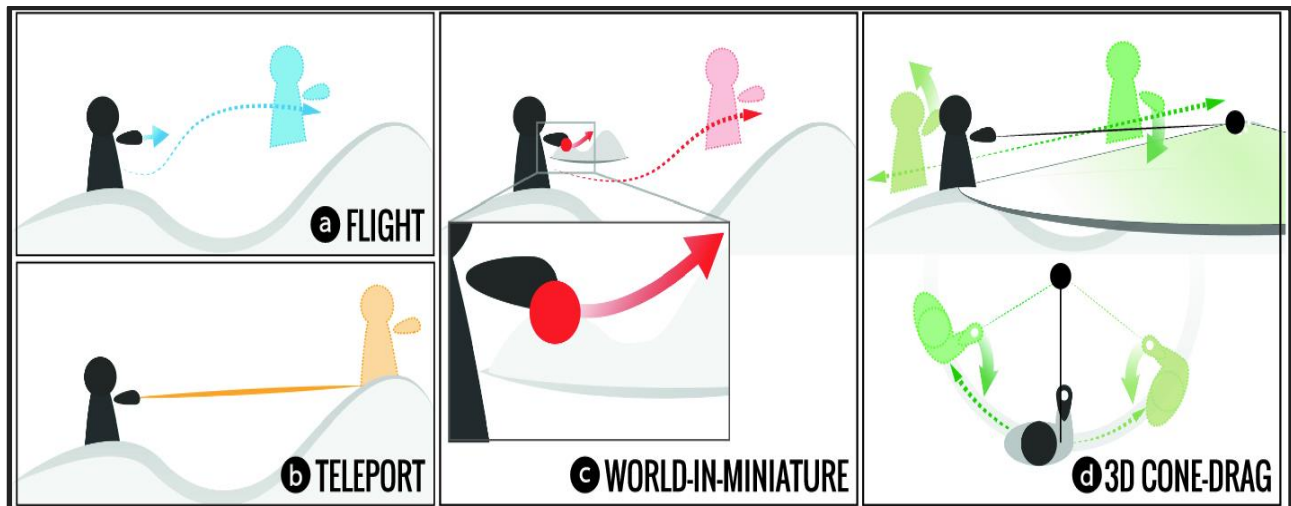


Figure 2.6: The four different VR navigation techniques: (a) flight, (b) teleport, (c) world in-miniature, (d) 3D cone drag. [12]

Flight: is one of the earliest navigation techniques for VR. When using this technique, a viewer moves through virtual environments as if they were controlling a flying vehicle, like a helicopter (Figure 2.6(a)). Flight involves a smooth, predictable, continuous, and reversible translation [12].

World-in-miniature allows the viewer to change their position by manipulating a version of themselves in a miniature version of the scene (Figure 2.6(c)). We use a 1m diameter mini-map, located at the center of the viewer's play area (Figure 2.8).

World In Miniature (WIM) gives the user another point of view from which to observe the scene, and the ability to change that point of view under direct manipulation as rapidly as the user can turn the model in his or her hands. [12]

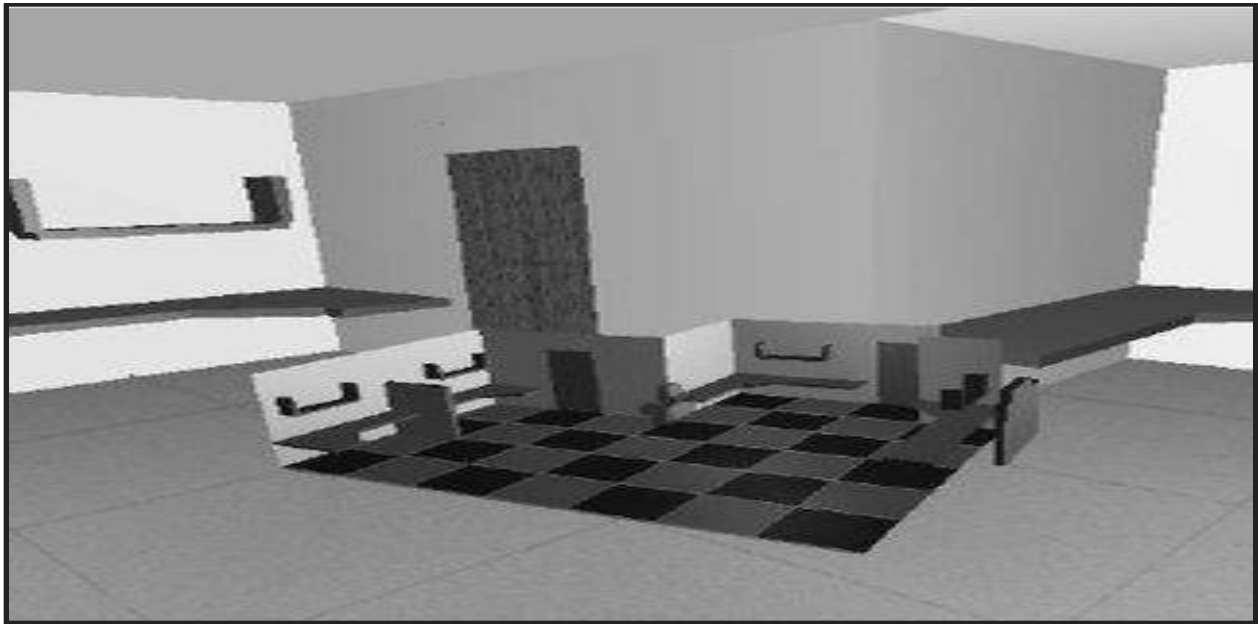


Figure 2.7:*The World in Miniature (WIM) viewed against the background of a life-size VR [12]*

The world-in-miniature (WIM) interface gives the user of an immersive three dimensional environment a chance to operate at several different scales through several different views at once without engaging explicit modes. From our informal user observations, we believe some of the most important features of this technique to be [12]:

- intuitive use of the WIM metaphor.
- multiple, simultaneous points of view.
- multiple scales for selecting and manipulating objects.
- easy manipulation and selection of objects at a distance.

- improved visual context.
- removal of occlusion and line of sight problems.
- general applicability across a wide range of application domains.

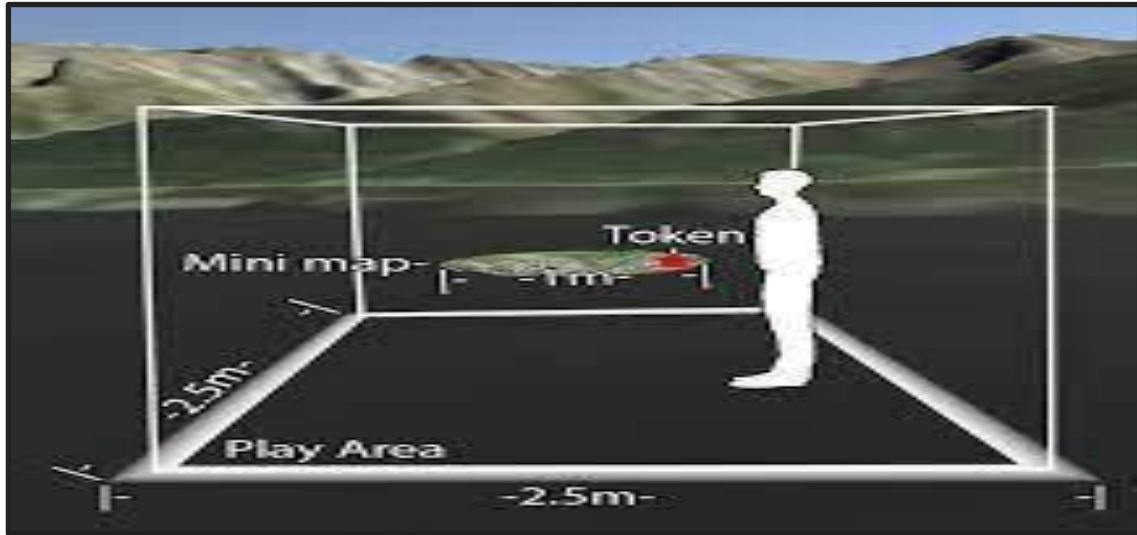


Figure 2.8: The 2.5m play area and 1m diameter mini-map configuration used in our test environment [12].

3D Cone Drag allows the viewer to directly manipulate the scene as though they were directly grabbing and dragging the landscape. We examine 3D Cone Drag because of its prominent use in Google Earth VR [5], a popular commercial application which involves exactly the type of large scene navigation tasks that we are interested in exploring. [12]

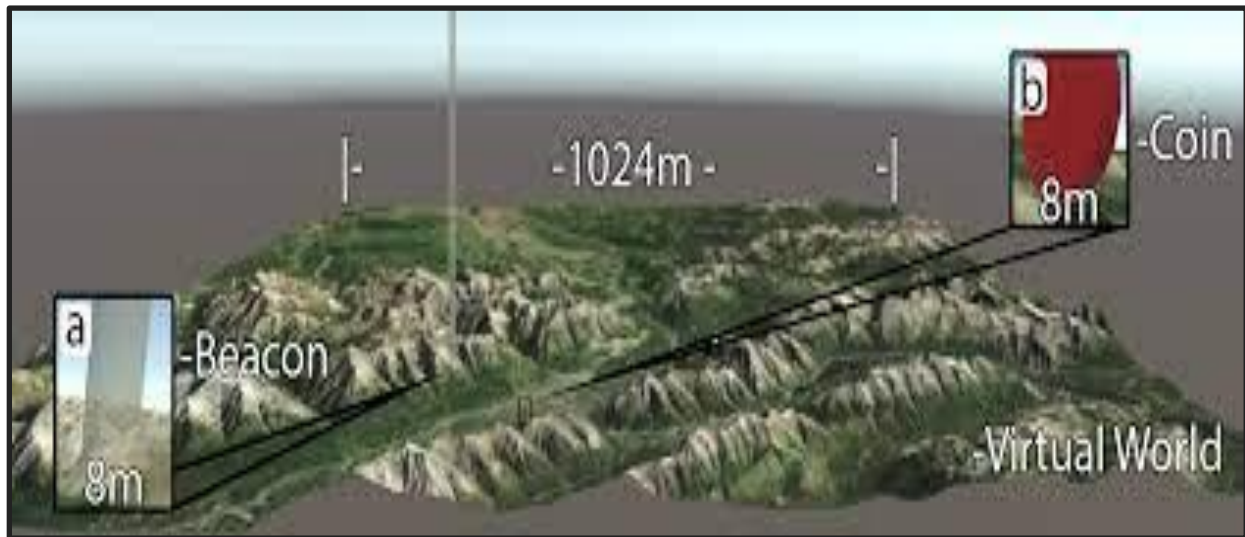


Figure 2.9: Overview of the entire virtual world used during the study tasks. Insets show beacons (a) used for navigation tasks and coins (b) used for search tasks [12].

Teleportation

Teleportation is a widely implemented virtual navigation technique that allows users to navigate beyond the confines of available tracking space with a low possibility of inducing virtual reality (VR) sickness.

Teleportation is a common virtual navigation technique used to traverse larger spaces. The user may specify a destination in the virtual environment by pointing. This is commonly done using a handheld motion tracked controller. When activated the user's viewpoint may instantly transition to the specified destination (Figure 2.6 b). [12]

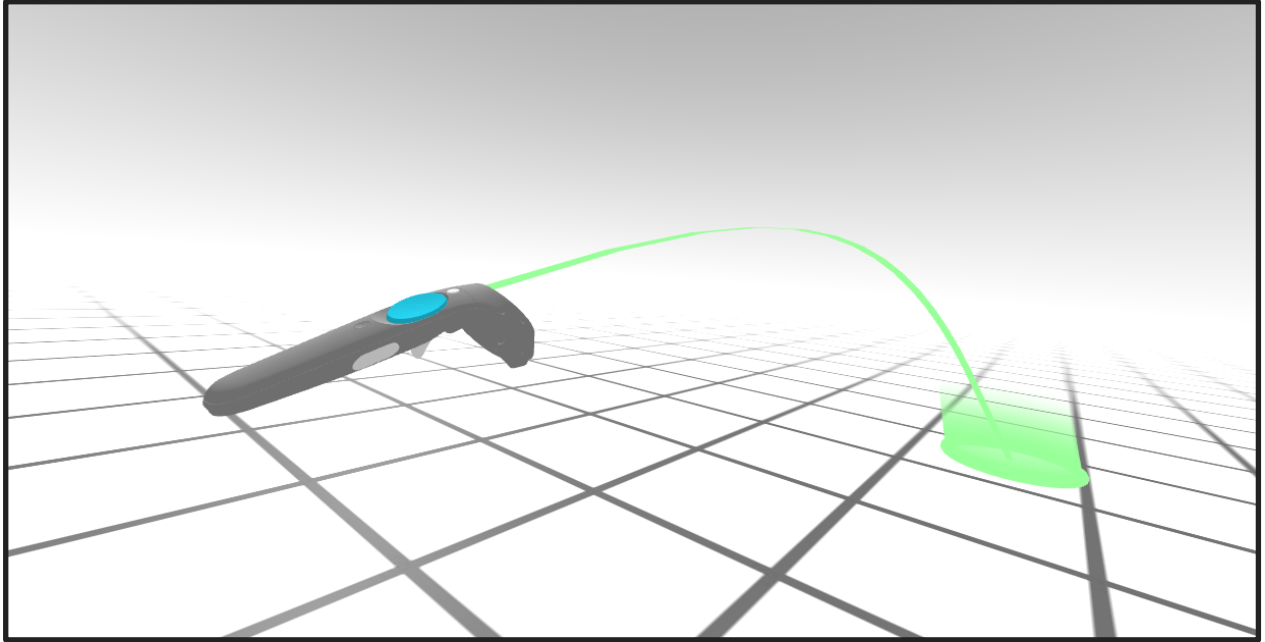


Figure 2.10: A picture showing the way of teleport navigation in virtual reality [12]

Teleportation has become the de facto navigation method for many VR experiences and is often used as a benchmark to compare new virtual navigation techniques against. There are different teleportation implementations used based on a few distinguishing features. We can classify them into:

operation: The user uses a motion sensing hand held controller to point towards a destination to teleport to and then activates teleport using a button press. Alternatively, hands-free approaches for selecting a destination are possible using eye gaze, head gaze or by pointing with the arm or hand which are tracked externally using a camera. [12]

Translation: When teleport is activated the user's avatar is instantly translated to the destination (discontinuous translation) or alternatively the user's avatar is moved rapidly maintaining a continuous representation. [12]

Concordance: A few implementations allow users to specify their orientation post-teleport (discordant) or they maintain their orientation (partially concordant). [12]

Range: Users can teleport to any destination within their visible range (commonly within a specific distance threshold). Contrarily, they may only be allowed to teleport to set locations in the environment. [12]

2.5 Factors influencing Navigation in Virtual Environments

In the field of visualization, it has been shown that effective navigation in a virtual 3D environment within an application area depends on a combination of characteristics stemming from the following six factors: environment, user, task, navigation strategy, navigation aid, and motion control. [14]

The factor environment describes structure and different attributes of VEs like display mode, topic, degree of abstraction for representation, size, density, activity and single-/multi-user environment. The choice of the virtual environment and the design of the environment with all of its attributes depends only on the application itself. The design of the environment may depend indirectly on the navigation aids and, to be more specific, if the user desires particular navigation aids which have to be integrated in the environment. [14]

The factor user defines the characteristics of the user (age, gender), as well as the abilities and weaknesses of a user. The user group is determined by the application, which fixes the general, application-oriented characteristics of the users. Application-oriented and individual characteristics of the users influence the choice of the task or goal the user pursues in a virtual environment, and the navigation strategies, which are necessary for performing the task or for reaching the goal. Likewise, they influence the selection and the design of the navigation aids and the motion control. [14]

The factor task defines the problem a user would like to solve in the environment. For solving a problem in VEs different navigation tasks must be performed, e.g. target search, exploration, etc. For an effective navigation technique in a VE, the task to be performed in the environment has to be selected in accordance with the application and the environment. For example, in designing the presentation of the tasks, additional user information has to be included. [14]

Navigation strategies describe concepts people use to navigate through an environment. Navigation strategies are complex strategies which consist of elementary navigational tactics like

'searching', 'vectorial navigation' and 'positional navigation' [15]. These in turn consist of basic actions (i.e. 'reaching target', 'course following' or 'wall following' [15]). The selection of the navigation strategy depends on the user, on the environment and on the task. Users can prefer particular strategies. If these preferred strategies are not supported, as for example through the supply of corresponding navigation aids, then the user must first become accustomed with those aids which can impair the effectiveness of navigation. However, not all strategies are sensible in a defined environment. Consequently, the parameters for the environment also determine the choice of strategy. [14]

Navigation aids factor are aids for determining direction and/or position. They can be both part of the environment (like landmarks, sound or signs) as well as also consisting of external objects (maps or compass). Navigation aids support orientation in space and therefore serve the purpose of finding the way. They include:

(a) aids which exclusively serve the function of determining position (maps showing actual position, coordinates or identification signs).

(b) aids which exclusively serve the function of determining direction (compass, directional signs).

(c) aids which are a combination of both (reassurance signs, sounds or markers). Navigation aids depend directly on the strategy selected, as the aid is implicitly included in the strategy. [14]

Motion controls are the hard- and software components for moving through a VE. Motion control is further differentiated into input devices, motion aids (e.g. cursor, direction buttons, avatars etc.) and interaction/motion technique. The interaction/motion technique defines motion metaphors like 'walk', 'fly', or 'jump' and different parameters of movement such as direction/target selection, velocity/acceleration selection and input condition [16]. The last parameter, motion control, specifies pure movement through a virtual environment. The choice and the design of motion control depends very substantially on the user and the navigation strategy, as well as the environment. In this context, the motor skills of the user, for example, are of particular importance. The navigation strategy can also influence the pure movement in an environment. The strategy of 'automatic path following' presumes that the user himself/herself does not control the movement, but that s/he is being 'transported' and so hands over control to the system. The display mode of a VE particularly influences the motion control. For example,

immersive environments demand other input devices and motion metaphors than desktop environments. [14]

Table 1 summarizes factors influencing the quality of navigation and shows example characteristics:

Factor	Example Characteristics
Virtual Environment	Display Mode: immersive VE, desktop VE, projected VE, augmented VE • Topic: city, building, landscape, sea etc., information visualization • Degree of abstraction for representation: very realistic up to abstract • Size: small, large, infinite • Density: sparse, dense, cluttered • Activity: static, dynamic • single-/multi-user environment
User	Characteristics: age, gender • Knowledge about: currently perceivable sub-section of the environment, current environment model, application task, application domain, real world and other environments [19] • Experiences in: use of computer, 3D environments • Abilities: motor, perceptual • Disabilities: motor, perceptual • Expectations
Task	Problem with primary and secondary navigation task: Primary navigation task: movement with wayfinding, that means searching (e.g. exploration, naive search, primed search [18]) • Secondary navigation task: application tasks with movement and wayfinding as secondary processes.
Navigation Strategy	Navigation using specific aids • Search strategies [8]: edge following search method,

	lawnmower search method, heuristic search method, area search method
Navigation Aids	Orientation: compass, directional signs • Position: maps showing actual position, coordinates, Identification signs • Combination: reassurance signs, sounds, markers
Motion Control	Input devices: keyboard, 2D/3D-mouse, joystick, spaceball • Motion aids: direction buttons, cursor, avatars • Interaction techniques: Motion metaphors: walk, fly, jump, teleport • Direction/Target Selection: gaze-directed steering, pointing/gesture steering, discrete selection (lists, environmental target) [16] • Velocity/Acceleration Selection: constant velocity/acceleration, explicit selection (discrete, continuous), user/environment scaling, automatic/adaptive [16] • Input Condition: constant movement/no input, continuous input, start and stop input, automatic start or stop [16]

Table 1. Factors influencing navigation and their example characteristics

2.6 Interaction in Virtual Environments

2.6.1 Introduction

In the real life we heard a lot the term interaction: in our relationship with other humans or what we call social interaction (the interaction between the manager and his workers, the interaction between family), in the political interaction (like the political conventions that took place during the presidential campaign), in the nature there are interaction (the melt of some rocks by heat create the magma that pushes through vent and fissures to the earth's surface (volcano)).even in

the technology there interaction and the one we interest in is the Human-Computer interaction especially the interaction in Virtual Reality domain. All of these examples of interaction are useful but our main goal is to understand what is interaction in general? What is interaction with virtual reality in-depth? [20]

2.6.2 Definition of Interaction

There a lot of general's definitions of interaction in general: reciprocal action or influence or communication, direct involvement with someone or something, but the one I found have a scientific concept is this: a particular way in which matter fields, and atomic and subatomic, particles affect one another through gravitation or electromagnetism, but these definitions are not our subject we interest in. Our main subject is the interaction in virtual reality, so we need to define it.

Virtual Reality has three distinct characteristics: interaction, immersion and imagination. Interaction refers to the natural interaction between the user (Human) and the virtual scene or the virtual environment (3D objects, virtual human...). [20]

Interaction in VR is the process by which people control their movement and their behaviors using environmental cues and artificial aids such as maps so that they can achieve their goal without getting lost.

Years of research have produced a myriad of interaction techniques for VR and these techniques support one of the three main types of action:

- Selection
- Manipulation
- Locomotion

Selection

The first type we are going to talk about is selection

Selection or object selection is an important task in a virtual environment and one of the most frequent activities for most common user interaction. Selection involves telling the system which object or UI element the user wishes to interact with. Once selection is confirmed by the user, the selected entity becomes the focus of further interaction inputs by the user. Selection can be performed using controller input, gestures or gaze.

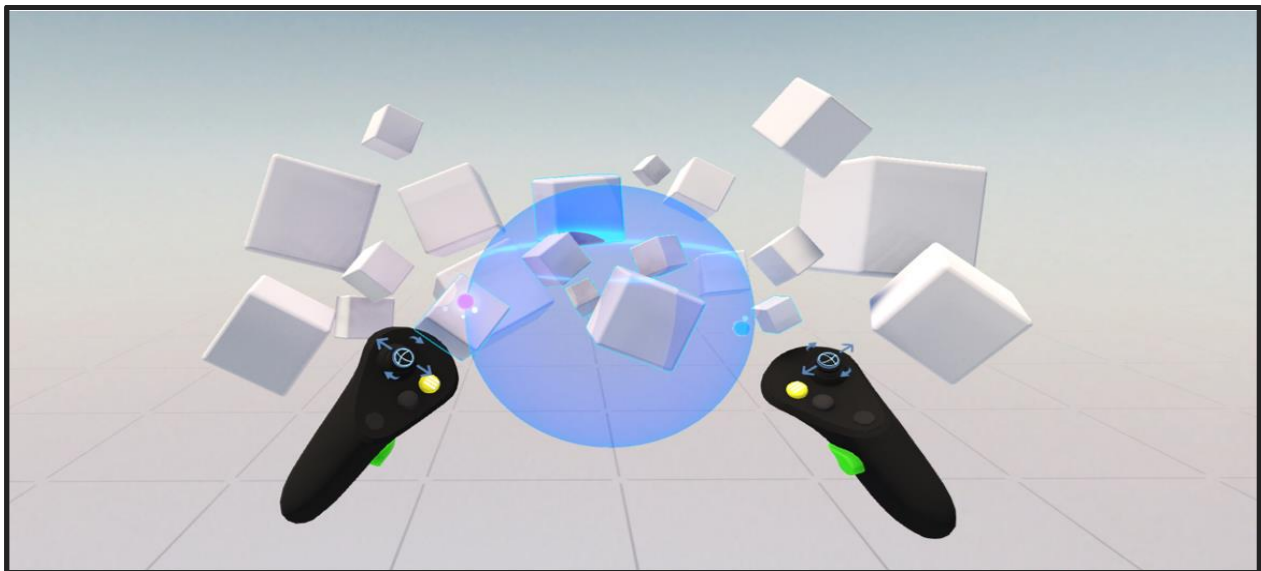


Figure 2.11: A picture showing selection objects using the controllers in a Virtual Environment [12]

The amount of data available around selection techniques inside a virtual environment (VE) is instead giant and describes in-depth the user of various interaction metaphors such as voice, gaze, laser, and virtual hand.

Controller Input (Laser-pointer or Rat-cast): This is the most famous selection technique.

A virtual laser-pointer is fired from the controller and the user changes the controller orientation to re-orient the laser-beam and intersect the pointer beam with a desired object.

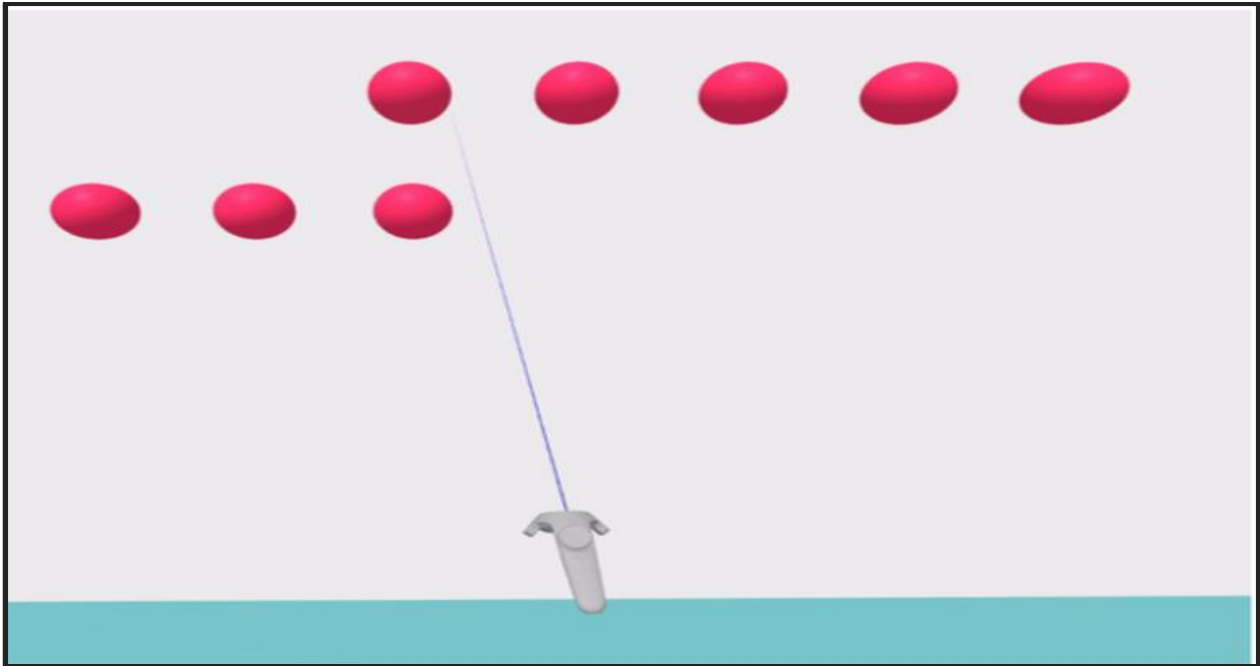


Figure 2.12: A picture showing Laser pointer selection in Virtual Environment [12]

Gesture: VR headsets which can capture and interpret actual hand gestures are becoming more common. Using hand-held passive props, active 6Dof controllers and camera capture, a wide range of gestures can be identified by the system. Nearby objects can be grabbed naturally if they are close enough.



Figure 2.13: A picture showing Gesture based selection in Virtual Environment [12]

Gaze: Gaze-based selection involves the head-pose or actual eye-gaze. Gyroscopes inside VR headset can easily track head-pose. Actual eye-gaze requires costlier and complex eye-tracking hardware. The selection interface element is a reticule (mainly for head-pose) or cursor displayed in the virtual environment. The position of the element is updated based on head-pose or gaze

of the user. The user starts the selection process by looking at the object directly. The selection can be confirmed by an external controller input or by dwelling on the target for a fixed period of time.

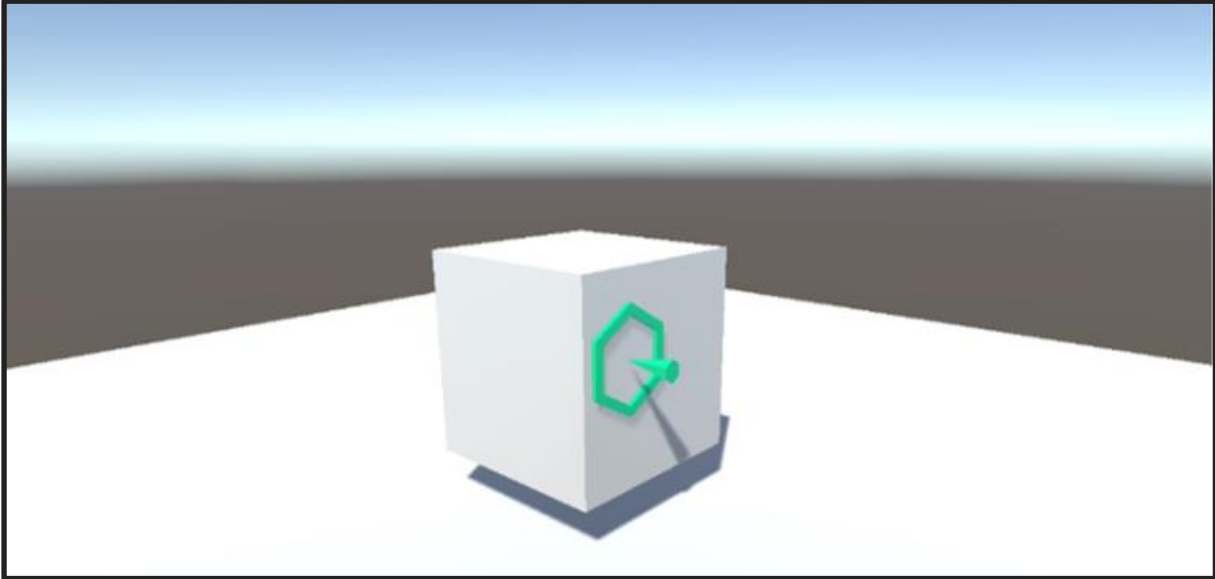


Figure 2.14: A picture showing Gaze base selection pointer with cursor in Virtual Environment [12].

Manipulation

The second type is Manipulation: This set of interaction actions occurs once an object is selected by the user. Manipulation can be performed using *S/R/T*, Affordance-driven manipulation.

***S/R/T*:** once an entity is selected by a user, they may want to manipulate it via: resize it, re-orient it or move it. These fundamental actions can scale, rotate, or translate the selected object. A variety of interaction techniques allow each individual action to be performed. The choice of technique depends on the available input controller's capability.

Affordance-driven manipulation: The user may also want to operate the entity based on the affordance it offers, e.g. push a virtual button or crank a lever. Care must be taken by the virtual world creator to expose the affordance in unambiguous ways and support the input capabilities of the hardware used by the user.

Locomotion

The third type is locomotion: This set of interaction techniques enable user movement within the virtual world. They reposition or re-orient the user in the virtual world. Locomotion can be performed using Locomotion and Vection, on rails, Gaze-directed steering, Teleport, *Real movement*.

Locomotion and Vection: The most important challenge of locomotion interaction is to reduce or eliminate vection. When a user performs locomotion in the virtual world while being still in the real world, they experience a feeling of disorientation or vection. The user's visual system sees movement while the body's balance apparatus indicates lack of movement. This manifests as visual-vestibular conflict and a common cause of nausea and simulator sickness. Locomotion interaction techniques manage vection to varying degrees.

On rails: The user's movement is controlled by the system in an 'on rails' fashion, e.g. a roller coaster simulator. This can result in vection very easily and needs to be controlled carefully.

Gaze-directed steering: On rails movement can be extended by allowing the user to look in the direction they wish to move. The movement can then either be at a constant velocity or triggered by another input.

Design considerations for interactivity: Modern day PCs and tablet devices have a standardized set of inputs (keyboard, mouse, touch-screen). They also implement a standardized set of interactions (e.g. Ctrl+C is recognizable as the copy command). In contrast, VR inputs and interactions are not standardized. The creator of a virtual world has to make critical decisions on how accessible their world is going to be to the user. These decisions are based on the choice of hardware and the interactions necessary for the VR experience.

Teleport: Users can use a gaze-based or controller-based raycast to select an area of the environment they want to move or 'teleport' to. This is often coupled with a rotation element, so that when selecting an area to teleport to, the user can also specify the direction they want to face when they teleport.

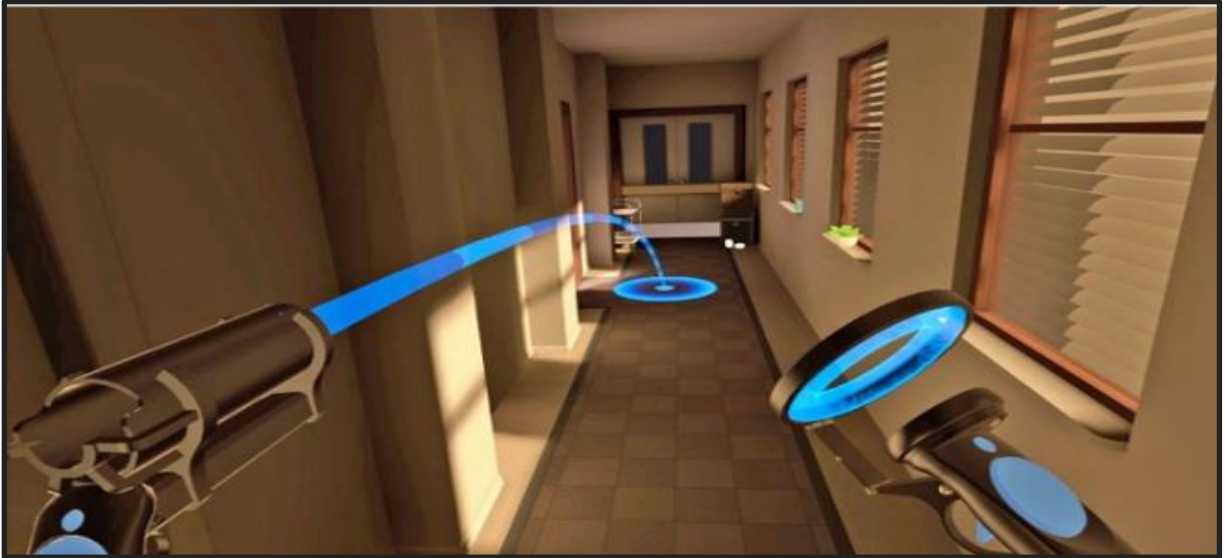


Figure 2.15: A picture showing Teleportation based locomotion in virtual environment [12]

Real movement: While using VR headsets like the Oculus Quest, users can simply walk around the virtual world. This reducesvection. However, walking around can increase the risk of collision with real-world objects. Manufacturers provide some form of proximity alert system to protect users in this scenario. This alleviates the collision risk in the absence of an omnidirectional treadmill.

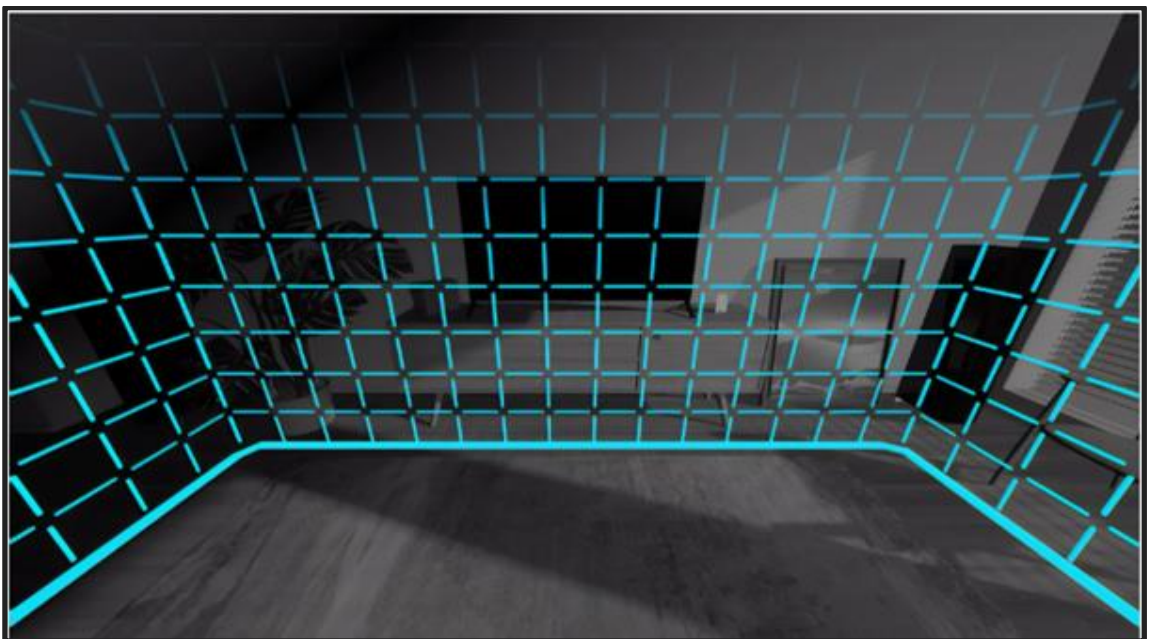


Figure 2.16: A picture showing Quest’s guardian protection system [12]

2.6.3 Collision Avoidance and Interaction

In the field of cars collision avoidance or collision avoidance system(CAS), also known as a pre-crash system, forward collision warning system, or collision mitigation system, is an advanced driver-assistance designed to prevent or reduce the severity of a Collision. In the field of Virtual Reality(VR) the collision avoidance aims to design a trajectory in which the player or the objects passes through a specific or random target at a safe distance and moves to a specific point. We also hear the term collision detection but Collision Avoidance Isn't the Same as Collision Detection Collision Detection lets the objects collide and bounce off each other but the collision avoidance tries to get the objects to change their paths so that they don't collide in the first place. Collision Avoidance(CA) it's an automated course planning hassle, this is due to a calculating for a specific player a path free of any Collision by having only the final destination. [21]

We have a lot a algorithm of collision avoidance we can use in our application:

Trial and Error algorithm(essais-erreurs): this algorithm calculates the volume generated by displacement of the moving object along the path, if there is an obstacle it determine the interaction between the generated volume and the obstacle, if it's necessary it suggest a new path and the object(Avatar or player) walk only if the obstacle are rare so that they can be treated once a time. This algorithm is more dedicated to Collision Detection (CD).

Algorithm of the graph of visibility of Lozano-Perez: In this algorithm we connect the vertices that can be seen and then take the shortest path collision free court from original to the goal.

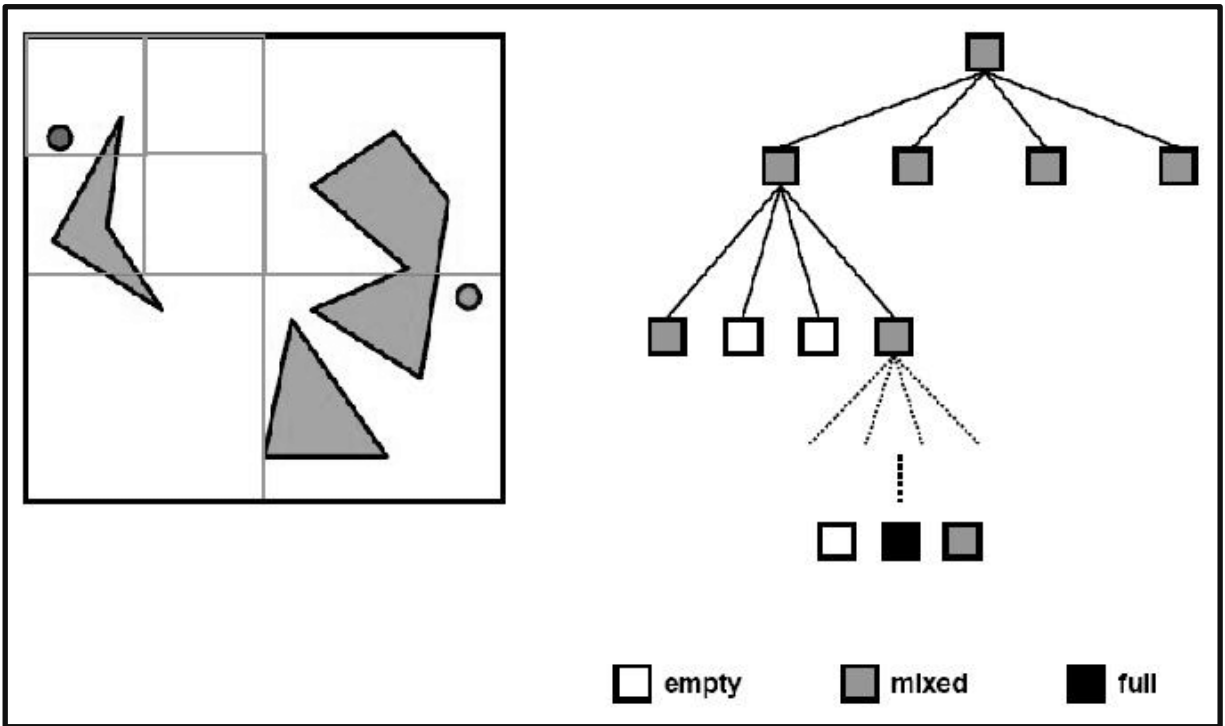


Figure 2.18: Image show an explication of the QuadTree algorithm

2.7 Conclusion

Virtual reality is an experience where a person is provided with the freedom of viewing and navigating in a virtual world. The experience is not constrained to a limited control. Navigation and interaction of all sorts is pervasive in the physical world, perhaps even central to our existence. We interact with other people, with other living beings, with objects, with information- we can even say that we interact with our (and other people's) thoughts and ideas. Interaction and navigation in VR can take place between the learner and the system, between several learners (and other teachers), and between learners and virtual objects.

Chapter03

System implementation and Architecture Design

3.1 Introduction

Virtual Reality (VR) research is continuously striving towards natural and seamless human-computer interfaces, and the existing interfaces for locomotion through virtual environment (VE) are still not quality and effective. Typically, participants navigate by using a handheld device such as a joystick or a wand.

They are then exposed to conflicting stimuli: the world around them seems as if they were moving and navigating in it and avoiding objects, but they feel their body is stationary and the experiments aren't reel. This results in a reduced sense of being present in the VE Virtual Reality (VR) technology provides a computer-generated 3D environment, in which the user can navigate in the virtual environment and interact with virtual humans (avatar) or virtual objects in real time and walk naturally with the locomotion techniques. However, the problem is the way of describing the mechanism of interaction between a real human (with an intelligent processor which is the brain) and the virtual human (Automated).

3.2 Presentation of system for collision avoidance

3.2.1 System objectives

The proposed solution, and the objective of this work is the development of a virtual reality experience game that simulates the interaction in a virtual environment between a real human and a virtual human or objects and find solution to the collision during the interaction.

In our game we will have 4 scenarios where every scenario represents an interaction type which are human and virtual human (Point of view of the human), human and objects (obstacles), virtual human and objects (obstacles) and virtual human and human (Point of view of the virtual human), we will use the continuous movement and teleportation methods of locomotion technology that enables movement from one place to another (locomotion) within a virtual reality environment and make some observations about the 2 methods and how they affect the interaction.

In that regard, this tool needs to encompass all capabilities of VR application: immersion, high-fidelity, presence and interactivity. Therefore, a focus on the visual design and creation of high fidelity, realistic assets, is required to improve the immersion of the game and to trigger interaction responses.

3.2.2 System Architecture Design

In this phase, we will take an overview of the general structure of our system and its components. Our system was built in order to include different determinants that influence the success of our experience in our VR game. It includes virtual world and human-virtual human characteristics and the virtual world-characters interaction and influential factors as determinants of a VR application’s quality.

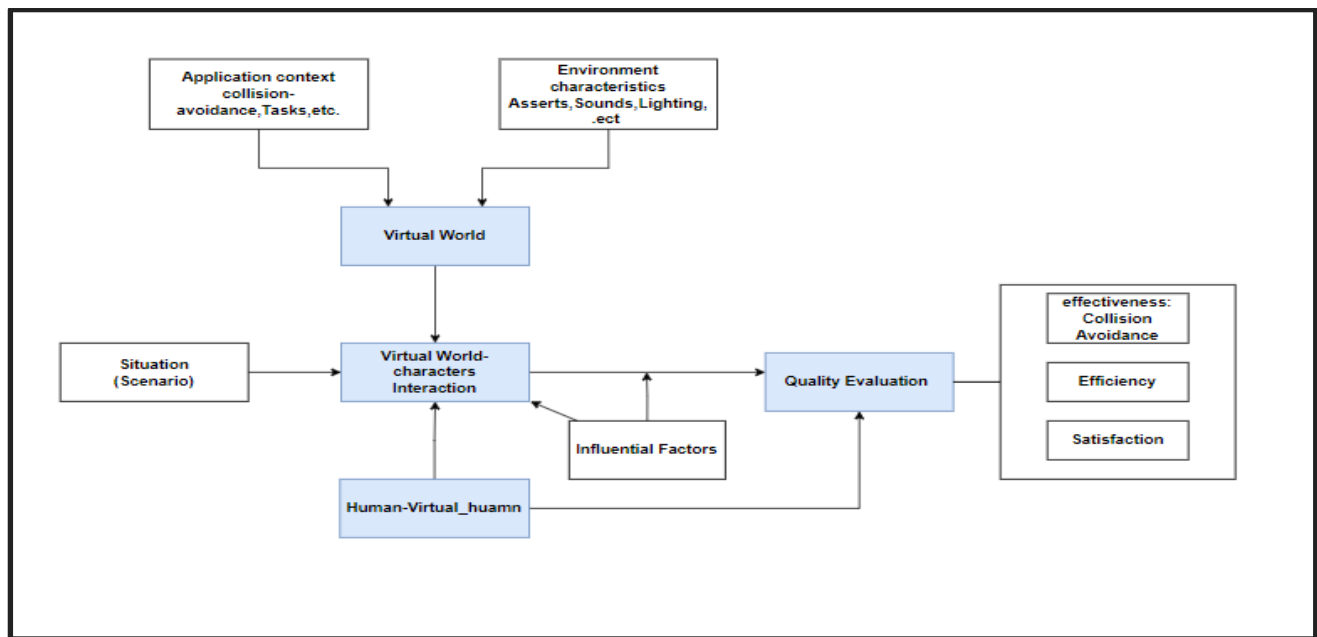


Figure3.1: Image show the System Architecture

3.2.2.1 Virtual World

Our system is primarily based on the reel immersive virtual reality experience because of this the player absolutely immerses into the computer-generated environment. Consequently, and to break his connection with the reel world, we have chosen a number of functions to be designed and manipulated according to the project's goals.

- **Application context:** specifies the purpose of the specific application, which in our situation is to prevent collisions between the human and the virtual human or the objects(obstacles). It also comprises the precise tasks that the player(humans) must do, as well as the task features (e.g., different situations) that are deducted from the unique context.
- **The Environment Characteristics: determines** how the environment should be built to get ready for designing and rendering in the next phases. Since we aim to offer a realistic interaction between the human and the virtual human that can simulate the same interaction between humans in reality, a suitable design is a necessary requirement for satisfactory use and outcomes of our system, for example, poor user interaction in VR can decrease performance so this requires a coherent environment with realistic objects.

3.2.2.2 Human-Virtual human

- **Human:** is the developer of the application or a player (real person) with self-intelligence who covers biological, physical, psychological, and social characteristics of participants and is based on the human factors definitions by [22]and It includes human capabilities as well as human limitations that are relevant for safe, comfortable, effective design and use of our application [22].
- **Virtual Human: The** Virtual Humans or Avatars are computer-generated characters who don't have any features of the human being except the acquisition of information by using artificial intelligence to do tasks that are usually done by humans because they require human intelligence and discernment.

3.2.2.3 Virtual world characters' interaction

The virtual world characters' interaction, which represents a player's real usage of the system, is the third component of our system design. Users traverse, interact with, and experience diverse circumstances in the virtual world based on the scenarios they choose, and as a result, they demonstrate particular behavioral responses that are connected to performance. The

interactions inside the usage setting must be consistent in order to have the same interaction impact as in the actual world.

3.2.2.4 Influential factors

The influential factors affect the virtual world characters' interaction and they can influence the quality measures of our VR application. These factors can be represented as the shape of the obstacles, the speed of the dynamic objects (human, virtual human) or in the hardware like the headset and the controller's configuration.

3.2.2.5 Quality Evaluation

The quality of our system indicates the VR application's output. The International Organization for Standardization (ISO) defines quality as "the degree to which a collection of inherent qualities meets standards" (International Organization for Standardization 1998). The degree denotes the degree to which a product or service satisfies, which can be expressed as good or bad product quality, for example. This may be further broken down for VR applications into usability-related characteristics of quality (effectiveness, efficiency, and satisfaction; (International Organization for Standardization 1998)). Increases in these three quality criteria are a measure of influence on usability, which may either improve or deteriorate the usefulness of an application. The following are the definitions for each of the attributes:

- **Effectiveness:** degree to which the player correctly and completely achieves the different scenarios.
- **Efficiency:** resources expended by the player to correctly and completely achieve the game goals.
- **Satisfaction:** degree to which user needs are satisfied by using the VR game application.

When the desired goal of the application is accomplished, which implies an improvement in performance with the fewest resources available, and when system usage is pleasurable for the

users. The capacity to recreate a realistic interaction between a human and a virtual human has been chosen as an outcome variable in our project.

3.3 Methodology Setting

Is the study of research methods, or, more formally, "a contextual framework for research, a coherent and logical scheme based on views, beliefs, and values, that guides the choices researchers [or other users] make". We need a process(methodology) for developing a virtual reality Interaction application that included a scenario-driven virtual environment. As a result, based on our system design and resources, we attempted to build an appropriate strategy for constructing a robust and engaging virtual environment with interacting objects. Our methodology is divided into 6 steps.

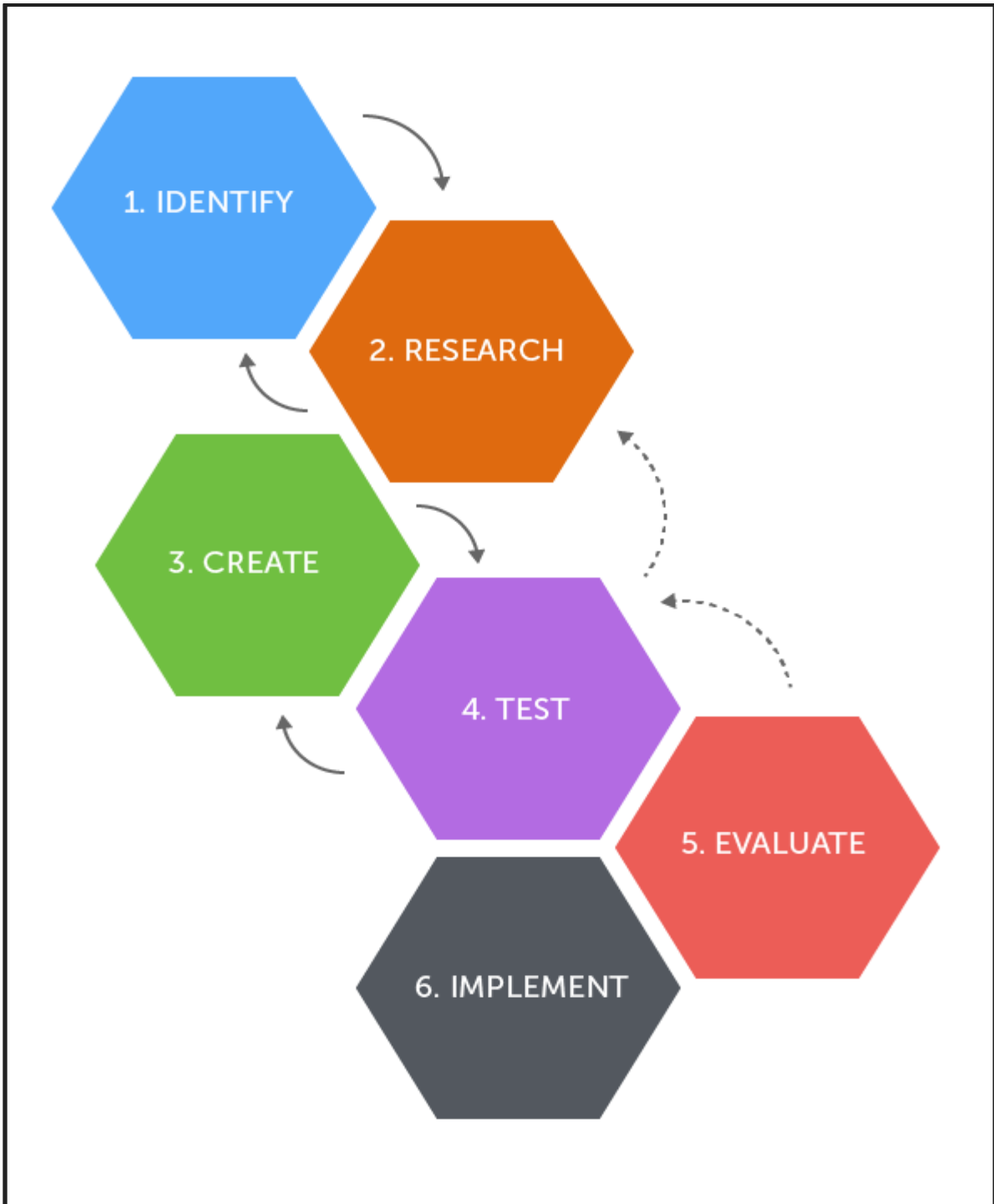


Figure3.2:Image show the steps of the methodology

3.3.1 First Step: Identify

Identify has a lot of meaning like distinguish, recognize, differentiate, characterize, set apart meanwhile, in our context it means all of these terms. It's the first step for any research methodology because it sets up the base ideas, data, resources and the main reasons and goals for the research. This step was completed as meticulously as possible, with all aspects taken into consideration, as it serves as the framework for future work. To express the requirements and objectives that needed to be fulfilled, we first developed an overall concept for the serious game application. The most crucial component and first step was to define the purpose of the VR application, which in our case was to simulate a realistic interaction between a real human and virtual human. The second step is to define the type of interaction which is in our case the collision avoidance with dynamic objects (real human and the virtual human) and static objects (obstacles). Then we explained how the environment should look, including objects, sounds, and lighting, as well as what should happen when users of the program provide input, so we can create a precise list of requirements for the development phase.

3.3.2 Second step: Research

The second step is research, divided into two phases where the first phases depend on making research for previous works to avoid the repetition of ideas and result and plagiarism or to upgrade a previous work with new ideas or data.

The second step after the research and the first step which is identified is analyzing what will be used as an input for the assignment phase to build the foundation for the entire application.

The environment description and the full scenario are used to generate a list of items as well as a list of actions and activities. The objects were divided into categories depending on their appearance and function. The next stage was to define the items that will be animated in the usage case, as well as their states. As an example in the scenario where the real human interacts with a virtual human and what meaning by interaction is how the real human avoids the collision

is up to the reel human to turn even right or left or even jump over the virtual human if it possible but in the other scenario when the virtual human avoids the reel human we have some constraints and conditions.

This indicates that we can pick which states should be joined with others. After that, we gave actions to the objects that lead to state changes. This allowed us to determine which sorts of objects share the same actions and whether or not two distinct types of objects' actions may be controlled by the same script.

Figure 3-11 illustrates the user's interaction with the VR system. When the player enters the game he can move with continuous movement with the left controller or teleportation with the right controller and navigate in the virtual world. Once he/she face an obstacle if the obstacle is a static or nor move obstacle he/she need to avoid it else he/she freeze at her/his place, if the obstacle was dynamic in our case a virtual human, he/she have two option, the first option is to stop and the avatar will avoid him/her, the second is to avoid the avatar and continue to navigate in the virtual world. When he/she want to end the game he/she press the oculus button and the game application appear with two button Exit or Resume.

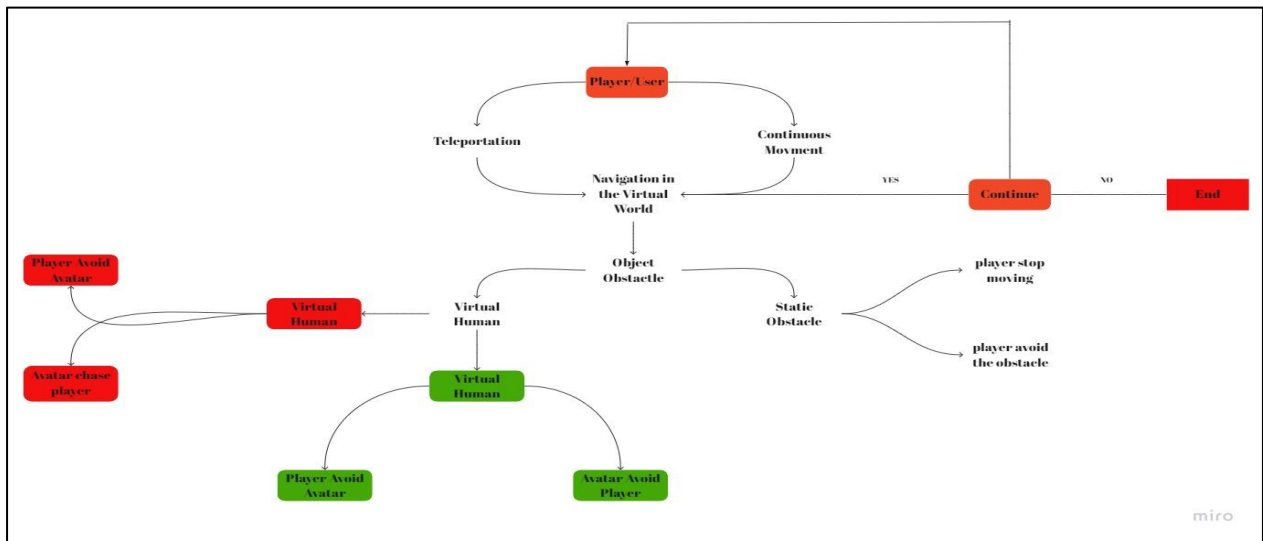


Figure3.3:Image show the Activity diagram of the user/VR system interactions

3.3.3 Third step: Create

In this step we focused on the coding process that involves developing the virtual environment's assets based on the list of items and scenario writing from the previous phases, which requires building the little bricks that make up the entire virtual world. To develop a realistic scenario for the interaction between our characters, a variety of assets, scripts, texts, visuals, animations have to be generated. By graphics, we mean both 2D graphics, such as graphical user interfaces (GUIs), and 3D models, which depict various elements found in environment architecture, such as high buildings.

The interactions between the reel human and the visual human in the virtual environment are based on animations. Almost every state change is followed by a matching animation, which may either be based on the state itself, such as the nature of the objects, etc. Then, to run the virtual environment, we backed up all of the activities using various types of scripts. There are scripts that hold the state of things, scripts that modify the state of objects, scripts that guide the reel human or the player through his or her experience, and so on. The state diagrams created based on the comprehensive scenarios serve as the scripts' basic foundation.

3.3.4 Fourth step: Test

We started the testing process as soon as we finished developing our virtual environment. This involved confirming that the work is advancing in compliance with the application requirements by testing all of the assets together, the scripts, the interactions, and the overall feel of the environment. We made sure that our simulation of the collision avoidance between the reel human and the virtual human fell realistic to the player.

3.3.5 Fifth step: Evaluate

In our game application this step needs participants but in our case due to the short time we just did a little evaluation with a few number of participants so we can't take the validate of the evaluation into account.

3.3.6 Six step: Implementation

The main focus of this phase was to synchronize the software with the hardware and test it for collision avoidance. Calibration is very important for stereoscopic projection and haptic interaction because the player or the real human must feel the realistic interaction in a virtual environment. Final testing was of course conducted after deployment

3.5 Conclusion

Collision Avoidance is considered one of important topics in the technology industry especially in the virtual reality thanks to the solution that provides humanity in avoiding materials and human loss. In this chapter we discussed how to develop a realistic experience that allows the player to interact in a virtual environment like he/she interacts in the real world. After that, we presented our proposed approach, which aims to provide the player or the participant with an effective VR application in the form of a game, as well as the architecture design description of the entire system, including how it works and the detailed phases(steps) of system development.

Chapter04

Implementation, Results and Evaluation

4.1 Introduction

VR application development differs significantly from that of a desktop program or a Web page. Furthermore, developing virtual reality game applications to simulate the interaction between the real human or the player with the environment and the virtual human, in which the respect of physics laws and behaviors are vital, is a complex process that necessitates strong skills in a variety of tools and techniques in order to achieve the required quality levels.

4.2 Technical Description

The creation of the VR game application was preceded by another step which is the description of what our application should be and what the offers it gives to users. After that as a basis for the design process, we pulled all of the application objects, activities, and requirements from that. Different 3D models and animations should be generated to reach a high degree of quality and realism for an efficient experience. The virtual environment should be able to be viewed, navigated, and interacted with by the player. To maintain the impression of presence and increase the player or user's immersion, realistic animations and interactions that obey the laws of physics, such as gravity and collisions, are required.

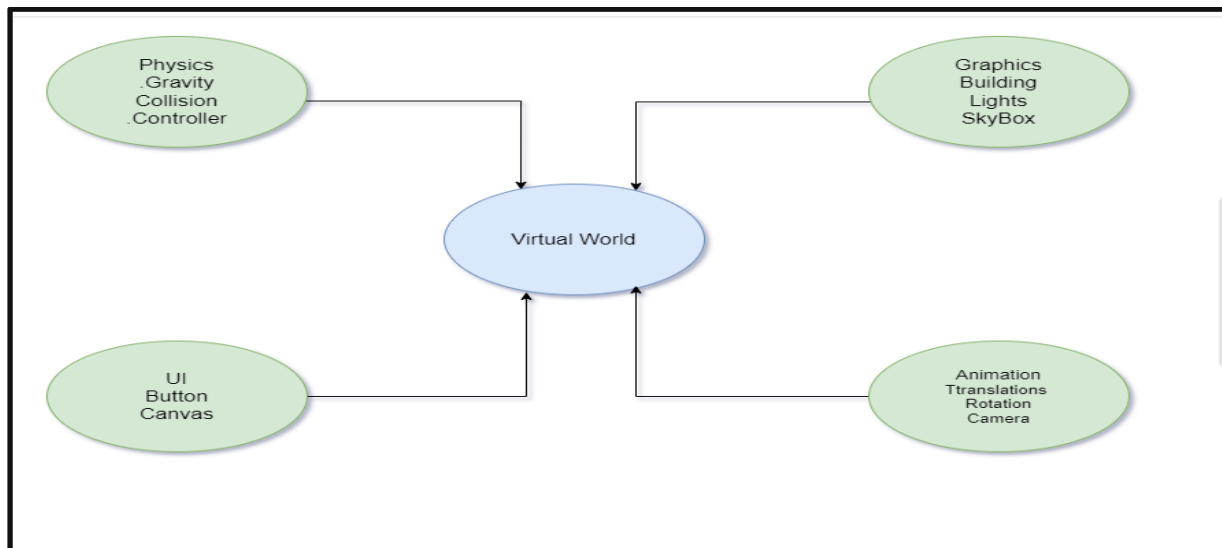


Figure 4.1: A diagram of Technical description of the application requirements

4.3 Hardware and Software for Development Environment

4.3.1 Hardware

The creation of the Virtual Reality game or application needs a necessary main device which is the headset with the two controllers. In our situation we did use the Oculus Quest 2.

Oculus Quest 2 (rebranded as Meta Quest 2 in November 2021) is a virtual reality (VR) headset developed by Facebook Reality Labs (formerly Oculus).



Figure 4.2: Image show the Oculus Quest 2 device with the controllers

As with its predecessor, the Quest 2 is capable of running as both a standalone headset with an internal, Android-based operating system, and with Oculus-compatible VR software running on a desktop computer when connected over USB or Wi-Fi. It is a refresh of the original Oculus Quest with a similar design, but with a lighter weight, updated internal specifications, a display with a higher refresh rate and per-eye resolution, and updated Oculus Touch controllers, it is lighter than the first-generation Quest, at 503 grams (17.7 ounces)[23]. The Quest 2 uses the Qualcomm Snapdragon XR2 SoC (which is part of a Snapdragon product line designed primarily for VR and

augmented reality devices)[24], with 6 GB of RAM — an increase of 2 GB over the first-generation model[23].

Its display is a singular fast-switch LCD panel with an 1832×1920 per-eye resolution, which can run at a refresh rate of up to 120 Hz (an increase from 1440×1600 per eye at 72 Hz)[25]. At launch, the display's refresh rate was locked at 72 Hz via software, with 90 Hz mode as an experimental option limited to the home area only. A software update in November 2020 enabled games to run in 90 Hz mode [23] [26] [27]. In April 2021, an update further added experimental support for 120 Hz mode in games [28] [29].

The headset includes physical interpupillary distance (IPD) adjustment at 58 mm, 63 mm and 68 mm, adjusted by physically moving the lenses into each position[25].

For the controllers : The included controllers with the Quest 2 are the third generation Oculus Touch controllers. The design of the new controllers was influenced by the original Oculus Rift controllers [26]. Their battery life has also been increased four-fold over the controllers included with the first-generation Quest [25] [30] [31].



Figure 4. 3:Image show the oculus quest 2 controllers

4.3.2 Software:

4.3.2.1 Visual Studio

Visual Studio is a comprehensive set of development tools for generating ASP.NET web applications, XML web services, office applications and mobile applications. Visual Basic, Visual C++, Visual C# all use the same integrated development environment (IDE), which gives them makes it possible to share tools and facilitates the creation of solutions using several languages. Moreover, these languages make it possible to take better advantage of features of the .NET framework, which provides access to key technologies simplifying the development of ASP web applications and XML web services thanks to Visual Web Developer.



Figure 4.4:Image show the visual studio 2019 Logo

4.3.2.2 Unity

Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Worldwide Developers Conference as a Mac OS X game engine. The engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game

development and is considered easy to use for beginner developers and is popular for indie game development[32].



Figure 4.5:Image show the unity logo

The engine can be used to create three-dimensional (3D) and two-dimensional (2D) games, as well as interactive simulations and other experiences[33][34]. The engine has been adopted by industries outside video gaming, such as film, automotive, architecture, engineering, construction. Oculus provides SDK for the Unity development environment. Using the SDK, you can use key VR features such as user input, controllers, and rendering to build immersive VR experiences for Oculus devices.

4.3.2.3 C# (Programming Language)

It is a general-purpose, multi-paradigm programming language. C# encompasses static typing, strong typing, lexically scoped, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines[35].

The C# programming language was designed by Anders Hejlsberg of Microsoft in 2000 and was later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO/IEC (ISO/IEC 23270) in 2003. Microsoft has introduced C# alongside .NET Framework and Visual Studio, which were both closed source. At the time, Microsoft had no products to open-source code. Four years later, in 2004, a free and open source project called Mono was born, providing a compiler and an environment cross-platform runtime for the C# programming language. Ten more years later, Microsoft launched Visual Studio Code (code editor), Roslyn (compiler) and

the .NET Unified Platform (software framework), all of which support C# and are free, open-source and cross-platform. Mono also joined

Microsoft but was not merged with .NET. As of 2021, the most recent version of the language is C# 10.0, which was released in 2021 in .NET 6.0[35].

4.3.2.4 Oculus Integration

The Oculus Integration is a package provided by unity to bring advanced rendering, social, platform, audio, and Avatars development support for Oculus VR devices and some Open VR supported devices. The Oculus Integration contains the following:

- Audio Manager - Contains scripts to manage all the audio and sound effects in your app.
- Avatar - Contains the scripts and prefabs to add Oculus Avatars to your app.
- Lip-sync - Lip Sync contains a set of plugins and scripts that can be used to sync avatar lip movements to speech sounds.

4.4 Development Process

In our VR game application situation, with the oculus quest 2 feature which is storing the data like image or videos even applications so all the development process was in the unity program and visual studio 2019 and the oculus quest 2 was just only for the storage of the game application and for the test.

4.4.1 Creation of the Virtual Environment

We used some external resources that we did find in unity asset store to build our scene which is a model of a city with buildings and roads and sidewalks, tunnels and other objects etc. The city model is the best model for navigation because it's large and there a lot of objects can be defined as an obstacle for the player(user) and the human virtual(avatar), and we did respect the laws of physics by adding a component called rigid body when we can find it in the unity component.

Rigid bodies enable your GameObjects to act under the control of physics. The Rigid body can receive forces and torque to make your objects move in a realistic way. Any Game Object must

contain a Rigid body to be influenced by gravity, act under added forces via scripting, or interact with other objects through the NVIDIA PhysX physics engine.

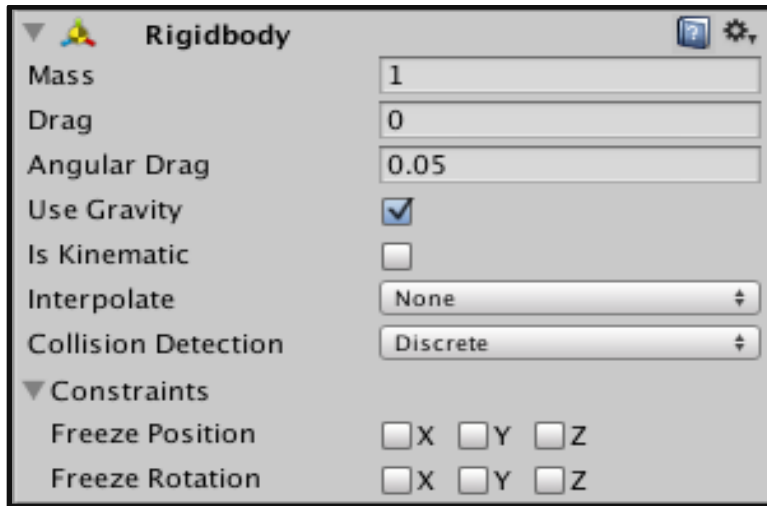


Figure 4.6:Image show the Rigidbody component

Property:	Function:
Mass	The mass of the object (in kilograms by default).
Drag	How much air resistance affects the object when moving from forces? 0 means no air resistance, and infinity makes the object stop moving immediately.
Angular Drag	How much air resistance affects the object when rotating from torque? 0 means no air resistance. Note that you cannot make the object stop rotating just by setting its Angular Drag to infinity.
Use Gravity	If enabled, the object is affected by gravity
Is Kinematic	If enabled, the object will not be driven by the physics engine, and can only be manipulated by its Transform. This is useful for moving platforms or if you want to animate a Rigidbody that has a HingeJoint attached.

Interpolate	None	No Interpolation is applied.
	Interpolate	Transform is smoothed based on the Transform of the previous frame.
	Extrapolate	Transform is smoothed based on the estimated Transform of the next frame.
Collision Detection	Used to prevent fast moving objects from passing through other objects without detecting collisions	
	Discrete	Use discrete collision detection against all other Colliders in the Scene Other colliders will use discrete collision detection when testing for collision against it. Used for normal collisions (This is the default value).
	Continuous	Use Discrete collision detection against dynamic Colliders (with a Rigidbody) and sweep-based continuous collision detection against static Colliders (without a Rigidbody). Rigidbodies set to Continuous Dynamic will use continuous collision detection when testing for collision against this rigidbody. Other rigidbodies will use Discrete Collision detection. Used for objects which the Continuous Dynamic detection needs to collide with. (This has a big impact on physics performance, leave it set to Discrete, if you don't have issues with collisions of fast objects)

	<table border="1"> <tr> <td>Continuous Speculative</td> <td>Use speculative continuous collision detection against Rigidbodies and Colliders. This is also the only CCD mode that you can set kinematic bodies. This method tends to be less expensive than sweep-based continuous collision detection.</td> </tr> <tr> <td>- Continuous Dynamic</td> <td>Use sweep-based continuous collision detection against GameObjects set to Continuous and Continuous Dynamic collision. It will also use continuous collision detection against static Colliders (without a Rigidbody). For all other colliders, it uses discrete collision detection. Used for fast moving objects.</td> </tr> </table>	Continuous Speculative	Use speculative continuous collision detection against Rigidbodies and Colliders. This is also the only CCD mode that you can set kinematic bodies. This method tends to be less expensive than sweep-based continuous collision detection.	- Continuous Dynamic	Use sweep-based continuous collision detection against GameObjects set to Continuous and Continuous Dynamic collision. It will also use continuous collision detection against static Colliders (without a Rigidbody). For all other colliders, it uses discrete collision detection. Used for fast moving objects.
Continuous Speculative	Use speculative continuous collision detection against Rigidbodies and Colliders. This is also the only CCD mode that you can set kinematic bodies. This method tends to be less expensive than sweep-based continuous collision detection.				
- Continuous Dynamic	Use sweep-based continuous collision detection against GameObjects set to Continuous and Continuous Dynamic collision. It will also use continuous collision detection against static Colliders (without a Rigidbody). For all other colliders, it uses discrete collision detection. Used for fast moving objects.				
Constraints	Restrictions on the Rigid body's motion.				
Freeze Position	Stops the Rigidbody moving in the world X, Y and Z axes selectively.				
Freeze Rotation	Stops the Rigidbody rotating around the local X, Y and Z axes selectively.				

Table 4.1: a table show the Properties of the rigid body component

Here some of image of the city:



Figure 4.7:Image show a part of the city model



Figure 4.8:Image show a part of the city model

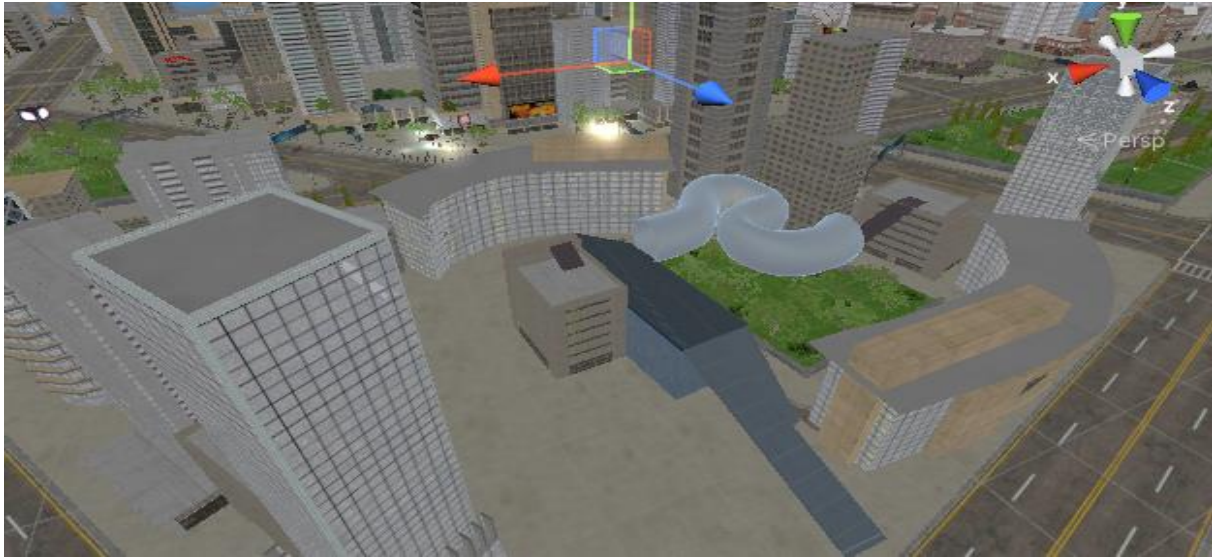


Figure 4.9:Image show a part of the city model

In our city model everything can be an obstacle for the user and the human virtual.

4.4.2 Creation of the Real human

The creation of the real human (the user) was too easy thanks to the Oculus integration asset that provided a lot of script and documentation to configure the headset to fit our demands in the virtual world.

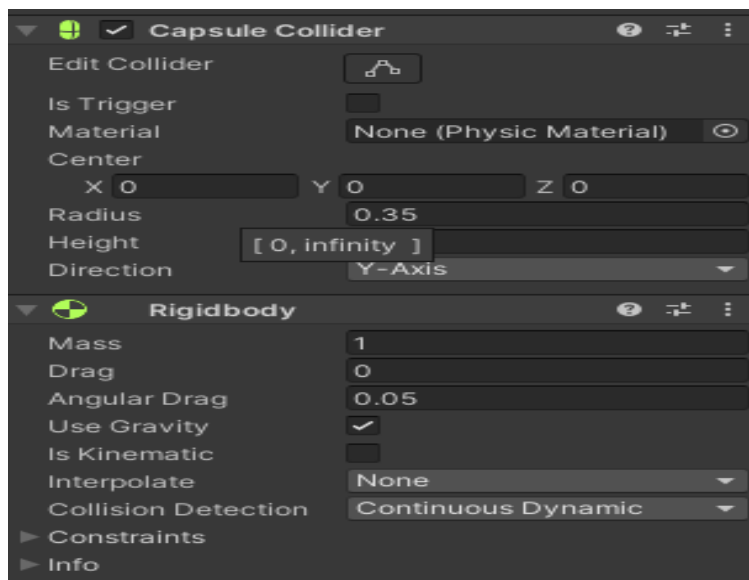


Figure 4.10:Image show the configuration of the user (the player)

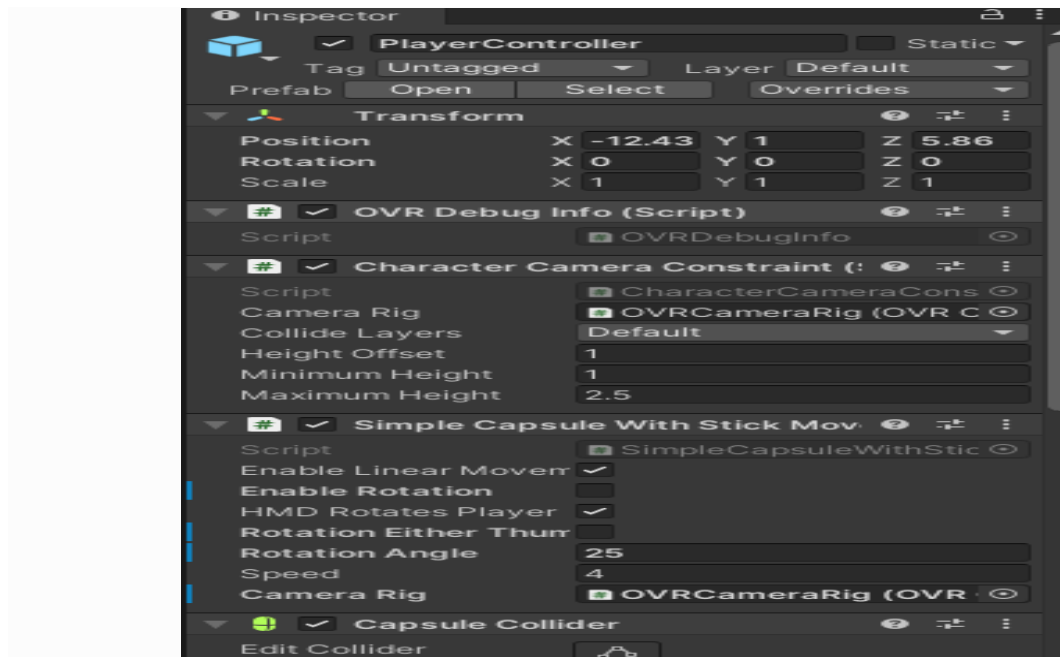


Figure 4.10: Image show the configuration of the user (the player)

OVR Debug Info script: Shows debug information on a heads-up display.

Character camera constraint script is responsible for moving the character capsule to match the HMD, fading out the camera or blocking movement when collisions occur, and adjusting the character capsule height to match the HMD's offset from the ground.

Simple Capsule with Sticks Movement (continuous movement): this script that allows the user to smoothly move their rig continuously over time.


```

}

1 reference
void StickMovement()
{
    Quaternion ort = CameraRig.centerEyeAnchor.rotation;
    Vector3 ortEuler = ort.eulerAngles;
    ortEuler.z = ortEuler.x = 0f;
    ort = Quaternion.Euler(ortEuler);

    Vector3 moveDir = Vector3.zero;
    Vector2 primaryAxis = OVRInput.Get(OVRInput.Axis2D.PrimaryThumbstick);
    moveDir += ort * (primaryAxis.x * Vector3.right);
    moveDir += ort * (primaryAxis.y * Vector3.forward);
    //_rigidbody.MovePosition(_rigidbody.transform.position + moveDir * Speed * Time.fixedDeltaTime);
    _rigidbody.MovePosition(_rigidbody.position + moveDir * Speed * Time.fixedDeltaTime);
}

```

Figure 4.11:Image show the script who controller the continuous movement

Capsule Collider: The Capsule Collider is made of two half-spheres joined together by a cylinder. It is the same shape as the Capsule primitive.

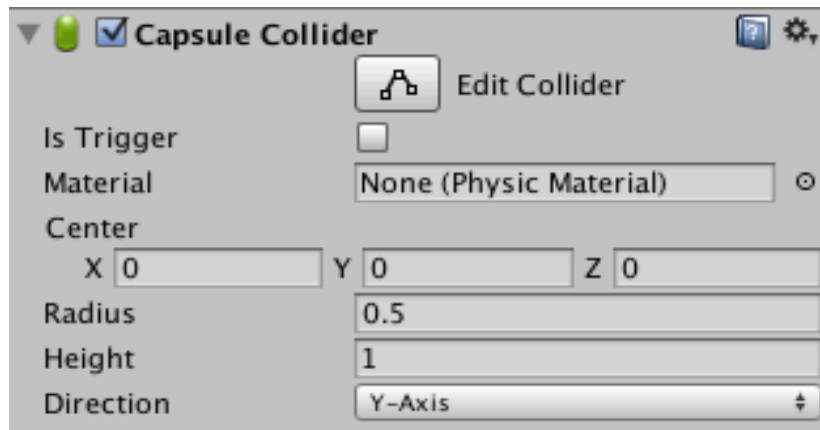


Figure 4.12:Image show the capsule collider propriety

Property	Function
Is Trigger	If enabled, this Collider is used for triggering events, and is ignored by the physics engine

Material	Reference to the Physics Material that determines how this Collider interacts with others.
Center	The position of the Collider in the object's local space.
Radius	The radius of the Collider's local width.
Height	The total height of the Collider.
Direction	The axis of the capsule's lengthwise orientation in the object's local space.

Table 4.2: a table show the propriety of the capsule collider

For the teleportation movement:

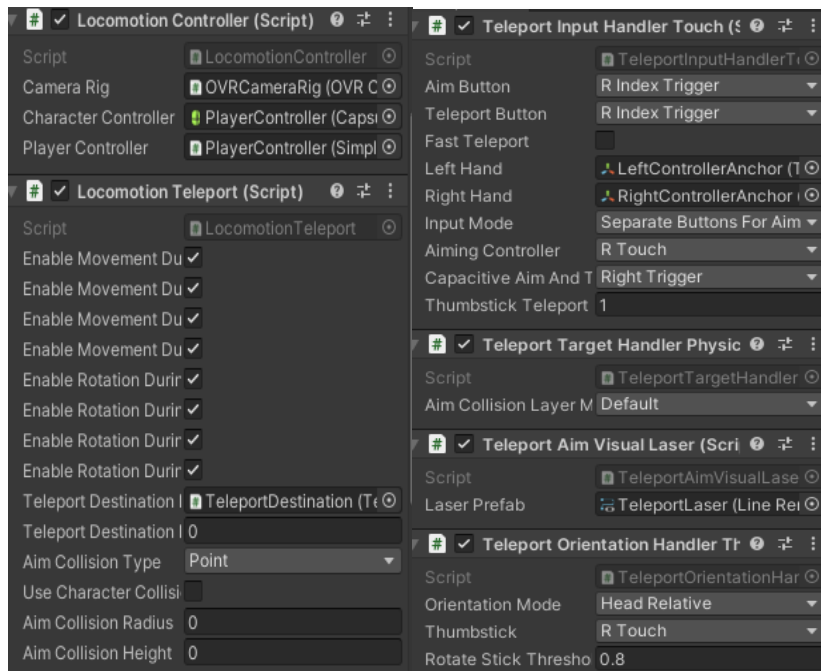


Figure 4. 13:Image show the configuration for the teleportation movement



Figure 4.13: Image show the configuration for the teleportation movement

4.4.3 Creation of the human virtual

We used the Maximo site web which provided us with a simple 3D character and the animation and the movement were done in unity. Our character is a model of a simple robot without any face details because our goal is to simulate movement not modeling 3D characters.

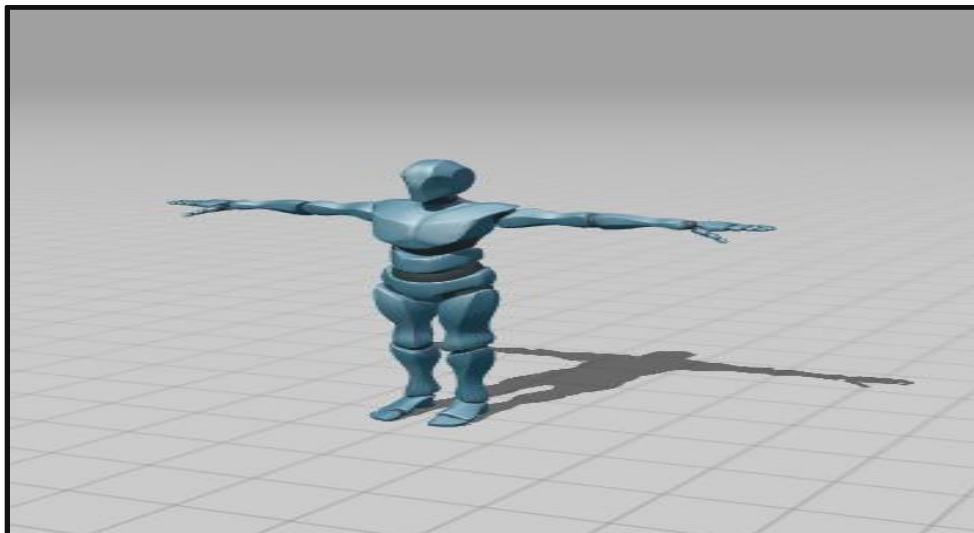


Figure 4.14: Image show the virtual human

We started by creating the Avatar animations: Walking, using Unity Mechanism's state machines and scripting to manage all the sequences of the animation clips.

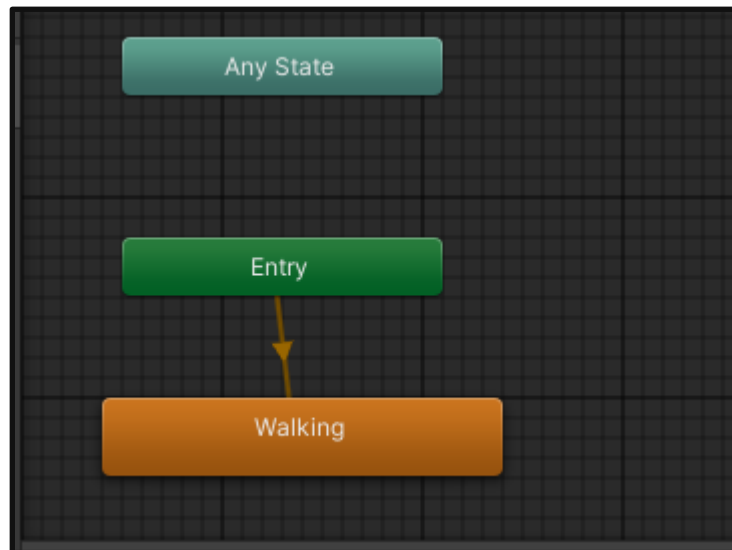


Figure 4.15: Image show the animation of the avatar

The Figure 4.16 show us the script who control the movement of the virtual human in our scene, we create a Vector3 (target), it the new coordinate of the VH to move there and by using the function Look At which Rotates the transform so the forward vector points at /target/'s current position.

```
public class Y_bot_Walking : MonoBehaviour
{
    Vector3 target;
    float speed = 1.0f;

    // Start is called before the first frame update
    void Start()
    {
        SetNewTarget(new Vector3(
            0.0f,
            0.0f,
            8.0f
        ));
    }

    // Update is called once per frame
    void Update()
    {
        Vector3 direction = target - transform.position;
        transform.Translate(direction.normalized * speed * Time.deltaTime, Space.World);
    }
    void SetNewTarget(Vector3 newTarget)
    {
        target = newTarget;
        transform.LookAt(target);
    }
}
```

Figure 4.16: Image show the script who control the movement of our avatar

For the Collision movement we did use the NavMeshAgent and we did some configuration to suit our avatar from speed to we of Collison.

The NavMeshAgent components help you to create characters which avoid each other while moving towards their goal. Agents reason about the game world using the **NavMesh** and they know how to avoid each other as well as other moving obstacles. Pathfinding and spatial reasoning are handled using the scripting API of the NavMesh Agent.



Figure 4.17: Image show the NavMeshAgent component

Property	Function
Radius	Radius of the agent, used to calculate collisions between obstacles and other agents.
Height	The height clearance the agent needs to pass below an obstacle overhead.
Base offset	Offset of the collision cylinder in relation to the transform pivot point.
Speed	Maximum movement speed (in world units per second).
Angular Speed	Maximum speed of rotation (degrees per second).
Acceleration	Maximum acceleration (in world units per second squared).
Stopping distance	The agent will stop when this is close to the goal location.
Auto Braking	When enabled, the agent will slow down when reaching the destination. You should disable this for behaviors such as patrolling, where the agent should move smoothly between multiple points.
Quality	Obstacle avoidance quality. If you have a high number of agents you can save CPU time by reducing the obstacle avoidance quality. Setting avoidance to none, will only resolve collisions, but will not try to actively avoid other agents and

	obstacles.
Priority	Agents of lower priority will be ignored by this agent when performing avoidance. The value should be in the range 0–99 where lower numbers indicate higher priority.
Auto Traverse OffMesh Link	Set to true to automatically traverse off-mesh links. You should turn this off when you want to use animation or some specific way to traverse off-mesh links.
Auto Repath	When enabled the agent will try to find a path again when it reaches the end of a partial path. When there is no path to the destination, a partial path is generated to the closest reachable location to the destination.
Area Mask	Area mask describes which area types the agent will consider when finding a path. When you prepare meshes for NavMesh baking, you can set each meshes area type. For example you can mark stairs with a special area type, and forbid some character types from using the stairs.

Table 4.3 : A table show the properties of NavMeshAgent

4.5 Result

In this section we will present you the results of navigation and the interaction in the virtual world. The images were taken from screenshot from a recording video because the taken images from the oculus quest 2 was difficult and impossible to take an image without interruption from the main menu of oculus quest 2. So the quality of the image may not be good (negative point for the oculus quest 2).

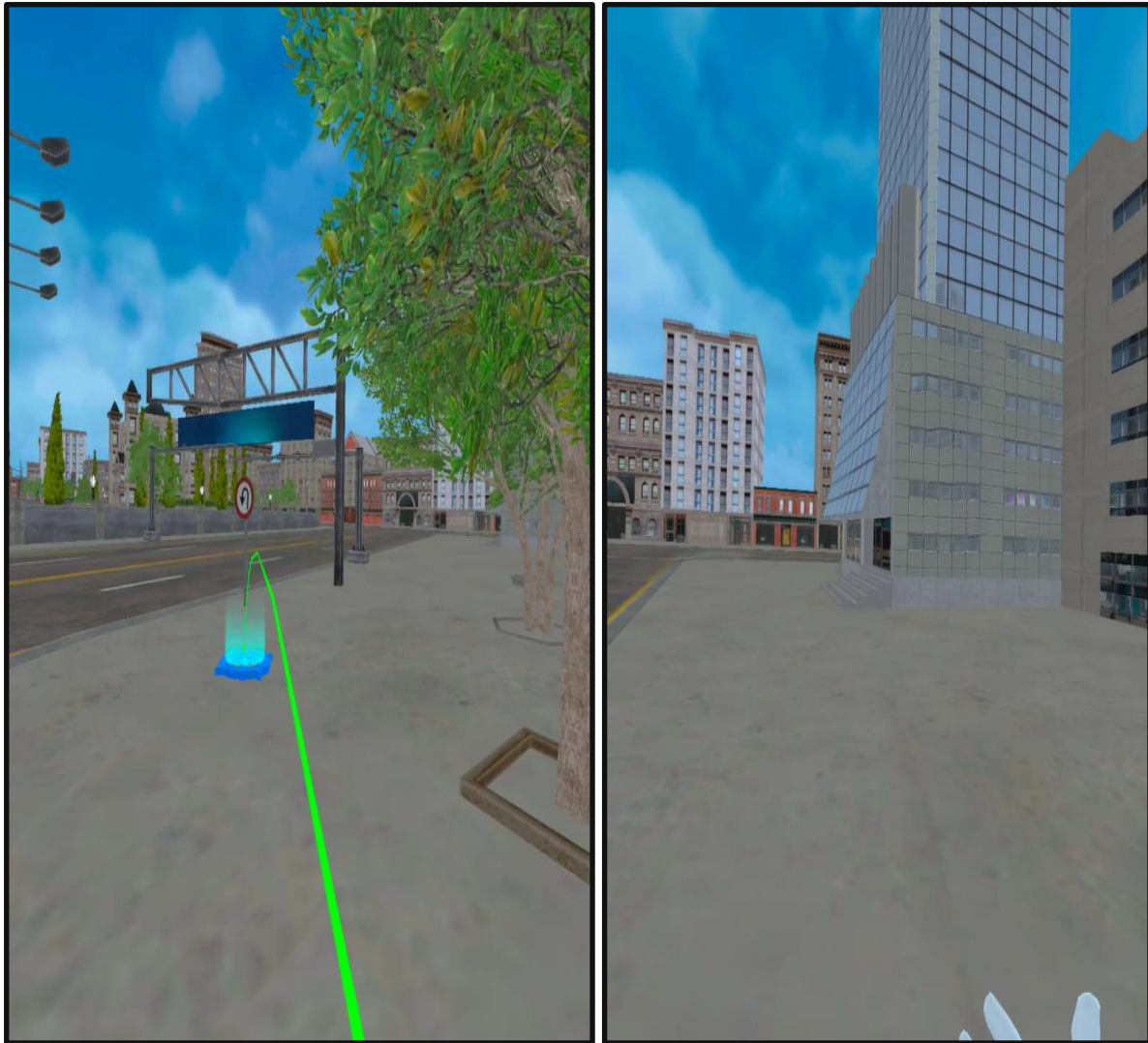


Figure 4.18: Image illustrate the difference between the teleportation movement and continuous movement in virtual world

In the movement and navigation: the teleportation technique doesn't feel realistic compared to continuous movement. It feels like we jump with closing eyes from side to side but there is one opposite side which respects the laws of physics.

The continuous movement feels like we are moving but in an irregular way, we don't feel realistic. It looks like a two years old baby walking with the help of a baby walker. But compared to the teleportation movement the continuous movement is more realistic than the teleportation movement.



Figure 4.19 : Image illustrate the difference between the continuous movement and the teleportation movement in the interact with objects (obstacle)

Avoiding obstacles are quite similar in both continuous and teleportation movement; the only difference is in the technique of avoiding.

For continuous movement if the obstacle has a large scale like a building or big tree the user needs to find another way.

For teleportation movement we don't consider the space of the obstacle because of teleportation techniques that allows the user to avoid the building by just aim the laser in roof of building and after that when he/she in the roof he aim the laser again in the other side.

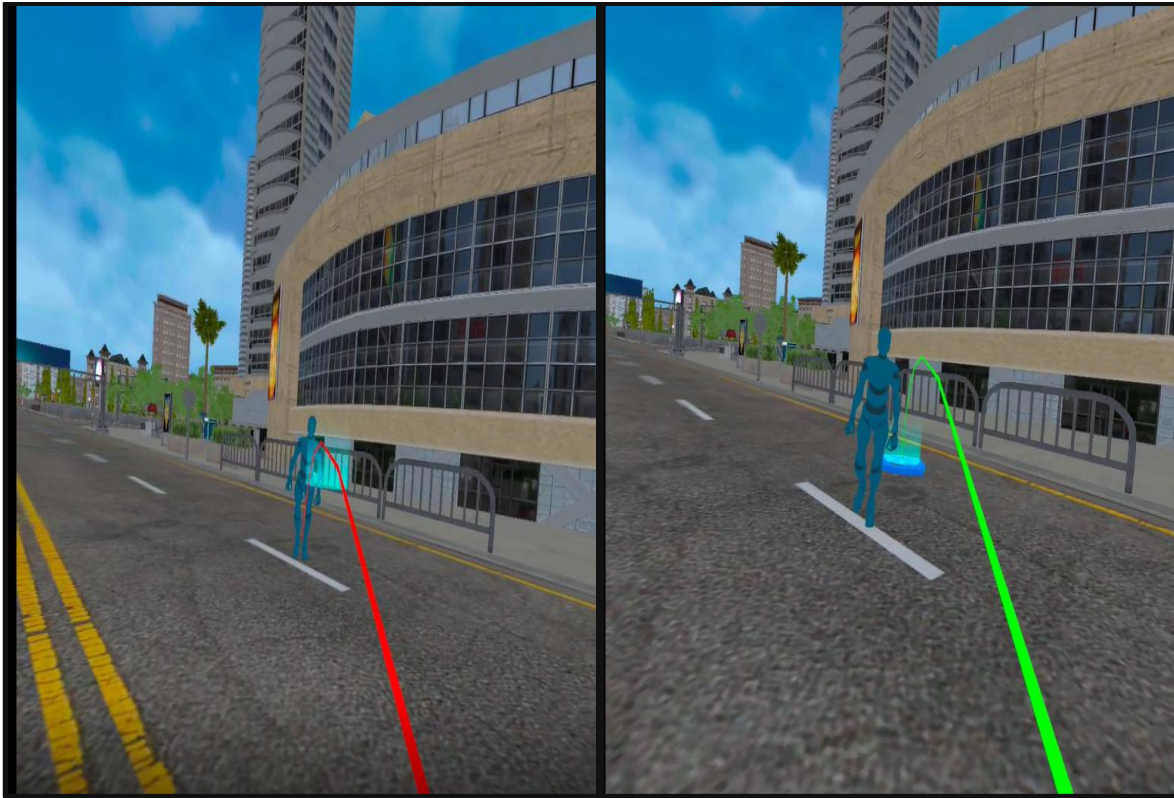


Figure 4.20 : Image illustrate the collision avoidance between a real human with teleportation movement and the virtual human



Figure 4.21 : Image illustrate the collision avoidance between a real human with continuous movement and the virtual human

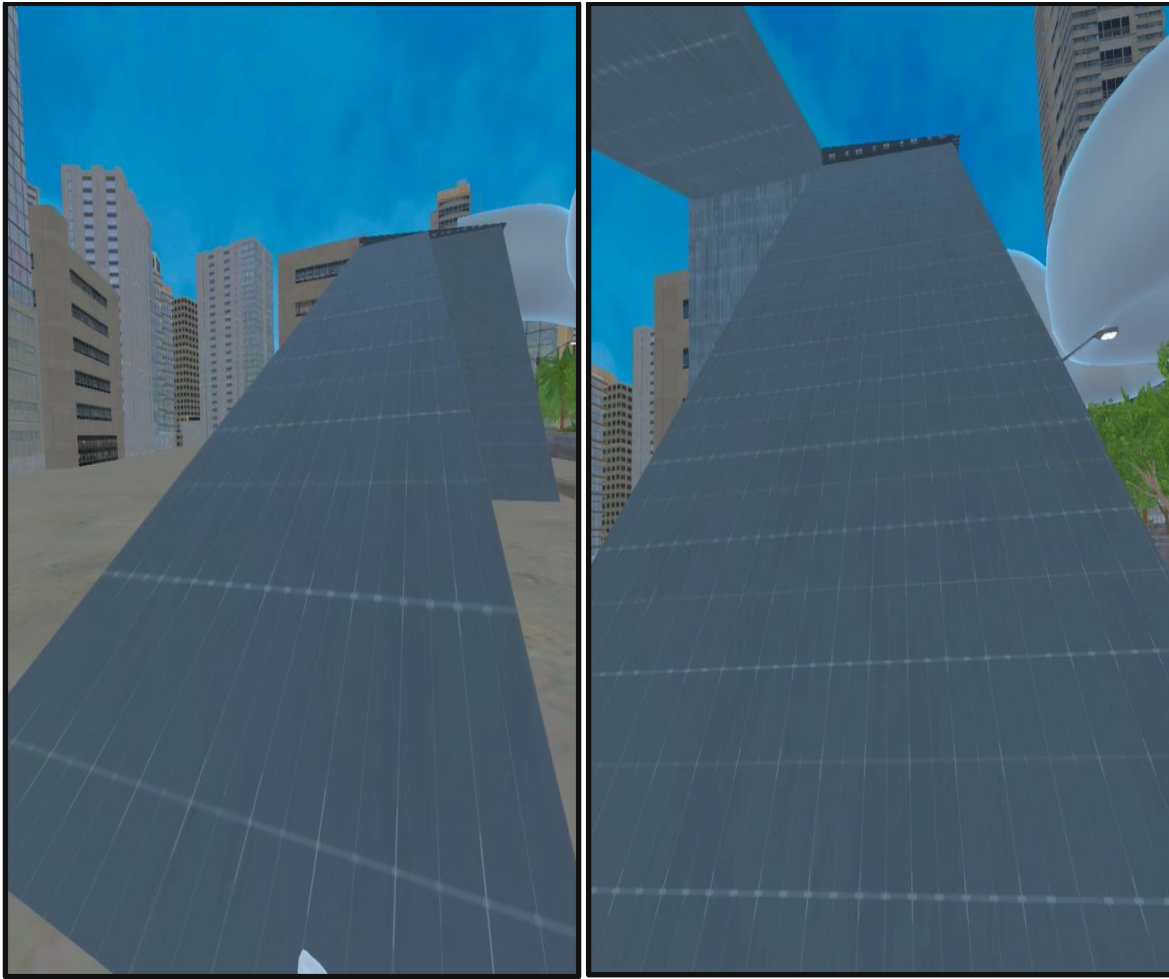


Figure 4.22: Image show the effect of the pitch of ladder and the gravity in the climbing of ladder
(Continuous Movement navigation with physics law)

The (figure 4.23) illustrate the pitch of the ladder which is The included (acute) angle between the horizontal and the ladder, which is measured on the side of the latter opposite the climbing side and it effects in the climbing process, left figure can climb the ladder but the right figure can't because the pitch and the gravity pull down the player to the ground.

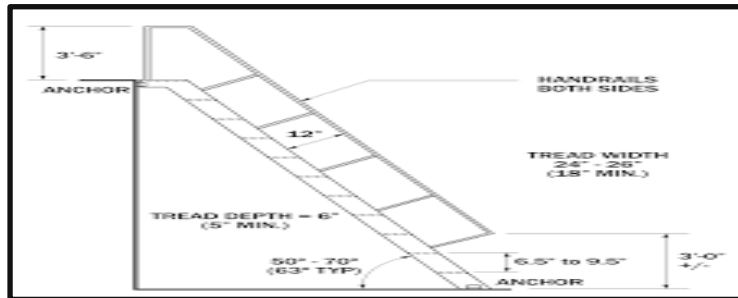


Figure 4.23: Image illustrate the pitch of the ladder

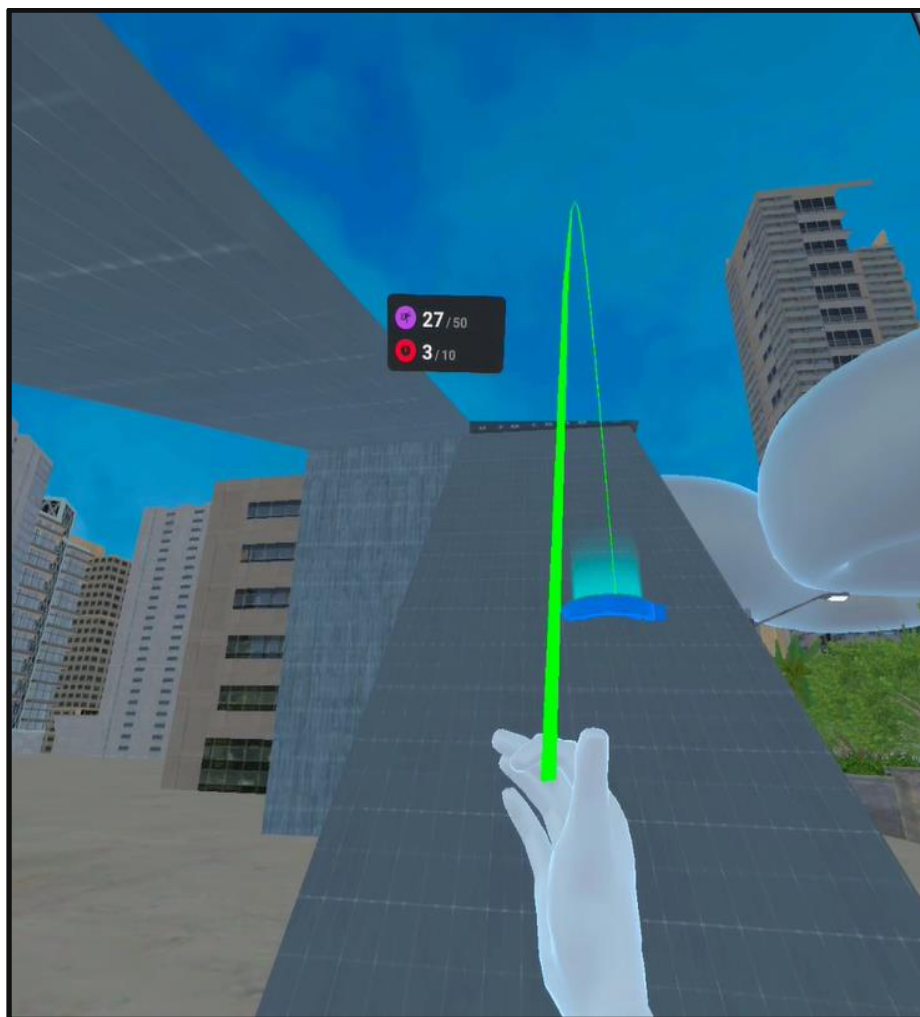


Figure 4.24 : Image shows the effect of the pitch of the ladder and the gravity in the climbing of the ladder(teleportation navigation with no physics law).

In (Figure 4.24) illustrates how the teleportation technique doesn't care about the physics law and we can climb the ladder by just modifying the angle of the laser to fit the length of the ladder.

Interaction and collision Avoidance: the continuous movement is better than the teleportation movement because it's more realistic.

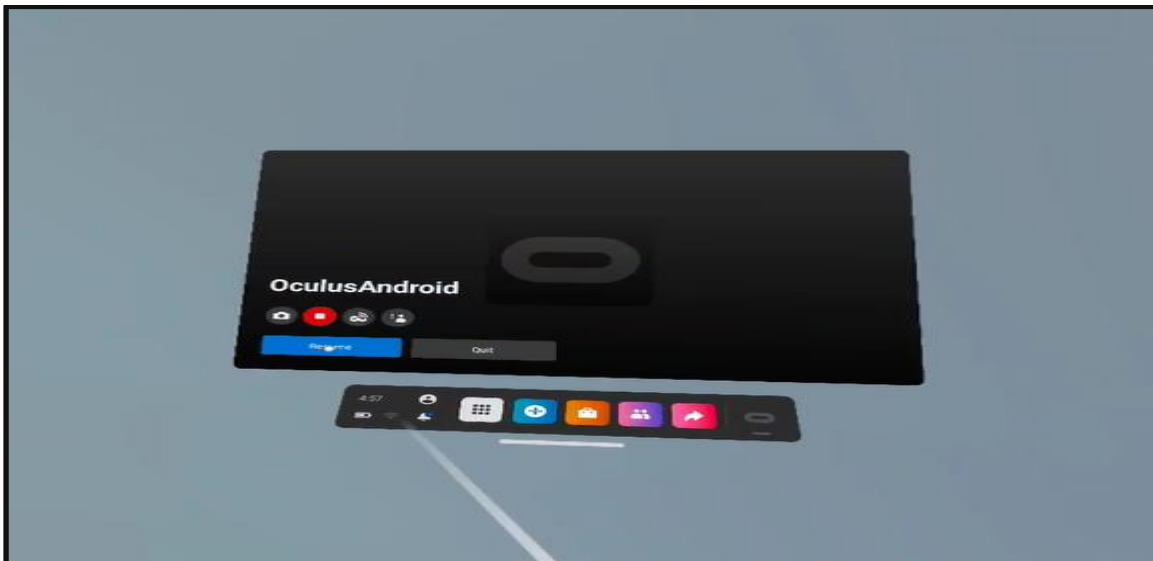


Figure 4.25: Image illustrate the Exit method of the game application

After all, the image doesn't give the experience that we are in a virtual world like the headset does. You should use the headset to feel the magnificence of the virtual reality experience.

4.6 Conclusion

This chapter's major purpose was to demonstrate the implementation process and outcomes of our proposed VR gaming application. We began by describing our system's technical specifications. After then, the development environment for dealing with hardware and software integration was introduced. The development method and data flow were further detailed in the next section. Finally, we demonstrated the prototype design and content, which included complete virtual worlds, graphical user interfaces, and interactions throughout navigation and interaction scenarios.

Conclusion and Perspectives

5.1 Summary

Virtual reality technology has grown in popularity over the last decade and it has emerged in a lot of fields and in many topics and one of these topics is interaction and collision avoidance. In this project, we explored the use of VR to study locomotion or navigation during collision avoidance between a real human (user/player) and a virtual human (avatar), where we try to make a realistic simulation between this two especially the virtual human.

In the light of the optimistic potential of virtual reality for creating a simulation with realistic interaction, a VR serious game application that takes advantage of the new technologies was devised in this project as a tool that can be used by users for the navigation and the interaction in a big virtual world(city).

We developed our game application in the unity Game Engine who provide us with different libraries and plugins that make our work easy and fast along with the new headset (Oculus quest 2) (latest technology), we start by creating our world which is a big city that allow the user to navigate without any constraints and then we create our virtual human and made him too interact with our virtual world in realistic way like a reel human.

Regardless of all the efforts made by us we didn't satisfy and reach our goals and purpose to make a realistic collision avoidance like the reel life and this due to several reasons:

- Lack of experiences in this field (virtual reality).
- lack of documentation and resources especially on the side of oculus quest 2 (new generation of headset).
- lack of time.

Future Work

We hope to continue developing our game application and make improvements in the realistic side and in the graphics side and make our game application a therapy application for a symptom called Topographical disorientation which refers to an acquired inability to navigate the environment in daily life. Patients with topographical disorientation are unable to find their way around their surroundings. This syndrome consists of several heterogeneous conditions. First, patients may have difficulty identifying familiar environmental landmarks such as buildings and street corners (topographical or landmark agnosia). This deficit is associated with the right posterior Para hippocampal gyrus and the anterior part of the lingual and fusiform gyri, which are critical for acquiring information and identifying buildings and landscapes. Second, landmarks are “recognized” but do not evoke a sense of direction (i.e., directional or heading disorientation). The patient fails to utilize previously learned routes when traveling from one landmark to another. This may be caused by posterior cingulate lesions. Third, patients have cognitive deficits that limit the development of a mental map of their environment. The lesions typically involve the hippocampus and retrosplenial cortex.

Bibliography

- [1] Rheingold, H. Virtual Reality. New York, New York: Touchstone. (1991)
- [2] Beier, K. P. Virtual Reality: A Short Introduction. Retrieved February 20, 2022, from [http://www.umich.edu/~vrl/intro/\(2008\)](http://www.umich.edu/~vrl/intro/(2008)).
- [3]. Dom, B. History of VR - Timeline of Events and Tech Development. Retrieved September 20, 2022, from <https://virtualspeech.com/blog/history-ofvr#:~:text=1960,wide%20vision%20and%20stereo%20sound>. (2019, AUGUST).
- [4]M Raja'a, F Farid. journal of Dentistry (Tehran, Iran)13 (3),215, (2016)
- [5] Juan, M. C., Baños, R., Botella, C., D, P., Alcañiz, M., & Monserrat, C. An Augmented Reality system for the treatment of acrophobia. *Presence*, 15 (4), 315–318, (2005).
- [6] Bryson, S. Knowledge-Based Augmented Reality. *Communications of the ACM*, 26 (7), (1993).
- [7] Gigante, M. Virtual reality: Definitions, history and applications. In *Virtual Reality Systems* (pp. 3-14). Academic-Press, (1993).
- [8]. Darken, R.P. and Sibert, J.L.: A Toolset for Navigation in Virtual Environments, *Proceedings of the ACM User Interface Software and Technology*, 157-165, (1993).
- [9]. Elvins, T.T: Wayfinding 2: The Lost World, *Computer Graphics*, Vol. 31, No. 4, 9-12, (1997).
- [10]. Maier, P. H: Spatial imagination. Components, gender-specific differences, relevance, development and realization in the Realschule, Frankfurt a. M., Peter Lang, (1994).
- [11]. Shi, P., and S. Zlatanova. Evacuation route calculation of inner buildings. *Geoinformation for disaster management*. Springer-Verlag, Heidelberg, pp. 1143-1161 (2005).
- [12]. Kurtis Danyluk, Wesley Willett University of Calgary, Calgary, AB, Canada, Evaluating the Performance of Virtual Reality Navigation Techniques for Large Environments, pp 203–215 (2019).

- [13]. K.H. Sharkawi, M.U. Ujang and A. Abdul-Rahman, Faculty of Geoinformation Science and Engineering, 3D NAVIGATION SYSTEM FOR VIRTUAL REALITY BASED ON 3D GAME ENGINE (2006).
- [14]. Tim R H Cutmore, TREVOR J. HINE, International Journal of Human-Computer Studies, Cognitive and gender factors influencing navigation in a virtual environment 53(2):223-249, (August 2000).
- [15]. Ware, C., Osborne, S.: Exploration and Virtual Camera Control in Virtual Three Dimensional Environments. SIGGRAPH Computer Graphics pp. 175–183 (1990).
- [16]. Kaser, D.P., Parker, E., Glazier, A., Podwal, M., Seegmiller, M., Wang, C.P., Karlsson, P., Ashkenazi, N., Kim, J., Le, A., Buhlmann, M., Moshier, J.: The Making of Google Earth VR. In: ACM SIGGRAPH 2017 Talks. pp. 63:1-63:2, (2017).
- [17]. Krieg-Bruckner, B., Rofer, T., Carmesin, H.-O., and Muller, R.: A Taxonomy of Spatial Knowledge for Navigation and its Application to the Bremer Autonomous Wheelchair, In: Freksa, C., Habel, C., and Wender, K.F. (Eds.): Spatial Cognition, Lecture Notes in Artificial Intelligence 1404, Springer Verlag, 373-397, (1998).
- [18]. Bowman, D.A., Koller, D. and Hodges, L.F.: Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques, Proceedings of the Virtual Reality Annual International Symposium (VRAIS), 45-52, (1997).
- [19]. Darken, R.P. and Sibert, J.L.: Navigating Large Virtual Spaces, International Journal of Human-Computer Interaction, 8(1), (1996).
- [20]. B. Katalin Szabó, Center for Energy Research Budapest Hungary, Interaction in an immersive virtual reality application, (October 2019).
- [21]. Kaur, K.: Designing Virtual Environment for Usability, Doctor Thesis, Centre for Human-Computer Interface Design, City University, London, (1998).
- [22]. Chapanis, A. To Communicate the Human Factors Message, You Have to Know What the Message Is and How to Communicate It. Human Factors Society Bulletin, 34, 1-4, (1991).
- [23]. Kuchera, Ben , "Oculus Quest 2 review: smaller, cheaper, better". *Polygon...* (2020).
- [24]. Smith, Ryan, "Qualcomm Announces XR1 Platform: Dedicated SoC for VR/XR Headsets, Coming Late 2018". *AnandTech*. Future plc, (2019).
- [25]. Lang, Ben, "Oculus Quest 2 Review – The Best Standalone Headset Gets Better in (Almost) Every Way", (2020).
- [26]. David, Heaney , "Oculus Quest 2 Specs: 'Nearly 4K' LCD, 90Hz, XR2 Chip, \$299". *UploadVR*, (2020).
- [27]. Wickens, Katie, "Oculus Quest 2 gets a cheeky 90Hz refresh rate update, finally rivaling competitors". *PC Gamer*, (2020).

[28]. INTRODUCING OCULUS AIR LINK, A WIRELESS WAY TO PLAY PC VR GAMES ON OCULUS QUEST 2, PLUS INFINITE OFFICE UPDATES, SUPPORT FOR 120 HZ ON QUEST 2, AND MORE, *Oculus Blog*. Oculus, (2022)

[29]. Robertson, Adi, "Oculus has turned on wireless PC streaming for the Quest 2"; *The Verge*, (2021).

[30]. Lang, Ben, *Road to VR*, (2020).

[31]. Machkovech, Sam. "Oculus Quest 2 as your next VR system". *Ares Technical*, (2020).

[32]. D'alessandri, Marie. "What is the best game engine: is Unity right for you?". GamesIndustry.biz. Gamer Network, (2020).

[33]. Axon, Samuel. "Unity at 10: For better—or worse—game development has never been easier". *Ares Technical*. (2016).

[34]. Takahashi, Dean. "John Riccitiello Q&A: How Unity CEO views Epic's Fortnite success". *VentureBeat*, (September 15, 2018).