VR embodiment of a virtual animal character with a virtual reality headset and two controllers

Théo Ducatez DUT, M.Eng

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Augmented and Virtual Reality)

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Supervisor: Rachel McDonnell

August 2022

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Théo Ducatez

August 19, 2022

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Théo Ducatez

August 19, 2022

VR embodiment of a virtual animal character with a virtual reality headset and two controllers

Théo Ducatez, Master of Science in Computer Science
University of Dublin, Trinity College, 2022

Supervisor: Rachel McDonnell

Virtual reality offers anyone the unique possibility of getting completely immersed in a virtual environment. The embodiment of a virtual avatar is part of the majority of these experiences and is an important factor in creating this feeling of virtual presence. When the bond between the players and their representation becomes strong enough, it leads to the illusion of body ownership (IVBO), a phenomenon where people start perceiving their virtual representation as their own body. However, most consumer-grade VR setups are not efficient enough to deliver sufficient sensory information to enable this phenomenon to appear. Furthermore, previous research on IVBO predominantly focuses on humanoid avatars instead of all the other possibilities these endless virtual worlds could enable. In this paper, we demonstrate the possibility of creating such an IVBO experience focused on the embodiment of an animal character with a typical consumer-grade VR setup composed of a headset and two controllers. It aims to prove these setups' compatibility with IVBO experiences and offers a list of important concepts for designing and implementing a successful animal embodiment-focused VR experience. Our empirical results show that creating a VR application involving the embodiment of a virtual animal character is totally possible with this consumer-grade setup. It also proved the importance of the environment and sound design of the virtual world in the appearance of IVBO. The results obtained from our user study align with the few previous papers on the subject using more complete setups, proving the potential of such simple setups in this domain.

Acknowledgments

I would like to thank Rachel McDonnell both as a teacher and supervisor for introducing me to computer animation and guiding me during the whole dissertation process, providing me with ideas and feedback from the start of this project.

I would also like to thank all the other lecturers and college staff that were involved in the Computer Science and Virtual Reality course this year at Trinity College Dublin. Especially Michael Manzke and Gaye Stephens for introducing me to research, as well as Carol O'Sullivan and Dónal Egan for guiding me during this project.

Finally, I would like to thank my friends and family for supporting me throughout the year giving me moral support and motivation.

Théo Ducatez

University of Dublin, Trinity College
August 2022

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Virtual Reality (VR) has been the subject of a growing increase in popularity over the years. It is now being used and explored in more industries than ever before, and it has become one of the most famous technologies in recent years. Its potential for creating realistic and immersive virtual experiences has been recognised since the 1960s (Sutherland (1965)). It is being heavily developed nowadays as many companies recently popularised the concept of the metaverse and their take on its future (Mystakidis (2022)). This recent increase in popularity also allowed for the creation of very accessible VR equipment and helped generalise its use to a wider audience.

One of the most important parts of an immersive experience is the illusion of belonging in the virtual world we are presented with, a phenomenon also known as *virtual presence* (Slater and Steed (2000)). This illusion is created by multiple aspects of the virtual experience, but one of its most important is the link between the user and its virtual representation, the character, more commonly called avatar. Due to its immersive properties, VR excels in amplifying the bond between us and our virtual self. So much that, under the right conditions, it can manage to make us feel like our virtual avatar is our own body. This phenomenon is known as *the illusion of virtual body ownership* (IVBO). Research showed that VR was an excellent candidate to induce IVBO in virtual experiences by giving the best results in creating this strong bond between the user and its virtual representation (Slater et al. (2010, 2009); Waltemate et al. (2018a)).

This dissertation takes into account these two important aspects of virtual reality and provides an in-depth exploration of the process of trying to create IVBO with a basic VR setup. Now that VR technology is available to more and more people, the most common setups used are simply composed of a 3-trackers system, a headset and two controllers. Inducing IVBO with such equipment is hard and is not usually done with so few trackers. It therefore seemed interesting to try creating an immersive experience focused on the embodiment of a virtual character with a similar setup. As for the character itself, most current and past research has been done on inducing IVBO on humanoid virtual avatars. In this dissertation, we will explore the process of creating an immersive experience based on an animal character and try to induce IVBO with this non-humanoid virtual representation. This represents a real challenge, as animals are morphologically different from humans and feeling like embodying such characters is difficult to achieve. Using a basic VR setup adds more difficulty to this immersion as well and represents an interesting challenge to overcome.

Therefore, this dissertation will try to address the following question: *How can virtual reality be used to induce IVBO on an animal character with a restricted VR setup composed of only three trackers, a headset, and two controllers?*

1.1 Motivations

With the development of virtual reality, virtual presence and virtual embodiment have been subject to a lot of research lately. Many of them proved that VR had an excellent potential in various domains like psychotherapy (Riva (2005); Parsons and Mitchell (2002)). Focusing on creating a strong bond between the user and its virtual representation is crucial to getting the best results in VR applications and experiments, especially in this field. The virtual avatar is one of the most important aspects of it but is also one of the hardest to get right. This is why research on its complexity is very important. VR also has a promising potential when coupled with the cinema industry and acting. With the recent development of virtual production, VR is already being used a lot in this domain for previsualization and environment creation. But with its immersive characteristics, research is being conducted to use this technology for immersive rehearsals or to be used on sets directly (Berthelot et al. (2016); Kammerlander et al. (2021)). Focusing on inducing IVBO and virtual presence is a key aspect of the use of VR in this industry. It is also from this idea that this project initially started, the use of VR on movie sets and how it could help actors embody non-humanoid characters. This is also why this project is focused on the embodiment of an animal character. It would have a great potential for immersing the motion capture actors in the world they would have to play in and in the skin of the virtual characters they would play as.

The choice of embodying an animal character is also an important aspect of this project. Research on IVBO and embodiment has mainly focused on humanoid characters in the past. But with the development of VR technology, using animals and non-humanoid characters as virtual experiences and video game avatars has become more and more popular. For its potential in video games, (Krekhov et al. (2019a,b)), in increasing our involvement with environmental issues (Ahn et al. (2016)) or in some psychological applications Oyanagi and Ohmura (2019). The main motivation of this project would be to experiment on how to create an animal embodiment centred experience with a basic VR setup to facilitate the creation of such experiences. Moreover, creating this illusion of embodying an animal is hard to achieve and logically unnatural for the human brain. It represents a very interesting challenge to overcome and is even harder to achieve with a very basic VR setup.

The complexity of the setup used when using VR is also a determining aspect of the quality of the embodiment-focused immersive experience. Most of today's popular VR equipment is made up of three main parts, the headset, and two controllers. Each of them enables the tracking of the user's movements in three different positions in space, usually the two hands and the head. More advanced setups are also available with more trackers, for the two legs and the waist, for example, but they are not as affordable and accessible as the first one. Therefore, to make such immersive embodiment experiences, the more complex the setup is, the better. It makes it very hard for anyone to be able to experience them as this equipment is not as affordable and available for the general public. Most of the research made on VR animal embodiment uses four or more trackers setups (Krekhov et al. (2019a,b)). This is therefore why this project is focused on using only three trackers. The second main motivation of this project would be to find a solution for opening VR embodiment to a wider audience and to different kinds of setups that are more accessible to the public.

1.2 Objectives

The main objective of this project is to create a VR immersive experience where the user embodies a virtual animal avatar. The main goal would be to try to induce IVBO with this very simple setup of three trackers only. As well as finding ways of immersing the user in the virtual world. Creating virtual presence in the environment is also an objective of the project, trying to use the environment to immerse the user even more. This dissertation will outline the process needed to create such immersive experiences, along with what does and does not work well in this process. It will take into account the reduced number of trackers and offer an evaluation of the viability of creating a successful embodiment experience with such a basic VR setup. The aimed contribution is to offer an outline of experimenting with creating this immersive experience focused on animal character embodiment. With an overview of what works and what does not and ideas on how to enhance them, taking into account the reduced number of trackers. Popularising such experiences with basic setups and their use in more situations and industries.

1.3 Outline

This dissertation has been written with the same layout as the workflow used to create the implementation. It is separated into 7 chapters dealing with either an important part of the workflow used for the implementation or the introduction, conclusions, or evaluation of the results. Chapters n°1 and n°2 detail the introduction and background of the research carried out before the implementation of the project. Chapter n°3 consists of an explanation of some relevant theories and concepts used in the implementation and the research as a whole. Chapter n°4 contains an outline of the design process of the implementation. It describes all the features and important aspects necessary for the implementation of the immersive experience, with a focus on the embodiment of the animal avatar. Chapter n°5 focuses on the implementation of the designed experience and how it came to life, with which tools and technologies. Chapter n°6 presents the results of the project along with its evaluation. The objective is to describe what has been done and judge the quality of the experience created along with its limitations. Finally, chapter n°6 is the final part of this dissertation. It concludes the document and offers an overview of the potential future work related to this project and the subject of VR animal embodiment.

Chapter 2

Background

In this chapter, multiple concepts used in the design of the implementation are presented in detail along with the research papers they originate from. The project is based on the papers presented in this chapter, each one of them influenced its design and implementation.

2.1 Illusion of body ownership and virtual presence

Virtual reality is well known for its immersive capabilities and the way it can alter human perception to create a feeling of belonging in a virtual world. This phenomenon has initially been described by Slater et al. (1995) with the term *virtual presence* during their research on walking techniques in virtual reality. It was then refined by Lombard and Ditton (1997) later on, where they outlined the basis of research in the domain, but with newer immersive technologies like VR as well as proposing a system to measure it.

Virtual presence is a very important concept for this project as it directly aims at inducing this immersion feeling. In order to facilitate its manifestation, it is crucial to rely on the embodiment of our virtual avatar. Especially in this project, as the embodied character is non-human. When this embodiment feeling is pushed to a certain degree, we refer to it as the *illusion of virtual body ownership* (IVBO) (Lugrin et al. (2015)). The feeling of owning the body of our virtual avatar and living the virtual experience through its representation. This concept was inspired by a research on *body ownership* conducted by Botvinick and Cohen (1998). Their experiment, called *the rubber hand illusion*, proved that it was possible to feel like owning an artificial hand. By hiding one of their participants' hands and replacing it with a fake rubber hand, touching both the fake and the real hand at the same time induced the feeling of owning this fake limb in all their participants. Further research has been conducted on this same experiment later on by Tsakiris and Haggard (2005), on different body parts by Lenggenhager et al. (2007), and on the whole body by Petkova and Ehrsson (2008).

At this point, body ownership experiments were mainly conducted in real-life conditions, but Slater et al. (2008) and Banakou et al. (2013) were the first ones to conduct these kinds of experiments with a virtual perspective. While this was great progress in the field, it was not closely related to VR as it did not take into account any tracking of the body or the hands. Sanchez-Vives et al. (2010) were the first to emphasise the significance of hands and fingers movement tracking in the sensation of ownership of virtual hands. This added tracking of body parts was proven to be so important that it was even said to be prioritised compared to visual hints only. This research led some more scientists to take an interest in body tracking for IVBO like Slater et al. (2010), Perez-Marcos et al. (2011) or Maselli and Slater (2013). The second one was centred on the importance of body continuity, posture alignment, and realism between the real-life movements and the movements of the virtual avatar. Maselli and Slater (2013) focused on proving that a first-person point of view with the virtual avatar was essential. At its beginning, research on IVBO has been mainly focused on the ownership of humanoid characters. For instance, Lin and Jörg (2016) worked on the importance of a realistic appearance in IVBO of a virtual hand. Jo et al. (2017) proved that body visual similarity was important in the embodiment of a virtual character. And finally, Waltemate et al. (2018b) proved that customisation of the virtual representation led to a higher IVBO. Although work on non-humanoid character embodiment changed people's minds a few years later, realistic virtual representations were seen as the simplest and most efficient way to induce IVBO.

The feeling of IVBO in an immersive experience has other benefits than just adding more realism to it. Numerous studies have been conducted on the direct effects it had on users and how it changed their behaviour according to situations. This change of behaviour when feeling like owning another body is called *the proteus effect* and has been explained by Yee and Bailenson (2007) in their excellent paper. According to Jun et al. (2018), IVBO is capable of changing our emotions very efficiently. For example, Lugrin et al. (2016b) revealed that embodying certain types of avatars in a VR experience could help with fighting acrophobia, the fear of heights. When playing as a robot or wooden character, the users felt that they had better stability and were less afraid of heights in the simulation. The work of Peck et al. (2013) proved that racial bias was highly reduced when users were embodying a black virtual character or even fighting gender stereotypes in Muller et al. (2017)'s paper. Also, Banakou et al. (2013) showed that embodying a child's body led to a change in the behaviour of the users. Making them feel and act more childishly.

2.2 Non-humanoid virtual character embodiment

In this project, the aim will be to use all of these IVBO characteristics to create an immersive experience centred around an animal character. This type of character is far from resembling a humanoid-like avatar. Therefore, we need to dive into the non-humanoid virtual character embodiment domain, which is also an area of research of high interest these days.

2.2.1 Extended Humanoid Avatars

Numerous studies have been conducted recently on augmenting the human body in virtual reality. Even though it does not represent a total change in human morphology, augmented humanoid characters still represent avatars rather distant from our own representation. For example, Won et al. (2015) worked on the human ability to inhabit non-humanoid characters with added body parts. Proving that we are capable of embodying such avatars. Ehrsson (2009) and Guterstam et al. (2011) proved that embodying a virtual character with a third arm was possible and that the users even felt like this third arm belonged to them in real life. In Kilteni et al. (2012)'s papers, the arms of a virtual character were stretched up to four times their initial realistic lengths without any loss in IVBO. In the same context as adding more body parts to virtual avatars, Sikström et al. (2014) Sikström et al. (2015) analysed different ways of creating IVBO of a pair of wings attached to the back of a virtual human character. They revealed that audio, visual, and sensory feedback added to the experience helped with inducing IVBO more efficiently while offering an analysis of different ways of controlling them. Finally, Steptoe et al. (2013) experimented with the control of a tail in virtual reality and analysed the feelings produced by controlling this extended humanoid avatar. Having additional or totally different body parts is very common among animals and insects. This research on extended humanoid avatar embodiment will be very interesting for this project.

2.2.2 Animal character embodiment

Embodying extended human characters is not the only subject of interest in this non-humanoid IVBO research domain. The embodiment of virtual avatars with a totally different morphology than humans has also been a subject of high interest these past few years. For instance, in 2014, Riva et al. (2014) stated, "*What if, instead of simply extending our morphology, a person could become something else, [...] a bat perhaps, or an animal so far removed from the human that it does not even have the same kind of skeleton...*". The democratisation of VR and its huge potential has forced many researchers to take an interest in the virtual aspect of this subject recently. Rhodin et al. (2015) were the first to study the case of animal embodiment with a 3D aspect. They mapped the participant movements with virtual animal characters, like spiders and horses, with different control methods. They then followed their research with Rhodin et al. (2015) where they controlled a caterpillar virtual character with a wave-like motion of their controllers to resemble the wave crawling movement these insects have.

Even though these two papers did not include any VR aspects, they paved the way for important future research in the domain. Two years later, Ahn et al. (2016) were the first ones to study the effects of embodying an animal character using an immersive technology like VR. They had participants embody animals like cows and fish, and gathered their opinions and feelings regarding the environment that was surrounding them. They found out that embodying these animals created a feeling of inclusion with nature and increased their feeling of involvement with nature in real life. Furthermore, Oyanagi and Ohmura (2019) experimented with the benefits of embodying a bird in VR to fight acrophobia, the fear of heights. They revealed that embodying such a character created all the characteristics of the *proteus effect* and made the participants less scared of the virtual height they faced. In another domain, Sikström et al. (2014) and Sikström et al. (2015) studied the effects of embodying characters with wings and found out that it was not preventing IVBO from appearing.

Closer to our subject, Krekhov et al. (2019a) and Krekhov et al. (2019b) focused their research on finding out if embodying a virtual animal character would still produce IVBO effects in participants. They also analysed the potential of using an animal character in VR video games and concluded that it was very promising. Participants even declared being more engaged with the game. They also found out that embodying animal characters could even be more efficient than embodying human-like avatars under certain conditions. The first paper focused on the viability of using animal characters in video games with IVBO questionnaires on participants as well as different ways of controlling them with different setups. They implemented the same principle Roth et al. (2017) used in their experiment by creating a virtual mirror to reflect the player's reflection to measure IVBO. The second paper focused on the viability and opportunities offered by using animal avatars in video games by experimenting with a game of the *escape room* genre. Finally, Andreasen et al. (2019) managed to create an immersive experience where users can embody a virtual bat in VR. They explained in detail the way they designed the experience and how they implemented the unique features of bats in the avatar. This last paper is particularly interesting to us as it closely resembles our project even though we took different directions when it comes to control and tools used. Research is not extremely developed on this topic, and not many papers are focused on the embodiment of non-humanoid characters in VR. The ones that focused on it used more trackers and different VR setups. Our goal in this project will be to extend their work and find a way to implement everything with a reduced number of trackers.

2.3 A measure of IVBO

Now that the different aspects and related work on IVBO and non-humanoid character embodiment have been presented, let's take a look at the way IVBO can be measured with Roth et al. (2017)'s work. This paper is based on an experiment conducted with a fake mirror and the analysis of how people react to a fake representation of themselves. They found out that control, acceptance, and change were the three most important aspects that defined IVBO. Out of these observations, they created a questionnaire that was able to measure the degree of IVBO created by a virtual experience. Krekhov et al. (2019b)'s paper extends this idea of a questionnaire to measure IVBO but for animal avatar centered experiences. They customised it, modifying and adding questions to suit their animal avatar experiments. Their goal was to prove that animal characters were a plausible and very suitable alternative for VR games. They also based their research on previous work that revealed that a majority of people were interested in embodying animal characters in video games as well as potentially feeling the effects of IVBO with such avatars. In this project, we will take example on Krekhov et al. (2019b)'s questionnaire and customise it to our specific needs to evaluate our resulting experience.

2.4 The importance of sound in VR embodiment experiences

For virtual experiences to be as immersive as possible, one key aspect of our reality has to be included in them too, sound. Interactive and passive sounds have been proven to be very important in inducing virtual presence in virtual reality situations. For instance, Wongutai et al. (2021)'s work demonstrated that interactive sound is a key aspect of immersion in VR serious games. Nordahl and Nilsson (2014)'s research proved that sound had a positive effect on VR experiments, increasing the appearance of "*the feeling of being there*", also called "*virtual presence*" while Serafin and Serafin (2004) focused more on the importance of sound in photo-realistic VR experiences. Closer to our project's subject, Lugrin et al. (2016a) revealed that audio feedback greatly increased the degree of IVBO felt while embodying a virtual character in mixed reality situations. This audio feedback can be described as the sound made by the character itself or when interacting with the environment. Finally, Sikström et al. (2014)'s work experimented with the role of sound in the ownership of a pair of virtual wings added to a human virtual character. It proved that audio feedback of the wings' movements increased the feeling of owning them and that they were easier to manipulate than without it. This aspect is very important for our project as it proves that adding audio feedback and interactions is important for increasing immersion or IVBO in VR.

2.5 The importance of the environment design

As important as sound is in virtual experiences, the design of the environment is also a key aspect in creating this immersive feeling. If the world you are immersed in is not well designed for the particular goal and situation intended, the feeling of belonging to it will be less present. For example, if you are supposed to embody a fish, the human brain will be more likely to feel like owning this virtual body if the environment is related to this fish body. Being in an aquarium or the water would make sense, as we generally see fish in water in real life. Research has been primarily conducted by Tanaka and Takagi (2004) on the best way to design virtual reality experiences' environments to increase virtual presence and reduce VR-related motion sickness. It proved that the design of the environment was very important in creating this immersive feeling. In Banakou et al. (2013)'s experiments, participants had to embody a child's body in VR and give estimates of the size of the objects surrounding them. Since the environment was not different than when they were embodying an adult character, they estimated the objects to be taller than they were. Even though this was not the first objective of the paper, it proved that the environment has to be well designed for embodiment experiences as the character and its environment are closely related. Berthelot et al. (2016)'s paper investigated the benefits of virtual reality (VR) in rehearsals for actors on green-screen movie sets. It demonstrated that the VR environment aided their practice by providing visual cues of the VFX that will surround their character in the final shots. Finally, Krekhov et al. (2019a) and Krekhov et al. (2019b)'s worked on animal embodiment by placing the participants in an animal body and emphasised the design of the environment they were in. For the first one, they placed them in a cage in a zoo environment, while for the second, they designed *virtual escape rooms* to resemble the natural animal habitats. Both these designs were thought of to help participants embody the virtual animal more easily. It will be important to think about this environmental design for this project too, as it plays an important role in the immersion.

2.6 Using VR and IVBO to help movie acting

For the final part of this background chapter, we will also analyse the work that has been done on the use of virtual reality on movie sets for actors. As the premises of this project were centred on the use of virtual reality to help motion-captured actors embody animals and creatures on movie sets, many of the concepts used in this domain helped the design of our implementation. As trying to create an immersive experience embodying an animal character is very close to helping actors immerse themselves in the skin of their played non-humanoid characters, we can get some interesting lessons from this research. For instance, Slater et al. (2000) worked on the challenges and difficulties of acting in a VR environment as well as the possibilities it could create. This paper was not specifically focused on the use of VR in movies but on the global spectrum of acting. On the other hand, Berthelot et al. (2016) explored the promising effects of using VR for rehearsals in movies with VFX where actors would be acting in front of green screens. By training in this virtual environment, actors had fewer problems visualising their imaginary environment and embodying their character on the day of the shooting. Furthermore, Kammerlander et al. (2021) and Kammerlander (2020) proved that the use of VR when doing motion capture could help with acting with different scaled virtual characters. When playing a very tall humanoid or non-humanoid character or even playing in front of another very tall character, it was easier to act accordingly as they did not have to imagine the other actor or themselves being very tall. Finally, Normand et al. (2012) also works on how VR could be used while acting, but with a focus on how it could be interesting to add a network functionality to it.

Chapter 3

Relevant Concepts

In this chapter, multiple concepts used in the design and implementation of this project are presented. They are essential to the understanding of the technical parts in chapters 4 and 5. Contrary to the concepts and work presented in the previous chapter, they are not directly related to the animal character embodiment subject as they only represent tools and technologies used in the implementation of these experiments.

3.1 Virtual Reality Tracking & Hardware

Virtual reality has been experiencing a tremendous resurgence in popularity in the past few years. At its beginning, the first commercial VR setups were not of good quality, and their hardware capacities were quite limited. Strong motion sickness, a nauseous feeling felt with some VR headsets, and high prices were the main causes of this low popularity. With the emergence of more developed and affordable HDM setups (Head Mounted Displays), as well as strong research and industry interest, virtual reality has once again become one of the most trending technologies on the market. Some of these new headsets are more portable than before, like the new *Meta Quest 2* (Occulus (2012)) which allows the users to use it anywhere with a very basic setup, one headset and two controllers. On the other hand, its main concurrent, the *HTC vive* (Vive (2015)), relies on more hardware power but less practicality with the necessity to use multiple tracking captors placed in the user's room. Nonetheless, both these setups produce the same effects in the end, tracking of the user's head and hands location and rotation in space. Using depth sensors and visual tracking technologies, these setups can track the users' movements in real-time and allow applications running on the device to use this data freely. See figure 3.1 below as an illustration of the way a VR setup tracks the user's head position and rotation in a room. Some setups may include more trackers for other

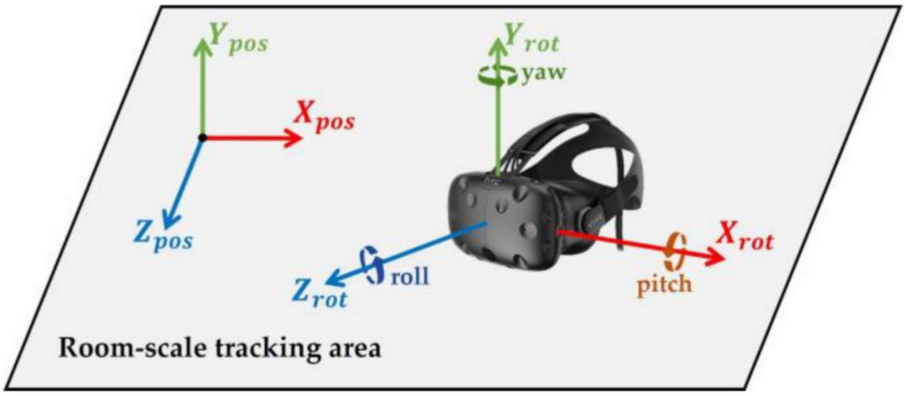

Figure 3.1: Tracking of position and rotation of a VR headset in a room via the *room-scale* technique. For instance, this is how the Meta Quest 2 tracks the user's head position.

parts of the body, like the legs or the waist, but simpler ones only track the head and the hands. This is the case of the *Meta Quest 2*, one of the most common VR setups used by the general public nowadays. This very useful setup can be used in many conditions but is limited by its tracking capability and its hardware capacity. Along with the fact that it can only track three points of the human body, this VR setup is of the *standalone* kind, meaning that it does not use any external computer to draw its power from. Working with this setup can be very challenging, but it is essential to know how to as it is the best-selling HDM on the market and could potentially be the future of VR setups, portable and practical.

3.2 Puppeteering

Puppetry, or puppeteering, is the act of manipulating inanimate objects called puppets as part of an artistic performance. This ancient practice has been historically used in theatre and is still very famous nowadays. Puppets are usually used to play with animals or imaginary characters on stage and are mostly controlled from the outside. Puppeteers control the puppets' movements with strings, their hands, or their body movements, but they hardly ever control them from the inside. Nonetheless, a new wave of puppetry is starting to appear with more advanced control of the puppets, requiring the action of multiple puppeteers, some of whom are sometimes completely inside them Gillinson (2022). This practice has also been the subject of a lot of research, especially on its potential use with computers and 3D technologies Husinsky and Bruckner (2018); Sturman (1998). Controlling a virtual character with any type of controller is very close to what puppeteers do to control their inanimate objects. More recently, the idea of using the concepts of puppetry in mixed reality has been explored by Shin et al. (2001), creating a *virtual stage*, proving that it could potentially be used with these immersive technologies. In the scope of this project, we will only be using simple concepts of this domain for small controls over our character's face. Just like puppeteers used strings to open the mouths or close the eyes of their puppets, we will be controlling them the same way, but with the buttons and triggers of our VR controllers.

3.3 Inverse Kinematics

In computer animation and robotics, given the parameters of a kinematic chain, the position and rotation of its end effector, for example, the hand of a robot or virtual character, can be effectively calculated with mathematical formulas, a process called *forward kinematics*. Its reverse operation is called *inverse kinematics* (Buss (2004)) and is in general more challenging to operate. Given the position and orientation of the start of the kinematic chain, for example, a shoulder, and the position and orientation of its end effectors, the objective is to calculate the in-between joint parameters that will enable these two conditions. See figure 3.2 for a simple illustration of how this process works.

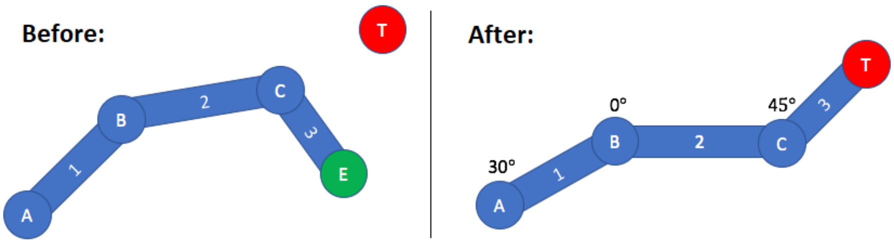

Figure 3.2: A simple example of how IK works in a three-joint scenario. The end effector is the green circle, and the target position is the red one. Before applying IK on the left and after on the right. Each pivot rotated by a certain amount to reach the final pose

This concept first originated in robotics but was quickly adopted in computer graphics for animation purposes. According to Aristidou et al. (2018), and their detailed overview of the IK methods used in computer animation, they can be divided into multiple categories.

Analytic solvers: These solvers are very fast and simple to use. They do not have the same convergence problems that numerical solvers can sometimes have, but it can be difficult to impose constraints on them. Analytical solvers can find solutions to an IK problem instantly by reducing it to a closed-form expression and are therefore very fast to compute. They do not rely on iterative calculations like numerical ones.

Numerical solvers: They are more versatile than analytical solvers and can be used on more complex and unusual structures. They either use Jacobian (first order) or Newton (second order) approximations of the forward kinematic to solve the problems. They produce a slow and smooth movement, leading to the desired final pose. However, they rely on an iterative process, requiring more computational effort than analytical solvers. This is an excellent candidate for our subject, as animals do have complex and unusual body structures.

Data-driven solvers: These solvers are based on the analysis of previously collected data, most often coming from motion capture. They try to find a similar solution to the current problem by comparing it to the previous correct data. The use of machine learning in this domain is very important and efficient for usual and well-known cases. But they are not optimised for specific and unusual cases like ours, based on animal movements.

Inverse kinematics has always been closely related to VR for enabling the possibility to animate parts of the virtual character's body that were not being tracked by the VR setup. Typically, legs, waist, and the whole lower body to then simulate full body movements virtually Parger et al. (2018); Caserman et al. (2019). Research in this specific area only takes into account humanoid virtual characters and not unusual characters like animals. Even though IK is used in the implementation of animal and non-humanoid virtual avatars, research is predominantly focused on these other scenarios. This is also why this project is also based on the use of IK for animal embodiment in VR, to give another perspective and piece of work on the problem using a real-time commercial engine and a very basic setup.

3.4 Blend shapes animation

Blend shapes animation, or also called morph targets animation, is a method used in computer animation to animate meshes by interpolating their vertices given multiple known "*faces (or targets)*", representing predefined positions. To create this effect, a deformed version of a mesh is created by storing its vertices positions for the target shape wanted. For each key-frames of a given animation, the vertices of the current face are interpolated between the neutral face and the stored deformed mesh to produce an animation Lewis et al. (2014). The amount of interpolation to produce is given by a coefficient. Multiple faces (or targets) can be used at the same time with their own coefficient to produce a blended animation of the neutral face. See figure 3.3 for an illustration of how a blendshape system works. This animation method is very versatile as it can create a huge number of different shapes and is very suited to facial animation thanks to its meticulous tweaking possibilities. Its only downside is that it is necessary to create all the different targets (or faces) meshes beforehand, requiring an extra amount of work and restricting the animation possibilities. This makes it quite incompatible with skeletal animation, which requires more freedom, but is excellent when used in addition to it. In our case, we will use blenshapes to help with puppeteering and control the subtle animation of our animal character's face and body, like the mouth, the eyes, etc.

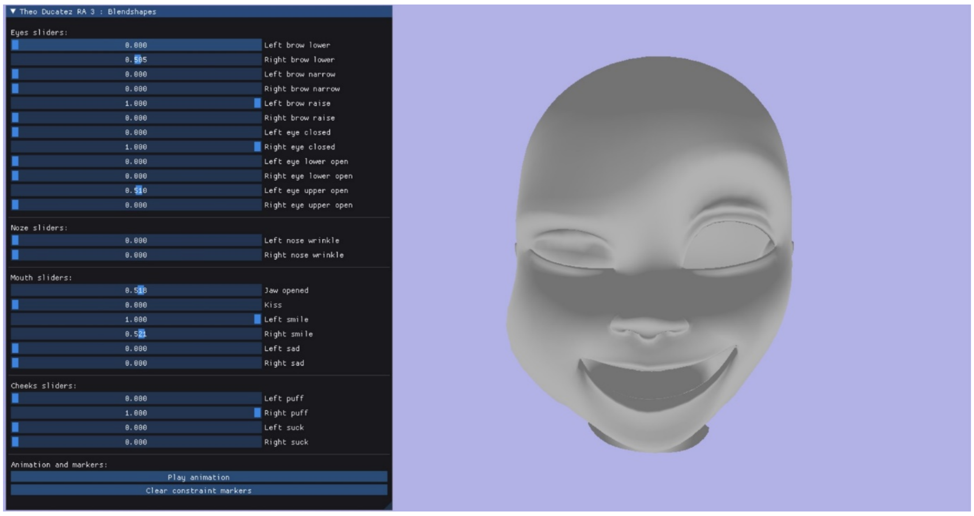

Figure 3.3: Previous work done as part of my Real-time Animation class illustrating how a blendshape system looks like. The slider on the left control the blendshape coefficients, creating this blended face expression on the right

3.5 Game Engines & Unreal Engine

In order to implement their real-time 3D experiments, most researchers often use free and publicly available graphic technologies like 3D computer graphics game engines. This allows them to focus on the core subject of their research, which sometimes is not directly related to its core technology, like animation or rendering, and save an incredible amount of time. These tools are widely adopted in research and are now used in all industries. They are optimised, easy to use and have a myriad of features and plugins to help anyone experiment quickly. In the research domain of virtual reality, *Unity* (Unitys Technologies (2004)) is the most used, with its wide VR support and simplicity. It is the best candidate for most experiments, but in this project, we will be using *Unreal Engine* (Epics Games (1991)), its principal concurrent. Unreal Engine has rarely been used in VR research because of its lack of VR support and its complexity. While Unity has many pre-built features, Unreal Engine forces users to implement them themselves. Nonetheless, Unreal Engine is a very interesting tool for experimenting, as it features a *Blueprint* system, a visual scripting mechanic allowing for easy and fast prototyping. I found the challenge of developing this project with Unreal Engine more interesting and wanted to contribute by implementing a VR solution with this engine. Moreover, no animal embodiment and very few VR experiments have been done using this 3D software. Using this tool would offer an overview of creating such an application with it for the first time.

3.5.1 Blueprints

In Unreal Engine, developers can choose between two different ways of implementing in-game features. They can either use the native C++ language, or they can use a visual scripting tool offered by Unreal Engine called "*Blueprints.* This visual scripting language is a node-based language that allows programmers to code visually by linking nodes to each other. This functionality is very useful when prototyping or experimenting, as it allows anyone to create new features quickly without any knowledge of the C++ language. However, blueprints are not as powerful as C++ as they rely on a virtual machine environment to be compiled during the game. The difference is very small and should not be taken into account when prototyping or experimenting. The animation and in-game assets all rely on the blueprint system in Unreal Engine. I found it interesting to present what blueprints were, as we will be using them a lot during the implementation phase. Most of the coding was done with these blueprints, and all Unreal Engine assets also include blueprint-specific functionalities. See figure 3.4 for an example of what a blueprint graph looks like.

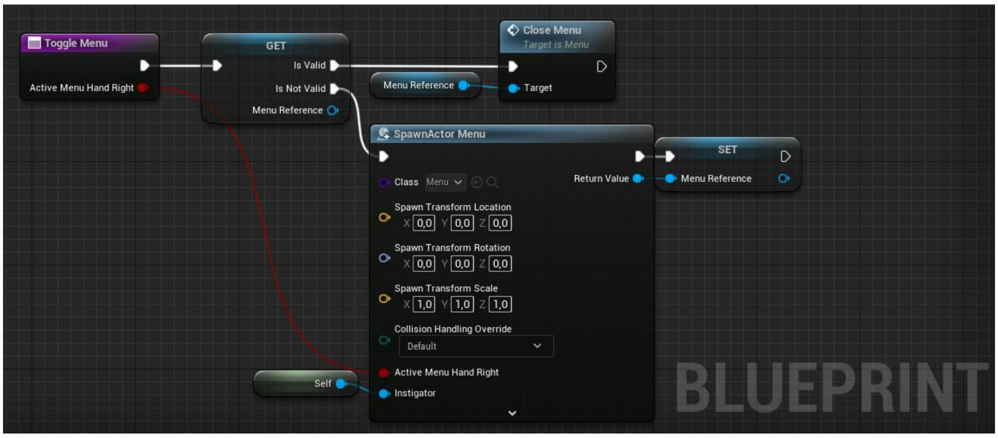

Figure 3.4: Simple example of a blueprint. The execution link between the nodes is represented in white. Each node is an operation. In this example, a UI menu is toggled. If the menu already existed, we call the "close menu" function, else, we spawn the menu and save its reference

3.5.2 Control Rig

Control Rigs in Unreal Engine are scriptable rigging systems based on blueprints and are mainly designed for controlling properties to drive animation. They are a specific asset provided by unreal engine and allow animators and designers to easily control characters. They can be used to control the animation of a character in-game, create IK systems, create animations with the sequencer, and many more. They work on the same basis as blueprints but have their own lighter-weight virtual machine called RigVM to provide highly efficient pose calculations. They also use a node-based visual scripting system called *Control Rig Graph*. When importing a skeletal mesh into Unreal Engine, a rig hierarchy will be created. Rig hierarchy's bones, along with additional bones, spaces, and controls created in the Control Rig Editor, can be used as inputs or outputs of control rig nodes. Control Rigs are not only limited to skeletal meshes and can be used to animate other components, such as static meshes and lights. In this project, two different control rigs are used together to create the final animation of the skeletal mesh animal character. I also found it interesting to present what control rigs were too, as they are used in the implementation phase. See figure 3.5 for an illustration of what a control rig system looks like.

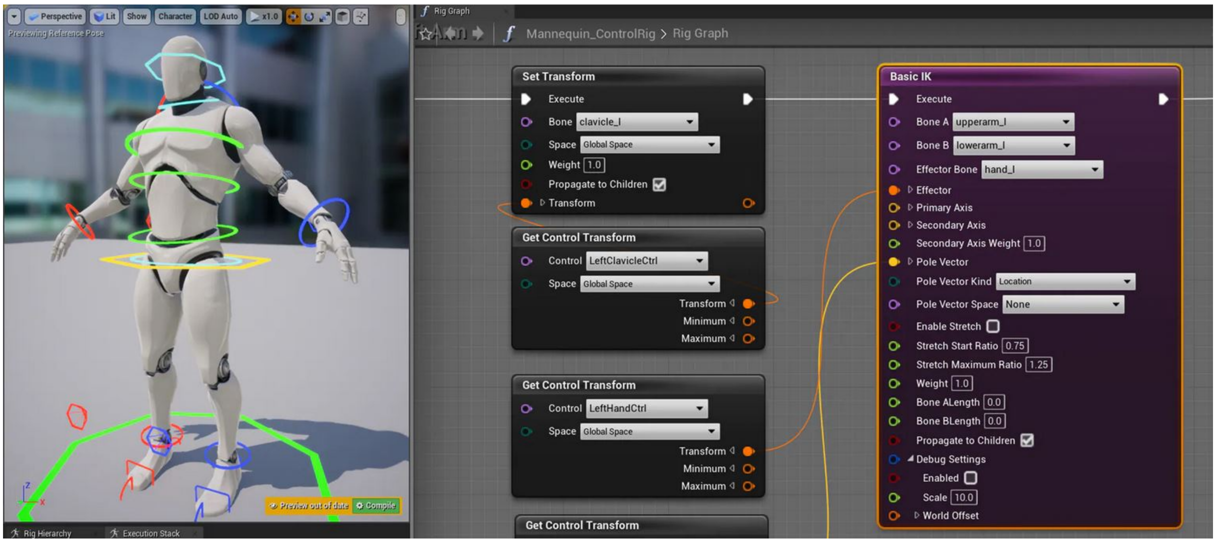

Figure 3.5: Illustration of what a control rig system looks like in Unreal Engine. We can see what controllers look like on the skeletal mesh character on the left. On the right, the Control Rig Graph controls the way the controllers are used to control the character's bones and its control system overall.

Chapter 4

Design

This chapter contains the design of the immersive embodiment experience created for this project. Different theories and technical choices are discussed along with the important features our experience will have to provide for a successful immersive experience. These choices are made with regard to the previous chapters' content and background.

4.1 Choice of Animal

One of the most important choices to make in an immersive experience focused on the embodiment of a non-humanoid character is the animal avatar players are going to embody. This choice will influence the whole experiment design and implementation, hence the importance of choosing it early in the project. It can be determined by the goal of the experiment developed, for instance Andreasen et al. (2019) chose a bat for their experiment to determine if echolocation could be simulated in VR, or in Oyanagi and Ohmura (2019)'s research, a bird avatar was chosen to help with fighting acrophobia. In the scope of our project, no specific goal has been determined for the embodiment experience except that a very basic VR setup has to be used and that the application will be developed with the Unreal Engine 3D engine Epics Games (1991). Therefore, the choice of animal to embody only comes down to the availability of the animal 3D asset, its compatibility with Unreal Engine, and a 3-tracker system. By browsing into Epic Game's free assets content, I came across Weta Digital's meerkat short movie demo project (Wetas Digital (1993)) assets available for free. This short demo was originally created by the VFX company to demonstrate the possibilities Unreal Engine could provide with real-time fur simulation for movie productions. The project assets were available for free on their marketplace, and the meerkat model they used was of very good quality. This meerkat was an excellent candidate for this project, knowing that this model was well rigged and that it would be compatible with Unreal Engine. Its morphology is quite close to the human one but still has enough differences to make it challenging to map it to human movements with only 3 trackers. And as Krekhov et al. (2019b) recommend in their paper, it is better to design the avatar such that the altered morphology feels like an extension of our own body, instead of being a restriction. A meerkat in a standing position could be used so that all its body parts could be seen as an extension of ours. See figure 4.1 for an illustration of Weta Digital's meerkat demo project.

Figure 4.1: Illustration of Weta Digital's short movie demo project. We can see the meerkat 3D model with fur selected on the right of the image

4.2 Mapping approach

Choosing the right mapping approach is the most important aspect of virtual embodiment. This choice will determine how the players will control their virtual representation and will directly influence their experience. In the case of a human virtual character, this mapping approach is straightforward as the two morphologies are usually similar. In our case, this mapping approach is directly influenced by the animal avatar chosen for the experiment, as they can have varying morphologies, sometimes very far from the human one. As mapping approaches have already been studied for virtual avatar embodiment, we will be using the knowledge gathered in previous research to decide which one would be optimal in our case. From this, three key concepts appeared to be crucial in creating a realistic immersive experience inducing IVBO:

First person point of view: The importance of a first-person perspective in VR IVBO experiments has been proven by Medeiros et al. (2018) and Maselli and Slater (2013).

These two papers experimented with different mapping approaches for embodying virtual human characters in VR and both came to the conclusion that first-person mappings were far superior in inducing IVBO. Even though these papers focused on humanoid avatars, this concept has also been proven valid for animal avatar embodiment. In their virtual animal body ownership paper, Krekhov et al. (2019b) experimented with multiple camera perspectives, including first person and third person, and concluded that first person was far superior in inducing IVBO.

Movement matching: Synchronisation of both the player and the virtual character's movements is a key part of virtual embodiment. As proven by Sanchez-Vives et al. (2010) in their research, it is very important to synchronise the movements of the player as precisely as possible with their virtual avatar to get the best IVBO results. Therefore, important body movements should be precisely matched. For instance, in our case, hands and head movements have to be synchronised with our meerkat character's body parts movement. For that, we will use our three trackers and map their movements to the corresponding parts of our meerkat.

Half body tracking: Related to the previous point, the position from which the players will be controlling our meerkat is also important to take into account. The mapping approach taken will influence the position they will have during the experience, which can result in negative feedback in certain cases. In their paper, Krekhov et al. (2019b) compared two different position and movement mappings for the embodiment of animal virtual characters, full-body and half-body tracking. Their full-body tracking approach involved the tracking of the participants' legs, arms, waists, and heads and enabled full control of the virtual character. This approach gave great results as the mapping was almost perfect between the two bodies' movements, but brought up some issues. First of all, this approach requires a specific VR setup with trackers for the legs and the waist. Then, as animal body structures can be quite different from the human one, an exact mapping could cause some control problems. For instance, animals that do not have upright postures (4-legged) would force the participants to crawl on the floor to control them, causing issues with the trackers touching the floor and induced exhaustion over a long period of time. For these reasons, they recommend using a *half-body mapping*.This option does not have any mapping of the user's legs and can be used for all types of animals while having the users stand straight and walk normally. In this setup, only the head, the hands, and the waist were tracked. They proved that this mapping did not cause any major decrease in IVBO and was easier to undertake for the participants.

These three important concepts are totally compatible with our choice of using a meerkat as a character. The first-person point of view will allow us to see the world directly from the meerkat's perspective, while also being able to see its body. The movement matching will allow us to move the important parts of its body, more precisely its hands and its head. And the half-body tracking will allow us to control it while standing up, without the need to crawl on the floor. Although meerkats are four-legged animals and crawl on the floor to move, they are famously known to stand up on their legs to look around them. See figure 4.2 for an illustration of the posture and the skeleton of a meerkat standing up. Without the legs or waist tracking, we could use this exact posture for the embodiment and have a nearly one-to-one mapping. This choice of animal is therefore perfect for this half-body mapping.

Figure 4.2: Illustration of the posture a meerkat's skeleton has while standing up. We can see that they have quite the same posture as a human, except for their shoulder bones, which are way more forward than ours. Picture taken from pterosaurheresies (2021)'s blog on meerkat morphology and body posture

4.3 User interactions

Now that the animal character and the mapping approaches have been discussed, we need to think about how the user will interact with its virtual representation. As seen in most of the VR related subject papers studied, interaction is a key part of embodiment. Embodiment will fail if a user does not receive visual feedback on its movements mapped on its character. Furthermore, it should have different ways of controlling certain parts of its avatar, just like in real life, as well as interacting with its surrounding environment. All of these concepts are very important in inducing IVBO and especially virtual presence.

4.3.1 Animal body movements

As stated in the previous paragraphs, we will be using a half-body mapping to control our animal character. The objective will be to map the user's movements to the virtual avatar with only three tracked positions, the head and the left and right hands. The simple solution to this approach would be to create a one-to-one synchronisation of the user's head and hands and its virtual representation. The virtual elements' position and rotation would be identical to those of our user. By adopting this simple solution, a problem arises. What happens when the user translates its head or hands from one place to another? These dynamic body parts will stretch indefinitely while the static parts will stay in the same place. To fix this problem, we could move the whole body according to these movements. But once again, another problem arises, the whole body will follow no matter where they move, left and right, up and down, which is very unnatural. The same applies to the hands. This is due to the fact that our animal character's body is static and the moving parts of it have no influence on the other ones.

To fix these problems, it is necessary to use inverse kinematics to move other body parts accordingly. While the head and hands movements will indeed directly move the head and hands of our virtual character, we will use inverse kinematics to indirectly move other parts like the arms, the legs, and the waist of our avatar. We will need multiple inverse kinematics systems applied to the skeleton of our animal. One for each limb, so that we can have a limb IK effect on the arms and legs, and one for the whole body movement to influence all of them. This way, moving the hands will have an indirect influence on the arms, and moving the head will have an indirect influence on the legs, the waist, and the trunk. This will allow the avatar to have a crouching movement when the user crouches, arms moving the same way when hands move, with respect to the natural body limitations and constraints if possible, and the body leaning forward and backward when the user does as well. See figure 4.3 for a summary of the interactions the user will be able to have in the experience.

Being able to rotate and move in the virtual environment will also be something to implement in the experience. This is where not having any trackers for the waist makes the implementation very difficult. Without it there is no way of differentiating when the user will be rotating their entire body and when the user will be rotating only their head. In the same way, there is no way of differentiating when the user will be only leaning or walking. Therefore, two different approaches to the body rotation could be adopted; following the head rotation movement, or using the controller's joystick. Rotating the whole body with the head direction would make it impossible for the user to turn their head around without making the whole character turn. This could be a very good option for unrealistic video games, but for an embodiment experiment, this could result in a lower IVBO feeling. As this result would be quite unnatural, the best option in our case would be to use the joystick for rotating the whole body and the headset's rotation for rotating the head only. This would allow the user to look around without turning completely and to turn only when necessary. However, something to take into account with this option is that it would force the users to stay in the same body orientation the whole time. This approach could also impact the IVBO effect but is far more realistic than the previous approach. Concerning the movement in the environment, two approaches can also be adopted. Rely on a teleportation system or on a locomotion system based on the joystick or the user's translation. Once again, not having any trackers for the legs or the waist is problematic in this case. For the same reasons as the rotation, using the user's translation in its room would be very hard to implement and quite unnatural. Both the other options would be correct in our case, but the more natural way would be a locomotion system based on joystick movement. This would obviously have to be accompanied by an animation system for the legs involving the IK system.

4.3.2 Puppeteering and Blendshapes

To create a believable illusion of body ownership, moving the arms and the head of the virtual character is not enough. The user has to believe in the total embodiment of its virtual character, and this also involves more subtle body movements as well. These subtle movements were not taken into account in previous research on the subject, and implementing a way of controlling them would be new and very interesting. For instance, the face is very static and never moves in the previous work about non-humanoid character embodiment we covered. Adding a functionality that lets the user control parts of the static body of the character could give it more life, leading to a potential increase in IVBO. This idea has been influenced by the puppeteering reading made during the background review of the project. The goal would be to replicate how puppeteers control some small parts of their puppet's body with their fingers or some light movements of other parts of their body. For instance, the mouth and the eyes of puppets are usually controlled directly by strings attached to the puppeteer's fingers. In our case, we will be mapping the buttons and triggers of our controllers to different motions of the face and body of our virtual character. We will be using blendshapes to control these subtle movements of the body. We will mostly be using the trigger buttons of the controllers as they will allow us to have more control over the blendshapes coefficients. A direct mapping of the intensity of the trigger to the coefficient of one or more blendshapes will therefore be possible. As for the different blendshapes triggered, knowing that most controllers only have 4 trigger buttons, the objective will be to be able to implement 4 different movements. Here are the movements chosen, also summarised in figure 4.3 below.

- Opening the mouth with a trigger button.
- Closing the eyes with a trigger button.
- Moving the ears with a trigger button.
- Make the meerkat breathe with a trigger button.

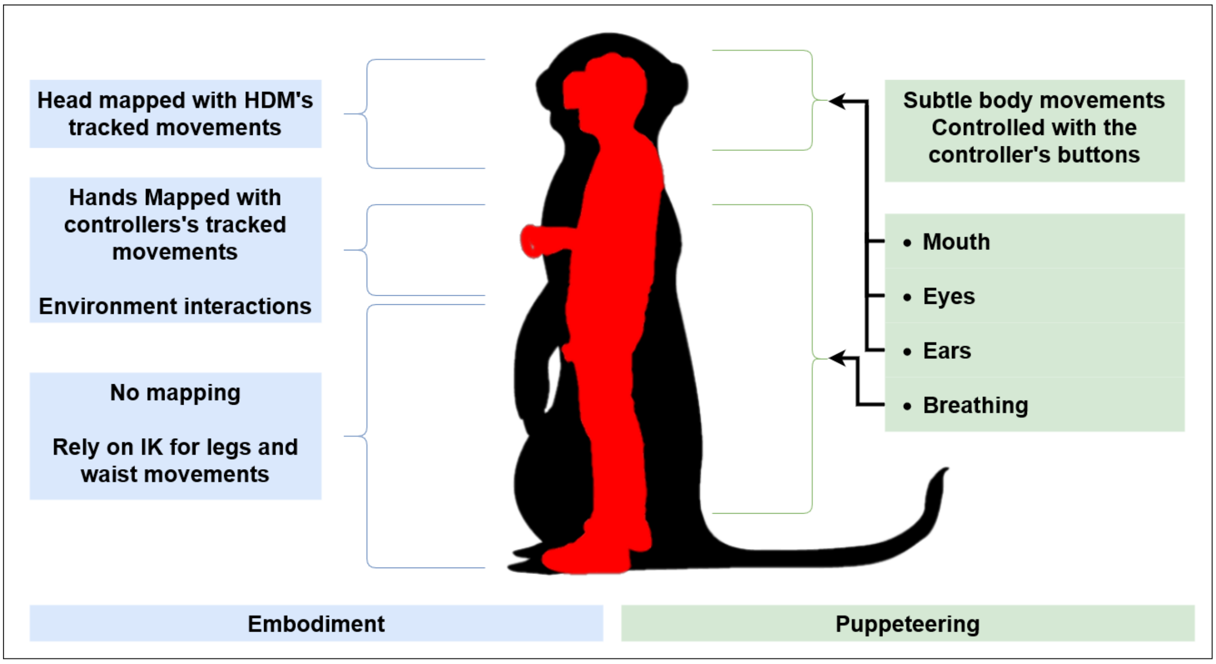

Figure 4.3: summary illustration of the different interactions the user will be able to have with our meerkat character, embodiment and puppeteering

4.3.3 Environment interaction

One last important detail about the interactions the users will be able to have with the game is the environment-related interactivity. Just like in real life, the user should be able to move or touch certain objects and interact with others. Our experiment will have to implement at least one interactive aspect in our environment. Inspired by Roth et al. (2017)'s IVBO questionnaire research, we will be implementing a mirror in the environment so that the players will be able to see their animal representation's reflection. This mirror concept was not at first created for non-humanoid characters but has later been proven to greatly increase IVBO in these conditions by Krekhov et al. (2019a)'s experiment. After conducting their research on IVBO for animal characters in VR, they advised the use of props like water reflections or mirrors to help users increase awareness of their virtual representation. They also proved that this small feature increased greatly IVBO in their experiment. Along with that, the environment will also have to include at least one prop the user will be able to pick up and throw away. As meerkats are known to be playful, being able to play with a part of the environment will be a plus; VR is also very suited for these kinds of interactions. All of these will have to blend into the environment so that the immersion is not impacted by their presence.

4.4 Environment design

The environment in which a VR experience takes place is very important in general. If not well designed, it could result in either VR-induced motion sickness or exhaustion. Moreover, it could also lead the user to a feeling of not belonging in the environment, leading to a decrease in virtual presence. For non-humanoid character embodiment VR experiences, this aspect is almost more important than any other. All the previous research on this topic involved the creation of a believable environment that would suit the subject. For instance, Krekhov et al. (2019b) decided to place the users in a zoo environment, because they were embodying different animals. Oyanagi and Ohmura (2019) decided to place them in a bird cage, as they were embodying a bird. Andreasen et al. (2019) placed them in a jungle because they embodied a bat. Krekhov et al. (2019a), while being focused on the "escape-room" side of their project, they decided to blend the escape room into the natural environment of the animals their participants embodied. This environment design is always present in these experiments as it helps increase their IVBO level. Embodying a non-humanoid character is a very challenging and unnatural process. Living this experience in a well-designed and engaging environment related to our animal character is necessary. It also palliates the potential weaknesses of the character's mapping and control. If the user could focus on something else than its imperfect virtual representation, it could help him reach a good enough IVBO level.

In our case, we first have to take into account that a meerkat is a very small animal. Therefore, our surrounding environment will have to be taller than us no matter what. Our players have to feel like they are small in their environment, just like what we think a meerkat should feel like. Since our virtual character will have to be upsized to match the size of the player, all the objects that would feel normally sized to us in real life will have to be upsized to create this illusion. This aspect of the environment is very important to take into account as it will directly influence the feeling of virtual presence. As proven by Banakou et al. (2013)'s in their child embodiment experiment, if a user embodies a small character in a VR experience, he will expect his surrounding environment to be bigger and taller than him. If it is not the case, just like in their experiment, it will produce an overestimation of the size of their surrounding objects, leading to an unnatural feeling about their environment. They will also start to question the size of their character. Respecting this size aspect is therefore very important for IVBO.

As seen previously, this environment will also have to match the natural habitat of the chosen animal. As our character is a meerkat, our experience will have to take place in one of its two natural habitats, a desert, or a grassland. Taking into account the hardware limitations of our standalone headset, the best choice would be to create a desert environment. Standalone headsets cannot handle the rendering of too much foliage in a scene. Since performance is also important for a successful embodiment, choosing a desert area would be the right choice to ensure a high FPS stability. The objective will be to design an environment resembling the famous South African Kalahari Desert. See figure 4.4 for an illustration of the objective environment.

Figure 4.4: Illustration of what the South African Kalahari-desert looks like. Photo taken by 959dscott and posted on TripAdvisor.

4.5 Sound design

Research proved that along with environment design, sound design was also an important aspect for VR embodiment experiments. Nordahl and Nilsson (2014) proved that sound effects and feedback had a positive effect of inducing a feeling of virtual presence in VR environments in general, and Lugrin et al. (2016a) revealed that audio feedback greatly increased the degree of IVBO felt by its participants. Sound feedback is even more important for animal embodiment cases as, just like the environment, it could help the users really feel like they belong in their virtual environment. Sikström et al. (2014) showed that audio feedback helped their participants embody a pair of virtual wings more easily. As advised by all these papers, we will be adding two types of audio to our experiment. A sound effect on some puppeteering interactions, and an ambient sound in the environment itself. For the first one, a sound effect of a meerkat's cry will be played when the player opens its mouth and a sound effect of a meerkat moaning will be played when the player makes it breathe. In the second, a desert ambient sound will be played at all times in the environment, along with meerkat cries in the distance. These two features will help the player feel included in its virtual environment.

4.6 Hypotheses & Evaluation criterias

The main objective of this project is to create an immersive VR experience focused on the embodiment of an animal character. Therefore, evaluating the amount of IVBO and immersion felt in this experiment will be one part of the evaluation. Along with identifying what features of the experiment have the most effect on it. For this, we will be making a trial with anonymous participants and make them fill out a questionnaire inspired by Roth et al. (2017) and Krekhov et al. (2019b)'s IVBO measurement questionnaires. Our second objective is to evaluate the viability of using a 3 tracker system and the Unreal Engine 3D engine to create an animal embodiment VR experience. Therefore, evaluating the difficulty of the process of creating such an experiment with these tools will also be part of the evaluation.

As for the hypotheses, I project that creating this VR experience will be difficult as the 3D engine used is not the most suited for VR and that having only 3 trackers can cause many problems during the implementation. Secondly, I also project that the mirror, the environment and sound design, and the hands and head mapping systems will be very important factors for inducing IVBO in our experience. I also project that participants will at least feel partial effects of IVBO and feel like embodying the meerkat.

Chapter 5

Implementation

This chapter deals with the implementation of all the features and concepts designed in the previous chapter. It is intended to be an overview of the implementation process as well as a discussion of the technical implementation choices made. This chapter is presented in the same chronological order as the experience has been developed during the implementation phase. The result of this implementation will then be evaluated in the next chapter.

5.1 Setup

5.1.1 Hardware

The hardware used for the implementation consisted of a laptop computer and a Meta Quest 2 VR headset setup. The laptop was used to create the application on Unreal Engine and the headset was used to run the application and interact with it. Both were connected with either a USB cable or the *Air-Link* functionality provided by the Quest 2. The VR setup was composed of a Meat Quest 2 HMD and two Occulus Touch controllers. Here are the specifications of the laptop and headset used:

Operating System: Windows 10 Home version 21H2

CPU: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz

Graphic Card: NVIDIA GeForce GTX 1060 (Laptop Version)

RAM: 16,0 GB

Headset Version: 39.0.0.88.336.363807235

See figure 5.1 below for an illustration of the setup used.

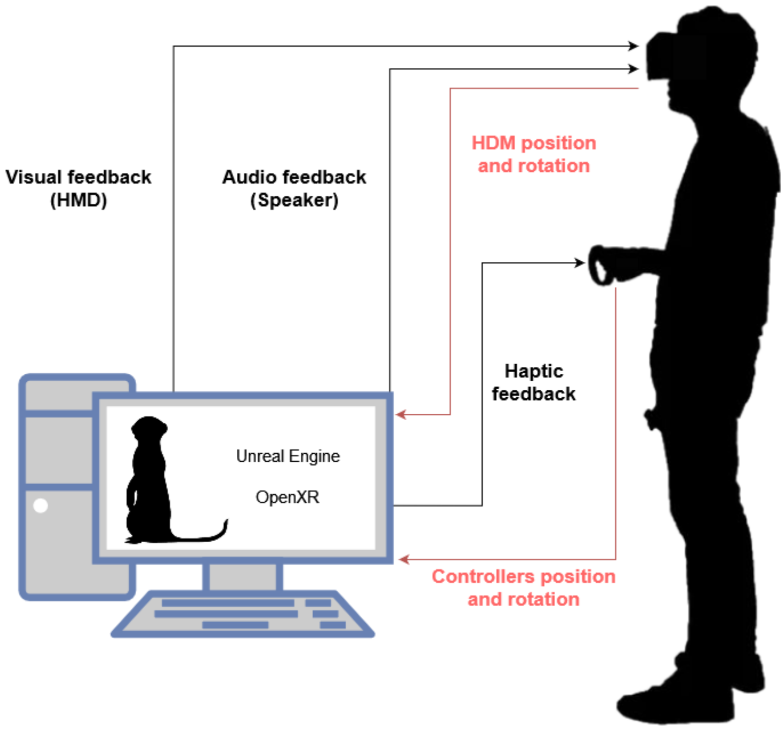

Figure 5.1: Setup used for the implementation and the information flows represented by arrows

5.1.2 Software & Assets

Multiple tools were used for the software setup used for the implementation. Unreal Engine version 5.0.2 was used to develop the entire application, along with all its necessary plugins for developing in VR (Quest VR, OpenXR, ...). The Occulus desktop app was used for the connection between the computer and the headset, allowing us to launch the app on the headset during the development phase. Quixel Bridge and its Megascan assets library were used to create the environment and import quality assets directly into Unreal Engine. As for the VR framework used to develop the application, OpenXR's SDK was used as it has now become the standard in VR/AR/XR software development. Finally, Audacity was also used to edit the different sound effects used in the environment.

Concerning the assets used, Weta Digital's free Meerkat video demo project assets were used as well as the Unreal Engine VR template project. The first one helped me with the meerkat 3D model and its textures, while the second helped me to start developing the basic VR functionalities in Unreal Engine. A detailed list of the assets used is available in the appendix.

5.2 Embodiment Features

As the main subject of this project is the embodiment of an animal virtual character, I first started working on creating all these features first and decided to implement the sound and environment parts later. This section therefore focuses on the development of this whole character embodiment aspect.

5.2.1 VR

The first step when creating a VR application is to start with the basic VR-related features of the game. These features include the ability to see the world through the HMD's screen with a camera moving along with the user's movement and being able to see the controller's motion in the game. These features are supported by Unreal Engine's Virtual Reality plugin and are also explained in detail in their documentation. To implement these features, we first need to create a PAWN blueprint. In Unreal Engine, PAWN actors are actors (i.e. game objects) that can be controlled by the players. Once created, I simply added a camera component to this actor and synchronised its position with the HDM's movements. With this implemented, the player can look around the scene and the camera component follows the exact same movements as he does in real life. For the controllers, Unreal Engine's virtual reality plugin already provides a solution for it. I just had to add two *motionController* components to the PAWN blueprint and set one of them to be the left controller to enable this functionality. With these two basic VR features added, we were now able to track the user's head and hand position and rotation, the foundation of our embodiment functionality.

5.2.2 Mirror

As the mirror was going to be useful for the whole embodiment implementation, I decided to create it before implementing these other functionalities. Unreal Engine offers multiple ways of creating a mirror effect in-game, but not all are compatible with the Occulus Quest 2. As this HMD is a standalone headset, its performances are very limited and some rendering techniques are not supported by its hardware. Therefore, relying on Unreal Engine's planar reflection effect was impossible. This component creates a perfect mirror effect very easily but is not compatible with mobile (android platform targets like the Quest 2) project configurations as it requires a lot of resources. Another option was to use a very metallic material and apply it on a plane. But Unreal Engine does not reflect dynamic objects on reflective materials' surfaces, leading to a very good reflection of the static environment, but no reflection of our moving character. As seeing our own dynamic movements in the mirror was very important, this was not a correct solution either. Finally, the solution adopted was to use a *sceneCaptureCube*. This component acts as another camera in the scene and is bound to a *RenderTarget*. It creates a reflection effect by rendering the scene from its own perspective and rendering it into a texture. This texture can be used in a material to then be applied to a plane shape, transforming it into a perfect mirror. The final step was to make the *sceneCaptureCube* follow the HMD's translation movements projected into the plane when the user was moving. Without this last modification, the resulting mirror effect was very static and did not feel very real. Following the user's head movement made it react like a real mirror. To create this effect, I simply used the blueprint system to change the capture component's location relative to the HMD's planar projection location on the plane. See figure 5.2 for an illustration of the result. As this solution renders the scene multiple times every frame, I ran into some performance issues later in the project when adding details to the environment. To fix this, I simply stopped rendering this mirror effect when the mirror actor was not in the user's sight, greatly increasing the overall FPS.

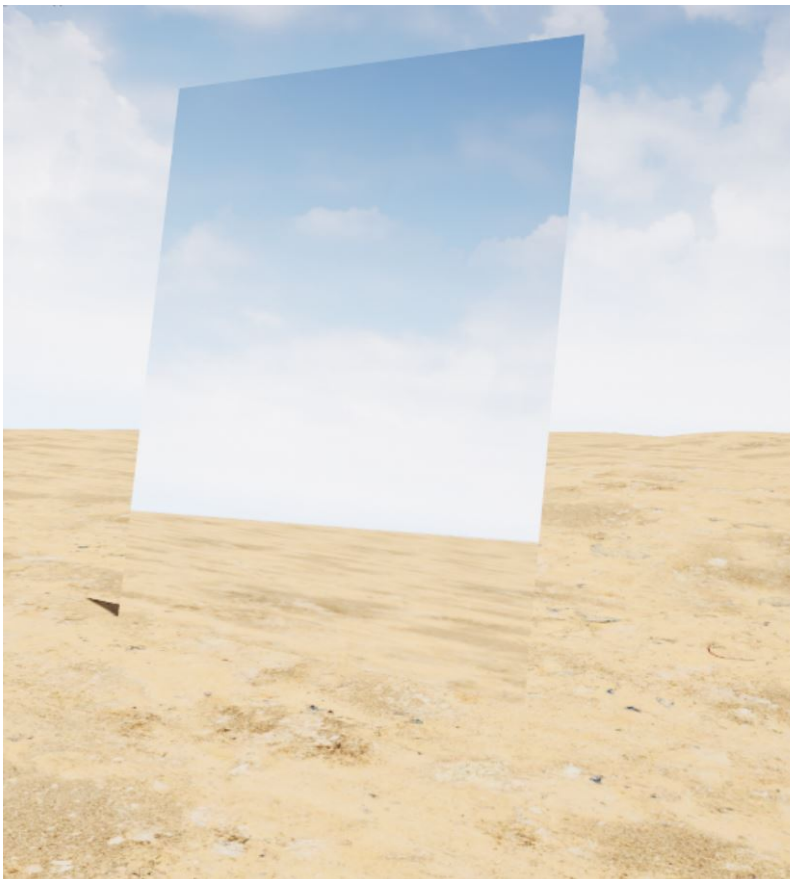

Figure 5.2: Image of the mirror created in an empty environment

5.2.3 Meerkat Avatar

With the mirror successfully created, being able to have direct visual feedback on the movement of our animal character was now possible. The next step in this embodiment implementation phase was the setup of the meerkat 3D model. As this 3D model asset was taken from another Unreal Engine project, I had to migrate it from one project to another. Due to the compatibility issues encountered when migrating a project from Unreal 4 to Unreal 5, many problems occurred with the migrated assets. In particular with the textures, materials, and Unreal's grooming (fur) assets. This issue slowed down my progress a lot and forced me to implement some of these assets again. Although this solved most of the problems, the results were not as good as in the original project but were good enough for the experiment. After many attempts, adding fur to the meerkat model ended in failure. Grooming rendering is not well supported by the Occulus and requires a lot of resources, which this HMD does not have. The numerous compatibility issues and the fact that the scene was rendered twice with the mirror effect in the scene did not help with it either. Therefore, this is why our meerkat does not have any fur in this experiment (see figure 5.3 for a picture of the model). After adding this new meerkat skeletal mesh model into our PAWN blueprint, implementing the hands and head mapping functionalities was now our next objective.

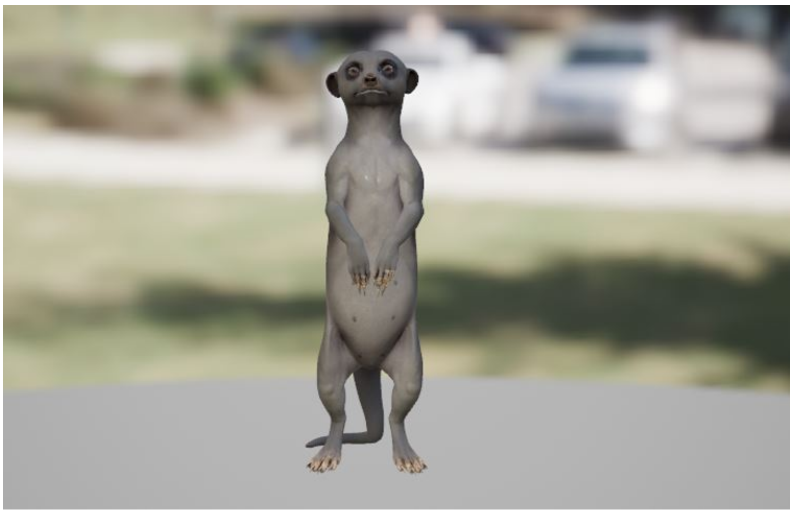

Figure 5.3: Meerkat character 3D model used, without any fur

Head Tracking

The objective of the head tracking functionality is to project the player's head movement onto the virtual character's head. To produce such an effect, we need to take care of two different types of head movements, translation and rotation. Handling the rotation of the head is a straightforward approach as it does not imply any implicit movement of other parts of the body. But handling its translation movement is more complicated as it is directly correlated to leg and trunk movements. The head does not move on its own but moves as a result of the whole upper and lower body moving.

To implement the rotation effect of the head, we will be adding two new components to our 3D model character, an *Animation Blueprint* and a *Control Rig*. The animation blueprint will be responsible for the overall animation of our character, while the control rig will simply help it control certain parts of the body dynamically. When the camera rotates when the player's head rotates, the new rotation value is transmitted to the animation blueprint. The animation blueprint then sends this value to the control rig, which handles the rotation of the "head bone" socket of the meerkat's body. This results in the rotation of the meerkat's head in the exact same way the player does in real life, without any movement of the rest of the body. See figure 5.4 below for an illustration.

Figure 5.4: Meerkat character reacting to the player's head rotation. The rotation of its head follows the same movements made by the HMD

As explained above, handling the head translation is far more difficult. Since we do not have any trackers for the waist, we cannot tell the difference between a bending movement and a ducking movement. We could be using the position of the controllers to try to make a difference, but this would not be accurate in the end. After many different experiments, I decided to treat the mapping of head movement differently. I decided to interpret a movement of the user's head as a ducking movement all the time and let inverse kinematics handle leaning movements indirectly. More precisely, I decided to move the whole upper body with any head movements from the user. As a rule of thumb, when our head goes in a certain direction, it means that the torso and upper body go in this direction as well. Therefore, this should not be too unnatural for our meerkat's character either. This then means that, when the user's head goes down, our meerkat will react by having its whole upper body go down more and more. The same applies for other directions. For example, when going up, its whole upper body will go up. This is a good solution, but a problem appears as a result of this decision. The whole lower body of our character is going to be static and follow the upper body movement when moving. This makes the body feel totally unnatural and makes our avatar go through the floor when going down and fly in the air when going too high up. The solution to this problem is to use Inverse Kinematics.

With the release of Unreal Engine 5, a new IK system called *IK rig* has been included in the software. This new feature allows for an easy application of inverse kinematics to skeletal meshes as well as new motion retargeting possibilities. I decided to use this new system to handle the inverse kinematics of the whole body of my character. This new system works by defining goals for certain parts of the skeletal mesh and assigning different solvers to them to create the IK effect on them. These goals can be translated and rotated at run-time the same way bones can be translated and rotated with a control rig, by using an animation blueprint. For the upper body movement, I defined a goal on the meerkat's neck and assigned it to a *Set Transform* solver. This solver's objective is to change the selected bone's location and rotation according to the position of the set goal. This solver does not involve any IK system and simply translates and rotates a single bone and its children. With this added, our character's upper body was now following the translations of the HMD. However, I also had to add an offset to the meerkat's body position so that the camera could stay inside its head when moving. To handle the legs' indirect reaction to this translation of the upper body, we will now need to use an IK solver. I assigned an IK goal to each leg and a *Limb solver* to each. A limb solver is a numerical solver that requires a root bone and a single IK goal to function. Its objective is to act as a single-chain IK Solver and is usually used for a character's limbs. By setting their goals to each of the two feet goals created and their root bone to each hip, it produces an IK effect on the legs. These solvers' calculations need to be performed after the transform solver so that they can react to a change in the hips' location. As a result, we now have a meerkat character that is able to rotate its head according to the HMD's rotation and move its whole body in a natural way when the HMD's location changes. When the user crouches, the meerkat does the same with the help of IK on both its legs (see figure 5.5 for an image of the crouching result). As decided in the design part, we will not be handling body rotation with the headset's movement but with the joystick. Therefore, I did not set up any body rotation functionality in this part.

Figure 5.5: Meerkat character reacting to the player's crouching movement. Crouching with the help of the IK system on the legs

Hands Tracking

Concerning the hand tracking and control mapping, I used the same process as for the legs in the previous sub-section. I used the IK rig I just set up to create two new goals, one for each hand. Along with these, I also created two new limb solvers, one for each arm. I then linked these solvers to their corresponding hand end goals and also set their root bones to be the shoulders of our meerkat. Even though meerkats' shoulders are more pushed in front of their bodies than us, the result was very satisfying and the arms reacted the exact same way the user's arms did (see figure 5.6 for the result).

Figure 5.6: Meerkat character reacting to the player's hand movements. The arm animation is made with the IK system

I also added a mapping of the user's hand rotation to our meerkat in the exact same way. As IK goals can be translated and rotated at the same time, this feature was simple to implement. The position and rotation of the hands are shared the same way the headset's position is shared with the IK rig, via the animation blueprint. The only difference was that the hands rotation was not correct at first as the IK rig handled rotation in bone space and not in world space. I had to invert and offset certain axes by some degrees, but it ended up working perfectly after a few attempts. See figure 5.7 for an image of the final animation blueprint system.

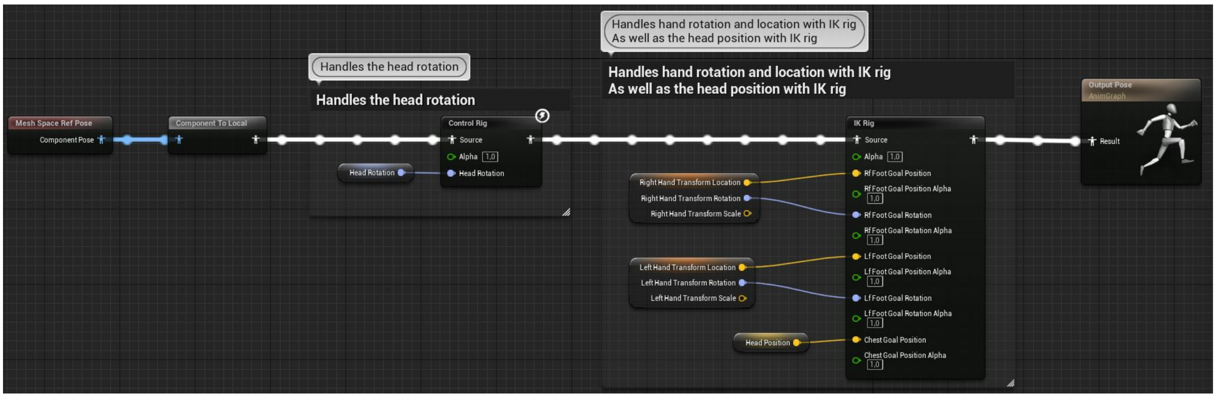

Figure 5.7: Final Animation Blueprint system. The execution starts with the control rig for the head rotation and then goes into the IK rig to handle the body position and hands transform

Puppeteering

In addition to the general body movements of our character, it has also been decided to add more subtle movements to some parts of its body. These additional movements will give more life to our character and make it less static. It will also allow for more expressiveness on the user's side. Being able to open its mouth or close its eyes is a good potential way of increasing IVBO. Influenced by puppeteering techniques, the objective is to be able to control these movements with the controllers and movements of our fingers. As a solution to implement this feature, we found out during the design phase that we could use blendshape animation concepts. Luckily, Unreal Engine enables the use of Blenshapes on its skeletal mesh characters, and the Meerkat 3D model we use already contains numerous quality blendshapes we can play with (see figure 5.8). In order to implement this feature, we first need to choose which keys of the Occulus Quest 2 controllers we will be able to map. To have full control over the blendshapes and a natural feeling, we need a way to gradually increase the coefficients of our blendshapes. Classic buttons are not suited for this purpose as they are either pressed or not pressed with no values in between the harder they get pressed. Luckily, the Occulus Quest 2 controllers have 4 trigger buttons we can use for this purpose, left and right grab buttons and left and right triggers. In Unreal Engine, in any Blueprints, we can get the value of these triggers, capped between 0.0 and 1.0, with the value varying depending on how much the triggers are pressed. Blendshape control is made inside an Animation Blueprint as well. Therefore, I simply had to use the same animation blueprint used in the previous sections to add this feature. Animations blueprints have their own event graph, where they can react to events and interactions. This event graph has a node specifically made to change a blendshape's coefficient. Therefore, simply linking the triggered amount to this node was enough to get the puppeteering effect desired. The mapping was done on the four selected blendshapes, opening the mouth, closing the eyes, moving the ears and breathing. A summary of the different key bindings is available in figure 5.12

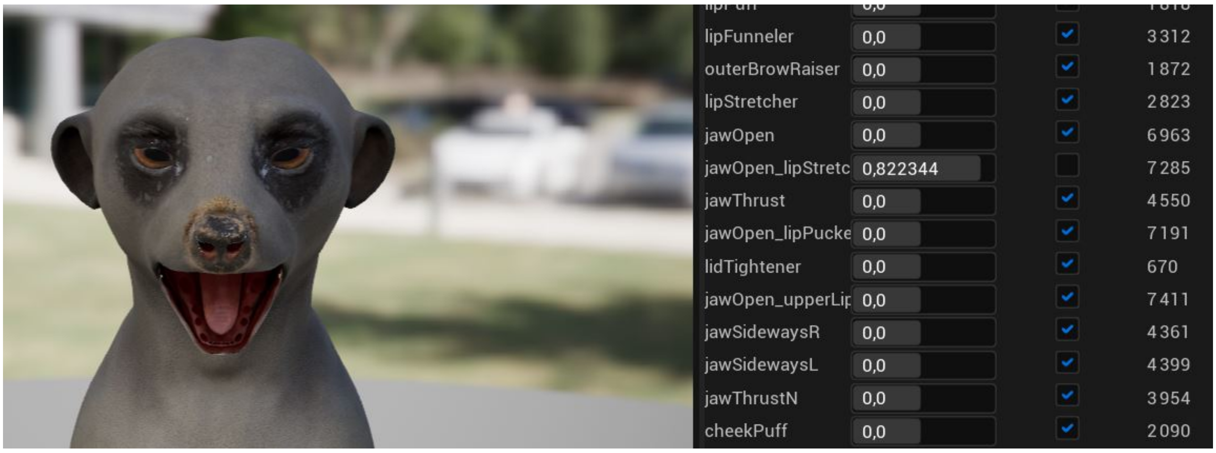

Figure 5.8: Image taken from the Unreal Engine interface showing the blendshapes available on our Meerkat 3D model. On the right, the blendshape list and their coefficients, and on the left, the meerkat model. The mouth open and eyes closed coefficients have been increased to 0.8 here

After implementing this feature, I ran into an unfortunate issue with the character's head. As this part of the meerkat's body was simply placed behind the camera with an offset, when some blendshape movements (opening the mouth and closing the eyes) were performed, they would show parts of the jaw and eyes on the screen. This is also when I realised that, when the user was turning its head, he could see the head of the meerkat on its side. This is also due to the offset applied. As the camera was always statically situated just in front of the meerkat's head, when turning around, it would stay in the same position. Thus, leading to this detached head vision when turning the head too much. As a solution to that, I simply needed to hide the character's head. Unreal Engine has a node in Blueprint that allows you to hide a bone and its children. But the problem was that, by doing so, the meerkat's head would also disappear from the mirror's reflection. Unreal Engine makes it possible to hide entire objects from reflection captures, but not bones and their hierarchy. Therefore, the only solution I found was to duplicate the meerkat's body, hide its head and its child hierarchy, and hide the previous meerkat character from the reflection capture. After mapping the whole movement system to this character too, the result was exactly what was wanted. The character in the real environment has now its head hidden, preventing the user from seeing it when turning, and the character reflected in the mirror has its head displayed, with blendshapes interaction activated.

5.2.4 Moving in the environment

Because we designed this application to be a static experience, meaning the user is not moving around or rotating while playing, we also needed to implement a way of moving and rotating in the scene. For the rotation system, we ended up relying on the joystick movement to turn our character around. As for the moving system, we decided to rely on a teleportation feature. Both these options were chosen for their simplicity, as a complete rotation and locomotion system relying on the user's HMD movement would have been hard to create with only 3 trackers. For the teleportation feature, Unreal Engine's VR template already comes with this exact functionality. I simply had to copy the code from their PAWN blueprint and tweak some small parts to adapt it to our meerkat character. This included changing the button used for teleportation as well as the origin of the teleportation ray. Having this movement feature is very interesting for immersion as it would allow the users to come closer to certain parts of the environment and navigate inside it. Being able to come closer to very tall parts of the scenery could create an even higher scale difference feeling, therefore potentially increasing virtual presence and IVBO.

Figure 5.9: Image of the teleportation aim trace when the player presses "A". When he releases it, the player is teleported to the centre of it

Along with being able to move, the ability to rotate the character's body was also necessary. Although our experience is static and does not allow the user to turn around completely, we still need a way of rotating inside the experience. Relying on the joystick movement to turn was straightforward. In the character's blueprint, I simply needed to get the value of the rotation of the left joystick via an event node, detect if this value is greater than a threshold (i.e. the deadzone), and then rotate the whole character in the direction of the joystick turn (see figure 5.10). The important thing here is to turn the whole PAWN and not only the meerkat character's model, or the camera will not turn as well. I also decided to turn with a snap motion and not a constant motion, as it was simpler and faster to turn around this way. These two movement functionalities might have a bad influence on IVBO and immersion, but unfortunately I did not manage to have the time to implement more advanced features for them.

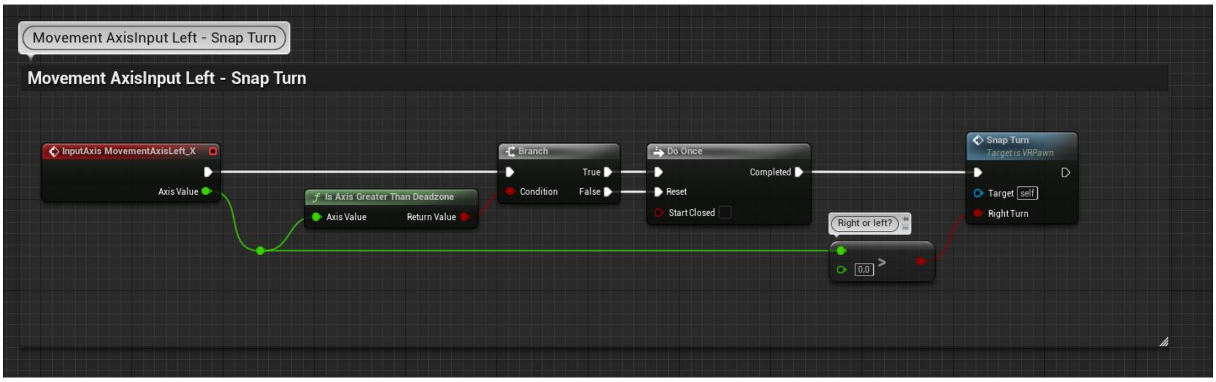

Figure 5.10: Image of the rotation system Blueprint. The node "Snap-Turn" is simply a function that turns the player and the character around in the given direction.

5.2.5 Grabbing objects

Adding some interaction with the environment was also an objective in this implementation. As an interactive system, the possibility to grab props from the environment was the one I personally preferred. It would add a bit of fun to the experience and allow the users to hold objects in their hands like in the real world. In order to implement this functionality, I used a very good tutorial from Unreal Engine's documentation pages and tweaked it specifically for our meerkat character use case. This feature was implemented by creating a whole new component, "*GrabComponent*" and making it interact with our PAWN character. This component is made to be placed inside any grabbable objects in the scene and would allow us to grab and release them. When the user presses the grab trigger, the PAWN class creates a detection sphere around the location where the grab has been executed. If one or more grabbable objects (i.e. has a GrabComponent) are detected in this sphere detector, they get added to a local array. This array is then scanned to find the closest grabbable object in it. Once this object is finally detected, if we do not have another object in this hand already, we simply attach this grabbable object to the correct "hand" bone of our character and disable its physics simulation. This results in the object being grabbed, and all its movements following the same ones as the hand bone, which is controlled by our own hand movement. When the trigger is released, the object is detached from the "hand" bone and simulation is enabled again. I added this feature to both hands and made sure that when an object is already held by one of them, it cannot be grabbed by the other one. As for the props that the user would be able to grab, I decided to use one of the 3D assets coming from Weta Digital's meerkat demo, an ostrich egg. Meerkats are carnivorous animals and mostly eat insects and eggs they find in the desert. Adding this as a prop would give the experience more realism and suit the animal's character. To implement it, I simply imported the 3D model to the project, activated its physics simulation, and added a "GrabComponent" component to it to make it grabbable. See figure 5.11 for the final result.

Figure 5.11: Image of the meerkat character grabbing an egg prop. The egg is added as a child of the "hand" bone to achieve this effect

To add a bit more interactivity and realism to the scene, I also activated the collisions between the eggs and the character's hand. This way, the player can also play with them without grabbing them. Moving them around and pushing them. See figure 5.12 below for a final summary of the different buttons of the controllers the player can interact with.

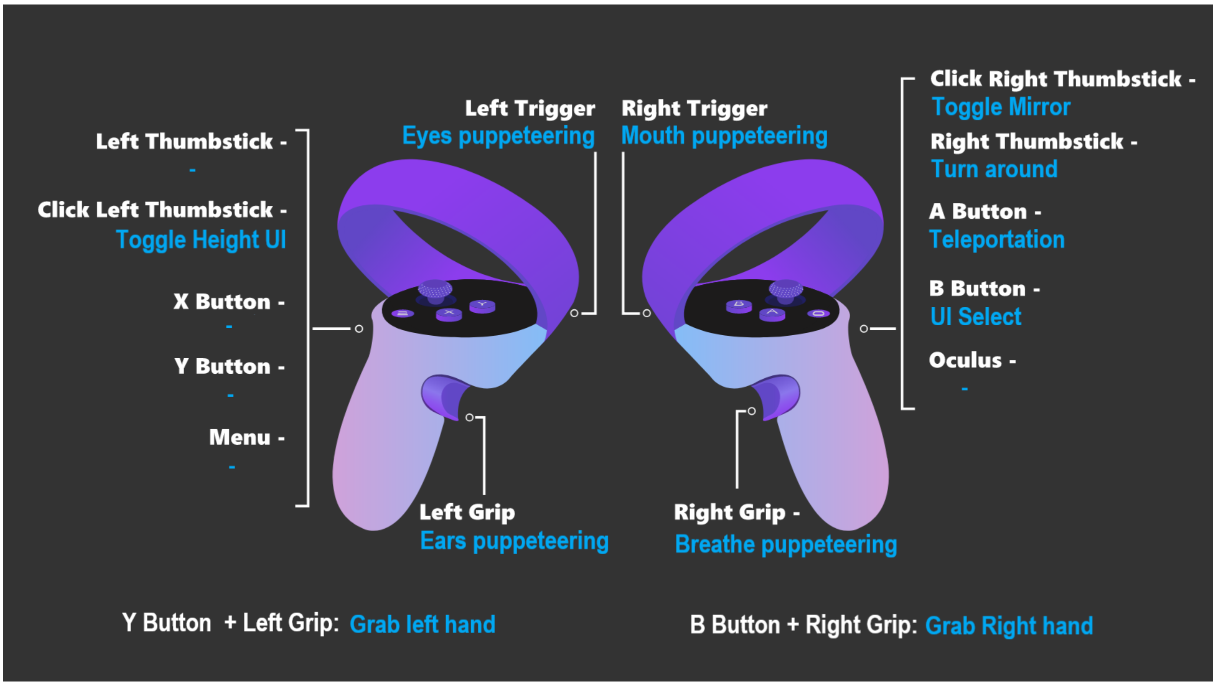

Figure 5.12: Key bindings of each button and triggers of the Occulus controllers

5.2.6 Users Height Management

After implementing these features, I tried to make a first small evaluation of the general embodiment feeling with a few people. While attempting to do so, I ran into a small problem. When other people tried the embodiment experience, the character model was always too large for them. It was constantly crouching even if they were standing straight and was almost going through the floor when they really did crouch. This problem originates from the way the whole character was set up. I used my morphology to set up the whole camera and embodiment system. However, by doing so, I hard-coded the offset and placement values I used to place the camera at the centre of the meerkat's head. These values were correct for me and my morphology, but not for someone smaller or taller. Therefore, I had to come up with a solution to that problem if I wanted to evaluate this experience with the opinions of some participants. It had to be suited to them as well. The solution adopted was to take into account the user's height and adapt the whole embodiment system to it. To implement this, I needed two different functionalities: a UI system to select the correct height and the body/offset scaling system.

For the UI system, I had to do some research on Unreal Engine's widget system as I was not familiar with it at all. I followed some tutorials and analysed the official documentation to get an idea of the best practices and started by creating the widget itself. The widget is composed of a 3D UI canvas floating in the air. This is usually the way UI is done in VR. It has three buttons: one for increasing the height, another one to decrease it, and a button to confirm (see Figure 5.13). When the user presses that last button, it changes the height value stored in the "*PlayerState*, a special blueprint made for storing player-related data. This height value will then be used by the whole PAWN blueprint to change the scale of the character. As for the interactivity, I calculate the direction in which the player is aiming at all times, and when he is aiming at the menu, I draw a line in this direction and display the pointer on the widget. The pointer follows the aim movement and does not leave the widget. When the player presses the "B" button on the right controller, an input event is fired to the widget he is aiming at. If it is a button, it receives the input and acts in consequence. The user also has the possibility of hiding this menu whenever he wants to by pressing the left joystick button. I added the same functionality to the mirror, but with a right joystick press. I wanted the users to be able to fully immerse themselves inside the environment if they wanted to. A mirror and a futuristic floating UI do not really have their place in an African desert.

Figure 5.13: The heigh modification UI menu. The player can increase or decrease the height value with the "+" and "-" buttons and confirm with the bottom one.

To implement the height matching feature, I decided to make all the placement and scaling coefficients dependent on that height value described previously. The PAWN blueprint has access to this value at any time in the game. Therefore, it could use it at any time to handle this functionality. Therefore, I simply replaced my previous hard-coded scaling and offset values by a coefficient multiplied by this value. By doing so, when the height value increases, all the other values increase accordingly. I just had to tweak the coefficients multiple times to get the perfect match, but it ended up working very well. The meerkat's scale is then dependent on this height value, as well as the offset used for the placement of the camera inside the head of the character. With this feature added, the meerkat's model now fits all the possible height mythologies. I tried it with a person as short as 1.50m and as tall as 1.90m and the result was always perfect.

5.3 Environment implementation

For the environment design, as mentioned in the previous chapter, my main inspiration was the Kalahari desert, situated in South Africa. Being meerkats' natural habitat, I really wanted it to look as close as possible to it. I did my own research on the different plants, types of rock and sand available there and started looking for free assets on the internet. Luckily for me, as Unreal Engine 5 comes with full support for Quixel's Megascans 3D assets, I was able to pick beautiful desert assets from their collection. Therefore, I used the Quixel Bridge application to get rocks, stones, cactus, and dead desert foliage assets. I also used the burrow asset from Weta Digital's demo project in the scene too for added realism. Although it had a very high polygon count, I still managed to include it in the scene without losing too much performance. I also decided to add other static meerkats to the environment, as meerkats are well known to live in large packs. Also, this way, our player would not feel too lonely in this vast environment. Finally, as a way of adding more life to it, I also played with Unreal Engine's foliage system. I added dead grass everywhere and especially in-between the rocks to hide any large gaps and fill the scene with more content. But I ended up needing to remove almost half of it because of performance issues. As the scene is rendered twice because of the mirror, the device could not handle this much detail while having high enough FPS. See figures 5.14 and 5.15 for the result.

Figure 5.14: Our desert environment. Cactus, rocks, foliage, burrow

Figure 5.15: Our desert environment seen from another perspective. We can see two of the 5 static meerkats I added to the environment on the right and far in the distance

When implementing this environment, I also took great care to respect the scale difference discussed in the design and background phase. As seen in previous research, it can have great effects on IVBO and virtual presence. As meerkats are very small animals, I scaled the rocks and the cactus to be very tall compared to our character. I also scaled some of the grass, but not all of it, or else we would not be able to see anything in the scene. As a result, we can really feel that feeling of being very small when we look at these objects up close. See figure 5.16 for an example.

Figure 5.16: One of the giant cactus seen from the bottom.

5.4 Sound implementation

The final part of the implementation process was the addition of ambient and interactive sound to the experience. I found a free desert ambient sound online and a recording of a meerkat pack cry and added them to the scene. As Unreal Engine allows for audio spatialisation, I emitted the cry ambient sound from the bottom of the burrow and the static meerkats I placed in the environment. As for the desert sound, I simply did not activate spatialisation for it, so it could be heard from anywhere. For the interactive audio, I used an audio editing software to isolate only two different cries and played them when the puppeteering actions were performed. One of the two sounds (a cry) is triggered when the meerkat opens its mouth, and the other one (a moan) is triggered when it is breathing. I randomised the pitch of both these sounds to add more variety to them and allow for fast triggering without any unnatural effects. To play a sound interactively, I used the PAWN blueprint once again. When it receives one of the two puppeteering events, it plays the correct sound at the meerkat's location with a randomised pitch. See figure 5.17 for an image of this feature in the blueprint.

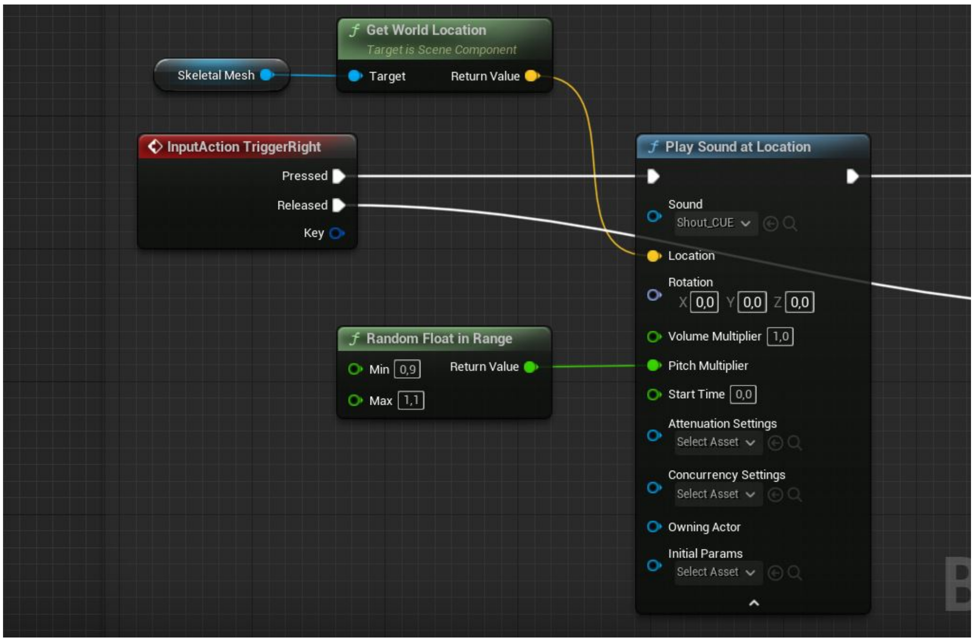

Figure 5.17: How an interactive sound is triggered in the PAWN blueprint. When the trigger is pressed, the sound is fired at the meerkat's location with a random pitch

Chapter 6

Results & Evaluation

This chapter deals with the results and evaluation of the implementation. The results section presents the final application and the different functionalities implemented in the previous chapter. As for the evaluation, it is presented in two phases. An evaluation of the implementation process and its final result from a technical point of view and a user experiment carried out with anonymous participants. A discussion on the limitations of this implementation is also presented at the end of the chapter.

6.1 Results

The final result of this implementation is a VR application focused on the embodiment of a Meerkat virtual avatar. It features all the functionalities listed during the design phase and is compatible with the Meta Quest 2 VR headset. It runs at an average of 30 FPS and uses the basic Meta Quest 2 hardware setup. It does not need any extra trackers or components to work. This application is meant to be an immersive virtual experience where the user can completely embody a meerkat character in their natural environment. It makes use of all the concepts studied in the background chapter and has been heavily inspired by Krekhov et al. (2019b) and Andreasen et al. (2019)'s papers. A list of all the features implemented in the final application is available in the table 6.1 below. They have been separated into three different categories as per the design chapter's layout. The "embodiment features" category represents all the functionalities influencing the control and embodiment of our character. And the "environment features" and "sound features" categories regroup all the environment and sound-related features. All the objectives set during the design phase have been reached and all the features implemented.

Embodiment features	Environment features	Sound features
Realistic Meerkat avatar	Realistic and captivating environment	Interactive sound effects
First-person perspective	Respect of meerkats' natural environment	Ambient sound effects
Control of the hands and the head movements	Interactive environment parts with a grab system	
Inverse kinematics system for the legs and and the arms	Scaling of the environment As seen by a real meerkat	
Mirror effect for enhanced IVBO		
Puppeteering of some body parts		
Teleportation system		
Body rotation system		
User height management system with a VR UI		

Table 6.1: All the features implemented in the final application, separated in three categories, embodiment, environment and sound effects.

Concerning the reusability of these features for other projects, for instance, creating the same experience but with another animal, it is estimated that 50% of them can be transferred without any modifications. The environment and sound features are easily transferable without modification, as well as the mirror effect, the teleportation and rotation system, and the height management system. But features involving the animal character itself, like the inverse kinematics and the first person perspective, would require a specific adaptation for each animal. A live video demonstration of the final application has been uploaded to YouTube and presents all the features implemented one by one. It has been recorded from the point of view of the user, and represents exactly what he sees while using the application. In the beginning, it also offers a second point of view with a live video of me playing the game with the headset on. The video can be accessed via this link:

`https://youtu.be/RNB92wA2iAU`

6.2 Evaluation

The evaluation of the implementation has been done in two different ways. The first one is more of a technical evaluation of the application and its development process, as well as its compatibility with a basic 3-tracker VR setup. The second one is a real user study carried out on anonymous participants based on a real academic IVBO questionnaire.

6.2.1 Personal & Technical evaluation

On the technical side, the application does very well and includes all the components desired. It runs on the Occulus Quest 2 and only requires the basic VR setup to work, which was one of the objectives of this project. On the performance side, it does not run as smoothly as initially wanted, but this is mainly due to the mirror system. As the scene is rendered twice, once normally and another time for the mirror reflection, any objects in the scene need to be rendered twice as well. The burrow and the eggs, which have a high polygon count, are the main culprits there. Rendering them twice every frame is very demanding, and as "nanite", the new Unreal Engine 5 feature that allows us to include high polygon models into scenes without performance issues, is not available for VR, I could not prevent it from happening. LODs were also not available for these assets. Secondly, the foliage was also one of the reasons for these performance issues. At first, I added too much of it into the scene, making the application very slow. I then decided to only keep the right amount to make the FPS higher than 30, and the problem was fixed. I also implemented the toggling of the mirror feature. When the user does not look at the mirror, it automatically toggles off, greatly increasing the FPS over 50. The same applies when the player toggles it manually. Technically, the implementation uses many good concepts like Inverse Kinematics, Control mapping, puppeteering, VR development and is very interesting from this point of view. It merges many different areas into a single experience and, even if it is not perfect, it has very nice potential on the subject.

During the design phase, we introduced the hypothesis that implementing this project with a basic 3-tracker VR setup and Unreal Engine 5 would be difficult. We also set one of the goals of this project to be an experiment on the viability of using a 3-tracker VR system with Unreal Engine 5 for embodiment experiences. After implementing everything, I believe that this first hypothesis can be validated as this embodiment experience would work better with four or six trackers. Also, using Unity would have been more straightforward and less complicated than Unreal Engine as it is more suited for VR development. Implementing this project would have also been faster. This hypothesis was therefore correct, but creating such an application with this setup and 3D engine is totally possible and could result in an excellent result with more work and knowledge about them. As for the goal, I also believe it has been reached as we created a very nice embodiment experience with this gear and software.

On a personal note, I believe that this project is a success overall. The embodiment is not perfect and would need some more work (see Future Work section), but it manages to merge many great concepts into a single experience, and this is what I like about this result. It also manages to do that with a very basic VR setup and using Unreal Engine as a development tool, which is the hardest engine to work with on VR development, at least compared to Unity. The embodiment part works very well and really feels like an embodiment. Even though I would have liked to add a better locomotion system with animation blended with IK for the legs and a better rotation system. The environment is realistic and appropriate for Meerkats, and the sound design adds a very nice touch. Although it is my personal opinion, it comes very close to the one deduced from the user study realised and presented in the next sub-section.

6.2.2 User study

Before conducting the user study, I carried out a first user trial in the middle of the implementation phase to help identify problems and gather some first impressions. This experiment was carried out just after finishing implementing the core of the embodiment system. The environment was not yet created and neither was the sound system. The objective was not to gather answers and opinions via a questionnaire or anything, it was simply to get some initial feedback about it overall. Two people were asked to try the first embodiment prototype and explain what they felt. Both of them felt like the absence of a real natural environment was not really immersive. By that time, I was still using the basic Unreal Engine props for prototyping. They also felt like the camera's first-person perspective was not enjoyable as they would see the head of the meerkat on their side when they turned their heads. This was because I had not yet implemented the invisible head feature. They also said that not being able to move around and rotate was not pleasant, but they really liked the fact that they could see the meerkat's body when they looked down. They also really liked the whole head/hand movement and rotation system, especially the IK leg, which appeared very realistic to them. This first partial experiment helped me a lot in identifying the first issues that appeared during the implementation. It also allowed me to validate the importance of some features I designed but have not yet implemented, like the environment, the head invisibility, and the teleportation system.

As for the final user study, it has been carried out after the whole implementation. Its goal was to evaluate the final implementation from an external, non-biased point of view and validate our hypotheses. In our case, I conducted this user study by performing a user trial on 7 anonymous people and collecting their opinions on it via a questionnaire given after the experiment. The users had a chance to try the embodiment experience for 5 minutes with the VR headset, a typical duration for IVBO studies (Tsakiris and Haggard (2005)). The size of the meerkat characters was adjusted to their real size with the scaling system, and they were introduced to the controls just before the experiment. They were also introduced to the concept of IVBO and embodiment in VR, as well as the goals of this experiment. This study was carried out on 7 subjects, of which 4 were women, aged 22 to 50. They all knew what VR was beforehand, but only 3 of them had already tried a VR headset before. 5 of them were familiar with the concepts of video games and reported playing occasionally.

After the experiment, the participants were asked to answer a questionnaire of 18 questions about their experience. This questionnaire is meant to measure IVBO and has been inspired by the "Alpha-IVBO" questionnaire created by Roth et al. (2017). There is no general procedure for measuring IVBO, but this questionnaire is widely used in the field and has been validated multiple times. Especially by Krekhov et al. (2019b), they customised and applied this questionnaire after an animal embodiment experiment to measure IVBO. They also verified its reliability and found out it worked well for non-humanoid characters as well. This questionnaire is rated with a 7 point "Likert" scale going from 0 (totally disagree) to 6 (totally agree). It aims at capturing the three dimensions of IVBO that Roth et al. defined in the same paper, acceptance, control, and change.

- **Acceptance** is about "self-attribution" and "owning of the virtual body" and is defined by statements like: *I felt as if the body parts I looked upon were my body parts*
- **Control** focuses on the correctness of the feedback and feeling of control with statements like: *I felt as if I was causing the movement I saw in the virtual mirror*
- Finally, **change** is about "self-perception" and is very present when the avatar is very different from the user. For instance, one of the statements of this category is *At a time during the experiment I felt as if my real body changed in its shape, and/or texture.*

Our questionnaire is very similar to this one except that a few irrelevant questions about human avatars have been replaced or removed, and 6 new questions in a new category called "Other" have been added. With it, I wanted to try to understand how important each feature group is in inducing IVBO in this embodiment experience. The list of all the questions and their categories is available in the table 6.2 below.

Category	Feature	Question
	Q1	I felt as if the body I saw in the virtual mirror might be my body.
Acceptance	Q2	I felt as if the body parts I looked upon where my body parts.
	Q3	The virtual body I saw was not humanlike.
	Q4	The movements I saw in the virtual mirror seemed to be my own movements.
Control	Q5	I enjoyed controlling the virtual body I saw in the virtual mirror.
	Q6	I felt as if I was controlling the movement I saw in the virtual mirror.
	Q7	I felt as if I was causing the movement I saw in the virtual mirror.
	Q8	The illusion of owning a different body than my real one was very strong during the experience.
	Q9	At a time during the experiment, I felt as if my real body changed in its shape, and/or texture.
Change	Q10	During or after the task, I felt the need to check that my body does really still look like me.
	Q11	I felt an after-effect as if my body had become taller/smaller.
	Q12	I felt an after-effect as if my body had become lighter/heavier.
	Q13	The puppeteering feature was important in the embodiment of the animal.
	Q14	The environment design was important in the embodiment of the animal.
	Q15	The sound effects were important in the embodiment of the animal.
Other	Q16	The movement synchronisation was important in the embodiment of the animal.
	Q17	The interactions with the environment was important in the embodiment of the animal.
	Q18	The mirror was important in the embodiment of the animal.

Table 6.2: The 18 questions asked to the 7 participants of my evaluation experiment. With four sub-components, Acceptance, Control and Change to measure IVBO, and the Other category to define the most influential parts of the experiment on IVBO. The grades go from 0 to 6 (from strongly disagree to strongly agree)

The same grading system as the "Alpha-IVBO" one has been used for our questionnaire. The results are presented as the average of the 7 participants' grades for each question. Questions in the Acceptance category have grades around 3.40, except for Q3, which has 4.5, with an average of 3.81. We can observe that the two first questions about thinking that the animal body was their body were mildly disagreed with. The results of Q3 were agreed with a lot, which is a good thing, as this is what we are trying to achieve, remove the human traits from the meerkat's character. Questions in the Control category have very similar grades, ranging from 5.15 to 5.57, with an average of 5.39. These grades are very good and demonstrate that the control part of the embodiment was very successful, even with only 3 trackers. The enjoyment statement was also strongly agreed with, which is a very good thing to notice. Questions in the Change category also have a similar trend ranging from 0.8 to 2.71, except for Q8 with 4.0 and with an average of 2.29. The fact that Q8 was mostly agreed with is a very good sign of a successful embodiment experience. The rest of the Change questions are very specific and push the embodiment specifications to the maximum. People did not feel any effects on their real bodies, which is very hard to achieve and may require a more realistic VR experience. Finally, questions in the Other category also have very similar results, ranging from 4.57 to 6.00 with an average of 5.36. These results show that all the features we designed as being important in the embodiment experience have been proven to be important for the users as well.

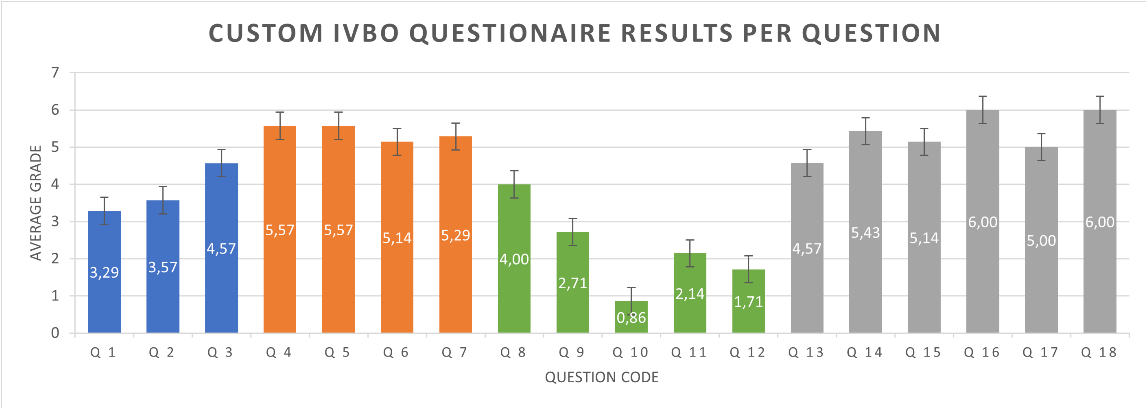

Figure 6.1: Results of the user study. The grade goes from 0 to 6 and each of the 18 questions was covered. These grades are the average of the grades given by 7 participants

It also shows that the movement synchronisation, the environment design, and the mirror were the most important features of the experience. This last category directly answers our second hypothesis defined in the design phase: "*I also project that the mirror, the environment and sound design, and the hands and head mapping systems will be very important factors for inducing IVBO in our experience.*". All of these aspects were indeed important in the overall experience, most importantly in the feeling of embodiment. Our last hypothesis, "*I also project that participants will at least feel partial effects of IVBO and feel like embodying the meerkat.*" can also be validated as the participants have globally felt at least some of the key feelings of IVBO. The results are presented in the figures 6.1 and 6.2 above and below as bar charts of the average for each question and the average per category.

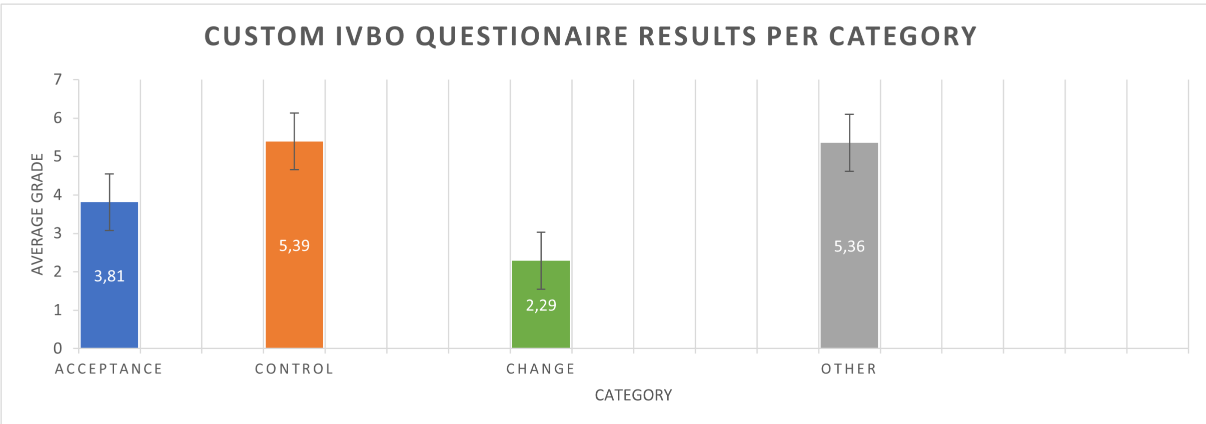

Figure 6.2: These grades are the average per category of the grades given by 7 participants. The grade goes from 0 to 6 and each of the 18 questions was covered

As a last evaluation, we can also compare these results with the results obtained by Krekhov et al. (2019b), who applied this questionnaire after an animal embodiment experiment. As they carried out many experiments on different body mappings, we will base our comparison on their best half-body experiment, the tiger experiment. One thing to remember however, is that they used a fourth tracker on the waist of their participants for this experiment, something that we did not have in our case. Their Acceptance average is around 3.50 while ours is a bit better, with 3.81. Their Control average is around 4.15 while ours is 5.39. It even beats their best Control average of 5.10 for their full-body tracking for a Bat character. Finally, their Change is around 1.90 while ours is 2.29. As we can see by this comparison, our scores follow the same trend as theirs and are even a bit better. This clearly shows that our project result is very similar to theirs and that it has been correctly carried out. It also shows that Control is the easiest grade to get while Change is very difficult to increase for everyone. With scores this close to each other, we can say that our experiment was a success and that creating an embodiment experience with only 3 trackers and Unreal Engine is totally possible, albeit difficult.

6.3 Limitations

While all the goals of this project have been reached and the final result is satisfactory enough, answering all the hypotheses, a few limitations have been observed in this work:

- This experience is a static experiment, meaning that the users cannot move around while they are in the environment. This is due to our implementation of IK on the legs and our design choice for the rotation of the body. The legs are always static in the experience and are not animated at all. This is mainly because we do not have any trackers to know when the user takes a step and also because we did not create a locomotion system that would make the meerkat walk. Instead, a teleportation system has been chosen for time-saving reasons as it was already provided by the VR template. Because of that, the users have to use the joystick to turn around and the teleportation system to move around the scene, causing a potential decrease in IVBO in my opinion.

- The meerkat character does not have any fur on it and could be more realistic. This is due to the import issues encountered when transferring the assets from one project to another. All the materials used by the Weta Digital's project 3D assets were empty and I had to create them again by hand. As I do not have any experience with materials in Unreal Engine and that they were very complicated, the result could have been better with the original ones. The fact that the character does not have fur is also due to the limitations of our setup. A standalone VR headset is very limited in performance and cannot handle grooming efficiently. Even though I tried implementing it for a long time, it kept either not showing or showing incorrectly while plunging the FPS count. With proper materials and fur, the meerkat would have been way more realistic and so would have been the experience overall.

- The application is not as fluid as it could be. Even though we reached a steady 25-30 FPS count, the limitations of our headsets made it hard to create the environment as realistically as possible. The rocks have a very low poly count and could look better and the same applies to the vegetation. The burrow has too many polygons and the eggs props as well, bringing the FPS count down. All of these issues are mainly due to the mirror effect that forces us to render the scene twice.

- The hand rotations are not perfect. When the user rotates their hands too much, the meerkat's hands start to fold in a very unnatural way. Adding constraints to this rotation could solve this problem. This is also due to the implementation of the IK on the arms, which could have been done better in my opinion. If I had to do it again, I would work more on this aspect to have a real arm feeling.

- The experience includes only one type of animal, and this animal is standing on its two feet. Very few animals stand on their two feet all the time. And taking a standing meerkat for this experiment was actually simpler than other types of animals as it vaguely resembles the morphology of a human. This experiment should be tried with more animals and also with animals standing on their four legs. Working on going from two-legged to four-legged positions in VR could also be very interesting.

- Finally, the user study did not have many participants. 7 people is enough, but more opinions would have been better for the results.

Chapter 7

Conclusions

This chapter provides a summary of the contributions of this project, followed by some possibilities for future work on the same topic, and ends with the conclusion of the whole project.

7.1 Contribution

This project has shown that creating a VR animal embodiment experience with a very basic VR setup composed of a standalone headset and two controllers, i.e. a three-tracker system, and Unreal Engine was possible and could give great results. All the important concepts of embodiment and virtual presence are respected, and it also uses fundamental animation ideas like inverse kinematics, blendshapes, and movement synchronisation. It also gave an overview of the design and implementation process of such an application by providing a list of all the important concepts to implement for a successful IVBO experience. People can use this paper in the future to get an idea of the important steps to follow and technologies to use for future applications. This project also provides a great summary of the research conducted on IVBO, avatar embodiment, and especially non-humanoid character embodiment in VR. As very little research focuses on this subject (only 3 or 4 papers focused on VR animal embodiment), it also provides a new point of view and implementation of the experience on the subject. It also proved that using a very basic setup for embodiment was possible. As this kind of setup is the most widespread among the public, it helps show that it can be possible to use it, popularising its use a bit more. Finally, with its user study, this project also proves its facts with real opinions and provides a new practical application of the Alpha-IVBO questionnaire on the animal embodiment subject. Only one paper has used this template for such experiments, and this project could be considered the second one.

7.2 Future Work

This project can be improved and extended in several ways. Either by directly improving some existing features or by implementing new ones. Here are some ideas of future work that could help directly improve the project as it is or help new projects on this topic:

Full-Body IK implementation: As explained in the limitation section of the previous chapter, the IK system the character uses is not totally complete. Unfortunately, only the four limbs have full IK functionality, not the entire body. Implementing a full-body IK system would be more realistic and allow for rotation of the entire body. While it is very difficult to implement with a 3-tracker system, focusing on finding a solution to handle this rotation correctly would be extremely beneficial to the overall IVBO experience. This would allow the user to turn around with the character's body following their movement while still being able to rotate their head independently. Finding a solution for whole-body rotation while still being able to rotate the head and implementing a full-body IK system for the character would be an exciting future task.

More realistic locomotion system: This aspect comes very close to the previous one but focuses on the locomotion system. As for now, the user uses a teleportation system to move around the environment. This is not very realistic and could impair IVBO inducement. As a very nice improvement, we could create a more realistic locomotion system relying on the joystick movements and inverse kinematics of the legs. By blending a walking animation with the IK system of the legs and moving the character according to the direction of the joystick, I believe that it would improve the realism of the experience. Teleportation was adopted because it was easy to implement and it is known to be suited for VR experiences and lower motion sickness. But for embodiment experience, a real locomotion system would be preferable. The best would be to have a locomotion system relying only on the walking movements of the user, but this is very hard, even maybe impossible with a 3-tracker setup (related to the previous point).

Different types of animals: For now, this experiment only allows for the embodiment of one single type of animal, a Meerkat. A good future work improvement would be to add more types of animals to the experiment, especially animals with different morphologies. Four-legged animals would be interesting to work with, just like in Krekhov et al. (2019b)'s experiment.

Going from 2 legs to 4 legs: In this project, the meerkat character is always in the same position, standing on its two legs. Few animals, especially Meerkats, are known to constantly walk on their two back legs. Some animals cannot even be in this position. Therefore, working on a way to go from this two-legged position to a four-legged position would be very interesting. The transition could be done by analysing the position of the headset as well as the hands and seeing if they are both close to the ground. If so, start the transition and rotate the entire body and ensure that the IK system still works well in this position.

More specific interactions: This point was also observed by Krekhov et al. (2019b) in their research. Their participants expressed that they would have liked to be able to perform actions specific to each animal. For example, with a bird, being able to fly, or with a meerkat, why not be able to dig into the ground. These animal-specific actions could greatly increase IVBO and would be a great addition to the project. For a meerkat, it could be digging, eating bugs, or climbing for example.

7.3 Conclusion

In this project, we successfully created a VR animal embodiment experience with a 3-tracker system and Unreal Engine and evaluated the resulting application with a user study based on an IVBO academic questionnaire, as well as comparing these results to previous work on the same subject. We proved that it was possible to create such an experience with a basic VR setup and Unreal Engine and that it could give great results. We also validated all our hypotheses and gave an overview of the application design and implementation process. With all these results, I believe that this project was a success, even though it could be improved in some aspects.

Bibliography

Ahn, S. J. G., Bostick, J., Ogle, E., Nowak, K. L., McGillicuddy, K. T., and Bailenson, J. N. (2016). Experiencing Nature: Embodying Animals in Immersive Virtual Environments Increases Inclusion of Nature in Self and Involvement with Nature. *Journal of Computer-Mediated Communication*, 21(6):399–419.

Andreasen, A., Nilsson, N. C., Zovnercuka, J., Geronazzo, M., and Serafin, S. (2019). What is it like to be a virtual bat? In Brooks, A. L., Brooks, E., and Sylla, C., editors, *Interactivity, Game Creation, Design, Learning, and Innovation*, pages 532–537, Cham. Springer International Publishing.

Aristidou, A., Lasenby, J., Chrysanthou, Y., and Shamir, A. (2018). Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 37:35–58.

Banakou, D., Groten, R., and Slater, M. (2013). Illusory ownership of a virtual child body causes overestimation of object sizes and implicit attitude changes. *Proceedings of the National Academy of Sciences*, 110(31):12846–12851.

Berthelot, R., Arnaldi, B., and Gouranton, V. (2016). Virtual reality rehearsals for acting with visual effects. pages 1–8.

Botvinick, M. and Cohen, J. (1998). Rubber hands 'feel'touch that eyes see. *Nature*, 391(6669):756–756.

Buss, S. R. (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16.

Caserman, P., Achenbach, P., and Göbel, S. (2019). Analysis of inverse kinematics solutions for full-body reconstruction in virtual reality. In *2019 IEEE 7th International Conference on Serious Games and Applications for Health (SeGAH)*, pages 1–8.

Ehrsson, H. H. (2009). How many arms make a pair? perceptual illusion of having an additional limb. *Perception*, 38(2):310–312.

Epics Games (1991). Unreal Engine 5. `https://www.unrealengine.com/en-US/unreal-engine-5`. Accessed: 2022-08-11.

Gillinson, M. (2022). No strings attached: the new wave of puppetry storming london's west end. *The Guardian*.

Guterstam, A., Petkova, V. I., and Ehrsson, H. H. (2011). The illusion of owning a third arm. *PloS one*, 6(2):e17208.

Husinsky, M. and Bruckner, F. (2018). Virtual stage: Interactive puppeteering in mixed reality. In *2018 IEEE 1st Workshop on Animation in Virtual and Augmented Environments (ANIVAE)*, pages 1–7.

Jo, D., Kim, K., Welch, G. F., Jeon, W., Kim, Y., Kim, K.-H., and Kim, G. J. (2017). The impact of avatar-owner visual similarity on body ownership in immersive virtual reality. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, pages 1–2.

Jun, J., Jung, M., Kim, S.-Y., and Kim, K. K. (2018). Full-body ownership illusion can change our emotion. CHI '18, page 1–11, New York, NY, USA. Association for Computing Machinery.

Kammerlander, R. (2020). Collaborative acting in virtual reality: Character embodiment for improving actor's immersion and minimizing scale mismatches for motion capture performances.

Kammerlander, R. K., Pereira, A., and Alexanderson, S. (2021). Using virtual reality to support acting in motion capture with differently scaled characters. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 402–410.

Kilteni, K., Normand, J.-M., Sanchez-Vives, M. V., and Slater, M. (2012). Extending body space in immersive virtual reality: a very long arm illusion. *PloS one*, 7(7):e40867.

Krekhov, A., Cmentowski, S., Emmerich, K., and Krüger, J. (2019a). Beyond human: Animals as an escape from stereotype avatars in virtual reality games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '19, page 439–451, New York, NY, USA. Association for Computing Machinery.

Krekhov, A., Cmentowski, S., and Krüger, J. (2019b). The illusion of animal body ownership and its potential for virtual reality games. In *2019 IEEE Conference on Games (CoG)*, pages 1–8.

Lenggenhager, B., Tadi, T., Metzinger, T., and Blanke, O. (2007). Video ergo sum: manipulating bodily self-consciousness. *Science*, 317(5841):1096–1099.

Lewis, J. P., Anjyo, K., Rhee, T., Zhang, M., Pighin, F. H., and Deng, Z. (2014). Practice and theory of blendshape facial models. *Eurographics (State of the Art Reports)*, 1(8):2.

Lin, L. and Jörg, S. (2016). Need a hand? how appearance affects the virtual hand illusion. In *Proceedings of the ACM symposium on applied perception*, pages 69–76.

Lombard, M. and Ditton, T. (1997). At the Heart of It All: The Concept of Presence. *Journal of Computer-Mediated Communication*, 3(2). JCMC321.

Lugrin, J.-L., Latt, J., and Latoschik, M. E. (2015). Anthropomorphism and illusion of virtual body ownership. *ICAT-EGVE*, 15:1–8.

Lugrin, J.-L., Obremski, D., Roth, D., and Latoschik, M. E. (2016a). Audio feedback and illusion of virtual body ownership in mixed reality. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, page 309–310, New York, NY, USA. Association for Computing Machinery.

Lugrin, J.-L., Polyschev, I., Roth, D., and Latoschik, M. E. (2016b). Avatar anthropomorphism and acrophobia. VRST '16, page 315–316, New York, NY, USA. Association for Computing Machinery.

Maselli, A. and Slater, M. (2013). The building blocks of the full body ownership illusion. *Frontiers in human neuroscience*, 7:83.

Medeiros, D., dos Anjos, R. K., Mendes, D., Pereira, J. a. M., Raposo, A., and Jorge, J. (2018). Keep my head on my shoulders! why third-person is bad for navigation in vr. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, New York, NY, USA. Association for Computing Machinery.

Muller, D. A., van Kessel, C. R., and Janssen, S. (2017). Through pink and blue glasses: Designing a dispositional empathy game using gender stereotypes and virtual reality. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '17 Extended Abstracts, page 599–605, New York, NY, USA. Association for Computing Machinery.

Mystakidis, S. (2022). Metaverse. *Encyclopedia*, 2(1):486–497.

Nordahl, R. and Nilsson, N. (2014). *The Sound of Being There: Presence and Interactive Audio in Immersive Virtual Reality*. Oxford Handbooks. Oxford University Press, United Kingdom.

Normand, J.-M., Spanlang, B., Tecchia, F., Carrozzino, M., Swapp, D., and Slater, M. (2012). Full body acting rehearsal in a networked virtual environment — a case study. *Presence*, 21(2):229–243.

Occulus (2012). Meta Quest 2. `https://store.facebook.com/ie/quest/products/quest-2/`. Accessed: 2022-08-11.

Oyanagi, A. and Ohmura, R. (2019). Transformation to a bird: overcoming the height of fear by inducing the proteus effect of the bird avatar. pages 145–149.

Parger, M., Mueller, J. H., Schmalstieg, D., and Steinberger, M. (2018). Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, New York, NY, USA. Association for Computing Machinery.

Parsons, S. and Mitchell, P. (2002). The potential of virtual reality in social skills training for people with autistic spectrum disorders. *Journal of Intellectual Disability Research*, 46(5):430–443.

Peck, T., Seinfeld, S., Aglioti, S., and Slater, M. (2013). Putting yourself in the skin of a black avatar reduces implicit racial bias. *Consciousness and cognition*, 22:779–787.

Perez-Marcos, D., Sanchez-Vives, M. V., and Slater, M. (2011). Is my hand connected to my body? the impact of body continuity and arm alignment on the virtual hand illusion. *Cognitive Neurodynamics*, 6(4):295–305.

Petkova, V. I. and Ehrsson, H. H. (2008). If i were you: perceptual illusion of body swapping. *PloS one*, 3(12):e3832.

pterosaurheresies (2021). Meerkat skeleton blog. `https://pterosaurheresies.wordpress.com/2021/07/18/the-meerkat-is-an-african-raccoon/`. Accessed: 2022-08-11.

Rhodin, H., Tompkin, J., Kim, K. I., de Aguiar, E., Pfister, H., Seidel, H.-P., and Theobalt, C. (2015). Generalizing wave gestures from sparse examples for real-time character control. *ACM Trans. Graph.*, 34(6).

Riva, G. (2005). Virtual reality in psychotherapy: Review. *CyberPsychology & Behavior*, 8(3):220–230. PMID: 15971972.

Riva, G., Waterworth, J., and Murray, D. (2014). *Interacting with Presence: HCI and the Sense of Presence in Computer-mediated Environments*. Walter de Gruyter GmbH & Co KG.

Roth, D., Lugrin, J.-L., Latoschik, M. E., and Huber, S. (2017). Alpha ivbo-construction of a scale to measure the illusion of virtual body ownership. In *Proceedings of the 2017 CHI conference extended abstracts on human factors in computing systems*, pages 2875–2883.

Sanchez-Vives, M. V., Spanlang, B., Frisoli, A., Bergamasco, M., and Slater, M. (2010). Virtual hand illusion induced by visuomotor correlations. *PLOS ONE*, 5(4):1–6.

Serafin, S. and Serafin, G. (2004). Sound design to enhance presence in photorealistic virtual reality. In *ICAD*.

Shin, H. J., Lee, J., Shin, S. Y., and Gleicher, M. (2001). Computer puppetry: An importance-based approach. *ACM Trans. Graph.*, 20(2):67–94.

Sikström, E., de Götzen, A., and Serafin, S. (2014). The role of sound in the sensation of ownership of a pair of virtual wings in immersive vr. AM '14, New York, NY, USA. Association for Computing Machinery.

Sikström, E., de Götzen, A., and Serafin, S. (2015). Wings and flying in immersive vr — controller type, sound effects and experienced ownership and agency. In *2015 IEEE Virtual Reality (VR)*, pages 281–282.

Slater, M., Howell, J., Steed, A., Pertaub, D.-P., and Garau, M. (2000). Acting in virtual reality. In *Proceedings of the third international conference on Collaborative virtual environments*, pages 103–110.

Slater, M., Perez-Marcos, D., Ehrsson, H., and Sanchez-Vives, M. (2009). Inducing illusory ownership of a virtual body. *Frontiers in neuroscience*, 3:214–20.

Slater, M., Pérez Marcos, D., Ehrsson, H., and Sanchez-Vives, M. V. (2008). Towards a digital body: the virtual arm illusion. *Frontiers in human neuroscience*, page 6.

Slater, M., Spanlang, B., Sanchez-Vives, M. V., and Blanke, O. (2010). First person experience of body transfer in virtual reality. *PLOS ONE*, 5(5):1–9.

Slater, M. and Steed, A. (2000). A virtual presence counter. *Presence*, 9(5):413–434.

Slater, M., Usoh, M., and Steed, A. (1995). Taking steps: The influence of a walking technique on presence in virtual reality. *ACM Trans. Comput.-Hum. Interact.*, 2(3):201–219.

Steptoe, W., Steed, A., and Slater, M. (2013). Human tails: Ownership and control of extended humanoid avatars. *IEEE transactions on visualization and computer graphics*, 19:583–90.

Sturman, D. (1998). Computer puppetry. *IEEE Computer Graphics and Applications*, 18(1):38–45.

Sutherland, I. E. (1965). The ultimate display. In *Proceedings of the Congress of the Internation Federation of Information Processing (IFIP)*, volume volume 2, pages 506–508.

Tanaka, N. and Takagi, H. (2004). Virtual reality environment design of managing both presence and virtual reality sickness. *Journal of PHYSIOLOGICAL ANTHROPOLOGY and Applied Human Science*, 23(6):313–317.

Tsakiris, M. and Haggard, P. (2005). The rubber hand illusion revisited: visuotactile integration and self-attribution. *Journal of experimental psychology: Human perception and performance*, 31(1):80.

Unitys Technologies (2004). Unity. `https://unity.com/`. Accessed: 2022-08-11.

Vive (2015). HTC vive. `https://www.vive.com/us/`. Accessed: 2022-08-11.

Waltemate, T., Gall, D., Roth, D., Botsch, M., and Latoschik, M. (2018a). The impact of avatar personalization and immersion on virtual body ownership, presence, and emotional response. *IEEE Transactions on Visualization and Computer Graphics*, PP:1–1.

Waltemate, T., Gall, D., Roth, D., Botsch, M., and Latoschik, M. E. (2018b). The impact of avatar personalization and immersion on virtual body ownership, presence, and emotional response. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1643–1652.

Wetas Digital (1993). Weta Digital Meerkat. `https://www.wetafx.co.nz/films/filmography/making-of-meerkat-a-real-time-animation-made-in-unreal-engine-4-26/`. Accessed: 2022-08-11.

Won, A. S., Bailenson, J. N., and Lanier, J. (2015). Homuncular flexibility: the human ability to inhabit nonhuman avatars. *Emerging Trends in the Social and Behavioral Sciences: An Interdisciplinary, Searchable, and Linkable Resource*, pages 1–16.

Wongutai, K., Palee, P., and Choosri, N. (2021). The effect of sound in vr exergame to adult player: A primary investigation. In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, pages 1–4.

Yee, N. and Bailenson, J. (2007). The proteus effect: The effect of transformed self-representation on behavior. *Human communication research*, 33(3):271–290.

Appendix

Appendix 1: Miscellaneous

Abbreviations:

- **IVBO:** Illusion Of Body Ownership
- **IK:** Inverse Kinematics
- **VR/AR/XR:** Virtual / Augmented / Mixed - Reality

Third party assets used:

- Meerakt 3D model, burrow, floor and eggs assets from Weta Digital's demo Unreal Engine project (Wetas Digital (1993)).
- Environment assets from Quixel Megascans' assets library.
- Royalty free sound assets from mixkit.co's website
- Unreal Engine 5 VR template and all its assets.

Appendix 2: Detailed results of the user study:

		Candidate 1	Candidate 2	Candidate 3	Candidate 4	Candidate 5	Candidate 6	Candidate 7		Total per Q	Total per T
	Q 1	2	3	4	5	3	2	4		3,28571429	
	Q 2	2	5	3	4	2	5	4		3,57142857	
Acceptance	Q 3	6	2	5	4	4	5	6		4,57142857	3,8095238
	Q 4	6	6	6	5	6	5	5		5,57142857	
	Q 5	5	6	5	5	6	6	6		5,57142857	
	Q 6	6	5	5	5	4	6	5		5,14285714	
Control	Q 7	6	6	6	6	4	4	5		5,28571429	5,3928571
	Q 8	5	3	4	5	3	4	4		4	
	Q 9	3	3	3	3	1	2	4		2,71428571	
	Q 10	2	0	1	0	0	1	2		0,85714286	
	Q 11	4	0	3	3	2	1	2		2,14285714	
Change	Q 12	2	1	2	1	2	3	1		1,71428571	2,2857143
	Q 13	4	6	5	4	4	4	5		4,57142857	
	Q 14	5	6	6	5	6	5	5		5,42857143	
	Q 15	4	6	5	5	5	6	5		5,14285714	
	Q 16	6	6	6	6	6	6	6		6	
	Q 17	4	6	6	4	5	5	5		5	
Other	Q 18	6	6	6	6	6	6	6		6	5,3571429

Figure 1: Detailed results of the user study with each grades given from each questions by each participants. The two last columns are the average for each questions and average for each categories