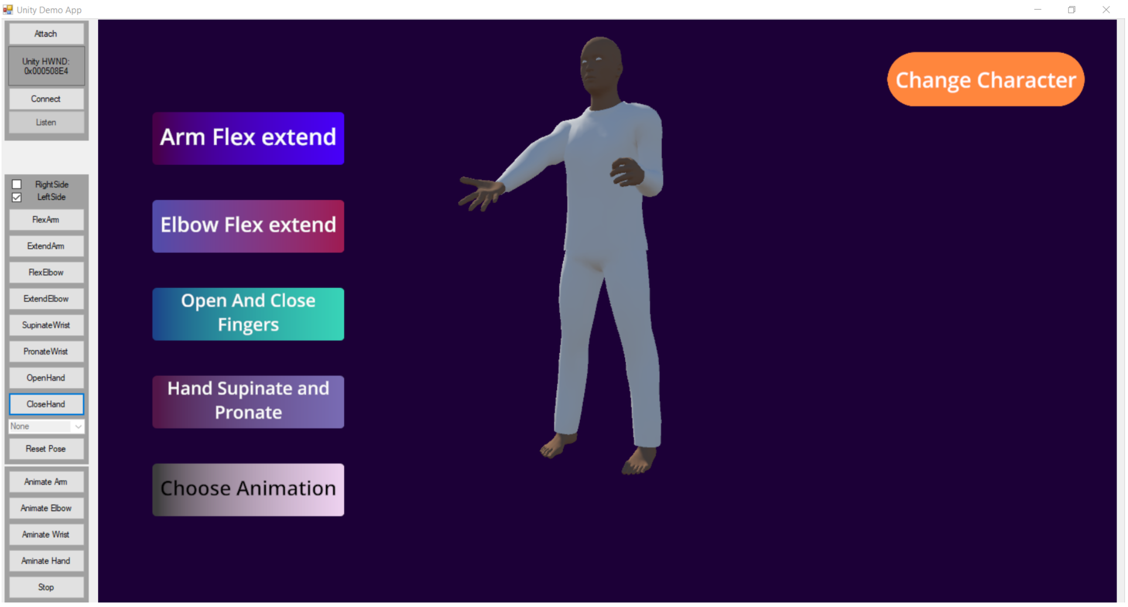


Development of a VR/3D Hand Prosthesis Software Application: Avatar Customization and Hand Motion Control

Master's thesis in Biomedical Engineering

FARYAL FATIMA SYEDA

DEPARTMENT OF BIOMEDICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Development of a VR/3D Hand Prosthesis Software Application: Avatar Customization and Hand Motion Control

FARYAL FATIMA SYEDA

Department of Electrical Engineering
Division of Signal Processing and Biomedical engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Development and Implementation of a VR/3D Hand Prosthesis Model: Avatar Customization and Hand Motion Control

FARYAL FATIMA SYEDA

Supervisor: Björn Davidsson, Integrum AB
Examiner: Silvia Muceli, Department of Electrical Engineering

Master's Thesis 2023
Department of Electrical Engineering
Division of Signal Processing and Biomedical engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Exploring the Future: Augmented Reality and Virtual Reality for Hand Prosthesis Training.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Development of a VR/3D Hand Prosthesis Software Application: Avatar Customization and Hand Motion Control
FARYAL FATIMA SYEDA
Department of Electrical Engineering
Chalmers University of Technology

Abstract

When a titanium device was first implanted into a thumb in Sweden in 1990, the osseointegration program for upper extremity amputation had its start there. Since then, below-elbow and transhumeral amputations have been performed using this technique. The e-OPRA system is an advanced version of a technique called the OPRA Implant System which is a technology in which it is possible to attach an artificial limb to the skeleton directly. It enables the acquisition of bioelectrical signals through implanted electrodes and nerve stimulation. The Artificial Limb Controller (ALC) System acts as a non-implanted device, facilitating communication between the e-OPRA system and a third-party robotic prosthetic for the upper limb. The e-OPRA/ALC system has software that is used for fitting, which is the process modifying and personalising the system's software to meet the recipient's unique needs and preferences. It includes setting up the software in order to make sure the recipient's prosthetic equipment is compatible with it. It involves configuring the software and ensuring its compatibility with the recipient's prosthetic device. It also has the potential to replicate movements for training and phantom limb reduction. Tools and software are used for configuring and maintaining the ALC system and e-OPRA implant system. The aim of the project is to explore and showcase the various functions and platforms that can be utilized to create and present interactive 3D models for software application. The project involves creating a human avatar that accurately represents behavior of hand movements, visual appearance, animation, and movement interaction. It is mainly focused on platform and tools investigation as a pre-study. Additionally, it aims to develop a software unit to enhance the control and interaction capabilities of prosthetic devices through virtual reality (VR) training. Furthermore, the model will be exported to different platforms to ensure compatibility with the existing software and hardware in the market.

The main objective of the software unit was to provide various upper limb movement control options, elbow flexion and extension, wrist supination and pronation, and hand opening and closing. Users could manipulate virtual items accurately and independently control each hand, enhancing the authenticity of hand movements. Additionally, a mouse-based control option was implemented to improve precision and ease of movement for delicate interactions or complex hand gestures. Although user feedback was not collected through surveys or interviews, valuable information was obtained from the comprehensive evaluation through the testing of the prototype in different conditions, which highlighted areas for improvement.

Areas for improvement in a future fitting application were identified, including diversifying training scenarios, enhancing the user interface, optimizing performance, and expanding platform compatibility. Future research and development suggestions involved incorporating electromyography (EMG) technology for dynamic control, improving usability and user experience. However, limitations should be considered, such as technical constraints and the potential disparity between virtual limb realism and real-life counterparts. The effectiveness of virtual limb visualization in reducing phantom limb pain may vary among individuals. Compatibility issues when exporting models to different platforms and the need for clinical validation should be considered. These limitations impact the interpretation and practical application of the project's outcomes.

In conclusion, the developed software unit has the potential to significantly enhance the training experience for individuals with prosthetic limbs. Through further research and development, incorporating user feedback and advancing technologies, the field of prosthetics can continue to evolve, providing amputees with improved mobility and quality of life.

Keywords: Augmented reality, virtual reality, hand prosthesis, training, rehabilitation, human-computer interaction, phantom limb pain

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Björn Davidsson, for all of his tremendous advice, assistance, and knowledge throughout the duration of this project. This research's direction and success have been greatly influenced by his astute criticism, persistent encouragement, and steadfast dedication to quality.

I also want to express my profound gratitude to Silvia Muceli, my examiner, for her insightful comments, helpful criticism, and careful assessment of this thesis. Her prosthetic knowledge and wise advice have made a significant contribution to the overall caliber and rigor of this study.

I sincerely appreciate the time, energy, and commitment Silvia Muceli and Björn Davidsson put into helping me with this project by serving as my mentors and advisors. They have been a tremendous source of support and encouragement, and I feel privileged to have had the chance to work under their direction.

The academic institution Chalmers University of Engineering and Technology, as well as my coworkers, friends, and family, deserve my sincere gratitude for their unflagging support, inspiration, and understanding throughout this research project.

Faryal Fatima Syeda, Gothenburg, May 2023

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

VR	Virtual Reality
AR	Augmented Realiry
ALC	Artificial Limb Controller
$OPRA^{TM}$	Osseointegrated Prostheses for the Rehabilitation of Amputees
TAC	Target Achievement Control
TCP	Transmission Control Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
MLA	Machine Learning Algorithm

Contents

List of Figures

1

Introduction

1.1 Background and rationale of the project

The e-OPRA Implant System is an improved version of the OPRA Implant System, which is a technique that allows a prosthetic limb to be directly attached to the skeleton. Through the use of neural cuff electrodes and implanted muscular electrodes, it enables the acquisition of bioelectrical signals as well as the stimulation of nerves [1]. The e-OPRA Implant System is utilised in conjunction with the Artificial Limb Controller (ALC), a control tool. It is intended to be fastened in between an upper limb robotic prosthetic device and the e-OPRA. The e-OPRA Implant System's ALC reads, decodes, and transmits the signals generated by the implanted electrodes, enabling two-way communication between the external prosthesis and the body. With the development of technologies like the Artificial Limb Controller (ALC) and the enhanced Osseointegrated Prostheses for the Rehabilitation of Amputees (e-OPRA) system, the field of prosthetics has made significant strides recently [2]. A more effective and comfortable prosthetic experience for amputees is what the e-OPRA Implant System is intended to give them. In order to collect bio-electrical impulses and stimulate nerves for tactile feedback, the device entails implanting muscular and neural electrodes. Compared with traditional socket prostheses, which may cause patients pain and suffering and restrict their mobility, the e-OPRA system is a major improvement [3]. The transhumeral configuration of the ALC system is depicted in Figure 1.1 [4].

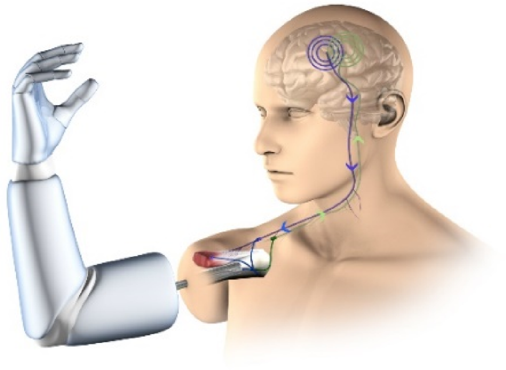


Figure 1.1: ALC system in trans humeral configuration (This figure has been taken from Integrum AB website; link: https://integrum.se/what-we-do/our-products-future-solutions/e-opra/)

The e-OPRA Implant system works in conjunction with the ALC system which gives the patient a much more intuitive and natural experience as compared to traditional prosthesis by enabling them to operate their prosthetic limb using their own muscle signals. Additionally, the ALC system enables two-way communication between the patient's body and the prosthetic limb, which enhances control and feedback [5]. These innovations collectively constitute a significant advancement in the field of prosthetics and have the potential to significantly enhance the lives of amputees. The ALC and e-OPRA systems can significantly increase mobility and improve quality of life by giving natural and intuitive control over prosthetic limbs [6]. Moreover, ongoing research into these technologies is expected to lead to the development of increasingly sophisticated prosthetic devices in the future, which will provide amputees with greater advantages [7].

1.1.1 Role of Prosthetic Technologies in Managing Phantom Limb Pain

In the context of amputation and rehabilitation, prosthetics, pain from phantom limbs, and therapeutic pain reduction are interlinked. Prosthetics offer individuals the opportunity to regain their mobility and independence through the substitution of artificial devices for missing body parts [8]. However, phantom limb pain, a syndrome where people experience pain, discomfort, or other feelings in the missing limb, might compromise the efficacy of prosthetics. It may be difficult to operate the prosthetic device efficiently since this pain can obstruct the signals that are transferred from the device to the brain [7]. It is thought that changes in the brain's processing of sensory information after amputation are responsible for the chronic pain syndrome known as "phantom limb pain," which can last for years after the amputation. Therefore, to reduce the pain that those who have had amputations endure, therapeutic pain reduction procedures are employed [8]. These methods can take many different forms, such as physical therapy, medication, psychological therapies, and others. People's quality of life can be enhanced by lowering pain so they can use their prosthetic devices more efficiently [9].

Additionally, more advanced prosthetic technologies are being created that include neuromuscular interfaces to give prosthetic limbs more precise control [10]. In order to control the prosthetic limb, these interfaces are made to collect myoelectric impulses from the muscles as shown in Figure 1.2 [11]. These interfaces enable patients to move their prosthetic limb in more intricate ways, which helps lessen the frequency and severity of phantom limb pain [4].

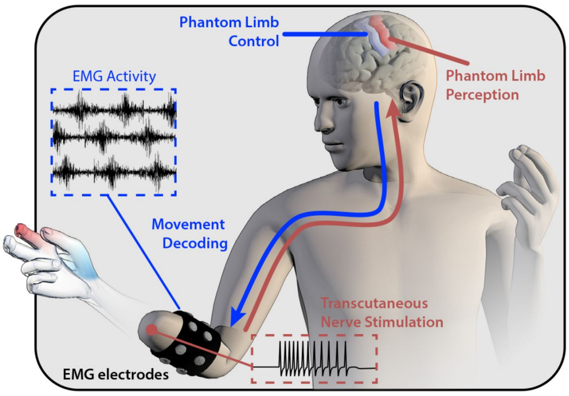


Figure 1.2: Perception and control of phantom limbs (This Figure is taken from the article [11] Link: https://www.medrxiv.org/content/10.1101/2020.05.22.20109330v1.full)

1.1.2 Advancements in Implantable Neuromuscular Interfaces for Robotic Prostheses

The e-OPRA Implant System represents a notable breakthrough in prosthetics, leveraging advancements in osseointegration. This innovative system addresses the challenge of establishing bi-directional communication between the human body and prosthetic limbs [4]. The technology uses neural cuff electrodes to stimulate nerves for tactile feedback and implanted muscular electrodes to collect bioelectrical information. By enabling robotic prostheses controlled by implanted neuromuscular interfaces, the device is poised to transform the field of prosthetics and make it a clinical reality as shown in Figure 1.3 [11].

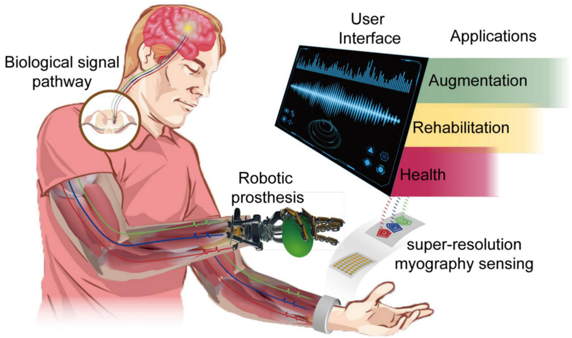


Figure 1.3: A diagram outlining the wearable super-resolution myography technology and its potential applications (This figure is taken from the article [11], link: https://www.frontiersin.org/articles/10.3389/fnins.2022.1020546/full)

According to numerous research, using prosthetics with neuromuscular interfaces can lessen the pain associated with phantom limbs. As an example, a study demonstrated that the utilization of a prosthetic device equipped with a neuromuscular interface resulted in a notable decrease in phantom limb pain and a significant enhancement in overall functionality [12]. A myoelectric prosthetic device was found to improve quality of life and reduce discomfort associated with phantom limbs, according to a study that was published in the BMJ open [13]. Overall, research into the creation of prosthetic systems that integrate neuromuscular interfaces has great promise for enhancing the function and quality of life of people who have had amputations. These technologies have the potential to greatly reduce pain from phantom limbs and enhance overall function, enabling people to carry out daily tasks more successfully [14].

Additionally, to enable bidirectional communication as shown in Figure 1.3 between the external prosthesis and the human body, the Artificial Limb Controller (ALC) System is employed in conjunction with the e-OPRA Implant System. The signals produced by the implanted electrodes are read, decoded, and transmitted by the non-implanted ALC System of the e-OPRA Implant System. Users' quality of life is enhanced by this system's smooth and intuitive control of their prosthetic limbs. Patients may benefit from a more natural and simple manner to operate their prosthetic limbs due to the integration of the e-OPRA Implant System and the ALC System. It has the potential to transform the lives of amputees all over the world and represents a significant advancement in the field of prosthesis [15].

1.2 Problem Statement

1.2.1 Challenges Faced by Amputee Patients Without Virtual Reality Tools

:

Without virtual reality tools for fitting, training, and rehabilitation, amputee patients confront a variety of difficulties. As a result of not having the opportunity to virtually train and rehearse using the actual devices beforehand, they frequently struggle to acclimate to their new prostheses at first. When obtaining the prosthesis, this may cause discomfort, pain, and difficulties doing chores [16]. These issues of training and fitting in e-OPRA patients have been addressed by investigating the possibilities of incorporating a virtual reality (VR) technology for training, rehabilitation and therapy software in this VR/3D model project. Additionally, fitting means that appropriate methods before constructing the devices physically, doctors may virtually fit and adjust them for certain patients in the virtual world. While training suggests that patients might practice using and adjusting devices in a virtual environment prior to obtaining the actual devices, this could enable each patient's fit and comfort could be optimized. By developing an interactive 3D-model that can be utilized in software applications to visualize virtual limbs for simulation, EMG training, and phantom limb pain reduction, this project seek to address these difficulties. In the field of e-OPRA, the development of this interactive 3D model is highly benificial since it will give medical experts an important tool for mimicking the behavior of prosthetic limbs (especially hand movements). Being able to train their prosthetic limbs to more closely resemble the movements of their natural limbs would improve the outcomes for patients who have undergone amputations or who have lost limbs as a consequence of trauma.

Future e-OPRA patients will also benefit even more from the employment of machine learning techniques in combination with the 3D model. These algorithms can assist in identifying trends and enhancing the precision of prosthetic limb control by examining the data produced during virtual simulations. Patients may experience an improvement in their quality of life as a result of being better able to manage their prosthetic limbs and carry out actions that were previously unachievable [16]. The VR/3D-Model project in eOPRA implant system represents an exciting new advancement in the field of e-OPRA and have the potential to dramatically enhance the lives of patients who have suffered amputations or lost limbs as a result of trauma. This thesis project of VR/3D model is still in the exploratory phase and aims to investigate the possibilities of applications using this technology in the context of aiding and supporting the clinical process by developing an interactive 3D-model that can be used for simulation, training, and phantom limb pain relief.

1.3 Aim of The Project

An advanced technology called the Virtual reality (VR) Model is utilized to get over the challenges associated with training and fitting e-OPRA patients. The VR/3DModel project and the eOPRA/ALC system, combined, seek to identify a platform for modeling and rendering virtual limbs and prostheses for simulation, EMG training, and alleviation of discomfort from phantom limbs. The main objective of this project is to thoroughly investigate and showcase the available functions and platforms for designing and rendering interactive 3D models for use in various software applications. The focus of the 3D models will be on visualizing virtual limbs, which can be utilized in simulation scenarios, training, and phantom limb pain reduction. Additionally, the project will involve the creation of a human avatar, considering aspects such as behavior, visual appearance, animation, and movement interaction. Furthermore, the project will explore the exporting of the model to multiple platforms, ensuring compatibility with the currently available software and hardware on the market. The investigation will specifically evaluate the ease of use, integration capabilities, and availability of the identified tools and platforms. Utilizing these investigations, the project aims to create a human avatar capable of moving and interacting with its surroundings, in addition to examining the exporting options of different rendering platforms to comprehensively understand the process.

The project will provide practical demonstrations of avatar modeling, allowing for interchangeable modifications of visual properties like gender, hair, color, and animations. Furthermore, the process of exporting the model to a rendering platform embedded within a software application will be showcased. Specifically, for the trans-humeral limbs, the application will demonstrate the ability to simulate the intended movement of an end user by injecting limb movements into the model. Another goal is to incorporate the model data and the rendering element into each software program. This will guarantee that the model is usable for a variety of applications and is simple to access. The project also seeks to show how to control and behave in at least 3DoF on the TH level, imitate desired end-user movements, and perform targeted movements of virtual 3D model avatar. The interactive 3D-model will give doctors and patients a more efficient approach to train and fit patients while also enhancing their experience.

1.4 Significance of the eOPRA/ALC system, and the VR/3D-Model

The $e-OPRA^{TM}$ Implant System is an upgraded edition of the $OPRA^{TM}$ Implant System, which enhances prosthetic limb control and sensory feedback. Additionally, the eOPRA/ALC system's capacity to improve end-user virtual control and enable the creation of a machine learning algorithm may completely alter how patients with upper or lower extremity limb loss undergo rehabilitation. Due to the lack of a smooth connection between the prosthetic device and the user's body sensations, motions, and thoughts, current prosthetic systems have limited functionality and integration. As a result, the prosthetic device cannot be used or controlled to its full potential. The system's cutting-edge technology has the ability to eliminate the barrier that now exists between humans and machines by enabling seamless prosthetic limb integration with the user's body and mind [20]. The different kind of amputations in OPRA/$e-OPRA^{TM}$ system is depicted in Figure 1.4 [19].

Figure 1.4: Apmutation levels of OPRA/e-$OPRA^{TM}$ system (This figure has been taken from Integrum AB website; link: https://integrum.se/what-we-do/our-products-future-solutions/opra-implant-system/)

The potential impact of the VR/3D-Model project is as important to that of the eOPRA/ALC system. In order to imitate end users' planned actions and carry out targeted movements of hand, the project aims to create a realistic virtual environment. Such a virtual setting could help with the treatment of pain from missing limbs and the rehabilitation of individuals who have lost their upper or lower limbs. In general, the VR/3D-Model software in the eOPRA/ALC system have enormous potential to lead to a paradigm shift in the field of prosthetics, improving the quality of life for amputees, and speeding up the recovery process.

1.4.1 Significance of the VR/3D-Model in e-OPRA Implant system

This MSc project's specific goal is to create an VR/3D model that e-OPRA/ALC users can utilise for training and rehabilitation. This model is significant in two ways:

1. For the purpose of helping upper limb amputees recover, it will offer a virtual en-

vironment for modelling deliberate actions and precise limb movements. This could accelerate recovery and lessen pain from phantom limbs.

2. To enhance the current neuromuscular training application, the AV/VR model and platform created in this research can be incorporated into a later edition of the e-OPRA/ALC system. End users will be able to train and manage their prosthetic limb more successfully as a result.

1.5 Innovative Prosthetic Solutions: The eOPRA/ALC System and VR/3D-Model Project

The VR/3D-Model project in the eOPRA/ALC system is crucial for tackling the issues in prosthetics, especially when it comes to educating and fitting e-OPRA patients. Patients who have lost an upper or lower extremity limb struggle to control their prosthetics, necessitating the need for a more effective and efficient alternative. Implanted electrodes in the muscle tissue of the eOPRA/ALC system collect myosignals in situ and enable prosthetic control, which may enhance patient rehabilitation [19]. Additionally, the software components of the ALC System allow for EMG streaming to the software applications, improving virtual user control and facilitating the training of a machine learning algorithm [20].
Furthermore, the VR/3D-Model's capacity to produce a lifelike virtual environment is essential for addressing the problem of phantom limb discomfort, which many amputees experience. The model can help in the treatment of phantom limb pain by offering a platform for simulating intended movements of end-users and carrying out targeted movements of virtual objects. The project's advantage is that it can be readily integrated with any available hardware and software platforms because it is extremely configurable to each software application. The project's importance stems from its potential for rehabilitation process of the patients who have lost an upper extremity limb, and help with the management of phantom limb discomfort. Many people's quality of life can be enhanced by the eOPRA/ALC system and the VR/3D-Model project, which provide them more control and less discomfort. The project has the potential to significantly benefit society by offering a more practical prosthetics solution.

1.6 Research questions

The prospective uses of the e-OPRA/ALC system containing virtual reality (VR) models for upper limb prosthetic rehabilitation are examined in this project. Determining the best modeling and rendering tools, hardware platforms, control and behavior practices will be necessary. This will also involve investigating potential limitations and challenges of implementing these systems in clinical settings. The goal of this study is to investigate the viability and potential benefits of using these technologies to improve virtual control for end-users and machine learning algorithms. The project addresses the following research questions in order to accomplish these objectives:

1. What modeling programs are appropriate for creating avatars and skeleton models in three dimensions?
2. What are the rendering platforms that can be utilized for 3D model import and execution?
3. Which 3D rendering systems can be supported by which hardware platforms?
4. What are the specifications for an ALC-TH-focused VR/3D model?

1.7 Theoretical framework

Architectural view of theoretical framework-VR Modelling has been shown in 1.5:

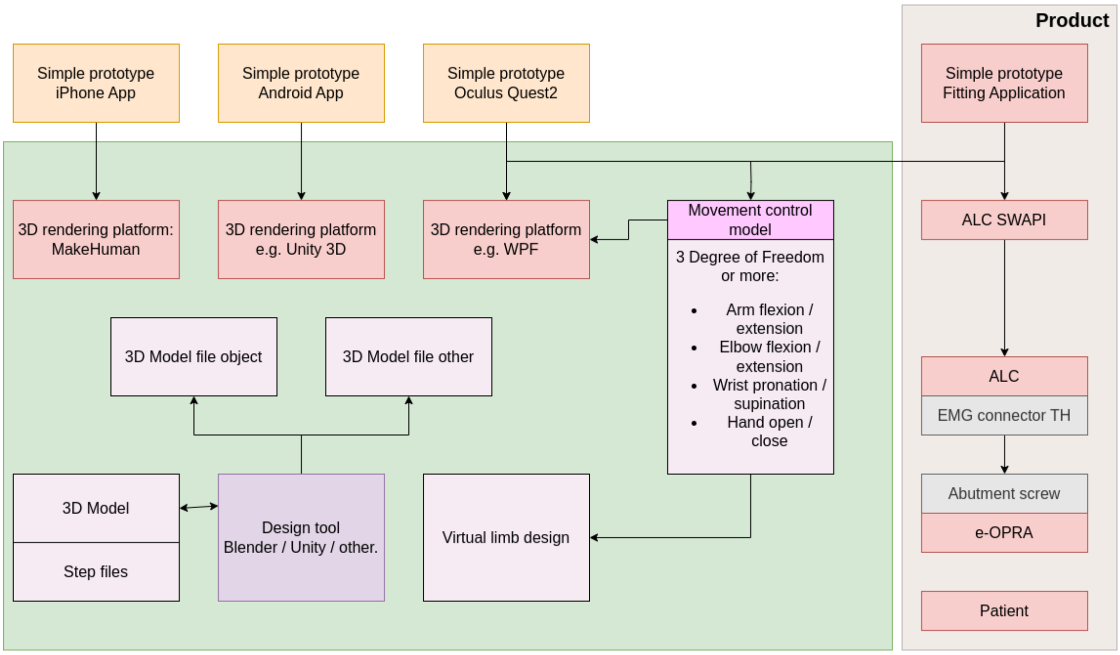


Figure 1.5: A depiction of the architectural perspective of the theoretical framework for VR modelling

With a particular emphasis on the users in visualizing virtual limbs for simulation, EMG training, and phantom limb pain reduction, the theoretical framework of this project explores the various functions and platforms available for designing and rendering interactive 3D models. This framework entails the development of a human avatar that can move about and interact with its surroundings, as well as research into hardware platforms like Android, iOS, Windows, and Oculus Quest 2 that can handle 3D rendering. To gain a thorough grasp of the procedure, the exporting capabilities of various rendering tools including Mixamo, Make Human, Blender, and Unity3D will also be investigated.
The framework will also use a Windows desktop application to demonstrate the full procedure, from design to encapsulation. The project will also concentrate on fulfilling the particular specifications for modeling, movement, behavior, exporting

and running, and application integration of VR/3D-Model. The project intends to uncover any gaps in existing research and offer a critical assessment of the present state of the art in this sector by closely examining these elements. In the end, the project's theoretical foundation offers a thorough framework for the creation and assessment of an VR/3D-Model for prosthetic design and training that may be utilized to improve the efficiency and usability of the market's current prosthetic training approaches.

1.8 Thesis Outline

There are seven chapters in the report. The study objectives are outlined in the first chapter , along with a description of the investigative tasks and an overview of the e-OPRA/ALC system and the VR/3D-Model. The literature on osseointegration, myoelectric control of prosthetic devices, and the application of VR/3D-Models in prosthetic training and rehabilitation is reviewed in Chapter 2 . It also discusses the drawbacks of the present approaches to prosthetic fitting and training.
Chapter 3describes the theoretical framework tools and platforms used to develop and render the 3D models, as well as the materials and processes utilized in the project. It also describes the parts of the e-OPRA/ALC system and the VR/3D-Model. Additionally, it describes the experimental setup and the methods used to gather data and assess the system. In Chapter 4, the study's findings and experimental methods are presented, including how the Avatar is created and gender selection has been made in Unity 3D, how accurate the VR/3D-Model was in mimicking movements and providing training, and how well the system was received by users. The results of the findings, overview of the application that has been made, the study's advantages and disadvantages, as well as prospective uses for the e-OPRA/ALC system and the VR/3D-Model, are covered in Chapter 5. It contrasts the system with current prosthetic fitting and training techniques, analyzes critical elements that could influence the system's adoption and execution, and discusses the project's constraints. The study's discussion section is presented in Chapter 6along with recommendations for additional research. It outlines the system's possible advantages and disadvantages while outlining the key contributions of the the VR/3D-Model application. It also makes suggestions for further research and development. Chapter 7contains the conclusion of the report.

2

Literature Review

2.1 Overview of the relevant literature on prosthetics, myoelectric control, and VR/3D modeling

Since their origin, prosthetic devices have advanced significantly, with improvements in materials, manufacturing processes, and control mechanisms making them more user-friendly and practical [21]. The incorporation of myoelectric control into prosthetics has been one of the most important advancements. Myoelectric control is a method that directs the motion of the prosthetic device using signals from the user's remaining muscles. Myoelectric control has showed promise in enhancing prosthetic users' functionality and happiness [22]. The range of motion and precision of the prosthetic limb can be enhanced by using myoelectric control to simultaneously and independently regulate numerous degrees of freedom in the limb. Real-time feedback to the user is one of the main advantages of myosignal-based prosthesis, enabling more accurate and effective control of the prosthetic limb [23]. This can greatly increase the prosthetic device's functionality and the user's quality of life [24]. According to research by Jiang et al. (2012), myoelectric control has also been proven to lessen the cognitive strain that comes with using a prosthesis, allowing users to carry out tasks more successfully and naturally[25].

Dr. Max Ortiz Catalan, Ph.D., at Centre for bionics pain and research has developed an innovative approach to treat Phantom Limb Pain (PLP) by utilizing virtual and augmented reality (VR/AR) in conjunction with phantom movements known as Phantom Motor Execution. This method involves anticipating intricate movements of the absent limb by utilizing machine learning algorithms that examine myoelectric patterns of muscular activity at the stump site [26].

Another developing field of study involves the incorporation of augmented reality (AR), virtual reality (VR), and 3D modeling into prosthetic training and rehabilitation. To imitate movements and give prosthesis users feedback during training and rehabilitation, VR/3D modeling can be employed, which will improve their overall results (Barbosa, A., 2012)[27]. Rutledge et al. has created a VR treatment for warriors who experience discomfort from phantom limbs. Following therapy, 71% of patients reported fewer phantom sensations and 57% reported less pain from phantom limbs. The results encourage further investigation into VR pain relief for phantom limbs [28]. In lower leg amputees, visual reality with tracking sensors lessened phantom limb pain. Following VR sessions, both subjects had rapid and long-lasting pain alleviation. According to preliminary findings, VR might be a cost-efficient and successful PLP treatment [29].

2.1.1 Designing Interactive 3D Models

The design and rendering of interactive 3D models for software applications have been addressed and the use of virtual reality (VR) and augmented reality (AR) technologies has gained significant attention in the field of limb visualization and rehabilitation. Techniques such as motion capture, skeletal modeling, and animation have been explored to create realistic and customizable human avatars [30]. These models allow for the modification of visual properties, including gender, hair, color, and animations, providing a versatile platform for users [31].

2.1.2 Rendering Platforms and Integration:

The process of producing visual information from 3D models is referred to as rendering. The essential tools and capabilities are offered by rendering platforms, which enable the creation and manipulation of the 3D models' visual components [30]. To produce realistic and compelling graphics, they perform duties including lighting, highlighting, and pattern mapping. The selection of appropriate rendering platforms is crucial for the successful implementation of interactive 3D models. Unity3D is the popular choices due to its extensive features, cross-platform compatibility, and user-friendly interfaces [30]. These platforms offer seamless integration with various software applications, facilitating the embedding of rendering components and model data [30].

2.1.3 Simulation and Training:

The simulation of limb movements plays a vital role in training and rehabilitation programs. By utilizing myoelectric pattern recognition techniques, it becomes possible to predict complex movements of missing limbs based on patterns of muscular activity [32]. Integration of these predictions into the virtual limb models allows for real-time movement simulation, enabling users to interact and train with the virtual limbs. This functionality can be implemented for single-arm simulations, bilateral arm training, as well as specialized tests such as the Target Achievement Control (TAC) test [33].

2.2 Evaluating Existing Research and Identifying Research Gaps

The existing literature [25] has provided valuable insights into prosthetics, myoelectric control, and the integration of VR/3D modeling in training and rehabilitation. However, there are several areas that require further exploration and research to advance the field. Firstly, in the domain of myoelectric control, while significant progress has been made in enhancing the functionality and precision of prosthetic devices, there is a need for more studies on the long-term usability and user satisfaction with these control mechanisms [31]. Understanding the impact of myoelectric control on prosthetic users' daily activities and quality of life would contribute to improving the design and usability of such devices [31].

Secondly, in the context of VR/3D modeling, there is a need for more research on the effectiveness of these technologies in prosthetic training and rehabilitation. Studies evaluating the impact of virtual limb visualization and feedback on users' motor skills, proprioception, and overall rehabilitation outcomes would provide valuable insights into the potential benefits of these approaches [33]. Furthermore, exploring the integration of myoelectric pattern recognition techniques with VR/3D modeling is an area that requires attention. Investigating the real-time synchronization of myosignal-based prosthetic control with virtual limb models and assessing its impact on users' motor learning and performance would contribute to advancing the field of prosthetic training.

Comparative studies exploring the ease of use, integration, and availability of different rendering platforms would assist developers and researchers in selecting the most appropriate tools for their specific needs. Overall, this literature review has identified gaps and research opportunities in the field of designing and rendering interactive 3D models for software applications, particularly in the context of virtual limbs and rehabilitation. By addressing these gaps, future research can contribute to the development of more effective and user-friendly tools and applications that enhance the training, simulation, and pain reduction strategies for individuals with limb loss or limb impairment [33].

By creating an VR/3D modeling application that uses various control methods and is created with a user-centered approach, this research will possibly contributes in closing these gaps. Through this application, amputees will be able to practice using their prosthetic limbs in a virtual setting that replicates situations and actions from the actual world. Additionally, the application will be more user-friendly and efficient if the preferences of the user are taken into account during the design and development process. The overall goal of this research is to use VR/3D modeling techniques to build more individualized and efficient prosthesis training and rehabilitation programs.

3

Theoretical framework

This project focuses on the utilization of virtual limbs for simulation, EMG training, and alleviating phantom limb pain. The theoretical framework of the study investigates different functions and platforms that can be employed to create and display interactive 3D models. The following section provides an overview of the investigational tasks and requirements for designing and rendering an interactive 3D model to be used in software applications. The aim of this investigation is to visualize virtual limbs for simulation, electromyography (EMG) training, and phantom limb pain reduction. This investigation includes various aspects such as modeling tools, exporting capabilities, rendering platforms, hardware platform compatibility, and the modeling of a human avatar with respect to behavior, visual properties, animation, and movement interaction.

3.1 Investigational Tasks:

1. Conduct research on suitable modelling tools for 3D-design of skeleton models and avatars.
2. Explore the exporting capabilities, such as the obj format, for transferring the 3D models to different platforms.
3. Investigate rendering platforms like Unity3D, WPF, and others to determine their compatibility with importing and running on hardware/software platforms.
4. Examine the hardware platforms (Android, iOS, Windows, Oculus Quest2) to identify which ones can support the chosen 3D rendering platforms.
5. Start the modelling process with a 3D skeleton model in the T/A (T-Pose/A-Pose) stance, commonly used in the gaming industry, and explore methods for interchangeable visual appearances.
6. Investigate how to apply control and behaviors, focusing on at least 3 degrees of freedom (DoF) for the upper extremity limbs, such as hands, optionally exploring additional DoF.
7. Report on the findings and conclusions of the investigations.
8. Demonstrate the entire process, from design to encapsulation, on at least a windows desktop application.

3.2 Requirements on VR/3D-Model:

3.2.1 Modelling:

1. Enable the modelling of a human avatar and provide the ability to interchange different avatars.
2. Allow the modification of selected properties, including gender, color, hair, weight, and length.

3.2.2 Movement and behavior:

1. Provide control over the movements of the upper limb extremities, including elbow flexion/extension, wrist pronation/supination, flexion/extension of the hand, and opening/closing of the hand because these are the most common and important ones and all other movements are the combinations of these movements .
2. Allow animation of the avatar with movements such as walking and sitting down/up.
3. Support the execution of the Target Achievement Control (TAC) test, simulating a target movement by overlapping the avatar with additional limbs controlled separately.
4. Enable training/demonstration movements with combinations of 3 degrees of freedom (DoF), including bilateral movement patterns.

Provide the ability to simulate and visualize prosthetic movements, with a virtual prosthesis attached to the residual limb of the avatar.

3.2.3 Export and Running:

1. Allow real-time control of the avatar from a software application using inputs such as mouse or electromyography (EMG).
2. Enable interaction with objects, including picking up, grabbing, and moving them, to facilitate user training and machine learning algorithms.
3. Support augmented reality, where the virtual limb is attached to the body of an end user.

3.2.4 Application integration:

1. Ensure the application can be invoked and run on a Windows desktop platform.
2. Provide compatibility with other platforms such as iOS, Android, and Oculus Quest2.

3.3 Conceptual model and hypotheses

The utilization of VR/3D-Model can considerably improve the efficacy and accessibility of current prosthesis training methods, according to the conceptual model of this project. The theoretical framework, which stresses the usage of several software programs like MakeHuman, Mixamo, Blender, and Unity 3D as well as research into the hardware platforms that may support 3D rendering, constitutes the theoretical foundation of this project [34]. With the use of these technologies, the project seeks to produce a human avatar that can move and interact with its surroundings, as well as look into the exporting capabilities of various rendering platforms to gain a thorough grasp of the procedure.

The project's hypotheses are predicated on the idea that using VR/3D models for prosthetic design and training will improve the training outcomes for amputees. In particular, it is believed that the usage of VR/3D-Model will improve amputees' ability to imitate and visualize their prosthetic limbs in a realistic way. This will then result in better muscular coordination, less pain from the phantom limb, and greater self-assurance when using their prosthetic limb [35]. It is also predicted that the usage of VR/3D-Models for prosthesis design and training will make training for amputees more accessible [36]. The goal of the project is to develop a framework that is affordable and simple to incorporate into current prosthesis training techniques. In general, the project's conceptual model and hypotheses seek to offer a thorough framework for the creation and assessment of VR/3D-Models for prosthesis design and training. It seeks to identify any gaps in existing products for example: Neuromotus; Integrum AB's unique, ground-breaking technology that is being utilized in therapy to lessen the pain associated with missing limbs following amputation, and offer a critical assessment of the present state of the art in this sector by looking into the specific requirements related to modeling, movement, behavior, export and running, and application integration of VR/3D-Model.

3.4 Software Platforms

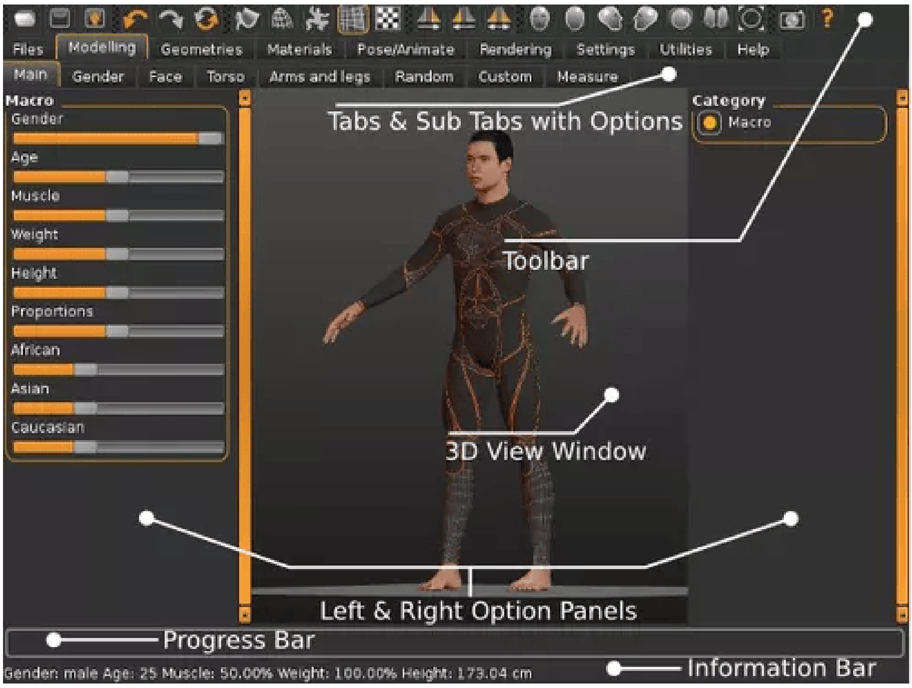
In this project, placing a strong emphasis on the investigative aspect is of utmost importance in addition to the production of artifacts. The investigative phase serves the purpose of thoroughly exploring and identifying a wide range of tools and platforms that are available on the market. This exploration goes beyond the initial selection made for the project, allowing for a comprehensive understanding of the diverse options that exist.

Insightful information about the most recent developments, trends, and advancements in the sector was gained by conducting this broader investigation on these platforms, valuable insights into the latest advancements, trends, and advancements in the field was gained. It enables them to stay informed about the ever-evolving landscape of tools and platforms, ensuring that the project's choices are well-informed and aligned with the current state-of-the-art. Moreover, this comprehensive investigation plays a crucial role in enabling informed decision-making. By considering a broad range of options, the project team can weigh the advantages, disadvantages, features, and capabilities of different tools and platforms. They can assess how well each option aligns with the project's objectives, requirements, and constraints. This informed decision-making process ensures that the final selection of tools and platforms is well-suited to meet the specific needs of the project, maximize efficiency, and deliver high-quality outcomes. This following broader investigation ensures a comprehensive understanding of the available options and enables informed decision-making:

3.4.1 MakeHuman

A free and open-source 3D modeling program called MakeHuman is made primarily to produce incredibly realistic human models for usage in a variety of industries, including animation, application development, and scientific study. Its main objective is to make it simple and quick for users to construct accurate 3D human models without the need for highly developed technical expertise or artistic abilities. MakeHuman accomplishes this by offering a variety of pre-built human models that are simple to edit to produce distinctive characters. Users can customize a model by changing different characteristics including age, gender, body shape, and facial traits using the software's relatively intuitive user interface.The model's realism is further increased by the software's inclusion of a library of pre-built apparel and accessories that can be added to it [37].

In order to make the model compatible with other 3D modeling and animation tools, the user can export it in a variety of formats, including OBJ, FBX, and Collada. Various plugins and scripts are also supported by MakeHuman, which can be used to increase its functionality and customization possibilities [38]. Figure 3.1 displays the essential characteristics of MakeHuman.

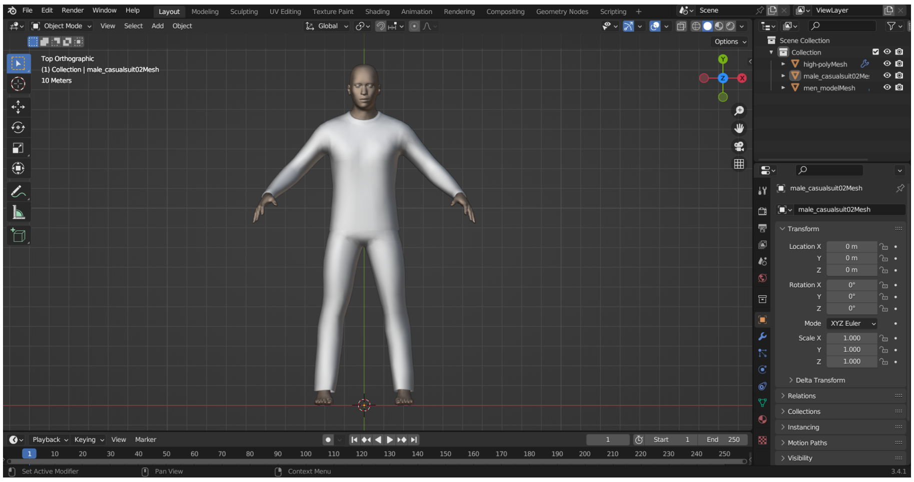


Gender: male Age: 25 Muscle: 50.00% Weight: 100.00% Height: 173.04 cm

Figure 3.1: Key features of MakeHuman (This Figure is taken from: http://static.makehumancommunity.org/makehuman/docs/the_interface_and_its_basic_functions.html)

3.4.2 Blender

The 3D modeling, sculpting, animation, rigging, rendering, and compositing tools in Blender are free and open-source. Its goal is to offer a full 3D creativity suite that is affordable and available to everyone, regardless of their degree of expertise or financial situation. With the help of Blender's modeling/animating features, artists and designers can produce intricate 3D models, animations, and realistic renders. With Blender, users can modify 3D objects and produce animations using a number of tools and methods thanks to a graphical user interface. Users can add vertices, edges, and faces to construct objects, which they can then edit using tools like extrude, bevel, and loop cut. Additionally, Blender offers sculpting capabilities that let users design organic shapes and textures. Blender not only offers modeling but also animation and rigging, enabling users to create intricate movements. Making a rigging system of bones and controllers enables users to control a 3D model's movement. Keyframe animation is another feature of Blender that enables users to make animations by specifying keyframes at various periods in time and letting Blender produce the interstitial frames on its own [39]. Figure 3.2 illustrates a 3D human model created using the Blender software.



male_casualsuit02Mesh

Transform
Location X 0 m
Y 0 m
Z 0 m
Rotation X 0°
Y 0°
Z 0°
Mode XYZ Euler
Scale X 1.000
Y 1.000
Z 1.000
Delta Transform
Relations
Collections
Instancing
Motion Paths
Visibility

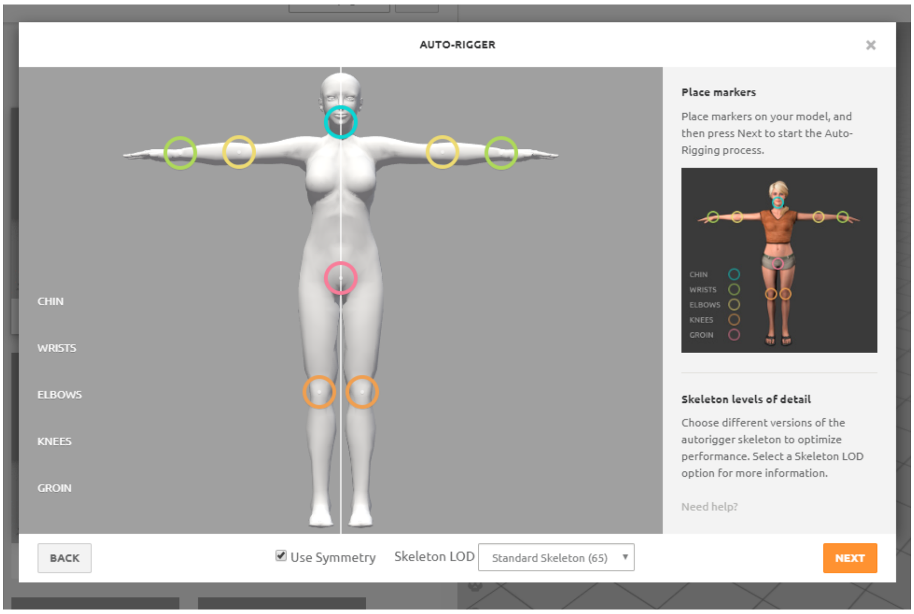
Current frame 1, Start 1, End 250

Figure 3.2: The 3D design of a human model in the Blender software

Users may produce photorealistic images and animations using Blender's potent rendering engine. The rendering methods supported by the engine include ambient occlusion, global illumination, and ray tracing. Users can add post-processing effects to their renders using the integrated compositor that is part of the software. Blender is a flexible 3D creation program that may be used for a variety of tasks, including the creation of games, movies and videos, buildings, and products. Worldwide, designers and artists favor it due to its open-source nature and vibrant development community [39].

3.4.3 Mixamo

Users of the web-based service Mixamo get access to a vast collection of 3D character models and animations. The same-named startup originally created the service, which Adobe later purchased. By offering pre-built elements that can be readily changed and included into a variety of projects, such as games, films, and simulations, Mixamo's main objective is to make it easier to create 3D character models and animations. Users of Mixamo have the option of uploading their own 3D character models or choosing from a large selection of already-built models that are offered on the website. Users can select animations from a sizable library after selecting a model and applying them to the model. Timing, intensity, and direction are just a few of the parameters that may be changed to create unique animations. Mixamo also offers tools for rigging 3D models, which is the act of giving the object a skeleton structure so that it may be animated as shown in Figure 3.3 below [40].



Place markers

Place markers on your model, and then press Next to start the Auto-Rigging process.

Skeleton levels of detail

Choose different versions of the autorigger skeleton to optimize performance. Select a Skeleton LOD option for more information.

☑ Use Symmetry

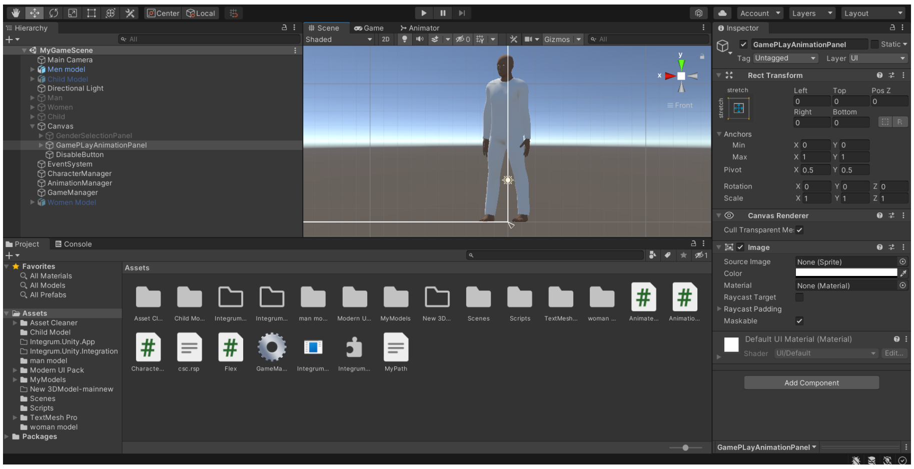
Skeleton LOD: Standard Skeleton (65)

Figure 3.3: Human Model Rigging through Mixamo

Overall, Mixamo offers an efficient workflow for building 3D character models and animations that can help designers and artists save time and money. The platform is renowned for other features as well, including a user-friendly interface and a sizable collection of pre-built components that make it a useful tool even for individuals without a lot of 3D modeling and animation knowledge [40].

3.4.4 Unity 3D

An integrated development environment (IDE) and cross-platform game engine called Unity 3D are used to make interactive 3D content. Although it was first designed to create video games, its capabilities have now been extended to include various types of interactive 3D experiences, including animations, virtual and augmented reality applications, and architectural visualizations. A component-based architecture is used by Unity 3D to create 3D environments and objects. Users can apply physics, lighting, and other effects to 3D models and textures imported from other tools like Blender to create dynamic and immersive experiences. The engine has built-in support for C# scripting, which enables users to design user interfaces, menus, and other features as well as program actions for objects and characters [41]. Avatar Creation: Based on the MakeHuman models, avatars can be made using Unity 3D. Users can load 3D models into the software, then add textures and motions to produce avatars that seem lifelike. Users can program the avatars' actions and produce interactive experiences by using scripting [41].



GamePLayAnimationPanel ☐ Static
Tag Untagged Layer UI

Rect Transform
stretch Left 0 Top 0 Pos Z 0
Right 0 Bottom 0
Anchors
Min X 0 Y 0
Max X 1 Y 1
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Canvas Renderer
Cull Transparent Me: ☑

☑ Image
Source Image None (Sprite)
Color
Material None (Material)
Raycast Target ☐
Raycast Padding
Maskable ☑

Default UI Material (Material)
Shader UI/Default Edit...

GamePLayAnimationPanel

Figure 3.4: Key features of Unity 3D

Hand Animation: Unity 3D can be used to develop animation patterns for the avatar's hands in the context of VR Modeling unit for Hand. Realistic hand movements can be produced using the software's features for manipulating the motion of various objects, including bones and joints. Using scripting, users can design intricate hand movements that can be controlled by user input [42].
Gender Selection: User interfaces that enable gender selection from the MakeHuman models can be made using Unity 3D. Users can design menus and buttons that let users select the avatar's gender. Using scripting, the software can alter the avatar's appearance in accordance with the user's preferences [42].
Application Development: For creating applications that can be used on a variety of platforms, such as desktop, mobile, virtual reality, and augmented reality devices, Unity 3D is a potent tool. Unity 3D can be used to build an application that lets users interact with the avatar, control its movements, and discover its features as part of your project. The completed item can either be included in a bigger project or exported as a stand-alone application [42].
Integration with Other Softwares: Transmission Control Protocol/Internet Protocol (TCP/IP) and other communication protocols are supported by Unity 3D, and it is also possible to interface with external systems through web services and RESTful APIs. Due to this, Unity applications can communicate with other programs and hardware, including databases, sensors, and other hardware. For your project, Unity 3D can be combined with additional software tools like Blender and MakeHuman to produce a thorough workflow for building and creating interactive 3D models [43].

3.4.4.1 Description of TCP/IP

Transmission Control Protocol/Internet Protocol is referred to as TCP/IP. It is a collection of protocols that are employed for online and off-line communication. Data between programs running on devices connected to the internet is reliably and sequentially sent thanks to the TCP protocol. It guarantees the delivery of all data and confirms that it is sent appropriately. Data packets must be routed between network devices using the IP protocol. It is in charge of addressing and packetizing data for network transmission. TCP/IP together offer a strong and dependable way to communicate across the internet. User Datagram Protocol, on the other hand, is what it stands for. A connectionless communication service over IP networks is provided through a more straightforward protocol. UDP does not offer error checking or dependable data transport like TCP does. Instead, it is frequently utilized for applications like video streaming, gaming, or live audio broadcasting where speed is more important than reliability. When data needs to be distributed simultaneously to numerous receivers via broadcast or multicast communication, UDP packets are frequently utilized [44].

Both TCP/IP and UDP can be utilized for communication between apps in terms of how they are used to communicate with other Windows applications and for avatar movement animation patterns of hands and gender selection from MakeHuman models. For instance, Unity 3D can communicate with other programs like servers and databases using either protocol [44]. This can be helpful when creating real-time data-exchange-dependent applications, like multiplayer games or social networking software. Unity 3D can communicate with the MakeHuman program using any protocol for hand movement animation patterns and gender selection. Models from MakeHuman may be exported in a number of formats, such as .fbx, which can then be loaded into Unity. The information from the MakeHuman models may then be used by Unity to produce animations and manage the movement of the avatars. The data from the MakeHuman models can be used to control how the Unity avatars look to implement gender selection [45].

3.4.5 Windows Presentation Foundation (WPF)

Windows Presentation Foundation (WPF) is a graphical subsystem developed by Microsoft for building user interfaces in Windows-based applications. It is part of the .NET framework i.e. a Microsoft-created development environment that enables programmers to create a range of applications, including Windows programs, web programs, and web services. It provides a rich set of tools, libraries, and APIs for creating visually appealing and interactive user interfaces. WPF utilizes XAML (eXtensible Application Markup Language) as its markup language, allowing developers to define the user interface and layout in a declarative and intuitive manner. This separation of UI design and logic promotes a more efficient and collaborative development process [46].

One of the key features of WPF is its ability to create highly customizable and visually stunning interfaces. It supports vector graphics, animations, multimedia, and 3D rendering, enabling developers to create visually rich and engaging applications. WPF also offers a variety of controls and layout panels, allowing for flexible and responsive designs [46]. The WPF UI library (3.5) is specifically created for constructing contemporary user interfaces within the Windows Presentation Foundation (WPF) framework [47].

3.4.6 Iterations: Selection of Tools

When selecting tools for a project, various methods can be employed to evaluate and compare them. In the case of selecting tools for the project at hand, the following

Build Fluent experiences on Windows using WPF UI.

Figure 3.5: WPF UI is a toolkit that's intended for developing sleek user interfaces within a Windows Presentation Foundation (WPF) framework (This figure is taken from: https://wpfui.lepo.co/documentation/)

methods and criteria were considered:

3.4.6.1 Usability

The ease of use and intuitive nature of the tools were evaluated. This involved assessing the user interface, available documentation, and community support to determine how quickly and efficiently developers could learn and work with the tools.

3.4.6.2 Integration

The compatibility and integration capabilities of the tools with other required technologies and platforms were examined. This included assessing whether the tools could seamlessly work together and integrate into the existing software and hardware infrastructure.

3.4.6.3 Compatibility

The compatibility of the tools with the target platforms, such as Windows, Android, iOS, and Oculus Quest2, was an important consideration. Compatibility ensured that the developed application could run smoothly and effectively on the desired platforms.

3.4.6.4 Features and Functionality

The features and functionality offered by the tools were compared to determine their suitability for the project's requirements. This involved evaluating whether the tools provided the necessary capabilities for modeling, rendering, animation, control, and behavior implementation.

3.4.6.5 Community and Support

The availability of a strong and active community and support system around the tools was assessed. This included considering factors such as online forums, documentation, tutorials, and the frequency of updates and bug fixes. A robust support system was crucial for troubleshooting issues and obtaining assistance during the development process.

3.4.6.6 Industry Adoption

The level of adoption and usage of the tools within the industry was taken into account. Tools with a significant user base and industry recognition were given preference as they often indicate reliability, stability, and long-term support.

3.4.7 Tool Evaluation and Selection

By evaluating the tools based on these criteria, the selection process involved weighing the strengths and weaknesses of each tool, and ultimately, Unity 3D and MakeHuman were chosen as suitable tools for the project. The selection of Unity, MakeHuman, and the decision to discard WPF, Blender, and Mixamo can be justified based on various factors. Unity was chosen as the rendering platform due to its extensive features, cross-platform compatibility, and user-friendly interface. MakeHuman, on the other hand, was selected for its capabilities in modeling human avatars and facilitating the interchangeability of different visual properties.
The decision to exclude WPF, Blender, and Mixamo might have been motivated by specific limitations or considerations. These could include factors such as limited integration capabilities with other software applications, complexity of use, lack of desired features, or compatibility issues with the target hardware or software platforms. The analysis of these tools and platforms is crucial as it provides insights into why certain choices were made, why particular options were preferred over others, and why some options were omitted. This analysis ensures that the selected tools and platforms align with the project's objectives and requirements for VR/3D-Model development in the context of prosthetic design and training.

4

Materials and Methods

The process for developing and implementing the VR/AR hand prosthesis model is described in this section. This project's goal was to create a Unity 3D application that would let users select the gender of their avatar and manage hand motions. The program was created with a variety of potential purposes in mind, such as pain management, training, and simulation of the movements of a virtual arm. Several software programs, including MakeHuman [37], Mixamo [40], Blender [38], and Unity 3D [41], were employed to accomplish this goal.

4.1 Creation of Avatar

The primary tool of software chosen for this project to produce 3D models of men, women, and kids is called MakeHuman. The program was chosen because it can create human models with correct proportions and features that are realistic in appearance. Age, gender, and body shape were only a few of the parameters used to build the models, enabling for the creation of a wide variety of models. To expedite production and guarantee that the models were anatomically accurate, the default cmu option was utilized in MakeHuman. The CMU default posture serves as a standardized, realistic, well-structured and legally permissible foundation for MakeHuman's human anatomy models as shown in Figure 4.1 below.

The models were finished and exported as .fbx files, which are a popular file type for 3D models. After that, the files were imported into Unity 3D, a popular game engine and development environment for 3D apps. A crucial part of the project's development was the ability to create dynamic, editable 3D avatars through the integration of MakeHuman and Unity 3D.

4.2 Rigging of 3D Models

The three 3D models of a man, woman, and child were exported from MakeHuman as.fbx files, and then they were easily imported into Unity 3D by dragging and dropping the files into the project folder. The models had to be chosen as humanoid in the Unity inspector window in order to use the Humanoid Rigging System. This method makes it possible to control the movements of the arm, hand, and fingers, which was crucial for this project because the application that was developed includes the real-time movement of human avatar. To use the Humanoid Rigging System, the models have to be set in the Unity inspector window as humanoids. For the regulation of arm, hand, and finger movements, this system offers a set of bones

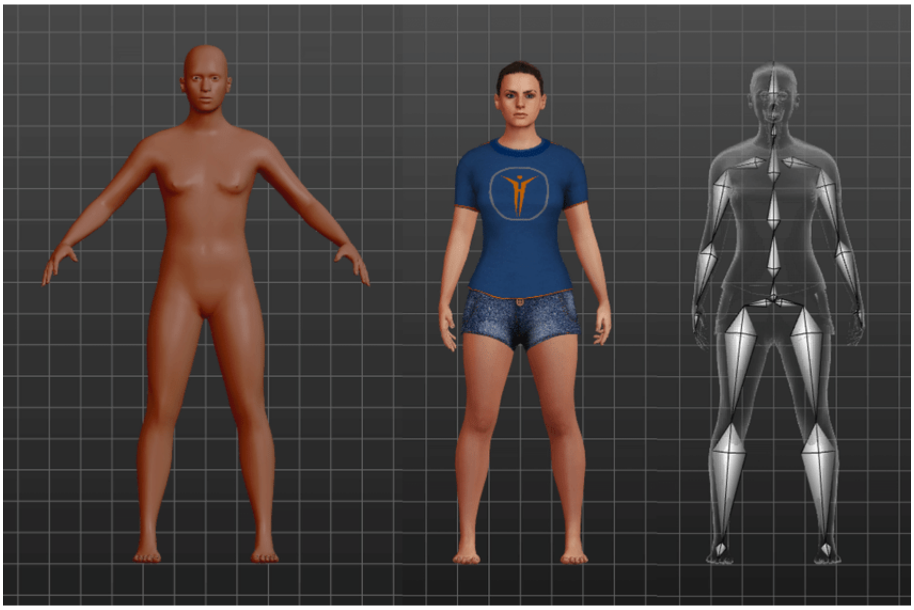


Figure 4.1: Creation of multiple avatar designs using MakeHuman

and restriction of angle.To ensure the avatar's proper alignment and movement, the humanoid option has to be chosen.

4.2.1 Humanoid Rigging System

Developers can rig 3D models with a preset bone structure that is best for human-like figures using Unity's Humanoid Rigging System feature as shown in Figure 4.2. For the 3D object to be animated, rigging entails building a virtual skeleton framework and controllers. To do this, virtual bones that could be controlled and animated to distort the model appropriately are attached to the model as manipulation points. The linked meshes deform when these bones rotate and move, resulting in animation of the 3D model [48]. By adopting this technique, the Humanoid Rigging technique's bone structure is automatically mapped to the bones of the 3D model, making the rigging process quicker and more effective. The Unity built-in animation tools, which are intended exclusively for human-like characters, can be used with the Humanoid Rigging System. This contains functions like animation retargeting, which enables animations to be applied to characters of various sizes and proportions without requiring manual modification, and inverse kinematics, which enables realistic movement of the 3D character's limbs and body [48].

The Humanoid Rigging System was required for this project in order to produce realistic and organic movements of the avatars for the virtual reality experience. The arm, hand, and finger movements needed to be carefully controlled in a way that would authentically mimic human behavior because the avatars were intended to represent real people. The Humanoid Rigging System was activated by choosing the models as humanoid in the Unity inspector window, allowing for the creation of expressive and lifelike animations for the avatars [48].

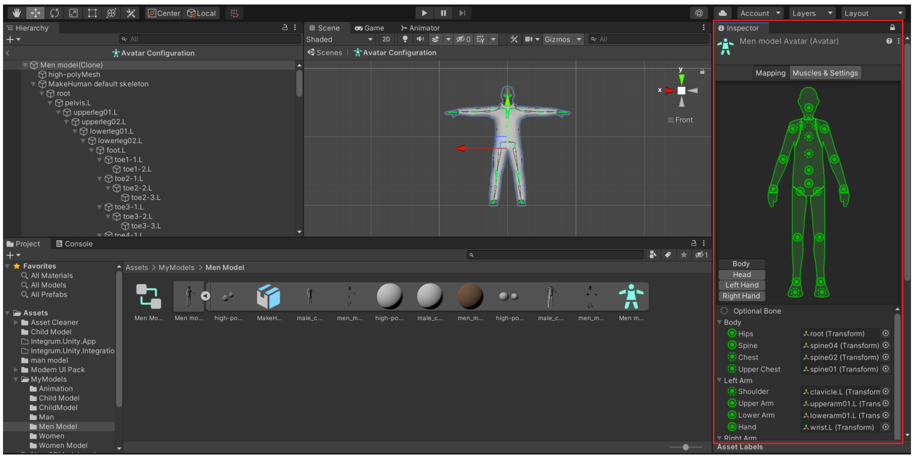


Figure 4.2: Humanoid Rigging System in Unity 3D

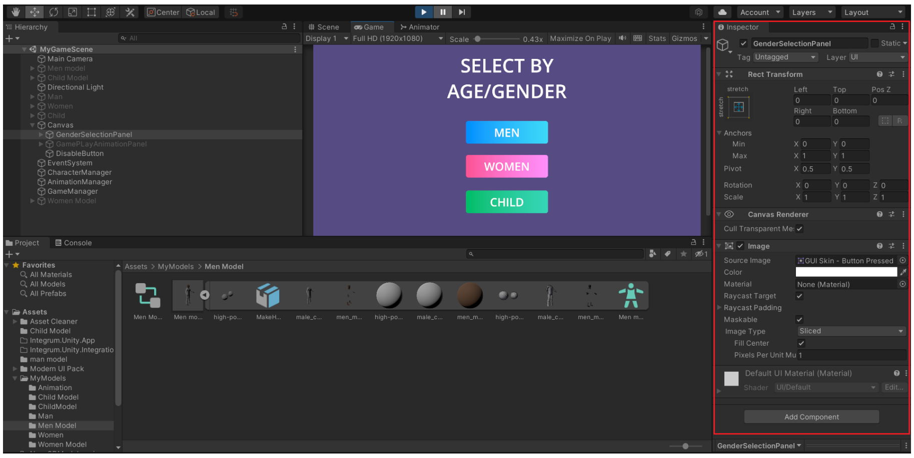
4.3 Implementation in Unity 3D

Additional features were added to Unity 3D to improve the user experience, including the choice of the 3D character's gender and the ability to animate hand gestures as follows:

4.3.1 a. Gender Selection

Unity 3D offers the ability to select a certain gender and age for the 3D models of a man, a woman, and a child. This procedure made use of the Humanoid Rigging System as well as a number of variables and scripts that enabled for the identification of gender and age. The first step was to create user interface (UI) elements, such as buttons or drop-down menus, that the user could interact with to select their chosen gender and age. Scripts tied to these UI elements were then used to control the 3D models in the scene. Designing UI elements that users might use to pick their gender and age was the first step in the procedure. A direct connection between the user's inputs and the corresponding changes in the 3D models was made possible by these scripts' connections to the UI elements. For instance, the associated scripts would carry out the required operations to modify the appearance of the 3D models in accordance with a user's selection of a given gender or age from a drop-down menu.

The user's choices might be used to manipulate the 3D models thanks in large part to the Humanoid Rigging System. This method made it possible to quickly transfer the UI input data to the 3D models' matching joints, bones, or morph targets. The gender and age of the 3D models could be calculated and changed dynamically based on user interactions with the UI components by leveraging this technology together with the parameters and scripts as highlighted in Figure 4.3. This made it easy to change the bones to reflect the user's preferred age and gender. To build the age selection function, the 3D models were resized based on the user's decision. For instance, selecting "child" would produce a 3D model that was scaled back, while selecting "adult" would produce a 3D model that was scaled up. Last but not least, the gender selection feature was implemented by altering the 3D model's mesh based on the user's preference. To do this, various game items (3D avatar) received male and female meshes (humanoid rigged models), which the user could enable or disable.



GenderSelectionPanel
Static
Tag Untagged
Layer UI

Rect Transform
stretch
Left 0
Top 0
Pos Z 0
Right 0
Bottom 0
Anchors
Min X 0 Y 0
Max X 1 Y 1
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Canvas Renderer
Cull Transparent Me: ✓

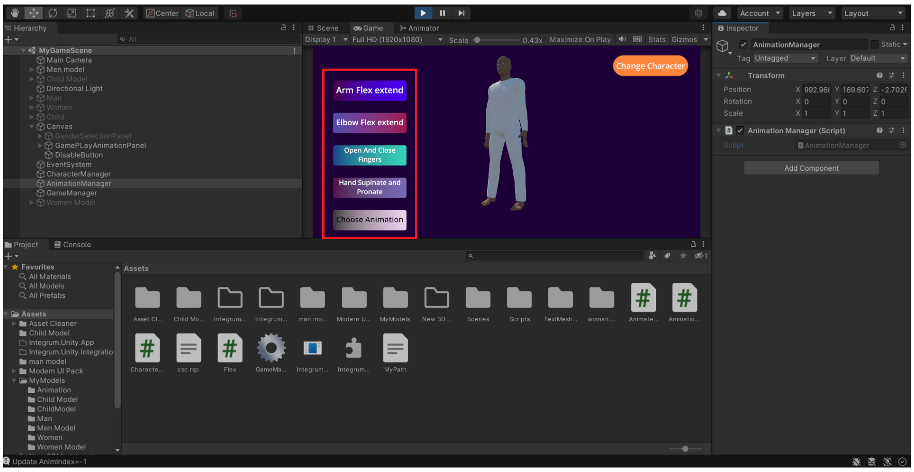
Image
Source Image GUI Skin - Button Pressed
Color
Material None (Material)
Raycast Target ✓
Raycast Padding
Maskable ✓
Image Type Sliced
Fill Center ✓
Pixels Per Unit Mu 1

Default UI Material (Material)
Shader UI/Default

Figure 4.3: The UI element feature of Unity 3D for gender selection

4.3.2 b. Hand Movement Animations

Several scripts were built using C# in Unity 3D in order to provide various hand movement animations to the 3D models. A different script was needed for each movement, and it was tied to the Hand Animator part of the 3D models. The Elbow Flexion/Extension movement script regulated the movement of the elbow joint, while the Flexion/Extension movement script controlled the bending and straightening of the fingers. The Hand Supination and Pronation script and the Finger Open and Close script were employed in a similar manner to rotate the hand to the left or right and to move the fingers respectively. To ensure that the hand and arm moved naturally and realistically, each script was meticulously planned and programmed. Once the scripts were associated to the Hand Animator component, user input from the keyboard or other input devices could cause them to run. The C# programming language script was created with the intention of animating an avatar in the Unity game engine. This code's function is to enable numerous hand movements for male, female, and child avatars, including flexion/extension of the wrist, elbow flexion/extension, hand supination and pronation, and finger opening and closing. LeftArmFlex, RightArmFlex, LeftElbowFlex, RightElbowFlex, LeftHandFinger, RightHandFinger, LeftAngleThumb, RightAngleThumb, LeftHandFlex, and RightHandFlex are just a few of the static float variables found in the code. The angle values of various hand movements are stored in these variables. The UnityServer class, which is used to link the server and client, is also present in the code. When a message is received, the server calls the ServerReceiveMessage function to handle it. The server listens for messages on port 12346. The function takes the AnimationCommand and angle values from the message and applies them to the command to take the necessary action. Additionally, it has a function called TriggerAnimationEvent that is called whenever a certain animation event is triggered. Based on the arguments supplied to the function, the function sets the AnimIndex and SelectedCharacter variables to the proper values. The SelectedCharacter variable is used to establish the avatar's model index, while the AnimIndex variable is used to set the avatar's animation index. This code aids in animating the avatar and giving it a variety of hand movements as shown in Figure 4.4 below.



AnimationManager
Static
Tag Untagged
Layer Default
Position X 992.96 Y 169.607 Z -2.702
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Script AnimationManager

```
Update AnimIndex=-1
```

Figure 4.4: Different Animation patterns for hand movement using scripting in Unity 3D

4.4 Development of User-Friendly Application

Creating a user-friendly application was a crucial first step in ensuring a satisfying user experience. The entire process of using the 3D model became more interesting and straightforward since it was simple for users to manage the hand movements of the avatar. Extensive testing was carried out to ensure a seamless user experience for the application. The hand movements were made natural and fluid, and the interface was made responsive. To avoid any unanticipated mistakes that can cause the application to crash or act in an unpredictable way, error handling was also implemented. When the program was finished, Unity 3D integrated it with the 3D model. This made it possible for the program to smoothly control the avatar's hand movements. The application now allowed users to control the hand movements of the avatar and view the outcomes in the 3D model in real-time as shown in Figure 4.5.

4.5 Integration with Windows Application

The Windows application being created as part of this project serves as a receiver for signals coming from different sources, such as the users, eOPRA implant system, or EMG from the amputees. There are several communication protocols like TCP/IP and UDP are used to integrate data from windows application into the Unity 3D. The following steps are part of the integration process:

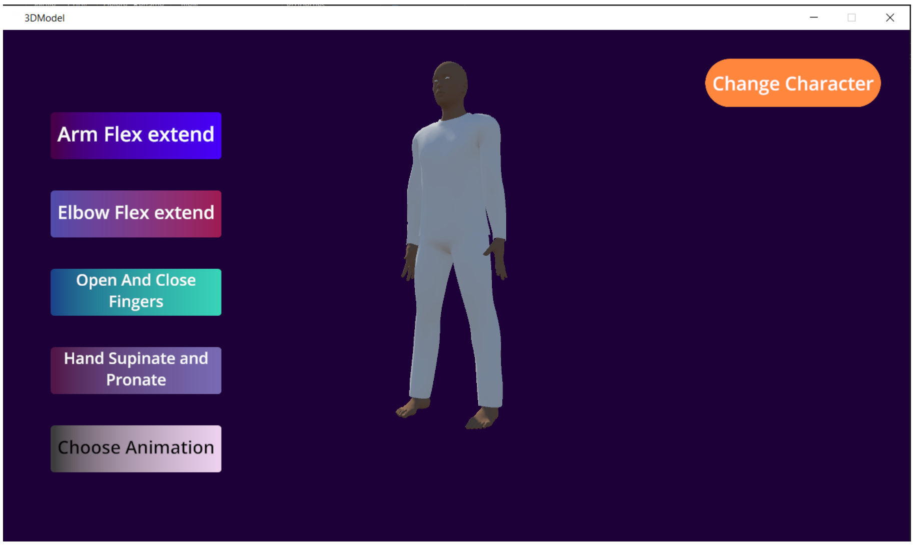


Figure 4.5: Real-time Movements of 3D model in Unity 3D application

1. **Signal Reception:** The eOPRA system, the EMG device, or the patient themselves can send signals/commands directly to the Windows application. These signals record the movements or instructions that the artificial limb controller is meant to receive.

2. **Processing with Artificial Limb Controller (ALC):** The Signals from e-OPRA implant system will transfer to the Windows Application through Artificial Limb Controller (ALC) and then this application processes the received signals and helps in visualizing it on canvas of the application. The signals are decoded by the ALC into usable directives for operating the artificial limb.

3. **Command Generation and Transmission:** Based on the signals that have been examined, the Windows application generates commands and gets them ready for transmission. These commands cause particular animations of the virtual avatar within the Unity 3D program and correlate to angular movements of limb.

4.6 Steps for integrating Windows applications with the Unity 3D application

The following steps are used to integrate Unity 3D with other applications:

1. **Avatar Selection from Unity 3D:** A library of avatars having different age and genders are available for the user to choose from in the Unity 3D application. The Unity 3D program allows the user to select the desired avatar for visualization through connection with the Windows application.

2. **Command Transmission through Windows Application:**
The Windows application serves as a conduit for sending the commands to Unity

3D application created by the ALC once the user has chosen an avatar. As seen in Figure 4.6, these instructions are communicated to the Unity 3D application, allowing for real-time control over the avatar's movements and the launching of animations.

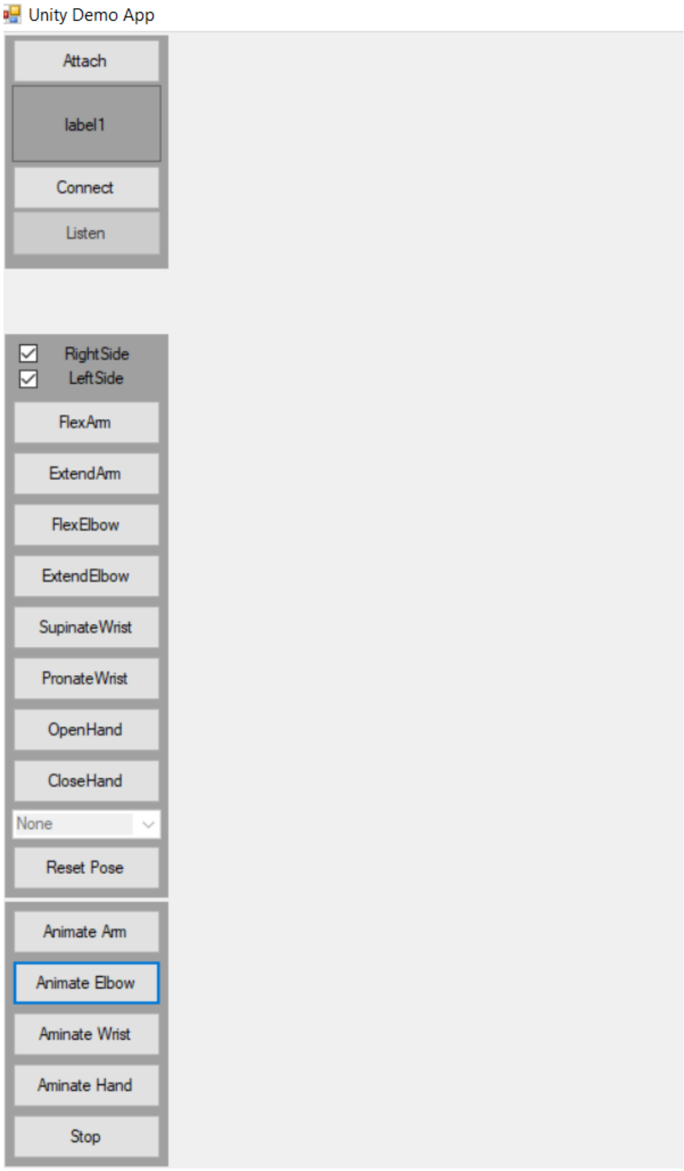


Figure 4.6: Overview of Windows Application for Integration in Unity3D's application

The final application includes an intuitive connection between the control signals received from outside sources and the visualization of those signals through the virtual avatar in Unity 3D by integrating the Windows program with the Unity 3D application.

5

Results

The research and implementation of various tools and platforms to build interactive 3D models for software applications and the creation of a software component that enables precise and autonomous control of upper limb movements led to the following main findings:

5.1 Description of the developed software unit

5.1.1 Software Functionality and Purpose

The generated software unit is VR modeling application made with Unity 3D that primarily aims to give users a thorough training environment. The software component tries to fill the gap between real-world training scenarios and virtual reality technology. A Windows platform has also been incorporated with the program, enabling fluid interaction and control. Future versions of the software unit may incorporate EMG (Electromyography) signals through Artificial Limb Control, allowing the system to react to muscle activity. To manipulate several parts of the virtual world, the software now accepts user commands. Avatar customisation is one of the software unit's key features. Users can customize their training experience by selecting the gender and age of the avatar. With this functionality, users can customize the program to meet their unique needs and learning goals.

The software unit's primary focus is on the exact control of the avatar's hand motions. Users can move the virtual hands in a variety of ways, including supination and pronation of the hands, finger opening and closing, elbow flexion and extension, and hand flexion and extension as shown in Figure 5.1. These hand movement controls are crucial for training users in tasks that call for dexterity and hand-eye coordination as well as for replicating real-life circumstances. The software unit provides a dynamic and immersive training experience by fusing VR technologies with simple hand movement controls. Users can engage with the virtual setting and control the hands of the avatar to carry out particular chores and actions for example: grabbing a ball, lifting a bag etc. This interaction encourages skill growth and improves the user's capacity to use newly acquired skills in practical contexts. The software unit enables users to participate in realistic training scenarios and gain fundamental skills that are applicable to a variety of sectors and businesses through avatar customisation and accurate hand movement control.

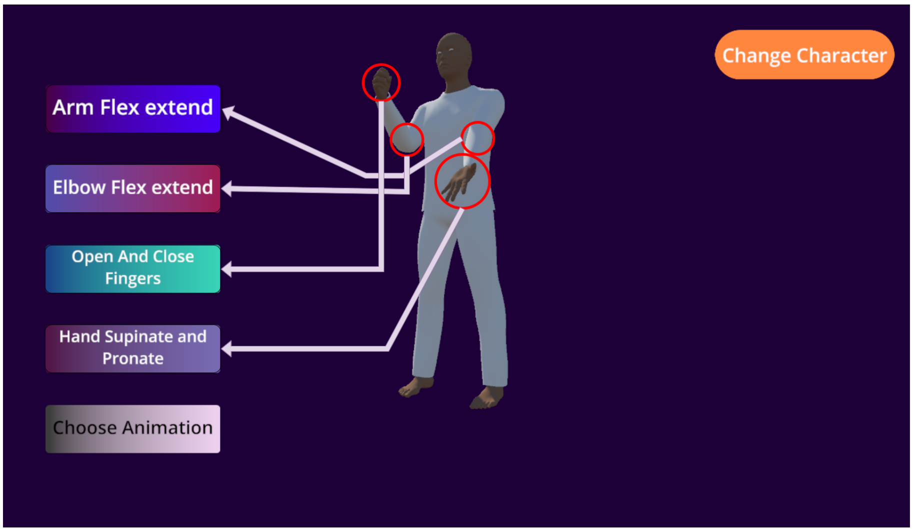


Figure 5.1: Hand Animation Patterns

5.1.2 Windows Application Integration and Avatar Control

A specific Windows application interface has been created to enable demonstration of the 3D-model and its integration to potential future EMG applications. By offering simple controls and customization choices, this interface acts as a conduit between the user and the virtual environment. The Windows program interface provides a number of buttons and menu items that make controlling the avatar's hand movements easier as shown in Figure 5.2. The "Attach" button is one of these options that is quite important. It connects the Windows app to the Unity software, which is used for VR modeling, when clicked. Real-time updates and synchronization are made possible by the flawless communication that is ensured by this connection between the two halves.

Users can click the "Connect" button to connect the chosen avatar to the Windows program once the connection has been made. With the use of this feature, users can tell the avatar's hands to move in whatever way they like. Users can specify and carry out sophisticated actions and gestures by connecting the avatar to the Windows software and taking exact control of the hand movements. The user's customization possibilities are further expanded by the Windows application interface. Users can choose between manipulating with their right or left hand through the interface.

5.1.3 Hand Movement Control Options: Windows Application Interface and Button Functions

The developed software unit's Windows application interface provides a number of hand movement control choices. Users have exact control over the avatar's hand movements because of the interface's design, which makes for a highly adaptable and realistic experience.

Users can choose either the left or right hand within the interface, giving them the ability to control each hand separately. Because users can concentrate on particular hand gestures or activities, this feature increases the amount of depth and realism in hand movements. The interface offers buttons for arm flexion/extension, elbow flexion/extension, supination/pronation, and hand open/close to start particular hand movements. Users can cause the matching action, which flexes the avatar's arm, by choosing a certain movement, such flexion. Users are given the ability to mimic natural hand gestures and movements by clicking these level of controlled movements in windows application as shown below in Figure 5.2.

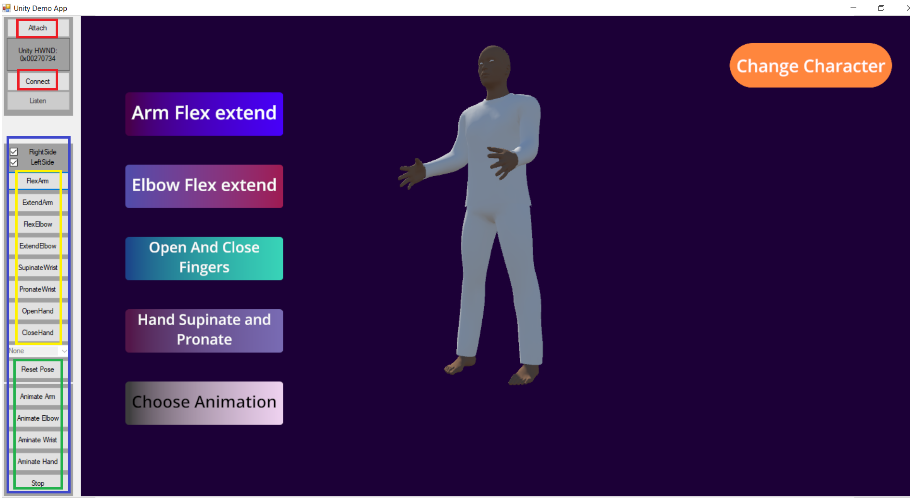

Unity HWND: 0x00270734

Figure 5.2: Animation controlled movement option in windows application

The software unit also provides a second way to employ mouse movements to regulate hand flexion. The flexion movement of the avatar's hand can be controlled by the user moving the mouse in the appropriate direction. Users may now move their hands more precisely and easily because to this feature's added flexibility and interactivity. Users have several options for interacting with the avatar and performing desired hand gestures using button and mouse-based control system. These choices provide for a more straightforward and engaging experience within the software unit and cater to various user preferences.

5.1.4 Pre-Defined Animations for Hand Movements

The created software unit includes a library of pre-defined animations to improve user experience and speed the hand movement selection procedure. These animations are made to offer users a variety of hand movement patterns that the avatar can choose from and carry out with ease.
Users have access to a selection of pre-defined animations for hand gestures within the software unit's interface. These animations feature a range of motions and gestures, including hand flexion and extension, finger opening and closing, elbow flexion and extension, and supination and pronation of the hands. Users can direct the avatar to continuously perform the chosen movement by selecting the right animation. When a certain animation is chosen, the avatar will do the associated hand movement without interruption until the user stops it by clicking the "STOP" button. This feature makes it possible for users to concentrate on watching and evaluating the hand movements without needing to continuously submit information,

making for a more engaging and dynamic training experience.

5.2 Different Camera positions

In addition to the functionality mentioned above, the developed application also allows users to visualize the 3D model from different camera positions. Users may view the avatar's hand movements more clearly and from different angles thanks to this feature. Users can interact with the 3D model by rotating it in the X and Y axes within the application's canvas as shown in Figure 5.3. Users can examine the avatar's movements and details from various perspectives thanks to this interactive feature, which offers a dynamic viewing experience.

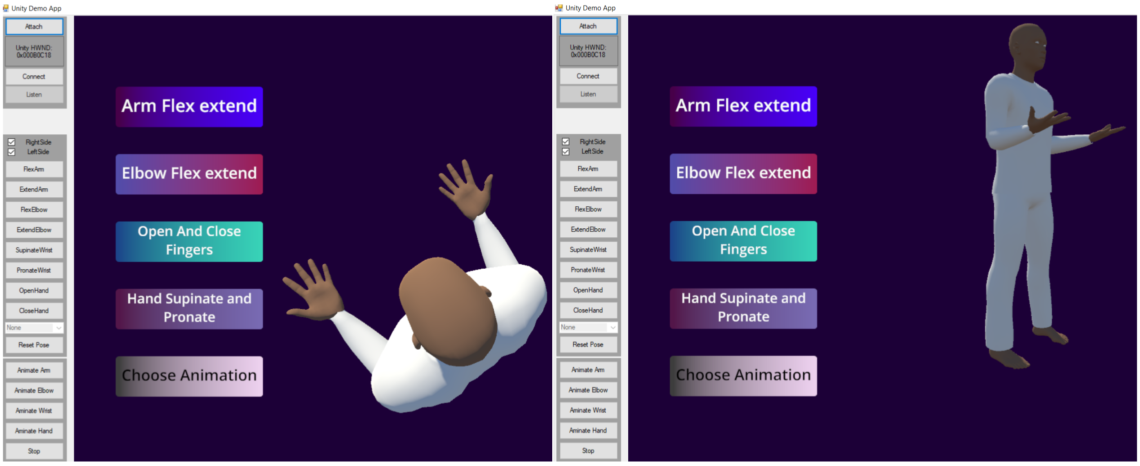


Figure 5.3: Different camera positions of 3D Model in Unity3D/Windows Application

This adaptability allows for particular training settings that need for concentrated control over a single hand.

6

Discussion

The discussion surrounding this project is covered in the following section.

6.1 Analysis of the project's results and findings

The subsequent section will delve into the project's results and findings, with a focus on highlighting the strengths and potential weaknesses of the software unit.

6.2 Analysis of the software unit features

The resulting software unit is a VR modeling application made with Unity 3D, with the main goal of giving users a thorough training environment. Its goal is to fill the gap between real-world training scenarios and virtual reality technology. The software unit offers seamless user interaction and control because it is integrated into a Windows platform. Although it may eventually incorporate EMG (Electromyography) impulses through an ALC (Artificial Limb Controller), the current functioning is dependent on user input.
Avatar customization is one of the software unit's key features; it enables users to tailor the learning experience by letting them choose the avatar's gender and age. With this functionality, the program can be tailored to meet certain needs and learning goals. The software unit's functionality and goal center on developing an intuitive and engaging VR modeling tool. The software unit enables users to participate in realistic training scenarios and learn key skills applicable to numerous domains of the prosthetic and medical industries. This is made possible by avatar customisation, fine hand movement control, and interaction with the Windows program. The software's adaptability and realism are increased by the pre-defined animations that are included, creating a smooth and engaging training environment. Furthermore, different camera positions improves the user's ability to carefully inspect and analyze the hand movements of the avatar by including these camera positioning and rotation features. Users can make precise assessments and get a thorough understanding of the model's behavior, which can be extremely helpful for training, research, and analysis in the field of prosthetic design and training.

6.2.1 Evaluation of Software Functionality

As an VR modeling application for thorough training, the evaluation of software functionality strives to ascertain the degree to which the built software unit successfully served its intended purpose and offered the expected functionality.The efficacy of two key components will be examined in this section: avatar customization and the accuracy and control of hand movements. It will focus on evaluating their performance.

6.2.1.1 Avatar Customization

A key component of the software component that seeks to give users a customized training experience is avatar customization. The software unit provides a level of personalisation that improves user engagement and immersion in the virtual environment by enabling users to personalize their avatars.

On the basis of a number of variables, the efficiency of the avatar modification option can be assessed. The variety of personalization possibilities that are offered to users is crucial, first and foremost. The software component ought to offer a wide range of options, such as the ability to choose the avatar's gender, age. Users can construct avatars that closely match their real-life counterparts or that reflect the traits important to their training objectives by providing a wide range of options.

The customizing interface's usability and intuitiveness should also be taken into account. The software component must have a user-friendly interface that makes it simple for users to browse the customisation choices and choose the features they want. The usability of the functionality can be further improved by providing clear instructions and visual cues, guaranteeing that users can effectively alter their avatars without becoming frustrated or perplexed.

Additionally, it is important to assess how the avatar modification option affects the training process as a whole. Virtual training settings can benefit from personalization in the same ways that other domains have shown to boost motivation and engagement. A well-designed avatar customisation tool can encourage a feeling of connection and ownership to the virtual world, enriching the training experience and making it more applicable to the specific user.

6.2.1.2 Precision and Control of Hand Movements

A key component of the software unit that adds to the realism and efficacy of the training experience is the accuracy and control of hand motions. In order to provide organic and intuitive interactions, the software unit should precisely capture and transform the user's hand gestures into the virtual environment.

Several aspects must be taken into account in order to assess the control and precision of hand movements. First and foremost, the hand tracking system's responsiveness is crucial. Any observable lag or delay between the user's actual hand movements and the corresponding movements of the virtual hands should be minimized by the software unit. A seamless and engaging training experience is made possible by the real-time tracking and representation of hand movements.

Another crucial factor to evaluate is the precision of the duplication of hand movements. The software unit must accurately mimic the user's hand movements in the virtual environment such that the virtual hands accurately imitate the user's gestures and activities. This accuracy is crucial for tasks that call for dexterous hand movements, accurate item manipulation, and fine motor coordination.

The software component should also offer simple controls for carrying out a variety of hand movements. A variety of hand actions, including hand flexion/extension, finger opening and closing, elbow flexion/extension, and hand supination and pronation, should be simple for users to understand and execute through the user interface. The user's capacity to imitate real-life situations and improve hand-eye coordination is improved through intuitive controls.

The software unit may give customers a realistic and immersive training experience that closely reflects real-world circumstances by successfully accomplishing precise and controlled hand movements. This level of control and realism improves the training platform's effectiveness by enabling users to hone their skills in a virtual setting before putting them to use in actual situations.

6.2.2 Integration with Windows Application

Numerous advantages result from the software unit's integration with a specific Windows application interface, and user interaction and control are significantly affected. This section covers the benefits of this integration and how it improves user experience overall by providing real-time updates and seamless communication between the Windows app and the Unity app. The "Attach" and "Connect" buttons' importance in creating connections and enabling precise control over hand movements is also highlighted.

There are various advantages to combining the software unit with a Windows application interface. First of all, it enables a more adaptable and natural user engagement. Utilizing user experience and lowering the learning curve, the specialized Windows software offers a recognizable interface to consumers. By using their familiar Windows conventions, users can explore menus, access settings, and carry out operations with ease, improving their comfort and productivity in completing daily tasks. Additionally, the integration enables friction-less communication between the Windows app and the Unity app. Changes performed in one application are quickly reflected in the other through real-time updates and data synchronization. For instance, changes made to the Windows app's avatar customization options can be immediately updated to the Unity app, creating a dynamic and synchronized experience.

Users gain access to a full and well-rounded training experience through the software unit's interaction with the Windows application interface. Users can easily transition between programs while keeping continuity in their training progress due to the smooth communication between the Unity app and the Windows app. Users have access to a variety of hand movement control choices through the Windows program interface, enabling complex and lifelike hand gestures and activities.

6.3 Available options for selection and control

This section examines the many options available to users for selecting and controlling each hand independently, emphasizing the operation of the buttons for arm and elbow flexion and extension, supination and pronation, and hand opening and closing. It also covers the alternative mouse-based control for hand flexion, which provides improved precision and ease of movement.

Users of the Windows program interface can choose and control each hand separately, giving hand movements a high level of freedom and realism. Users can accurately control virtual items and make complex gestures by selecting the desired hand. The interface includes buttons for elbow flexion and extension as well as arm flexion and extension. Users can practice bending and straightening their arms with these buttons, simulating the motion of real arms. Users can change the arm position and angle by pushing the corresponding buttons, which allows them to interact with the virtual environment realistically.

Likewise the interface has buttons for pronation and supination, which regulate how the forearm rotates. Users can replicate the rotation of their forearms by pressing these buttons, which makes it easier to do operations that call for twisting or turning motions. The overall immersive experience is improved by this functionality, which gives the hand movements an additional depth of authenticity. Another crucial component of the control options is the hand open/close buttons. By pressing these buttons, users can mimic the grip and release motions of real hands by opening or closing their hands. Users may make fists by pressing the hand close button, and they can extend their fingers by pressing the hand open button. Users can interact with virtual items using this functionality, grasping and manipulating them as needed for the training scenario, and using a variety of hand motions.

The Windows program interface offers mouse-based control as an alternative to the aforementioned control methods for controlling hand flexion. For applications requiring fine manipulation, in particular, this alternative offers higher precision and mobility. Users can more precisely alter the finger placements by using the mouse to flex their hand. The overall control and realism of the hand movements are improved by this feature, which is especially helpful in situations that call for delicate interactions or complex hand gestures. The Windows application interface gives users the ability to perform intricate and realistic hand gestures and actions by providing a wide variety of hand movement control options, such as buttons for arm flexion and extension, elbow flexion and extension, supination and pronation, and hand opening and closing. These choices increase training and encourage better hand-eye coordination by allowing users to engage with the virtual environment.

6.4 Performance and Effectiveness Evaluation of the Software Unit

The functionality and user experience of the software unit were evaluated in terms of performance and effectiveness using a variety of techniques. The evaluation techniques employed and proposed areas for improvement in light of found constraints and difficulties are covered in this section. Additionally examined were technical performance measures like load times, responsiveness, and stability. The software unit is tested several times during the evaluation phase by carrying out a number of predetermined activities and situations. The results and experience shed important light on the operation and performance of the software unit. Careful consideration has been given to factors including usability, realism, and the success of the training results. In addition, the application includes a modeled avatar that performs hand movements based on user commands. However, due to time constraints, certain aspects outlined in the original scope were not fully implemented. These include:

1. Execution of the Target Achievement Control (TAC) test, which involves mimicking a target movement by overlaying additional limbs onto the avatar's existing limbs, with separate control for each limb.

2. The ability to pick up, grab, and move objects for user training and machine learning purposes.

3. Integration of augmented reality, where the virtual limb is attached to the body of the end user.

While these functionalities were initially planned, they were not implemented within the given time limitations. Even though surveys, interviews, or usability tests were not used to collect user feedback, the evaluation procedure nonetheless yielded useful data for pinpointing areas that needed improvement. The restrictions and difficulties found during the tests can act as a springboard for more software unit improvements.

6.5 Ethical, Societal, and Ecological Considerations

The following are the ethical, Societal, and ecological implications of utilizing software unit development for the eOPRA/ALC system:

6.5.1 Ethical Considerations

The use of VR/AR/3D modeling with the eOPRA system-hand prosthesis creates significant ethical challenges that must be addressed in addition to the technical ones. The collection and utilization of personal information from amputees in future for modeling and software development raises ethical issues, as with any new technology. Consent, privacy, and data security all become crucial factors in assuring the proper handling of sensitive data. Concerns exist over the pricing and accessibility of these technologies as well, particularly for people residing in disadvantaged or marginalized areas. The price of acquiring and maintaining the required hardware and software may prevent individuals who might most benefit from them from having access to these solutions.The appropriate and inclusive use of these technologies depends on addressing these ethical concerns and guaranteeing equitable access to them. To enable people from different backgrounds to benefit from improvements in VR/AR/3D modeling and prosthetic technologies, it requires strong data privacy measures, open consent procedures, and initiatives to minimize the division due to digitization.

6.5.2 Societal Considerations

Additionally, the use of VR/AR/3D modeling and eOPRA system-hand prosthesis has important societal ramifications, especially for people who have lost an upper limb and depend on prosthetic devices. By increasing the design and functionality of hand prosthesis, these technical developments have the potential to significantly affect their quality of life, sense of freedom, and self-esteem. The creation and use of these technologies have effects that go beyond the level of the individual. Supporting the ongoing advancement and responsible application of VR/AR/3D modeling becomes more crucial as the societal impact and possible economic benefits become clear.

6.5.3 Ecological Considerations

It is critical to take into account any potential ecological effects connected to the creation and application of the eOPRA system, hand prosthesis, and VR/AR/3D modeling. Although these technologies have many advantages, they could potentially have negative effects on the environment. Environmental issues might arise during the production and disposal of prosthetic devices. However, the switch from analog to digital models and prototypes has the potential to produce less material waste. Utilizing telemedicine and virtual simulations can reduce the need for actual travel, which reduces carbon emissions and has a positive effect on the environment overall. It's critical to establish a balance between environmental sustainability and technical progress.

Overall, the application of VR/AR/3D-Modeling for the eOPRA system-hand prosthesis presents both opportunities and challenges in terms of sociological, ethical, and ecological ramifications. It is essential to consider these aspects during the design, development, and use of these technologies in order to guarantee their responsible and sustainable use.

6.6 Limitations of the project

Following sections describes the limitations of the project:

6.6.1 Lack of EMG Integration

The software unit's current iteration only relies on user commands for control, not EMG inputs. The software's inability to react to muscle action, which may offer a more dynamic and intuitive training experience, is hampered by this absence. By incorporating EMG technology, the software would be able to recognize and decipher muscle signals, allowing users to control hand movements more intuitively and naturally.

6.6.2 Platform Limitations

One of the project's main objective was to showcase the capabilities of the chosen rendering platforms, particularly Unity3D because it can operate independently of other Windows applications. Although this platform's integration and compatibility with the Windows environment were carefully examined, the study did not go into great detail regarding its potential for use with other platforms, such as iOS and Android. The evaluation gave the 3D model and avatar in the Windows desktop program a higher priority due to time restrictions and the project's scope.

6.6.3 Complexity of Hand Movement Controls

Although the software unit allows for fine control over hand movements, certain users, particularly those with minimal experience in VR environments, may find it challenging to operate the controls. The many combinations of button presses, mouse clicks, and hand motions required to perform specific hand movements may be too much for some users to handle. By providing users with additional instructions, tutorials, or interactive prompts inside the software, the usability and learning curve may be improved and the learning curve may be shortened.

6.6.4 Limited Hand Gesture Recognition

At the moment, the software unit mainly relies on human input through button presses and mouse movements to control hand movements. This approach provides the user control, but it cannot instantly detect and comprehend hand actions. If hand gesture recognition technology, such as machine learning algorithms or depth-sensing cameras, were integrated, the software would be able to recognize and comprehend hand motions. Users would be able to interact with the virtual environment in a more natural and intuitive way as a result. This would make the training experience more immersive and realistic overall, simulating real-world environments.
The functionality and user experience of the software unit would be improved by addressing these issues, making it more usable and immersive for a wider range of users.

6.7 Areas for Improvement

Several areas for development have been identified to strengthen the software unit and solve the noted limitations and issues based on the evaluation results and the team's observations. These include hand movement control options, user interface, functionality, and overall performance. Potential improvements that might be taken into account include the following:

6.7.1 Functionality

One improvement would be to look at various procedures to increase the variety of training scenarios and activities the software unit offers. To give users a more thorough training experience, this can entail adding extra modules, activities, or simulations.

6.7.2 User Interface

Enhance the user interface by making it more logical and appealing to the eye. To simplify user interactions and improve the overall user experience, think about implementing user-friendly design elements, such as simple navigation, instructive tool-tips, and adjustable settings.

6.7.3 Hand Movement Control Options

Enhance and broaden the range of options for controlling virtual hand movements so that users can make them more accurate and believable. This can entail experimenting with various input techniques or gadgets that support intricate and lifelike hand movements and activities.

6.7.4 Performance Optimization:

Continue to improve the software unit's performance to provide responsiveness and a seamless user experience. To deliver a seamless training experience, this may entail enhancing system stability, lowering latency, and optimizing resource consumption. The software unit can be improved to better match user wants and preferences and offer a more thorough and fulfilling training experience in VR settings by addressing these areas for development. Iterative reviews and regular user input gathering will help to spot and address new areas for development in subsequent versions.

6.8 Suggestions for future research and development in the field

Future research and development opportunities are many in the fast developing field of VR modeling and training. This section covers prospective research topics and offers ideas for future development for the software unit.

6.9 Exploring the Potential of Unity 3D: Compatibility, Portability, and Distributed Communication

In this evaluation, it is important to consider the potential of moving Unity to other platforms such as iOS and Android. One of the unforeseen positive effects of selecting Unity as the rendering platform is its compatibility with TCP/IP and UDP communication protocols, which enable distributed software applications. The implementation of TCP/IP and UDP communication between the Windows application and Unity 3D was a requirement for the project. This communication mechanism allows seamless data exchange and control between the two software components. Additionally, this implementation has the added benefit of making it possible to move the Unity application to different Windows systems or any other hardware platform that is compatible with Unity.

By leveraging TCP/IP and UDP communication, the project gains the flexibility to extend the Unity application to other platforms beyond the initial Windows environment. This opens up possibilities for deploying the application on mobile platforms such as iOS and Android, providing a wider reach and accessibility to users. Furthermore, the distributed nature of the software architecture enables the Windows application and Unity 3D to communicate and collaborate effectively, even if they are running on separate systems. This allows for scalable and flexible deployment scenarios, where the Windows application can be hosted on one machine while the Unity application runs on another. The selection of Unity as the rendering platform not only fulfills the project's immediate requirements but also offers the potential for future expansion to other platforms, due to its compatibility with TCP/IP and UDP communication protocols. This unforeseen advantage adds versatility and adaptability to the project's software ecosystem, enhancing its portability and extending its reach to a broader range of hardware platforms.

6.10 Evaluation of the software unit's performance and effectiveness

Several aspects can be taken into account while assessing the functionality and performance of the generated software unit, including:

6.10.1 User Feedback

Collecting input from software users can provide valuable insights into its performance and functionality. User interviews, surveys, and usability testing are effective methods for identifying areas of improvement and validating the software's ability to meet user requirements. However, it's important to clarify that the evaluation in this case focused on assessing the extent to which the original function scope was achieved and how effectively the chosen technology/tool facilitated the desired functions and overall project goals. It should be noted that gathering user feedback through user interviews, surveys, or usability testing was not part of the evaluation process.

6.10.2 Task Completion

An evaluation of the software unit's capacity to assist users in effectively completing training assignments can reveal its efficacy. Quantitative information on the software's performance can be obtained by monitoring task completion rates and accuracy.

6.10.3 Realism and Immersion

It is critical to assess the software unit's capacity to produce an immersive and realistic training environment. Through subjective assessments or questionnaires, user input on the degree of immersion, engagement, and perceived realism can be acquired.

6.10.4 Training Outcomes

It is crucial to examine how the software unit affects users' skill and knowledge growth. This can be determined by comparing assessments taken before and after training or by analyzing how well users perform in real-world situations after using the software.

6.10.5 Technical Performance

It is crucial to evaluate the technical performance of the software unit, including its responsiveness, stability, and compatibility with various hardware setups. Monitoring parameters like frame rate, loading times, and resource utilization can reveal information about the effectiveness and dependability of the system.

6.10.6 Usability and User Experience

The total effectiveness of the software unit can be determined by assessing the usability and user experience. User testing and surveys can be used to evaluate elements including usability, control clarity, and user satisfaction. Performance and efficacy of the software unit can be evaluated by taking into account these evaluation elements and using the proper evaluation techniques. The evaluations' findings will aid in pinpointing areas that require improvement and serve as a roadmap for further work on the software's usability and functionality.

6.10.7 Integration of EMG Technology

The incorporation of EMG technology into the software unit is one interesting area for future research. The program can offer a more dynamic and intuitive training experience by detecting and deciphering muscle signals [?]. By employing their own muscle activity to direct hand movements in the virtual environment, users would be able to closely resemble real-world situations. The software's interactivity and realism could be improved with more study and development in this area, increasing its usefulness for teaching.

6.10.8 Expansion of Platform Compatibility

Future development efforts should concentrate on expanding platform compatibility outside of the Windows operating system in order to increase the software unit's accessibility and reach. The development of versions that are compatible with various platforms would enable a wider range of users to take advantage of the software's teaching capabilities, especially in light of the popularity of other operating systems like macOS and Linux. The software's effect and acceptance would increase as a result of this expansion's ease of integration into various contexts and workflows.

6.10.9 Improved Usability and User Experience

Although the software unit allows for fine control over hand movements, there is still potential for usability development, especially for individuals with little prior experience in VR environments. Users can more easily comprehend and master the hand movement controls by being given additional instruction, interactive lessons, or intuitive prompts within the software, which lowers the learning curve and improves the user experience overall. Future studies can investigate creative methods to enhance the software's usability, making it more approachable and user-friendly for people with a range of skill sets [?].

6.10.10 Hand Gesture Recognition

Another area for future growth is the incorporation of hand gesture recognition technology into the software unit. The program can directly recognize and understand user hand motions by utilizing machine learning techniques or depth-sensing cameras. This development would make it possible to interact with the virtual environment in a more natural and intuitive way, increasing the training experience's realism and immersion. Future research should concentrate on investigating and putting into practice reliable hand gesture recognition methods, enabling users to perform actions in the virtual environment by employing gestures that closely resemble actual movements.

6.10.11 Target Achievement Control (TAC) test

In addition to the aforementioned recommendations, running a Target Achievement Control (TAC) test might offer insightful information for upcoming study and development. The TAC test entails studying and categorizing the particular actions carried out within the software unit, assessing their degree of difficulty, and pinpointing areas that need improvement. A thorough TAC test can help developers better grasp the software's advantages and disadvantages, which will guide further design choices and improvements. A useful methodology to assess the performance and efficiency of the software unit in accomplishing particular targets and tasks is the TAC test. Developers can make wise decisions about upcoming design modifications and enhancements by running a TAC test to acquire a deeper understanding of the software's strengths and shortcomings [?].
The following actions are part of the TAC test:

6.10.11.1 Task Identification

Within the software unit, particular tasks are identified and categorized at this step. These tasks might be simple ones like moving objects around in a virtual world or more intricate ones requiring deft hand-eye coordination [?].

6.10.11.2 Difficulty Assessment

Each recognized task is rated according to its degree of complexity. This evaluation assists in identifying potential difficulties users may experience while carrying out the tasks and sheds light on areas that need to be improved. The complexity of the required hand movements, the amount of time available, and the level of precision can all be used to classify difficulty levels [?].

6.10.11.3 Performance Evaluation

Participants in the TAC exam are required to carry out the specified tasks inside the software unit. Their performance is carefully monitored and documented, including details like completion time, accuracy, and the number of attempts necessary to hit the goal. This information offers numerical metrics that can be examined to evaluate the efficiency and performance of the software [?].

6.10.11.4 Feedback and Analysis

Participants' experiences, difficulties, and recommendations for improvement will be collected by developers after the TAC test has been completed. Future design decisions are influenced by this qualitative information, which when combined with quantitative performance data, aids in identifying areas that need improvement [?].

6.10.11.5 Iterative Development

This software unit could be improved using the information gleaned from the TAC test. Based on the shortcomings and potential areas for development, developers can do improvements [?]. Overall, the interaction, realism, usability, and accessibility of the software unit should be improved by further study and development. The field of VR modeling and training can continue to develop by taking into consideration the comments made and implementing cutting-edge technologies, giving users access to more immersive and effective training experiences [?].

7

Conclusion

The development of prosthetic technologies and applications such as this 3D/VR hand prosthesis software in e-OPRA Implant System and the Artificial Limb Controller (ALC) has led to significant progress in the field of prosthetics, offering amputees more natural and intuitive control over their prosthetic limbs, improving their mobility and quality of life. The effectiveness of prosthetics can be hampered by phantom limb discomfort, which neuromuscular connections in prosthetic systems have shown promise in alleviating.

Future advancements in prosthetic devices due to ongoing research in this area are anticipated to benefit amputees even more. The project's evaluated software unit showed encouraging results in terms of customizing avatars and demonstrating accuracy and control over hand movements, all of which helped to create a highly immersive and interesting virtual reality training experience. However, there is still potential for development, including enhancing the user interface, enhancing system efficiency, improving the functionality of the avatar customization feature, and expanding the hand movement control options and gesture detection capabilities. Addressing potential issues or constraints would also benefit from taking user feedback into account.

In summary, the software unit has the ability to provide an outstanding virtual reality training experience for a variety of sectors and applications, enhancing the development of prosthetic technologies to enhance the quality of life for those who have had amputations.

Bibliography

[1] Jönsson, S., Caine-Winterberger, K., Brånemark, R. (2011). Osseointegration amputation prostheses on the upper limbs: methods, prosthetics and rehabilitation. Prosthetics and Orthotics International.

[2] Hoyt, B., Walsh, S., and Forsberg, J. (2020). Osseointegrated prostheses for the rehabilitation of amputees (OPRA): Results and clinical perspective. Expert Review of Medical Devices.

[3] Brånemark, R., and Aszmann, O. (2018). Myoelectric signals and pattern recognition from implanted electrodes in two TMR subjects with an osseointegrated communication interface. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).

[4] Raspopovic, S., Capogrosso, M., Petrini, F., Bonizzato, M., Rigosa, J., Pino, G., Carpaneto, J., Controzzi, M., Boretius, T., Fernandez, E., Granata, G., Oddo, C., Citi, L., Ciancio, A., Cipriani, C., Carrozza, M., Jensen, W., Guglielmelli, E., Stieglitz, T., Rossini, P., and Micera, S. (2014). Restoring Natural Sensory Feedback in Real-Time Bidirectional Hand Prostheses. Science Translational Medicine.

[5] Engdahl, S., Christie, B., Kelly, B., Davis, A., Chestek, C., and Gates, D. (2015). Surveying the interest of individuals with upper limb loss in novel prosthetic control techniques. Journal of NeuroEngineering and Rehabilitation.

[6] Pasquina, P. F., Carvalho, A. J., and Sheehan, T. P. (2015). Ethics in Rehabilitation: Access to Prosthetics and Quality Care Following Amputation. AMA Journal of Ethics, 17(6), 535-546. doi: 10.1001/journalofethics.2015.17.6.stas1-1506.

[7] Watson, K.. What Causes Phantom Pain, and How Is it Treated? Healthline. Retrieved from https://shorturl.at/dlJ04. Accessed on 22nd April 2023.

[8] Fuchs, X., Flor, H., and Bekrater-Bodmann, R. (2018). Psychological Factors Associated with Phantom Limb Pain: A Review of Recent Findings. Pain Research and Management, 2018, Article ID 5080123, 12 pages. https://doi.org/10.1155/2018/5080123

[9] Mastinu, E. (2017). Embedded Controller for Artificial Limbs (Technical Report No. R008/2017). Retrieved from https://publications.lib.chalmers.se/records/fulltext/251770/251770.pdf

[10] Osborn, L. E., Ding, K., Hays, M. A., Bose, R., Iskarous, M. M., Dragomir, A., Tayeb, Z., Lévay, G. M., Hunt, C. L., Cheng, G., Armiger, R. S., Bezerianos, A., Fifer, M. S., and Thakor, N. V. (2020). Sensory stimulation enhances phantom limb perception and movement decoding. medRxiv.

https://doi.org/10.1101/2020.05.22.20109330 (Published in Journal of Neural Engineering, doi: 10.1088/1741-2552/abb861)

[11] Osborn, L. E., Ding, K., Hays, M. A., Bose, R., Iskarous, M. M., Dragomir, A., Tayeb, Z., Lévay, G. M., Hunt, C. L., Cheng, G., Armiger, R. S., Bezerianos, A., Fifer, M. S., Thakor, N. V. (2020). Sensory stimulation enhances phantom limb perception and movement decoding. medRxiv. doi:10.1101/2020.05.22.20109330.

[12] Petrini, F., Bumbasirevic, M., Valle, G., Ilić, V., Mijovic, P., Cvancara, P., Barberi, F., Katić, N., Bortolotti, D., Andreu, D., Lechler, K., Lešić, A., Mazic, S., Mijovic, B., Guiraud, D., Stieglitz, T., Alexandersson, A., Micera, S., and Raspopovic, S. (2019).Sensory feedback restoration in leg amputees improves walking speed, metabolic cost and phantom pain. Nature Medicine.

[13] Lendaro, E., Hermansson, L., Burger, H., Van der Sluis, C. K., McGuire, B. E., Pilch, M., Bunketorp-Käll, L., Kulbacka-Ortiz, K., Rignér, I., Stockselius, A., Gudmundson, L., Widehammar, C., Hill, W., Geers, S., and Ortiz-Catalan, M. (2018). Phantom motor execution as a treatment for phantom limb pain: protocol of an international, double-blind, randomised controlled clinical trial. BMJ open, 8(7), e021039. https://doi.org/10.1136/bmjopen-2017-021039

[14] Dietrich, C., Walter-Walsh, K., Preissler, S., Hofmann, G., Witte, O., Miltner, W., and Weiss, T. (2012). Sensory feedback prosthesis reduces phantom limb pain: Proof of a principle. Neuroscience Letters.

[15] Ahmad, A. (2011). PROSTHETIC PROBLEMS OF TRANSTIBIAL AMPUTEE. Journal of Postgraduate Medical Institute.

[16] Jiang, N., Rehbaum, H., Vujaklija, I., Graimann, B., and Farina, D. (2014). Intuitive, Online, Simultaneous, and Proportional Myoelectric Control Over Two Degrees-of-Freedom in Upper Limb Amputees. IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[17] Mastinu, E. Embedded Controller for Artificial Limbs. Ph.D. Thesis, Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, 2017

[18] Xiong, J., Hsiang, E. L., He, Z., Qian, Y., Cheng, S., Tang, J., ... and Hu, Y. (2021). Augmented reality and virtual reality displays: emerging technologies and future perspectives. Light: Science and Applications, 10(1), 216. https://doi.org/10.1038/s41377-021-00658-8.

[19] Lu, Y., Mao, P., Wang, G., Li, W., Shao, J., Huang, Y., ... and Shao, X. (2020). Spinal cord stimulation for chronic intractable trunk or limb pain: study protocol for a Chinese multicenter randomized withdrawal trial (CITRIP study). Trials, 21(1), 834. https://doi.org/10.1186/s13063-020-04768-3

[20] Gull, M. A., Bai, S., and Bak, T. (2020). A Review on Design of Upper Limb Exoskeletons. Robotics, 9(1), 16. https://doi.org/10.3390/robotics9010016

[21] Westberry, D. (2017). State of the Art: Amputation and Prosthetics.. Journal of Pediatric Orthopaedics.

[22] Powell, S., Cruz, R., Ross, M., Woodruff, M. (2020). Past, Present, and Future of Soft Tissue Prosthetics: Advanced Polymers and Advanced Manufacturing. Advanced Materials.

[23] Mastinu, E. (2017). Embedded controller for artificial limbs. (Master's thesis, Department of Electrical Engineering). Retrieved from https://publications.lib.chalmers.se/records/fulltext/251770/251770.pdf

[24] Ku, I., Lee, G., Park, C., Lee, J., Jeong, E. (2019). Clinical outcomes of a low-cost single-channel myoelectric-interface three-dimensional hand prosthesis. Archives of Plastic Surgery.

[25] Jiang, N., Rehbaum, H., Vujaklija, I., Graimann, B., and Farina, D. (2014). Intuitive, Online, Simultaneous, and Proportional Myoelectric Control Over Two Degrees-of-Freedom in Upper Limb Amputees. IEEE Transactions on Neural Systems and Rehabilitation Engineering.

[26] Ortiz-Catalan, M., Håkansson, B., Brånemark, R., and Delbeke, J. (2014). An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs. Science Translational Medicine, 6(257re6), 257re6. https://doi.org/10.1126/scitranslmed.3008933

[27] Barbosa, A., Lamounier Junior, E. A., Oliveira Andrade, A. de, Cardoso, A. (2012). Virtual and Augmented Reality: A New Approach to Aid Users of Myoelectric Prostheses. InTech. doi: 10.5772/50600 Fleming, A., Stafford, N., Huang, S., Hu, X., Ferris, D. P., and Huang, H. H. (2021). Myoelectric control of robotic lower limb prostheses: a review of electromyography interfaces, control paradigms, challenges and future directions. Journal of neural engineering, 18(4), 10.1088/1741-2552/ac1176. https://doi.org/10.1088/1741-2552/ac1176.

[28] Rutledge, Thomas Velez, Deborah Depp, Colin Mcquaid, John Wong, Garland Jones, R Atkinson, J Giap, Bosco Quan, Alex Giap, Huan. (2019). A Virtual Reality Intervention for the Treatment of Phantom Limb Pain: Development and Feasibility Results. Pain medicine (Malden, Mass.). 20. 10.1093/pm/pnz121.

[29] Ambron, E., Miller, A., Kuchenbecker, K. J., Buxbaum, L. J., Coslett, H. B. (2018). Immersive Low-Cost Virtual Reality Treatment for Phantom Limb Pain: Evidence from Two Cases. Frontiers in neurology, 9, 67. https://doi.org/10.3389/fneur.2018.00067

[30] Juan, W. (2011). A platform to render 3D model of mechanical production dynamic based on PaaS. Modern Manufacturing Engineering.

[31] Boonbrahm, P., Sewata, L., and Boonbrahm, S. (2015). Transforming 2D Human Data into 3D Model for Augmented Reality Applications. Procedia Computer Science.

[32] Boschmann, A., Neuhaus, D., Vogt, S., Kaltschmidt, C., Platzner, M., and Dosen, S. (2021). Immersive augmented reality system for the training of pattern classification control with a myoelectric prosthesis. Journal of neuroengineering and rehabilitation, 18(1), 25. https://doi.org/10.1186/s12984-021-00822-6.

[33] Simon, A. M., Hargrove, L. J., Lock, B. A., and Kuiken, T. A. (2011). Target Achievement Control Test: evaluating real-time myoelectric pattern-recognition control of multifunctional upper-limb prostheses. Journal of rehabilitation research and development, 48(6), 619–627. https://doi.org/10.1682/jrrd.2010.08.0149

[34] Sekhavat, Y., and Naamani, M. (2018). Projection-Based AR: Effective Visual Feedback in Gait Rehabilitation. IEEE Transactions on Human-Machine Systems, PP, 1-11. doi: 10.1109/THMS.2018.2860579.

[35] Carrino, F., Rizzotti, D., Gheorghe, C., Kabasu Bakajika, P., Francescotti-Paquier, F., Mugellini, E. (2014). Augmented Reality Treatment for Phantom Limb Pain. In: Shumaker, R., Lackey, S. (eds) Virtual, Augmented and Mixed Reality. Applications of Virtual and Augmented Reality. VAMR 2014. Lecture Notes in Computer Science, vol 8526. Springer, Cham. doi: 10.1007/9783319-074641_23.

[36] Phelan, I., Arden, M., Matsangidou, M., Carrion-Plaza, A., and Lindley, S. (2021). Designing a Virtual Reality Myoelectric Prosthesis Training System for Amputees. In Y. Kitamura, A. Quigley, K. Isbister, and T. Igarashi (Eds.), CHI EA '21: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-7). ACM.

[37] Briceno, Leyde and Paul, Gunther. (2019). MakeHuman: A Review of the Modelling Framework: Volume V: Human Simulation and Virtual Environments, Work With Computing Systems (WWCS), Process Control. 10.1007/978-3-319-96077-7_23.

[38] Joepal1. (2015). Documentation: Big dump from drupal. Retrieved from https://shorturl.at/gDLT9. Accessed on 3rd May 2023.

[39] Charnas, J., and Hartmann, T. (2023, March 27). What Is Blender (Software)? – Simply Explained. All3DP. Retrieved from https://shorturl.at/INVX5. Accessed on 3rd May 2023.

[40] Wikipedia contributors. (2023, May 1). Mixamo. In Wikipedia, The Free Encyclopedia. Retrieved from https://shorturl.at/opNS0. Accessed on 3rd May 2023.

[41] Schardon, L. (2023, January 13). What is Unity? – A Guide for One of the Top Game Engines. GameDev Academy. Retrieved from https://shorturl.at/eoILP. Accessed on 3rd May 2023.

[42] Kerckhove, E. V. d. (2022, March 14). Creating Reusable Characters With Blender and Unity. Kodeco. Retrieved from https://shorturl.at/mCL47. Accessed on 3rd May 2023.

[43] Vishnu Sivan. Device communications in Unity 3D. Retrived from: https://shorturl.at/djmJU. Accessed on 3rd May 2023.

[44] Forouzan, B. (1999). TCP/IP Protocol Suite (4th edition). McGraw Hill. Retrieved from https://shorturl.at/qrIQZ. Accessed on 5th May 2023.

[45] MakeHuman. (2021). Saving models for Unity and how to import them there. Retrieved from https://shorturl.at/gwxOV. Accessed on 5th May 2023.

[46] Jones, A., Freeman, A. (2010). Windows Presentation Foundation. In: Visual C# 2010 Recipes. Apress. https://doi.org/10.1007/978-1-4302-2526-3_17.

[47] WPF UI Documentation. Retrieved from: https://wpfui.lepo.co/documentation/. Accessed on 20th May 2023.

[48] Unity Technologies. (2023). Rig tab (Unity Manual). Retrieved from https://docs.unity3d.com/Manual/FBXImporter-Rig.html. Accessed on 20th May 2023.

A

Appendix 1

A.1 User Manual

Prosthetic Training in Virtual Reality

A.1.1 Introduction

Thank you for choosing our Prosthetic Training in Virtual Reality software. This user manual will guide you through the setup, usage, and features of the software. It is designed to provide a comprehensive training experience for amputees using virtual reality technology. By following the instructions in this manual, you will be able to customize avatars, control virtual hand movements, and engage in immersive training scenarios.

A.1.2 System Requirements

Before getting started, please ensure that your computer meets the following system requirements:
1. Windows operating system (compatible with Windows 10 or later)
2. VR headset (compatible with Oculus Rift or HTC Vive)
3. Motion controllers (compatible with Oculus Touch or Vive controllers)
4. High-speed internet connection

A.1.3 Installation

To install the Prosthetic Training in Virtual Reality software, follow these steps:
1. Insert the installation disc or download the software from our website.
2. Run the installer and follow the on-screen instructions to complete the installation process.
3. Once the installation is complete, launch the software.

A.1.4 Software Interface

Upon launching the software, you will be greeted with the main interface, which consists of the following elements:

A.1.5 Avatar customization panel:

Allows you to customize the appearance of your virtual avatar.
Hand movement control options: Provides buttons and controls to manipulate virtual hand movements.
Training scenarios: Offers a selection of training scenarios and activities to engage in.
Performance and settings: Allows you to adjust performance settings and access additional features.

A.1.6 Avatar Customization

To customize your avatar's appearance, follow these steps:
1. Click on the avatar customization panel to open it.
2. Choose from a range of options, including men, women and child.
3. Use the available controls and sliders to fine-tune the selected options.
4. Preview the changes in real-time and make adjustments as desired.
5. Once satisfied with the customization, click "Connect" and "Attach" to save the changes.

A.1.7 Hand Movement Control

The software provides various options to control virtual hand movements, including buttons for arm flexion/extension, elbow flexion/extension, supination/pronation, and hand open/close. Additionally, mouse-based control is available for precise manipulation. Follow these instructions to control hand movements:
Arm Flexion/Extension: Press the corresponding buttons to mimic the motion of your arms, changing the position and angle as necessary.
Elbow Flexion/Extension: Use the buttons to adjust the forearm's rotation, enabling twisting or turning motions.
Hand Open/Close: Press the hand open button to extend your fingers and the hand close button to make a fist.
Mouse-Based Control: Use the mouse to flex your hand and change finger placements for precise manipulation.

A.1.8 Training Scenarios

The software offers a variety of training scenarios and activities to improve hand-eye coordination and simulate real-life interactions. To access and engage in these scenarios, follow these steps:
1. Select the "Training Scenarios/Animation Pattrens" option from the main interface.
2. Browse through the available animations and choose one that suits your training goals.
3. Follow the on-screen instructions to complete the training scenario, using the hand movement controls to interact with virtual objects and perform required tasks.

4. Monitor your performance, completion time, and accuracy to track your progress.

A.1.9 Performance and Settings

The performance and settings section allows you to optimize the software according to your preferences. Here are some options you can explore:

A.1.10 Graphics settings:

Adjust the graphics quality to optimize performance based on your computer's capabilities.

A.1.11 Calibration:

Calibrate your VR headset and motion controllers for accurate tracking.

A.1.12 Help and Support:

Access user guides, FAQs, and contact information

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY