Physio-Fitness in VR: A New Dimension for Health Improvement

Bachelor Thesis

Author: Emmanouil Platakis (Student ID: 4730)
Supervisor: Prof. Papagiannakis George
Co-Supervisor: Dr. Kamarianakis Manos

Department of Computer Science
School of Sciences & Engineering
University of Crete
Heraklion, Greece

June 20, 2024

Abstract

In this thesis, I present the development of a Unity-based application designed for use in the fields of physiotherapy and fitness. Leveraging the MAGES 4.0 SDK, the application aims to assist users in selecting, learning, and implementing various exercises by providing both real-life instructional videos and 3D animated models for guidance. This dual approach allows users to gain a comprehensive understanding of proper exercise techniques, thus enhancing the effectiveness of their workouts and reducing the risk of injury.

Upon selecting an exercise within the application, users are presented with a video tutorial to their right, demonstrating the correct execution of the movement. To the left, a 3D model on a rotating platform allows users to examine the exercise from multiple angles. This model includes visualizations of the specific muscle groups engaged during the exercise, providing an additional layer of understanding. Additionally, the application offers detailed information and instructions for proper implementation, including recommended repetition ranges and safety precautions.

The central objective of this application is to empower users with the knowledge and tools needed to perform exercises correctly, whether for fitness training or physiotherapy rehabilitation. By incorporating virtual instruction, 3D modeling, and real-time feedback, the application seeks to provide a comprehensive resource for both fitness professionals and individuals seeking to maintain or regain physical health. The flexible and interactive design allows users to access exercise information on demand, reducing the need for repeat consultations with physical therapists or fitness trainers.

Furthermore, the application facilitates self-assessment by allowing users to view their own movements in a virtual mirror, enabling them to compare their form with the correct techniques displayed by the 3D model. This feature enhances user confidence and promotes adherence to proper exercise protocols, reducing the likelihood of injury and improving overall fitness outcomes.

In summary, this application serves as a versatile tool for fitness instructors, physiotherapists, and individuals undergoing physical therapy, providing a user-friendly platform for exercise instruction and muscle visualization. It has the potential to significantly improve exercise accuracy and safety, contributing to better health outcomes and greater accessibility to reliable fitness and rehabilitation resources.

Declaration of Authorship

I, Emmanouil Platakis, declare that this thesis titled, "Physio-Fitness in VR: A New Dimension for Health Improvement", and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signature: Emmanouil Platakis

Date: June 20, 2024

Acknowledgements

I would like to express my deepest gratitude to my supervising professor, Prof. George Papagiannakis, for his continuous support, invaluable guidance, and insightful feedback throughout the course of this project. His expertise and encouragement have been instrumental in shaping the direction and execution of this research.

I am also profoundly thankful to my supervisor, Dr. Emmanouil Kamarianakis, for his meticulous supervision, constructive criticism, and unwavering support. His contributions have significantly enriched my understanding and have been crucial to the successful completion of this thesis.

Special thanks to my technical supervisor, Nikolaos Marmaras, for his technical assistance, practical advice, and for sharing his extensive knowledge. His support has been indispensable in navigating the technical challenges encountered during this project.

Additionally, I would like to acknowledge the support of my family and friends, whose patience and encouragement have been a constant source of motivation.

Thank you all for your support and belief in my work.

Contents

List of Figures

1 Introduction

In today's fast-paced world, maintaining physical health and fitness is more important than ever. However, achieving this goal can often be challenging, particularly for individuals who lack access to professional guidance. To address this need, I have developed a cutting-edge application based on the Unity platform, specifically designed for use in physiotherapy and fitness training. This innovative application is intended to bridge the gap between users and expert instruction, providing a versatile and comprehensive tool for exercise education and implementation.

The core functionality of my application lies in its ability to assist users in selecting, learning, and performing various exercises correctly and safely. By leveraging both real-life instructional videos and advanced 3D animated models, the app offers a dual approach to exercise guidance. Users can view a video tutorial that demonstrates the correct execution of each movement on one side of the screen, while simultaneously examining a 3D model on the other side. This model is displayed on a rotating platform, allowing users to scrutinize the exercise from multiple angles and gain a deeper understanding of the proper techniques.

The application's primary objective is to empower users with the knowledge and tools necessary for effective and safe exercise performance. This is particularly beneficial for individuals undergoing physiotherapy rehabilitation or those aiming to improve their fitness levels. By incorporating virtual instruction, 3D modeling, and real-time feedback, the app provides a holistic resource for fitness professionals and individuals alike.

In summary, FitPhysioXR is a versatile and user-friendly tool designed to enhance exercise accuracy and safety. It serves as a valuable resource for fitness instructors, physiotherapists, and individuals seeking to maintain or regain physical health. By providing accessible and reliable exercise instruction and muscle visualization, the application has the potential to contribute significantly to better health outcomes and greater accessibility to high-quality fitness and rehabilitation resources.

1.1 Advanced Frameworks in Virtual Reality

Virtual reality (VR) is a technology that creates a simulated environment, allowing users to immerse themselves in and interact with a three-dimensional world that can mimic or differ significantly from the real world. VR's evolution has been remarkable, transforming from a futuristic concept into a tangible and increasingly ubiquitous tool across various industries.

The 21st century has witnessed exponential growth in VR technology, driven

by advancements in computing power, graphics processing, and sensor technology. The release of affordable and high-quality consumer VR headsets, such as the Oculus Rift, HTC Vive, and PlayStation VR in the 2010s, democratized access to VR, making it accessible to a broader audience beyond niche markets. These headsets featured improved resolution, reduced latency, and enhanced tracking capabilities, significantly enhancing the user experience. Alongside hardware advancements, software development has also flourished, with VR applications expanding into fields such as education, healthcare, real estate, fitness, and remote collaboration. The integration of VR with other emerging technologies like artificial intelligence (AI) and augmented reality (AR) continues to push the boundaries of what is possible, paving the way for more immersive, interactive, and realistic virtual environments. As VR technology continues to evolve, its potential to transform various aspects of daily life and industry grows, heralding a future where virtual experiences become an integral part of the human experience.

Inverse kinematics (IK) is a computational technique used to determine the positions and angles of joints in an articulated figure or robot to achieve a specific task, such as positioning an end-effector or tracker at a designated location. This process involves solving a system of nonlinear equations that link the joint parameters to the end-effector's spatial position and orientation. Several algorithms have been developed to solve the IK problem, such as Cyclic Coordinate Descent (CCD), Jacobian Inverse Method, Jacobian Transpose Method, Pseudo-Inverse of the Jacobian, FABRIK (Forward and Backward Reaching Inverse Kinematics), Analytical Methods, Heuristic and Optimization-Based Methods (including Genetic Algorithms, Particle Swarm Optimization, and Simulated Annealing) etc.

For instance, in the case of a two-link robotic arm, the IK problem involves finding the angles θ_1 and θ_2 that place the end-effector at a target point $p = (x, y)$. This requires solving equations that relate these angles to the end-effector's position based on the lengths of the links, l_1 and l_2.

In recent advancements, researchers have introduced MAGES 4.0 (Zikas et al., 2023), an innovative software development kit aimed at streamlining the creation of collaborative medical training applications in virtual and augmented reality (VR/AR). This novel solution operates as a low-code metaverse authoring platform, allowing developers to swiftly prototype high-fidelity, intricate medical simulations. MAGES 4.0 transcends conventional authoring limitations in extended reality, enabling networked participants to collaborate seamlessly using various VR/AR devices, as well as mobile and desktop platforms, within a unified metaverse environment.

This platform represents a significant upgrade to the traditional 150-year-old master–apprentice medical training model. Among its notable features are 5G

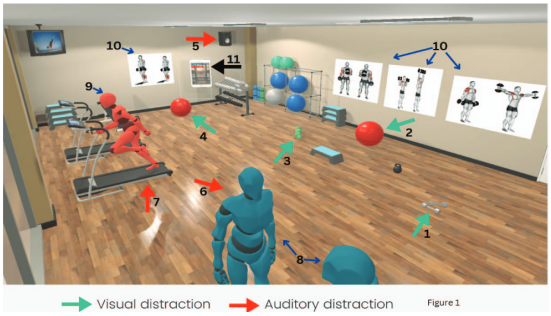

Figure 1: Virtual Gym Environment.

edge-cloud remote rendering and a physics dissection layer, realistic real-time simulation of organic tissues as soft bodies in under 10 milliseconds, a highly realistic cutting and tearing algorithm, neural network assessment for user profiling, and a VR recorder for recording, replaying, and debriefing training simulations from any perspective. These innovations collectively aim to revolutionize the field of medical training by providing a more interactive and realistic learning experience.

1.2 Fitness/Physiotherapy Applications

This section provides a comprehensive review of various studies and applications utilizing Virtual Reality (VR), Augmented Reality (AR), and other Extended Reality (XR) technologies in the fields of fitness and physiotherapy. These applications aim to enhance user engagement, provide tailored exercise experiences, and improve overall well-being.

Chong et al. describe a VR human-centric application designed for users with attention-deficit/hyperactivity disorder (ADHD). The application employs a virtual gym environment and fitness app as a motivating example scenario, as illustrated in Figure 1. In another study by Xu et al., a six-week intervention using an immersive VR (iVR) exergame was conducted to reduce anxiety, depression, and perceived stress among university students. This study also examined the usability and acceptability of such games, with the students engaging with the FitXR app.

Xu et al. developed a full-motion two-mode VR game that features both single- and multi-tasking modes. The study utilized a VR head-mounted display (HMD) and a 50-inch large display to collect data on users' game experiences, simulator sickness, and brainwave responses. In a related approach, Nair, Azman, Rahim, and Cheng introduced Endure, an AR horror-themed game that motivates users to stay fit by incorporating running into the gameplay.

Hansson presented a VR app that integrates several types of exercises, all connected through a gamified system of gaining points and trophies, which en-

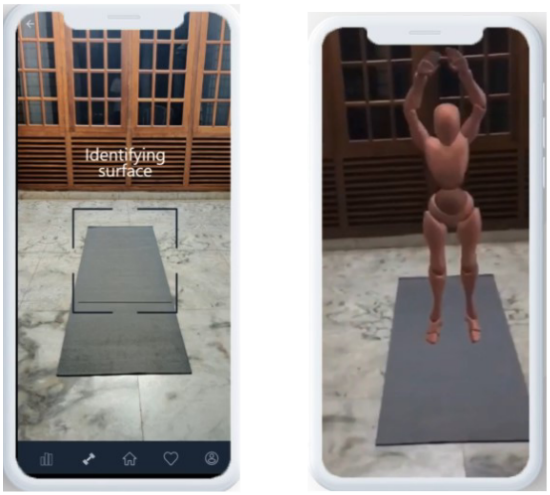

Figure 2: PumpFit app.

Figure 3: JARVIS app.

hances user motivation and engagement. Complementing this, Koech focused on a mobile augmented reality interface that allows users to interact and engage with online fitness data. The study defines the requirements for developing a mobile application for AR fitness activities.

Binoy and Srivastava developed a mobile AR app aimed at improving gym exercises and workouts, making them more effective and less strenuous. This app features a virtual instructor that demonstrates exercises and provides corrections if the exercises are performed incorrectly, as shown in Figure 2. Additionally, Rabbi, Park, Fang, Zhang, and Lee introduced JARVIS, a virtual exercise assistant that provides an immersive and interactive gym exercise experience. JARVIS guides users through the proper way of performing exercises in a virtual environment, depicted in Figure 3.

Taware, Kharat, Dhende, Jondhalekar, and Agrawal described a system that tracks body movements and counts exercise repetitions using the Mediapipe Pose Estimation Model. This AI-based workout tracking app is illustrated in Figure 4. In another medical application, Jin, Monge, Postolache, and Niu developed a mobile-AR application for stroke patients, immersing them in a mixed environment that combines real-world and virtual objects. The interface is materialized through an iPhone, a head-mounted 3D display, and a set of wireless sensors for measuring physiological and motion parameters.

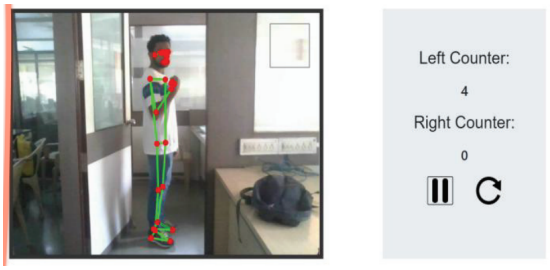

Figure 4: AI-based workout app tracking.

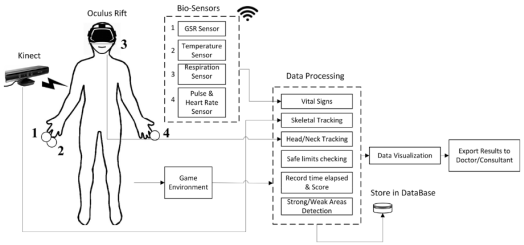

Figure 5: Physiotherapy system block diagram.

Baqai, Memon, Memon, and Shah designed a VR game for patients to perform routine exercises without requiring physical intervention. This game enhances joint and muscle mobility while providing continuous monitoring of vital signs using biosensors, as shown in Figure 5. Similarly, da Silva, Pacagnelli, de Almeida, and Siscoutto introduced PhysioAR, an AR app designed to assist in respiratory therapy treatments for hypertensive patients. This app uses three-dimensional virtual environments and specific hardware to measure vital signs, enabling physiotherapists to create and apply therapy sessions.

Postolache et al. developed a VR physiotherapy game characterized by a Kinect natural user interface. The game is accompanied by a mobile application that provides patient electronic health records, game remote configuration, and data presentation for physiotherapists. In a subsequent study, Postolache et al. presented an XR serious game designed to complement regular clinical training, where mechanical equipment is commonly used. These XR-based solutions are part of an IoT Physical Rehabilitation ecosystem that ensures data storage for individual exercises performed by users.

The *FIT XR* app blends gamification with boxing, dance, and high-intensity interval training (HIIT) movements to encourage users to adopt a healthy lifestyle. This VR fitness game/app is illustrated in Figure 6. Similarly, *Fyter: AR Fitness Workouts* is a mobile-AR fitness app that provides access to an extensive library of workouts. It monitors each repetition with skeletal tracking to ensure proper form and technique.

iMuscle 2 is a health and fitness app that assists users in planning and tracking their fitness programs. The app uses an anatomically correct model to iden-

Figure 6: FitXR app.

Figure 7: iMuscle 2 app.

tify muscles and demonstrate exercises, as shown in Figure 7. Additionally, *Gymnotize Fitness Workout App* is a mobile-AR app offering a variety of weight-based exercises. It tracks repetitions, allows users to create customized workouts, and uses AR to demonstrate proper lifting techniques, as depicted in Figure 8.

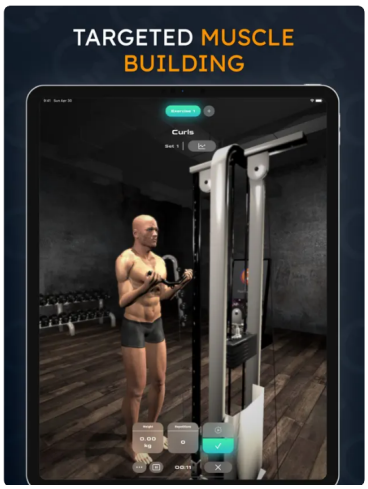

Figure 8: Gymnotize app.

A series of papers by Alturki and Gay, Barsasella et al., Liu, Menhas, Dai,

Saqib, and Peng, Bansal, Rajgopal, Chamola, Xiong, and Niyato, and Hsiao highlight the benefits of using XR technology for healthcare and well-being in general. Additionally, research by Menhas, Qin, Saqib, and Younas and Peng, Menhas, Dai, and Younas discusses the advantages of VR applications during the COVID-19 pandemic, emphasizing their support for mental and physical health during periods of social isolation.

Finally, studies by Qian, McDonough, and Gao, Lucena-Anton, Fernandez-Lopez, Pacheco-Serrano, Garcia-Munoz, and Moral-Munoz, Brepohl and Leite, Brady, Dejaco, Lewis, McCreesh, and McVeigh, and Viglialoro et al. review the use of AR and VR applications in physiotherapy, detailing the advantages and effectiveness of these methods in clinical settings.

1.3 Challenges and Proposed Solutions

In developing the Unity-based application for physiotherapy and fitness, several key solutions were implemented to address various challenges and enhance user experience and effectiveness. One of the primary challenges was effectively illustrating muscle engagement during exercises. To solve this, I implemented a muscle highlighting framework that gradually highlights muscles instead of using a single color or static animations. This solution provides users with a more intuitive and dynamic understanding of muscle activation throughout the movement. By gradually highlighting muscles, users can better visualize how different parts of the body are engaged during each phase of an exercise, which fosters a deeper comprehension of proper techniques and the physiological impacts of their workouts. This method also makes the learning process more engaging and less overwhelming, as it allows users to see the progression of muscle activation in a more natural and informative manner.

In addition to the muscle highlighting framework, integrating both real-life instructional videos and 3D animated models addressed the challenge of catering to diverse learning styles and providing a comprehensive educational experience. Real-life videos offer a relatable and realistic depiction of exercises, helping users to connect with the material on a personal level. Conversely, 3D models provide a level of detail and flexibility that videos alone cannot achieve, such as multi-angle views and interactive features. This dual approach ensures that users can fully understand the mechanics and nuances of each exercise, leading to better performance and reduced risk of injury. By combining these two mediums, I created a robust solution that enhances user comprehension and engagement.

Another significant challenge was enabling users to monitor and correct their form in real-time, which is often facilitated by a personal trainer or physiothera-

pist. To solve this, I developed the application's interactive design, particularly the inclusion of a virtual mirror for self-assessment. This feature empowers users to compare their movements directly with the correct techniques displayed by the 3D model, enhancing user confidence and adherence to proper exercise protocols. The virtual mirror solution facilitates immediate correction of mistakes and encourages continuous improvement and self-directed learning, thus mimicking the experience of receiving live feedback.

Furthermore, the challenge of providing accessible and convenient exercise guidance was addressed by ensuring on-demand access to detailed exercise information and instructions. Frequent consultations with physical therapists or fitness trainers can be time-consuming and costly, so my solution was to create a resource that users can utilize at their own pace and convenience. This flexibility is particularly beneficial for individuals with busy schedules or those who may have limited access to professional fitness and rehabilitation services. By making high-quality exercise guidance readily available, the application supports users in maintaining or regaining their physical health more effectively and independently.

In summary, the solutions incorporated into this Unity-based application were driven by a commitment to enhance user experience, improve learning outcomes, and increase accessibility to reliable fitness and rehabilitation resources. Through dynamic muscle highlighting, the integration of real-life videos and 3D models, interactive self-assessment features, and on-demand access to information, the application provides a comprehensive and user-friendly platform that meets the diverse needs of its users.

2 FitPhysioXR

The Unity-based application developed in this thesis embodies a dynamic solution tailored to the realms of physiotherapy and fitness. It stands as a robust tool facilitating exercise selection, learning, and execution through a novel dual instructional paradigm. This innovative approach seamlessly integrates real-life instructional videos with immersive 3D animated models, offering users comprehensive insights into exercise mechanics and muscle engagement from various perspectives. Complemented by detailed guidance on exercise implementation, including recommended repetition ranges and safety precautions, the application empowers users with the expertise required for precise exercise execution, thereby mitigating the necessity for frequent professional consultations. Moreover, the application introduces a cutting-edge self-assessment feature, enabling users to scrutinize their movements in a virtual mirror and compare them with the ideal techniques illustrated by the 3D model. In essence, this application represents

a sophisticated platform for exercise instruction and muscle visualization, catering to diverse stakeholders, including fitness instructors, physiotherapists, and individuals undergoing physical therapy, with the potential to markedly enhance exercise precision and safety.

2.1 Scope/Description

Utilizing the developed Unity-based application offers numerous compelling advantages for individuals engaged in physiotherapy or fitness endeavors. Firstly, it provides a comprehensive and user-friendly platform that streamlines the process of selecting, learning, and executing exercises. With its unique blend of real-life instructional videos and interactive 3D animated models, the application offers unparalleled insights into exercise mechanics and muscle engagement, fostering a deeper understanding of proper techniques. This not only enhances exercise efficacy but also reduces the risk of injury, making it an invaluable tool for individuals seeking to optimize their workout routines or undergoing rehabilitation. Moreover, the application's flexibility and accessibility empower users with the knowledge and tools needed to achieve their fitness goals independently, minimizing the reliance on costly and time-consuming professional consultations. Furthermore, the innovative self-assessment feature facilitates continuous improvement by allowing users to monitor and refine their form in real-time, thus promoting adherence to proper exercise protocols and bolstering confidence in one's abilities. In summary, the developed application represents a transformative solution that not only enhances exercise precision and safety but also empowers individuals to take control of their physical well-being, making it an indispensable asset for fitness enthusiasts, physiotherapists, and individuals on the path to recovery alike.

2.2 Features and Novelties

The application exhibits a range of distinctive features and innovations aimed at transforming the exercise experience. Notably, it combines real-life instructional videos with immersive 3D animations, offering users a multifaceted learning environment. Unlike conventional approaches, this integration provides users with dynamic perspectives, enhancing comprehension and engagement. Moreover, the application introduces a pioneering muscle highlighting framework (See implementation details in Section 3.1) that dynamically illuminates specific muscle groups during exercises. This unique feature offers users a comprehensive understanding of muscle engagement, promoting safer and more effective workouts.

Furthermore, the application incorporates an interactive virtual mirror for

real-time self-assessment, allowing users to compare their movements with correct techniques demonstrated by the 3D models. This hands-on approach facilitates immediate feedback and adjustment, fostering user confidence and adherence to proper form. Additionally, the application offers detailed exercise information, including recommended repetition ranges and safety precautions, ensuring users have access to comprehensive guidance for optimal performance and injury prevention.

The integration of inverse kinematics (IKs: See implementation details in Section 3.2) and upper body tracking technologies represents a pivotal advancement in the development of my application. These sophisticated techniques profoundly enhance the realism, interactivity, and effectiveness of the user experience. By leveraging IK algorithms, the application simulates natural human movement with unparalleled accuracy, ensuring lifelike animations of exercises. Additionally, the incorporation of upper body tracking enables users to interact with the application using natural gestures and motions, further enhancing immersion. Collectively, these advanced technologies empower users with more engaging and immersive exercise demonstrations, leading to a deeper understanding and improved execution of proper exercise techniques.

In essence, the application represents a groundbreaking solution that redefines exercise instruction and visualization. By leveraging innovative features such as real-life videos, dynamic muscle highlighting, interactive self-assessment, inverse kinematics and upper body tracking, it empowers users to elevate their fitness journey and achieve their goals with precision and confidence.

3 Software Design and Results

In this section, I present the software development process and the results achieved through the implementation of my Unity-based application. This application was designed with a focus on cross-platform compatibility, allowing it to be exported to various platforms including Windows, iOS, Android, and more. The flexibility and versatility of Unity as a development environment enabled us to integrate several advanced features to enhance the user experience. These features include the muscle highlighting framework, inverse kinematics for upper body tracking and the fact that the framework is easily extendable for adding more exercises.

3.1 Muscle Highlighting Framework

The muscle highlighting framework implemented in the application serves as a versatile and dynamic solution designed to enhance users' understanding of exer-

cise mechanics and muscle engagement. Leveraging a rigged model and employing a modular architecture consisting of various interconnected classes, this framework offers extensive customization options and facilitates seamless integration of new features. At the heart of this framework lies the `TextureMuscleActivator` class, serving as the central component responsible for orchestrating muscle activation and visualization.

This framework is built upon a robust architecture comprising multiple classes that interact with each other to achieve the desired functionality. A fundamental component of this architecture is the Body class, which serves as a mapping entity for references to the joints of the rigged model. While not the main function, the `Body` class plays a crucial role in providing essential infrastructure for accessing joint references and facilitating communication between different components of the framework. The framework's diagram is illustrated in Figure 9 and an example of its use is shown in Figure 10.

The muscle highlighting framework is highly extensible, allowing developers to easily incorporate new muscles, joints, and activation criteria. Through the use of inheritance and composition, different classes within the framework collaborate to achieve complex behaviors while maintaining modularity and flexibility. This extensibility enables users to tailor the framework to their specific requirements and integrate additional functionalities as needed.

The heart of the muscle highlighting framework lies in its ability to dynamically adjust the color of muscle materials based on joint rotation. The selection of rotational thresholds for muscle activation was a critical step in the development of this feature. These thresholds were chosen based on a combination of factors: the anatomical position of the joints, my personal experience with exercise and muscle engagement, and the expert advice provided by my gym coach. This multi-faceted approach ensured that the thresholds accurately reflect realistic and effective muscle activation levels during various exercises.

Through a systematic evaluation process, each joint's rotation is compared against predefined rotation thresholds for maximum muscle intensity. As illustrated in Figure 21 in the Appendices section, the `AddMovement` function systematically associates each muscle with the respective minimum (left) and maximum (right) rotational thresholds of the joints upon which the muscle activation is contingent. Depending on the outcome of the evaluation of the comparison between current joint's rotation and the thresholds for maximum intensity values, the corresponding muscle's color is modified accordingly, providing users with real-time feedback on muscle activation levels.

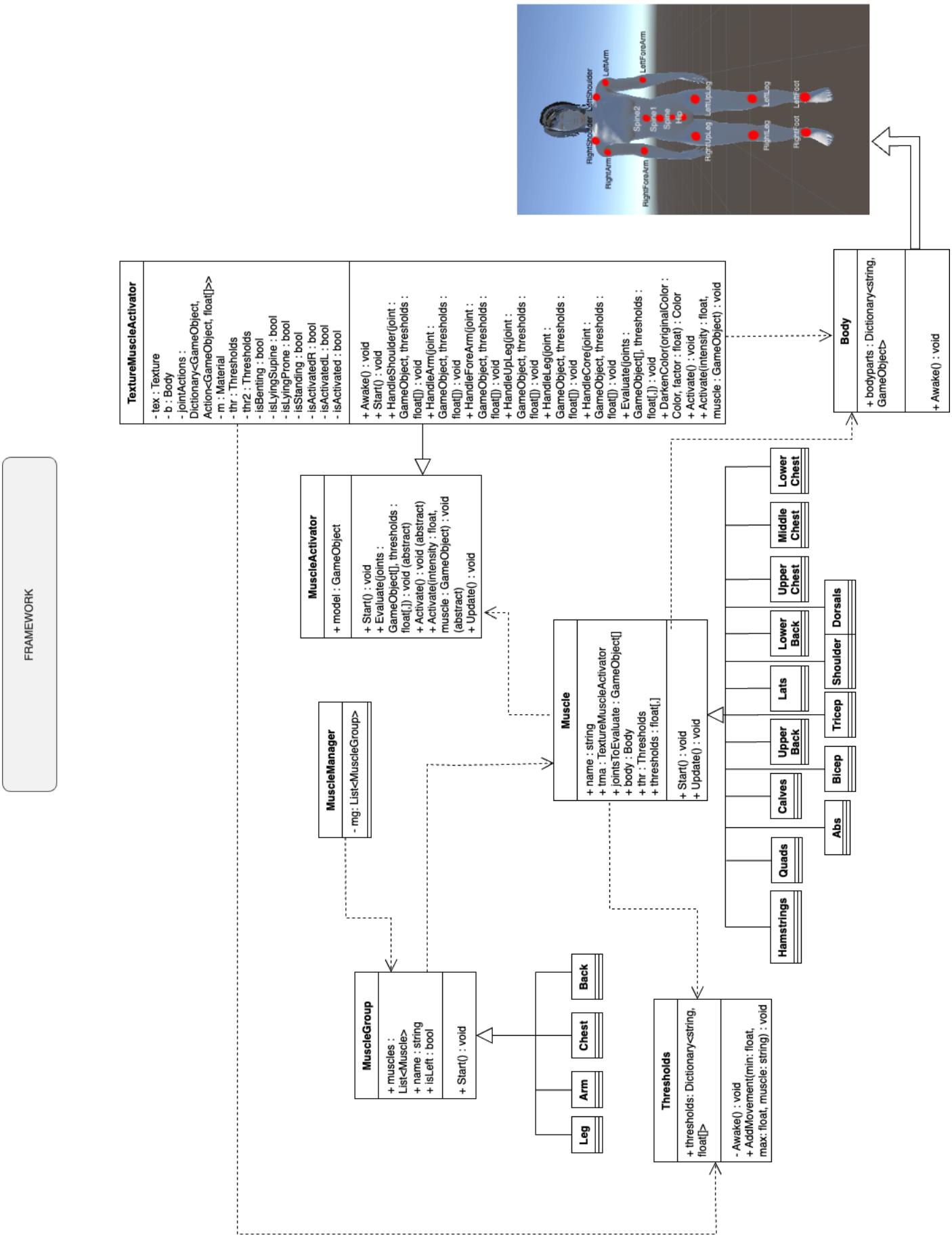

Figure 9: Framework diagram.

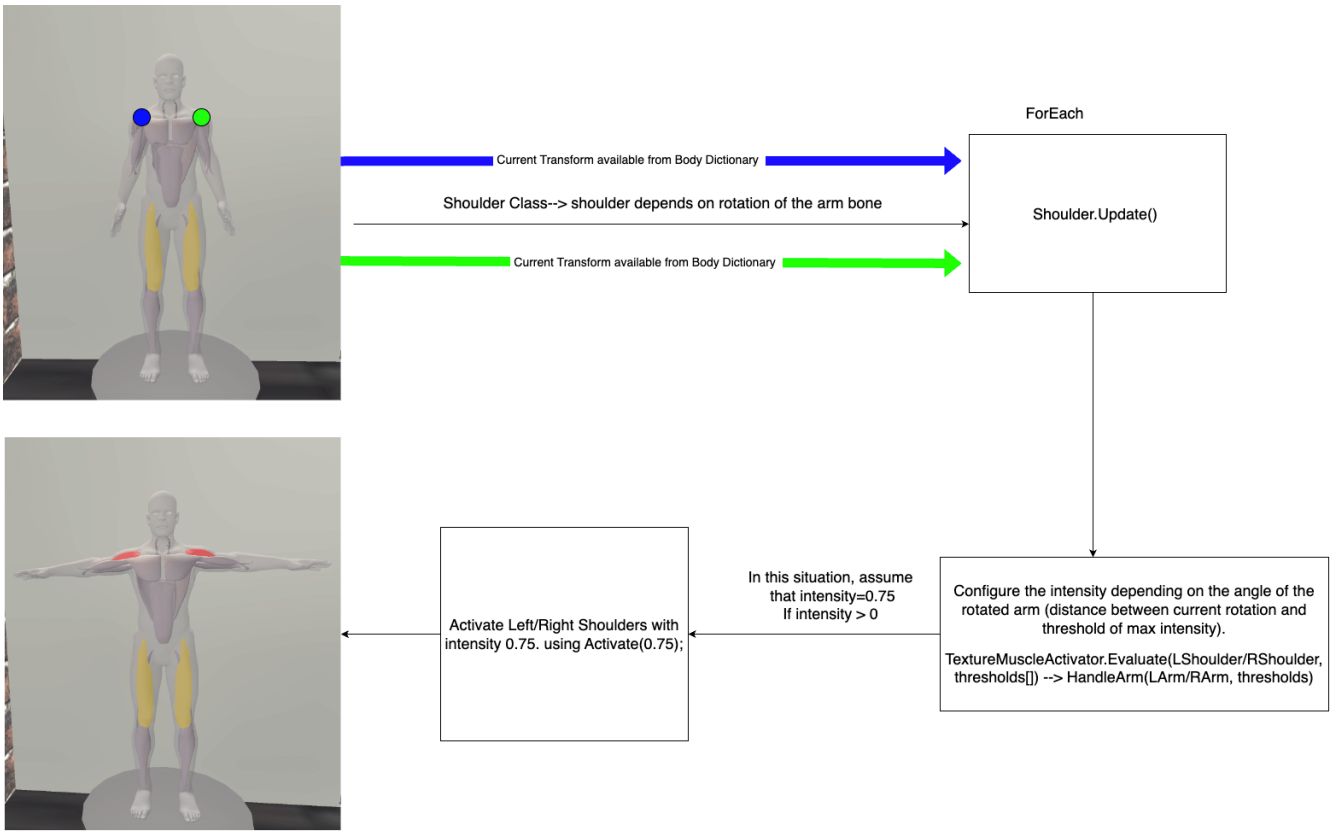

Figure 10: Framework Example.

In summary, the muscle highlighting framework represents a sophisticated and adaptable tool for visualizing muscle activation dynamics in real-time. Through its extendable architecture and dynamic color adjustment capabilities, the framework empowers users with valuable insights into exercise mechanics, facilitating a deeper understanding of proper technique and muscle engagement.

3.2 Inverse Kinematics and Upper Body Tracking

In my Unity-based application, the power of inverse kinematics (IK) is harnessed to enrich the user experience, particularly within virtual reality (VR) environments. Leveraging the capabilities of the FINAL IK package, advanced IK solutions are integrated to achieve precise and realistic animation control. Specifically, the VR IK module within FINAL IK is employed to simulate basic upper body tracking using only VR headsets and controllers as trackers.

Despite the inherent limitations posed by relying solely on VR headsets and controllers for tracking, the VR IK module offers a compelling solution for implementing immersive interactions within the application. Through dynamic adjustments to the position and orientation of virtual limbs based on the movements of VR controllers, users are provided with a rudimentary form of upper body tracking. This functionality allows users to engage in exercises and observe real-time

muscle highlighting, thereby enhancing their understanding of proper technique and muscle engagement.

The integration of inverse kinematics with the FINAL IK package and VR IK module results in a unique and immersive experience that effectively bridges the gap between virtual and physical fitness training. Despite the challenges associated with limited tracking capabilities, the utilization of IK enables the application to overcome these obstacles and deliver a compelling and interactive solution for users to engage with exercises and visualize muscle activation dynamics.

3.3 Extending the set of exercises

Within the app, the process of extending the set of exercises requires a hands-on approach to Unity development and C# coding. This section elucidates the methodologies and steps involved in empowering users to expand the repertoire of available exercises through direct engagement with Unity's development environment. Moreover, this application is open source, providing users with full access to the application's source code and allowing for transparency and community-driven innovation. By embracing an open source model, users are empowered to not only extend the set of exercises but also contribute to the ongoing development and improvement of the application.

To add a muscle into the framework, the following steps should be undertaken. First, within the Framework Scripts folder, a new class that inherits from the `Muscle` class must be created, for example, a class named `Gluteus` (See Figure 11). This new class should be structured similarly to existing muscle classes as illustrated in the accompanying Figure 12. Next, minimum and maximum thresholds for the muscle's activation need to be defined within the `thresholds.cs` script (See Figure 13). This ensures that the activation parameters are properly set for the new muscle. Subsequently, the handling of the muscle should be integrated into the appropriate handle function within the `TextureMuscleActivator` class, allowing the framework to process the muscle's activation correctly. Finally, the new muscle class created in the first step must be added as a component to the corresponding muscle GameObject on the avatar, thereby linking the new muscle script to the specific muscle on the model (See Figure 14).

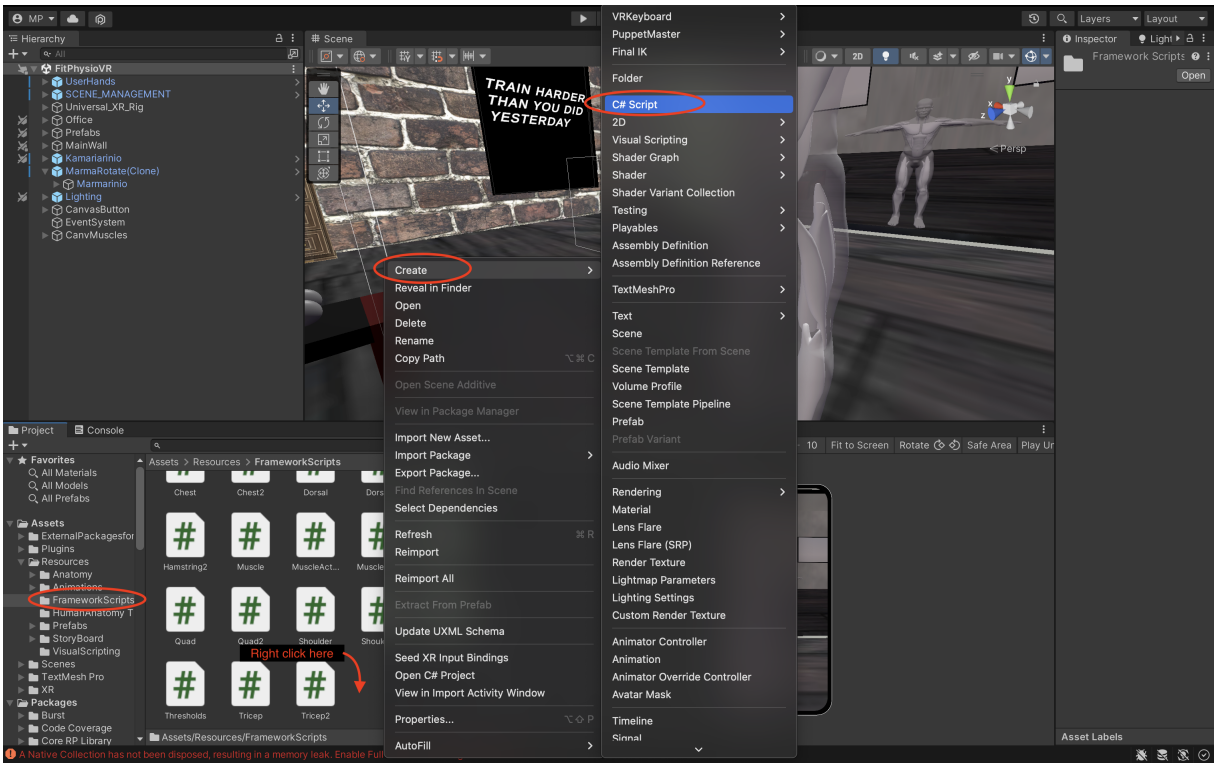

Figure 11: Create C# Script for new muscle class.

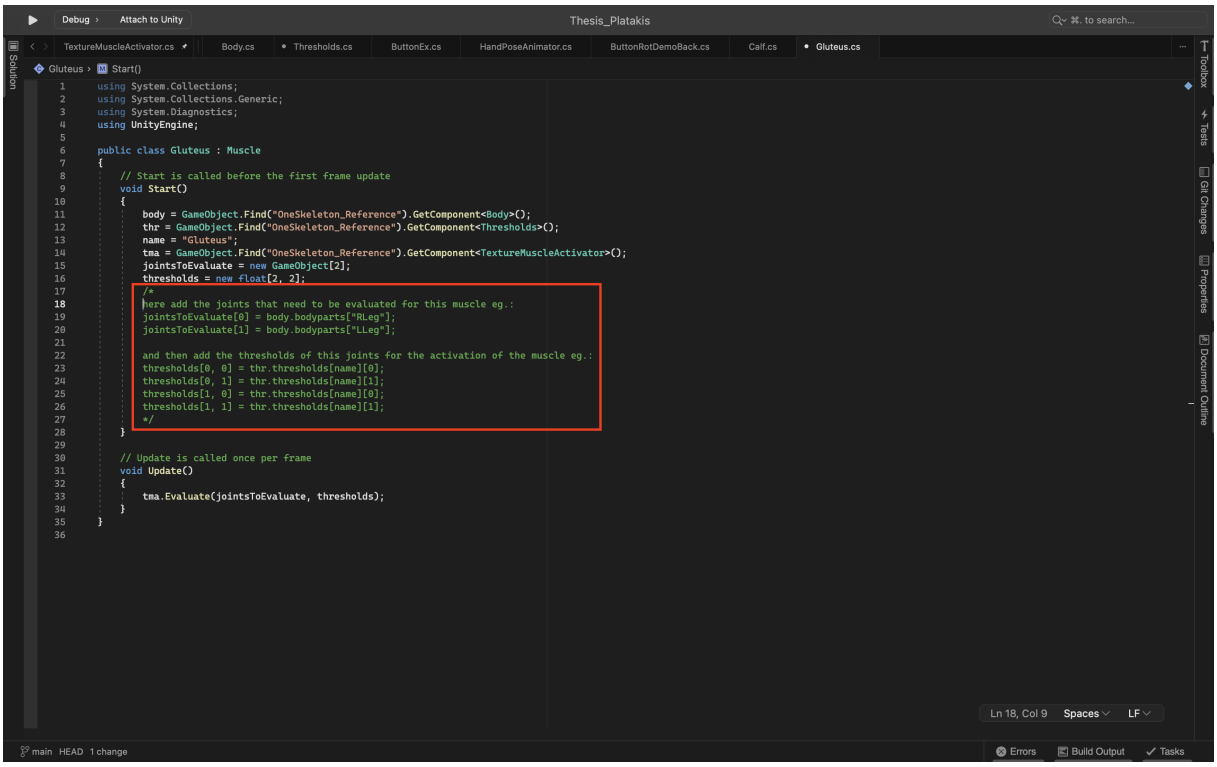

Figure 12: Gluteus class.

To incorporate a new exercise for a muscle, the following procedure should be followed. Initially, within Unity, the desired animation clip should be recorded by adding it to the **Marmarinio** GameObject's animator component and capturing

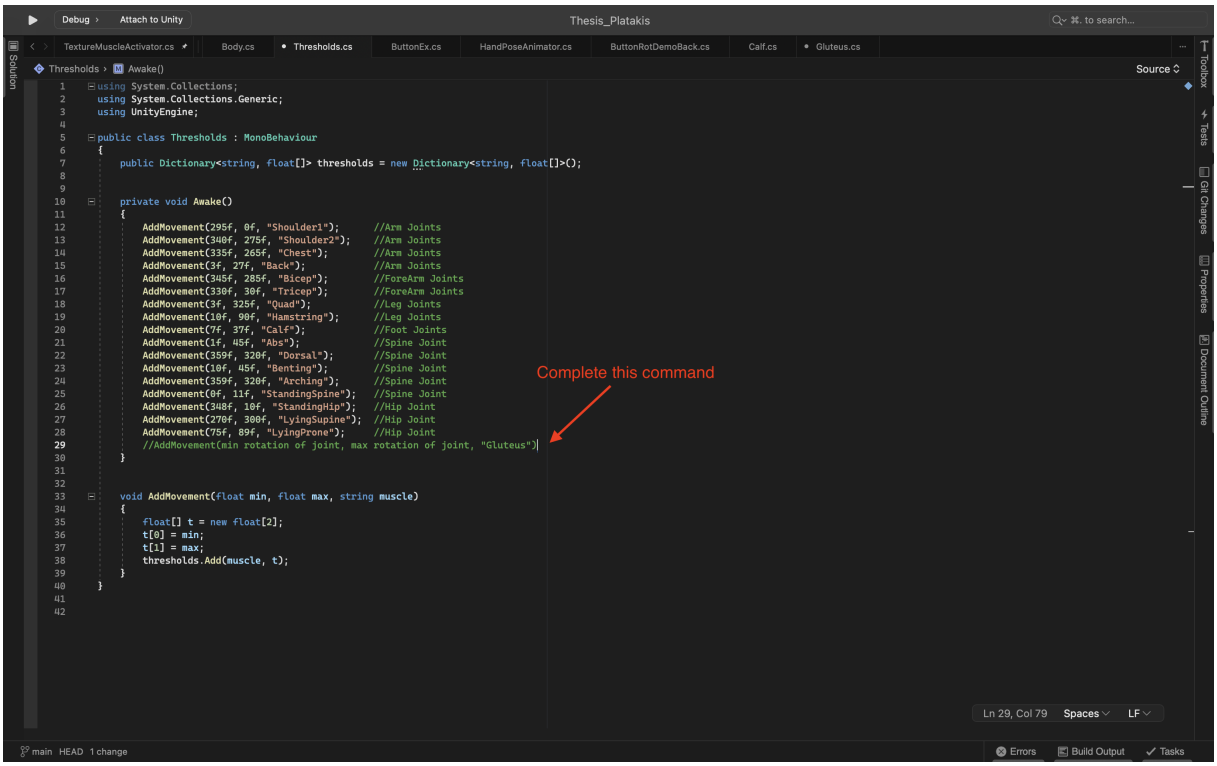

Figure 13: Set threshold values for the muscle activation(min,max).

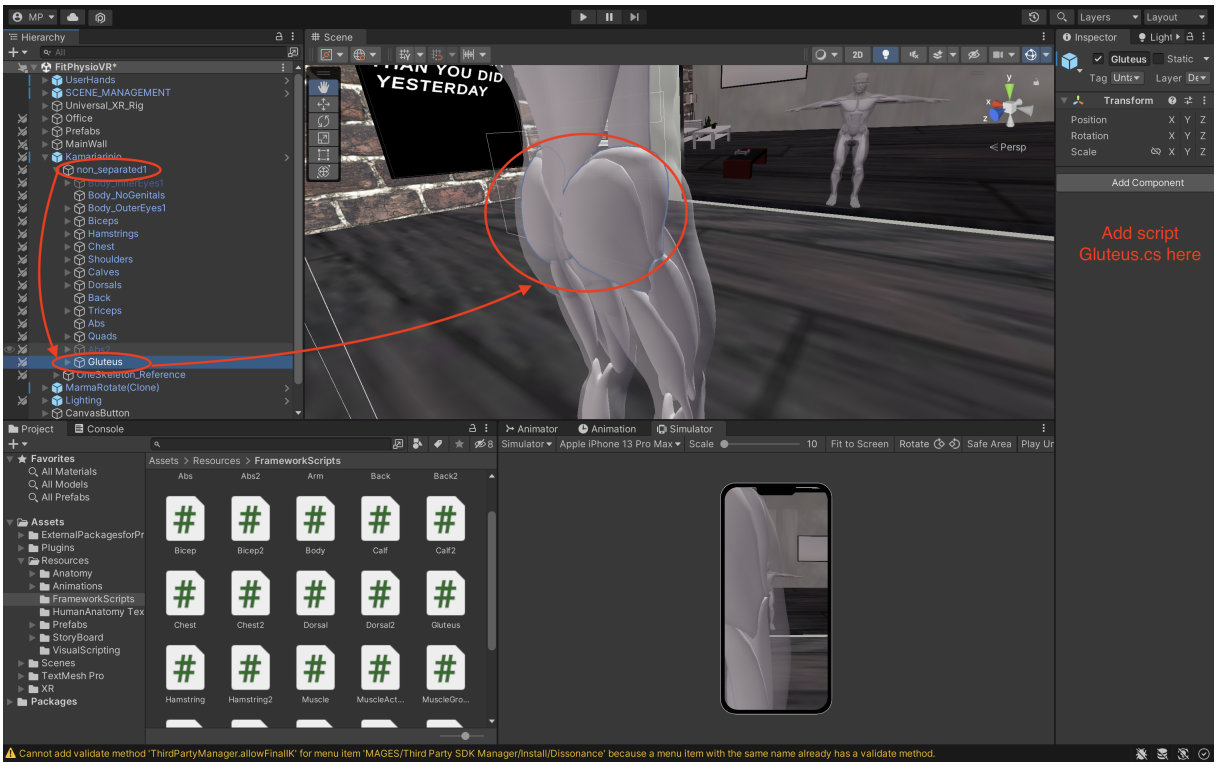

Figure 14: Gluteus GameObject in Unity.

the necessary movements within this clip. Subsequently, if the exercise pertains to an existing muscle UI, a new button should be created by duplicating the components of the existing buttons eg. **BCurls** (See Figure 15). If it involves

a new muscle, a new UI must be created (eg. duplicate of **UIBiceps**), and a button added to it following the same duplication process. Moving forward, the `buttonex.cs` script should be modified to include a new function named after the exercise being added, ensuring the new exercise is properly integrated into the system (See Figure 16). Lastly, within the button behavior component of the newly created button, the newly defined function must be linked in the OnClick() section, thereby enabling the functionality of the new exercise within the application.

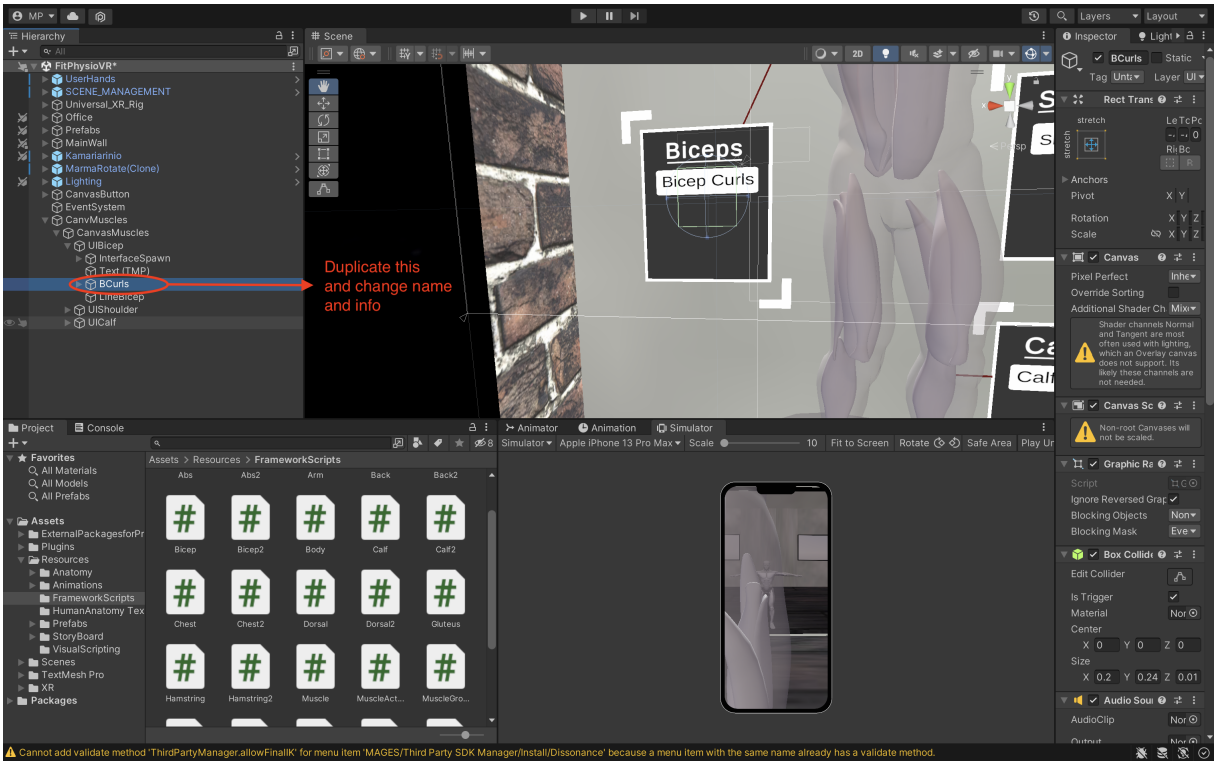

Figure 15: Duplicate an existing exercise gameobject and adapt it for the new exercise.

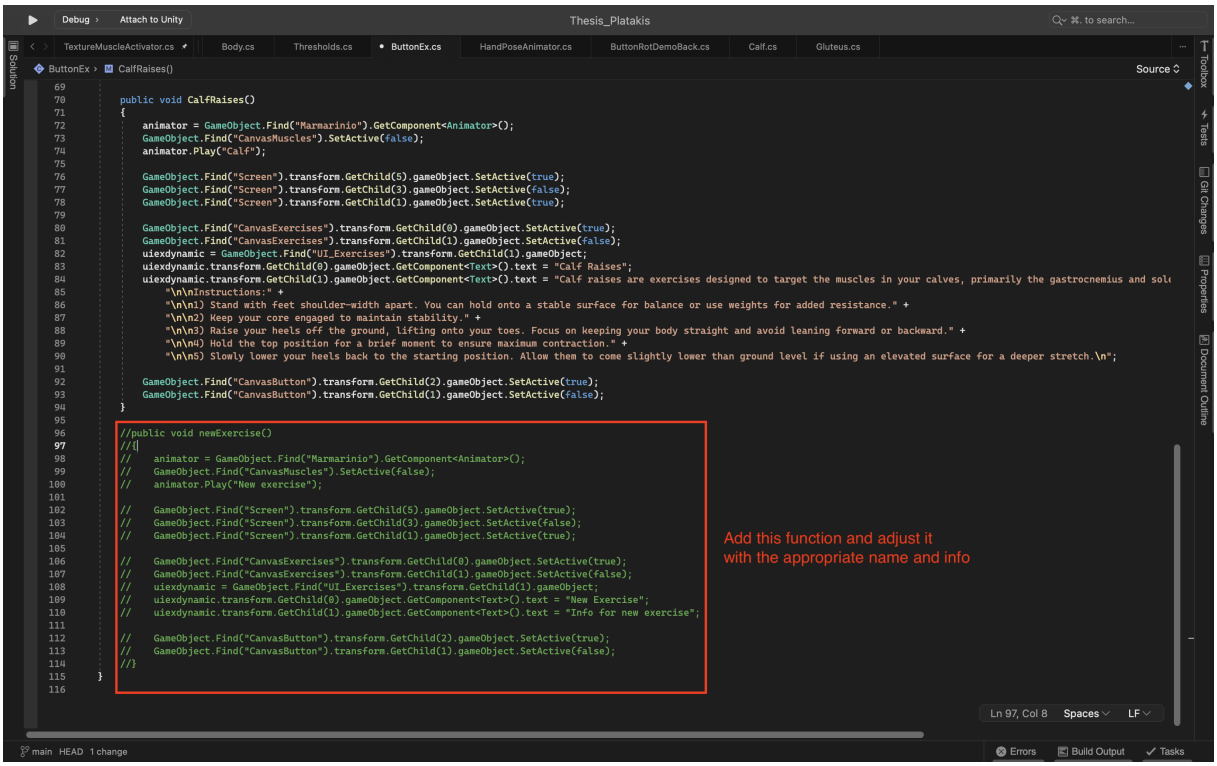

Figure 16: newExercise function.

3.4 Results

The culmination of extensive development efforts and user engagement within the app has yielded compelling results in the realms of fitness training and physiotherapy rehabilitation. This section provides an overview of the outcomes and impacts observed following the deployment and utilization of the application.

Through the integration of real-life instructional videos, 3D animated models, and interactive feedback mechanisms, users have reported a significant improvement in the effectiveness of their workouts. The comprehensive understanding of proper exercise techniques facilitated by the application has enabled users to optimize their training regimens and achieve better fitness outcomes. In addition, by providing detailed instructions, safety precautions and feedback it has contributed to a notable reduction in the risk of exercise-related injuries. Users report feeling more confident in their ability to perform exercises safely and correctly, leading to a decrease in the incidence of strains, sprains, and other exercise-related ailments.

The inclusion of a virtual mirror feature has helped users to perform self-assessments of their exercise form and technique. By comparing their movements to the correct techniques displayed by 3D models, users can identify areas for improvement and make real-time adjustments to their exercise execution, fostering a sense of self-efficacy and autonomy in their fitness journey. Finally, the application's user-friendly interface, flexible design, and comprehensive exercise

library have made fitness training and rehabilitation resources more accessible to a wider audience. Fitness professionals, physiotherapists, and individuals alike benefit from the versatility of FitPhysioXR app, which offers a wealth of exercise options tailored to diverse needs and goals.

In the following link, you will find a video demonstrating a presentation of the app: FitPhysioVR Demo.

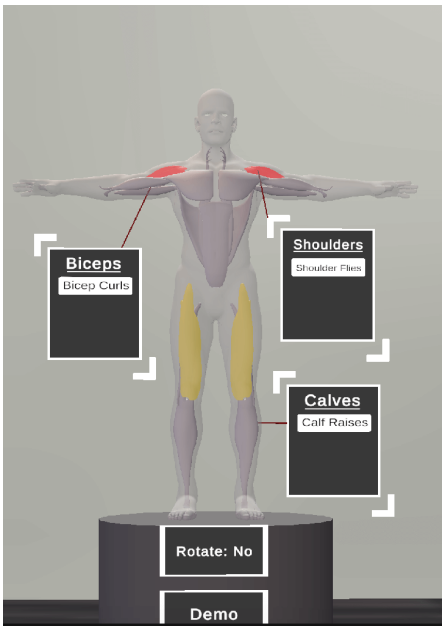

Figure 17: Example in which Shoulders and Quads are activated.

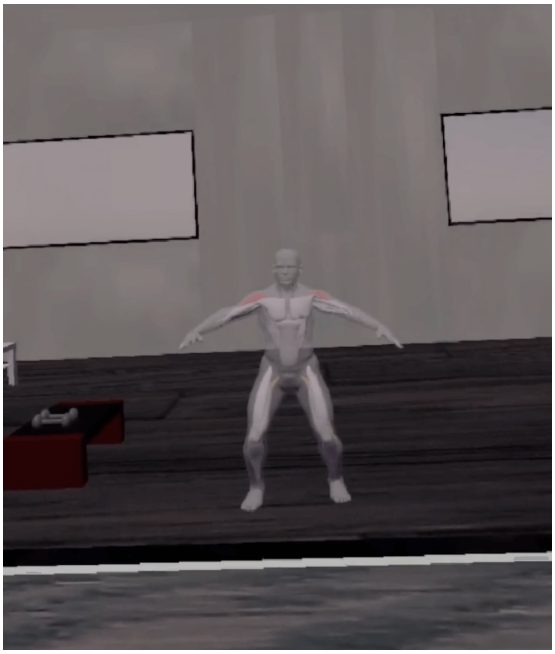

Figure 18: Self-assessment through mirror.

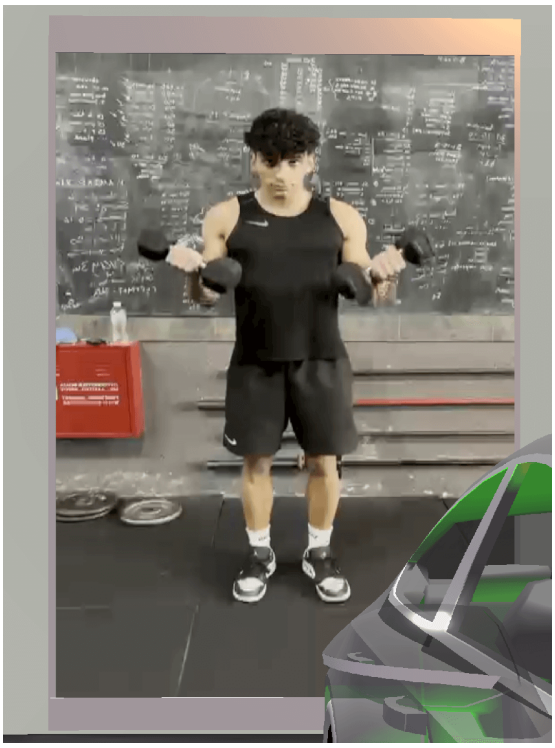

Figure 19: Real-life video.

3.5 Code Availability

I provide a link to the GitHub repository associated with this thesis: `https://github.com/platak1sm/Thesis`. This repository serves as a comprehensive resource for readers interested in exploring the source code, accessing additional materials, or furthering their understanding of the concepts discussed in the thesis. By providing access to the repository, I aim to promote transparency, reproducibility, and collaboration in the realm of academic research and software development. More specifically you will find the framework scripts here: Framework Scripts.

4 Conclusions, Limitations and Future Work

As we approach the culmination of the journey in developing and deploying my application for fitness training and physiotherapy rehabilitation, it becomes essential to engage in a reflective discourse. This entails a meticulous examination of the achievements, a candid appraisal of encountered limitations, and a forward-looking discourse on prospective avenues for growth and innovation.

4.1 Conclusions

The conclusions drawn from the development and deployment of the application reflect a profound understanding of the transformative power of technology in the realms of health, fitness, and rehabilitation.

Central to the conclusions is the recognition of the symbiotic relationship between technology and expertise in fostering optimal health and fitness outcomes. By harnessing cutting-edge technologies such as virtual reality, inverse kinematics, and muscle highlighting frameworks, I have created a platform that complements the expertise of fitness professionals and physiotherapists, empowering users with the knowledge and tools needed to achieve their health and fitness goals. Moreover, the application's emphasis on education and visualization has emerged as a cornerstone of its effectiveness in facilitating exercise instruction and rehabilitation. By providing users with real-life instructional videos, 3D animated models, and dynamic muscle highlighting, they are empowered with the knowledge and visual feedback needed to perform exercises correctly, reduce the risk of injury, and accelerate rehabilitation progress.

At the heart of the conclusions lies a commitment to user-centric design principles and accessibility. By prioritizing ease of use, flexibility, and inclusivity in the design, fitness training and rehabilitation resources more accessible to a wider audience, including individuals with diverse needs and abilities. Additionally, I recognize the importance of continuous improvement and innovation in meeting the evolving needs and expectations of users. The conclusions underscore the iterative nature of software development, as I remain committed to gathering feedback, iterating on features, and exploring new technologies to enhance the user experience and drive better health outcomes.

4.2 Limitations

FitPhysioXR application, while robust and versatile, is not without its limitations. These constraints inform the boundaries of its functionality and usability, providing insights into areas for potential improvement and development.

One of the primary constraints lies in its reliance on limited full body tracking capabilities. With only VR headsets and controllers serving as hand trackers, the application may face challenges in accurately capturing full body movements. This limitation restricts the granularity of motion analysis and may impact the accuracy of exercise feedback and visualization. Additionally, the VR IK module utilized within the application may present limitations in accurately simulating upper body tracking using VR controllers. Factors such as tracking accuracy, occlusion, and calibration issues may impact the fidelity of virtual limb movements,

leading to discrepancies between real-world and virtual interactions.

Another significant limitation is the prerequisite of C# programming and Unity development skills for extending the application's framework and adding new exercises. While these technical requirements empower users with the ability to customize and expand the application's functionalities, they also present a barrier to entry for individuals without programming experience or familiarity with Unity's development environment. This complexity may pose challenges for users seeking to customize or add new exercises, necessitating significant time and effort investment in learning and implementation.

The application's compatibility is constrained by the hardware requirements of VR headsets and controllers. Users must ensure compatibility with their specific hardware setup, which may limit accessibility for individuals with incompatible devices or limited access to VR equipment. Finally, the resource intensiveness of virtual reality applications may present limitations in terms of hardware performance and system requirements. Users may experience performance issues or compatibility challenges on devices with limited processing power or memory resources.

In summary, while this application offers a powerful platform for fitness training and physiotherapy rehabilitation, it is essential to acknowledge and address its limitations. By recognizing these constraints and actively seeking solutions and workarounds, I can strive to enhance the application's functionality, accessibility, and user experience.

4.3 Future Work

As I look to the future, there are several exciting avenues for further development and enhancement of FitPhysioXR application. These prospective areas of focus hold the potential to enrich user experiences, expand functionalities, and drive continued innovation in the fields of fitness training and physiotherapy rehabilitation.

One of the primary objectives for future development efforts involves improving full body tracking capabilities. This may entail exploring advanced motion capture technologies or integrating additional tracking sensors to enhance the accuracy and granularity of motion capture. By enabling more precise tracking of user movements, users can be provided with a more immersive and engaging exercise experience. Moreover, expanding the application's muscle highlighting framework to include more specific muscle groups represents another area for future development. By incorporating detailed anatomical models and muscle visualization techniques, users can gain deeper insights into the specific muscles

engaged during each exercise. This enhanced muscle visualization can aid in targeting specific muscle groups for strength training or rehabilitation purposes.

Continuously expanding the application's exercise library to include a wider range of exercises tailored to diverse fitness goals and rehabilitation needs is paramount. This may involve collaborating with fitness experts, physiotherapists, and exercise specialists to curate specialized workout programs targeting specific populations or addressing specific health conditions. Additionally, integrating gamification elements such as challenges, achievements, and progress tracking can further enhance user engagement and motivation. Additionally, introducing an exercise scoring system that evaluates users' performance based on their adherence to proper form and technique represents a valuable feature for future development. By providing users with real-time feedback and scores based on their execution of exercises, they will be more focused on proper technique and continuous improvement. This scoring system can also serve as a motivational tool, rewarding users for their progress and achievements.

Incorporating biometric monitoring capabilities into the application represents a promising avenue for future development. By integrating sensors or wearables that track vital signs such as heart rate, oxygen saturation, and movement metrics, users can be provided with comprehensive insights into their physiological responses during exercise. This biometric data can inform personalized exercise recommendations, track progress over time, and enhance the overall effectiveness of fitness training and rehabilitation programs.

In conclusion, the future work outlined above represents a multifaceted approach to further advancing my Unity-based application and maximizing its impact on user health and well-being. Through ongoing innovation, collaboration, and a commitment to user-centric design principles, we can continue to push the boundaries of technology in the service of promoting healthier, happier lives.

References

Alturki, R., & Gay, V. (2019). Augmented and virtual reality in mobile fitness applications: a survey. *Applications of intelligent technologies in healthcare*, 67–75.

AR Fitness App using ARkit. (2020). `https://www.youtube.com/watch?v=Qj05vSfaPws&ab_channel=JayeshRathod%7CAR%7CVR%7C`. (Accessed: 2023-09-06)

Bansal, G., Rajgopal, K., Chamola, V., Xiong, Z., & Niyato, D. (2022). Healthcare in metaverse: A survey on current metaverse applications in healthcare. *Ieee Access*, *10*, 119914–119946.

Baqai, A., Memon, K., Memon, A. R., & Shah, S. M. Z. A. (2019). Interactive physiotherapy: an application based on virtual reality and bio-feedback. *Wireless Personal Communications*, *106*(4), 1719–1741.

Barsasella, D., Liu, M. F., Malwade, S., Galvin, C. J., Dhar, E., Chang, C.-C., ... Syed-Abdul, S. (2021). Effects of virtual reality sessions on the quality of life, happiness, and functional fitness among the older people: a randomized controlled trial from taiwan. *Computer Methods and Programs in Biomedicine*, *200*, 105892.

Binoy, A., & Srivastava, A. (2020). Pump fit: An augmented reality application which helps people with their workout more efficiently. In *Companion proceedings of the 2020 conference on interactive surfaces and spaces* (pp. 91–93).

Brady, N., Dejaco, B., Lewis, J., McCreesh, K., & McVeigh, J. G. (2023). Physiotherapist beliefs and perspectives on virtual reality supported rehabilitation for the management of musculoskeletal shoulder pain: A focus group study. *Plos one*, *18*(4), e0284445.

Brepohl, P. C. A., & Leite, H. (2023). Virtual reality applied to physiotherapy: a review of current knowledge. *Virtual Reality*, *27*(1), 71–95.

Chong, N., Chu, E., Nadonza, A., Rodriguez, S. M., Tith, S., Shan, J., ... Hoang, T. (2023). An empathetic approach to human-centric requirements engineering using virtual reality. In *2023 ieee 47th annual computers, software, and applications conference (compsac)* (pp. 1717–1724).

da Silva, M. V., Pacagnelli, F. L., de Almeida, L. L., & Siscoutto, R. A. (2021). Physioar: An augmented reality system applied in respiratory physiotherapy for hypertensive patients. In *Proceedings of the brazilian symposium on multimedia and the web* (pp. 37–44).

Fitness APP in Augmented Reality. (2020). `https://www.youtube.com/shorts/7gTybp7-8Qk`. (Accessed: 2023-09-06)

FIT XR. (2023). `https://fitxr.com/`. (Accessed: 2023-09-06)

Fyter: AR Fitness Workouts. (2021). `https://apps.apple.com/us/app/fyter-interactive-ar-fitness/id1502336594`. (Accessed: 2023-09-11)

Gymnotize Fitness Workout App. (2020). `https://apps.apple.com/us/app/gymnotize-fitness-workout-app/id773676355`. (Accessed: 2023-09-06)

Hansson, A. (2018). *Fitness and exercising in virtual reality.*

Hsiao, K.-F. (2013). Using augmented reality for students health-case of combining educational learning with standard fitness. *Multimedia tools and applications*, *64*, 407–421.

iMuscle 2. (2011). `https://3d4medical.com/apps/imuscle-2`. (Accessed: 2023-09-06)

Jin, Y., Monge, J., Postolache, O., & Niu, W. (2019). Augmented reality with application in physical rehabilitation. In *2019 international conference on sensing and instrumentation in iot era (issi)* (pp. 1–6).

Koech, I. (2020). *Interactive mobile augmented reality for fitness activities.*

Liu, R., Menhas, R., Dai, J., Saqib, Z. A., & Peng, X. (2022). Fitness apps, live streaming workout classes, and virtual reality fitness for physical activity during the covid-19 lockdown: an empirical study. *Frontiers in public health*, *10*, 852311.

Lucena-Anton, D., Fernandez-Lopez, J. C., Pacheco-Serrano, A. I., Garcia-Munoz, C., & Moral-Munoz, J. A. (2022). Virtual and augmented reality versus traditional methods for teaching physiotherapy: a systematic review. *European Journal of Investigation in Health, Psychology and Education*, *12*(12), 1780–1792.

Mejtoft, T., Lindahl, H., Norberg, O., Andersson, M., & Söderström, U. (2023). Enhancing digital social interaction using augmented reality in mobile fitness

applications. In *Proceedings of the 2023 5th international conference on image, video and signal processing* (pp. 95–100).

Menhas, R., Qin, L., Saqib, Z. A., & Younas, M. (2023). The association between covid-19 preventive strategies, virtual reality exercise, use of fitness apps, physical, and psychological health: testing a structural equation moderation model. *Frontiers in Public Health*, *11*.

Nair, Y. N. I., Azman, F., Rahim, F. A., & Cheng, L. K. (2019). Endure: Augmented reality fitness mobile application. In *2019 ieee 4th international conference on computer and communication systems (icccs)* (pp. 419–424).

Peng, X., Menhas, R., Dai, J., & Younas, M. (2022). The covid-19 pandemic and overall wellbeing: mediating role of virtual reality fitness for physical-psychological health and physical activity. *Psychology Research and Behavior Management*, 1741–1756.

Postolache, O., Monge, J., Alexandre, R., Geman, O., Jin, Y., & Postolache, G. (2021). Virtual reality and augmented reality technologies for smart physical rehabilitation. *Advanced Systems for Biomedical Applications*, 155–180.

Postolache, O., Teixeira, L., Cordeiro, J., Lima, L., Arriaga, P., Rodrigues, M., & Girão, P. (2019). Tailored virtual reality for smart physiotherapy. In *2019 11th international symposium on advanced topics in electrical engineering (atee)* (pp. 1–6).

Qian, J., McDonough, D. J., & Gao, Z. (2020). The effectiveness of virtual reality exercise on individual's physiological, psychological and rehabilitative outcomes: a systematic review. *International journal of environmental research and public health*, *17*(11), 4133.

Rabbi, F., Park, T., Fang, B., Zhang, M., & Lee, Y. (2018). When virtual reality meets internet of things in the gym: Enabling immersive interactive machine exercises. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, *2*(2), 1–21.

Taware, G., Kharat, R., Dhende, P., Jondhalekar, P., & Agrawal, R. (2022). Ai-based workout assistant and fitness guide. In *2022 6th international conference on computing, communication, control and automation (iccubea* (pp. 1–4).

Viglialoro, R. M., Condino, S., Turini, G., Carbone, M., Ferrari, V., & Gesi, M. (2019). Review of the augmented reality systems for shoulder rehabilitation. *Information*, *10*(5), 154.

Xu, W., Liang, H.-N., Baghaei, N., Ma, X., Yu, K., Meng, X., ... others (2021). Effects of an immersive virtual reality exergame on university students' anxiety, depression, and perceived stress: pilot feasibility and usability study. *JMIR Serious Games*, *9*(4), e29330.

Xu, W., Liang, H.-N., Yu, Y., Monteiro, D., Hasan, K., & Fleming, C. (2019). Assessing the effects of a full-body motion-based exergame in virtual reality. In *Proceedings of the seventh international symposium of chinese chi* (pp. 1–6).

Zikas, P., Protopsaltis, A., Lydatakis, N., Kentros, M., Geronikolakis, S., Kateros, S., ... Papagiannakis, G. (2023). Mages 4.0: Accelerating the world's transition to vr training and democratizing the authoring of the medical metaverse. *IEEE Computer Graphics and Applications*, *43*(2), 43-56. doi: 10.1109/MCG.2023.3242686

A Appendix

In this appendix section, I provide supplementary materials to complement the discussions and analyses presented throughout the thesis. Included here are screenshots of code snippets, providing readers with visual aids to better understand the implementation details and technical aspects discussed in the main text. These screenshots offer a glimpse into the underlying codebase of my Unity-based application, offering insights into the intricacies of its design and functionality.

Through the combination of code screenshots and GitHub repository access, I endeavor to enrich the reader's understanding of the technical underpinnings of the application and facilitate further exploration and engagement with the materials presented.

```
public class Body : MonoBehaviour
{
    public Dictionary<string, GameObject> bodyparts = new Dictionary<string, GameObject>();

    private void Awake()
    {
        bodyparts.Add("LShoulder", GameObject.Find("OneSkeleton_LeftShoulder"));
        bodyparts.Add("RShoulder", GameObject.Find("OneSkeleton_RightShoulder"));
        bodyparts.Add("LArm", GameObject.Find("OneSkeleton_LeftArm"));
        bodyparts.Add("RArm", GameObject.Find("OneSkeleton_RightArm"));
        bodyparts.Add("LForeArm", GameObject.Find("OneSkeleton_LeftForeArm"));
        bodyparts.Add("RForeArm", GameObject.Find("OneSkeleton_RightForeArm"));
        bodyparts.Add("LUpLeg", GameObject.Find("OneSkeleton_LeftUpLeg"));
        bodyparts.Add("RUpLeg", GameObject.Find("OneSkeleton_RightUpLeg"));
        bodyparts.Add("LLeg", GameObject.Find("OneSkeleton_LeftLeg"));
        bodyparts.Add("RLeg", GameObject.Find("OneSkeleton_RightLeg"));
        bodyparts.Add("LFoot", GameObject.Find("OneSkeleton_LeftFoot"));
        bodyparts.Add("RFoot", GameObject.Find("OneSkeleton_RightFoot"));
        bodyparts.Add("Spine", GameObject.Find("OneSkeleton_Spine"));
        bodyparts.Add("Spine1", GameObject.Find("OneSkeleton_Spine1"));
        bodyparts.Add("Spine2", GameObject.Find("OneSkeleton_Spine2"));
        bodyparts.Add("Hips", GameObject.Find("OneSkeleton_Hips1"));
        bodyparts.Add("LShoulder2", GameObject.Find("OneSkeleton_LeftShoulder2"));
        bodyparts.Add("RShoulder2", GameObject.Find("OneSkeleton_RightShoulder2"));
        bodyparts.Add("LArm2", GameObject.Find("OneSkeleton_LeftArm2"));
        bodyparts.Add("RArm2", GameObject.Find("OneSkeleton_RightArm2"));
        bodyparts.Add("LForeArm2", GameObject.Find("OneSkeleton_LeftForeArm2"));
        bodyparts.Add("RForeArm2", GameObject.Find("OneSkeleton_RightForeArm2"));
        bodyparts.Add("LUpLeg2", GameObject.Find("OneSkeleton_LeftUpLeg2"));
        bodyparts.Add("RUpLeg2", GameObject.Find("OneSkeleton_RightUpLeg2"));
        bodyparts.Add("LLeg2", GameObject.Find("OneSkeleton_LeftLeg2"));
        bodyparts.Add("RLeg2", GameObject.Find("OneSkeleton_RightLeg2"));
        bodyparts.Add("LFoot2", GameObject.Find("OneSkeleton_LeftFoot2"));
        bodyparts.Add("RFoot2", GameObject.Find("OneSkeleton_RightFoot2"));
        bodyparts.Add("Spine21", GameObject.Find("OneSkeleton_Spine2_1"));
        bodyparts.Add("Spine22", GameObject.Find("OneSkeleton_Spine2_2"));
        bodyparts.Add("Spine23", GameObject.Find("OneSkeleton_Spine2_3"));
        bodyparts.Add("Hips2", GameObject.Find("OneSkeleton_Hips2"));
    }


}
```

Figure 20: Body Class (Mapping)

```
public class Thresholds : MonoBehaviour
{
    public Dictionary<string, float[]> thresholds = new Dictionary<string, float[]>();


    private void Awake()
    {
        AddMovement(295f, 0f, "Shoulder1");      //Arm Joints
        AddMovement(340f, 275f, "Shoulder2");    //Arm Joints
        AddMovement(335f, 265f, "Chest");        //Arm Joints
        AddMovement(3f, 27f, "Back");            //Arm Joints
        AddMovement(345f, 285f, "Bicep");        //ForeArm Joints
        AddMovement(330f, 30f, "Tricep");        //ForeArm Joints
        AddMovement(3f, 325f, "Quad");           //Leg Joints
        AddMovement(10f, 90f, "Hamstring");      //Leg Joints
        AddMovement(7f, 37f, "Calf");            //Foot Joints
        AddMovement(1f, 45f, "Abs");             //Spine Joint
        AddMovement(359f, 320f, "Dorsal");       //Spine Joint
        AddMovement(10f, 45f, "Benting");        //Spine Joint
        AddMovement(359f, 320f, "Arching");      //Spine Joint
        AddMovement(0f, 11f, "StandingSpine");   //Spine Joint
        AddMovement(348f, 10f, "StandingHip");   //Hip Joint
        AddMovement(270f, 300f, "LyingSupine");  //Hip Joint
        AddMovement(75f, 89f, "LyingProne");     //Hip Joint


    void AddMovement(float min, float max, string muscle)
    {
        float[] t = new float[2];
        t[0] = min;
        t[1] = max;
        thresholds.Add(muscle, t);
    }
}
```

Figure 21: Thresholds Class (min-max)

```
public class TextureMuscleActivator : MuscleActivator
{
    Texture tex;
    Body b;
    private Dictionary<GameObject, Action<GameObject, float[]>> jointActions = new Dictionary<GameObject, Action<GameObject, float[]>>();
    Material m;
    Thresholds thr, thr2;
    bool isBenting = false, isLyingSupine = false, isLyingProne = false, isStanding = true, isActivatedR = false, isActivatedL = false, isActivated = false;
    bool isBenting2 = false, isLyingSupine2 = false, isLyingProne2 = false, isStanding2 = true, isActivatedR2 = false, isActivatedL2 = false, isActivated2 = false;
    private void Awake()
    {
        b = GameObject.Find("OneSkeleton_Reference").GetComponent<Body>();
    }

    // Start is called before the first frame update
    void Start()
    {
        thr = GameObject.Find("OneSkeleton_Reference").GetComponent<Thresholds>();
        jointActions[b.bodyparts["LShoulder"]] = HandleShoulder;
        jointActions[b.bodyparts["RShoulder"]] = HandleShoulder;
        jointActions[b.bodyparts["LArm"]] = HandleArm;
        jointActions[b.bodyparts["RArm"]] = HandleArm;
        jointActions[b.bodyparts["LForeArm"]] = HandleForeArm;
        jointActions[b.bodyparts["RForeArm"]] = HandleForeArm;
        jointActions[b.bodyparts["LUpLeg"]] = HandleUpLeg;
        jointActions[b.bodyparts["RUpLeg"]] = HandleUpLeg;
        jointActions[b.bodyparts["LLeg"]] = HandleLeg;
        jointActions[b.bodyparts["RLeg"]] = HandleLeg;
        jointActions[b.bodyparts["LFoot"]] = HandleFoot;
        jointActions[b.bodyparts["RFoot"]] = HandleFoot;
        jointActions[b.bodyparts["Spine"]] = HandleCore;

        //2nd model
        thr2 = GameObject.Find("OneSkeleton_Reference2").GetComponent<Thresholds>();
        jointActions[b.bodyparts["LShoulder2"]] = HandleShoulder;
        jointActions[b.bodyparts["RShoulder2"]] = HandleShoulder;
        jointActions[b.bodyparts["LArm2"]] = HandleArm2;
        jointActions[b.bodyparts["RArm2"]] = HandleArm2;
        jointActions[b.bodyparts["LForeArm2"]] = HandleForeArm2;
        jointActions[b.bodyparts["RForeArm2"]] = HandleForeArm2;
        jointActions[b.bodyparts["LUpLeg2"]] = HandleUpLeg2;
        jointActions[b.bodyparts["RUpLeg2"]] = HandleUpLeg2;
        jointActions[b.bodyparts["LLeg2"]] = HandleLeg2;
        jointActions[b.bodyparts["RLeg2"]] = HandleLeg2;
        jointActions[b.bodyparts["LFoot2"]] = HandleFoot2;
        jointActions[b.bodyparts["RFoot2"]] = HandleFoot2;
        jointActions[b.bodyparts["Spine21"]] = HandleCore2;
    }

```

Figure 22: Texture Muscle Activator Initialization

```
private void HandleArm(GameObject joint, float[] thresholds)
    {
        string jname, jname2;
        GameObject muscle, muscle2, muscle3;
        bool isLeft = false;
        float distance, distancel;
        muscle3 = GameObject.Find("Back");


        if (joint.name == "OneSkeleton_LeftArm")
        {
            jname = "LArm";
            jname2 = "LForeArm";
            isLeft = true;
            muscle = GameObject.Find("LShoulder");
            muscle2 = GameObject.Find("LChest");
            isActivated = isActivatedL;
        }
        else
        {
            jname = "RArm";
            jname2 = "RForeArm";
            muscle = GameObject.Find("RShoulder");
            muscle2 = GameObject.Find("RChest");
            isActivated = isActivatedR;
        }

        Vector3 rotation = b.bodyparts[jname].transform.localRotation.eulerAngles;
        Vector3 rotation2 = b.bodyparts[jname2].transform.localRotation.eulerAngles;
        Vector3 hiprotation = b.bodyparts["Hips"].transform.localRotation.eulerAngles;
        Vector3 spinerotation = b.bodyparts["Spine"].transform.localRotation.eulerAngles;

        if ((thresholds[1] > thresholds[0] || thresholds[1] == 275) && rotation.z > 280 && rotation.z < 290)
        {
            if (thresholds[1] < 275)
            {
                distance = thresholds[1] - rotation.x;
                if (distance <= thresholds[1] - thresholds[0] && distance >= 0)
                {
                    //print("Distance of joint " + jname +": "+ distance);
                    distancel = Mathf.InverseLerp(0f, thresholds[1] - thresholds[0], distance);
                    //print("Distance 0-1 of joint " + jname + ": " + distancel);
                    Activate(1 - distancel, muscle3);
                }
                else
                {
                    Activate(0, muscle3);
                }
            }
            else
            {
                distance = rotation.x - thresholds[1];

                if (distance <= thresholds[0] - thresholds[1] && distance >= 0)
                {

                    distancel = Mathf.InverseLerp(0f, thresholds[0] - thresholds[1], distance);
                    //print("Distance 0-1 of joint " + jname + ": " + distancel);
                    Activate(1 - distancel, muscle);
                    if (isLeft) isActivatedL = true;
                    else isActivatedR = true;
                }
                else
                {
                    if (isLeft) isActivatedL = false;
                    else isActivatedR = false;
                    Activate(0, muscle);
                }
            }
        }
        else
        {
            if (thresholds[1] == 0)
            {
                distance = (thresholds[1] - rotation.z + 360) % 360;
                if (distance <= (thresholds[1] - thresholds[0] + 360) % 360 && spinerotation.x >= thr.thresholds["StandingSpine"][0] && spinerotation.x <= thr.thresholds["StandingSpine"][1])
                {
                    distancel = Mathf.InverseLerp(0f, (thresholds[1] - thresholds[0] + 360) % 360, distance);
                    Activate(1 - distancel, muscle);

                }
                else
                {
                    Activate(0, muscle);
                }
            }
            else if (thresholds[1] == 265)
            {
                distance = rotation.y - thresholds[1];

                if (hiprotation.x <= thr.thresholds["LyingSupine"][1] && hiprotation.x >= thr.thresholds["LyingSupine"][0])
                {
                    if (distance <= thresholds[0] - thresholds[1] && distance >= 0)
                    {
                        distancel = Mathf.InverseLerp(0f, thresholds[0] - thresholds[1], distance);
                        //Debug.Log("Distance 0-1 of joint CHEST: " + distancel);
                        Activate(1 - distancel, muscle2);
                    }
                    else
                    {
                        Activate(0, muscle2);
                    }

                }
                else
                {
                    Activate(0, muscle2);
                }
            }


        }
    }
```

Figure 23: Texture Muscle Activator.HandleArm()

```
public override void Evaluate(GameObject[] joints, float[,] thresholds)
    {
        float[] thresholdsarr = new float[2];
        if (2 * joints.Length != thresholds.Length)
        {
            Debug.LogError("The number of joints and thresholds should be the same.");
            return;
        }
        for (int i = 0; i < joints.Length; i++)
        {
            if (jointActions.ContainsKey(joints[i]))
            {
                thresholdsarr[0] = thresholds[i, 0];
                thresholdsarr[1] = thresholds[i, 1];
                jointActions[joints[i]](joints[i], thresholdsarr);
            }
            else
            {
                Debug.Log("This joint is not yet implemented.");
            }
        }
    }

    Color DarkenColor(Color originalColor, float factor)
    {
        // Clamp the factor between 0 and 1
        factor = Mathf.Clamp01(factor);

        // Darken the color by reducing the red and green components
        float newRed = originalColor.r * (1 - factor);
        float newGreen = originalColor.g * (1 - factor);

        // Keep the blue and alpha components unchanged
        float newBlue = originalColor.b;
        float newAlpha = originalColor.a;

        // Return the new color
        return new Color(newRed, newGreen, newBlue, newAlpha);
    }



    public override void Activate(float intensity, GameObject muscle)
    {
        m = muscle.GetComponent<Renderer>().material;
        if (intensity > 0)
        {
            Color darkerYellow = DarkenColor(Color.yellow, 0.5f);
            Color lerpedColor = Color.Lerp(darkerYellow, Color.red, intensity);
            //Debug.Log("Activated " + muscle.name + " with intensity " + intensity);
            m.color = lerpedColor;
            m.SetColor("_EmissionColor", lerpedColor);
            m.EnableKeyword("_EMISSION");
            m.globalIlluminationFlags = MaterialGlobalIlluminationFlags.RealtimeEmissive;
        }
        else if (intensity == 0)
        {
            m.color = Color.white;
            m.SetColor("_EmissionColor", Color.black);
            m.EnableKeyword("_EMISSION");
            m.globalIlluminationFlags = MaterialGlobalIlluminationFlags.RealtimeEmissive;
        }
        if (transform.childCount != 0)
        {
            foreach (SkinnedMeshRenderer renderer in muscle.transform.GetComponentsInChildren<SkinnedMeshRenderer>())
            {
                Material m1 = renderer.material;
                if (intensity > 0)
                {
                    Color darkerYellow = DarkenColor(Color.yellow, 0.5f);
                    Color lerpedColor = Color.Lerp(darkerYellow, Color.red, intensity);
                    //Debug.Log("Activated " + muscle.name + " with intensity " + intensity);
                    m1.color = lerpedColor;
                    m1.SetColor("_EmissionColor", lerpedColor);
                    m1.EnableKeyword("_EMISSION");
                    m1.globalIlluminationFlags = MaterialGlobalIlluminationFlags.RealtimeEmissive;
                }
                else if (intensity == 0)
                {
                    m1.color = Color.white;
                    m1.SetColor("_EmissionColor", Color.black);
                    m1.EnableKeyword("_EMISSION");
                    m1.globalIlluminationFlags = MaterialGlobalIlluminationFlags.RealtimeEmissive;
                }
            }
        }

    }
```

Figure 24: Evaluate and Activate functions