UNIUNEA EUROPEANĂ
Fondul Social European POSDRU 2007-2013
Instrumente Structurale 2007-2013
MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE
OIPOSDRU
Universitatea POLITEHNICA din Bucuresti

Proiect PROSCIENCE - POSDRU/187/1.5/S/155420
Promovarea științei și calității în cercetare prin burse doctorale

UNIVERSITATEA POLITEHNICA DIN BUCUREŞTI

Facultatea de Automatică şi Calculatoare

Departamentul de Calculatoare şi Tehnologia Informației

TEZĂ DE DOCTORAT

Soluții Bazate pe Realitate Virtuală şi Augmentată in Domeniul Medical

Solutions Based on Virtual and Augmented Reality in Healthcare

Autor: Oana Alexandra Voinea

Conducător de doctorat: Prof. Dr. Ing. Florica Moldoveanu

COMISIA DE DOCTORAT

Președinte	Prof. dr. ing. Adina Magda Florea	de la	Universitatea POLITEHNICA din București
Conducător de doctorat	Prof. dr. ing. Florica Moldoveanu	de la	Universitatea POLITEHNICA din București
Referent	Profesor dr. ing. Vasile Manta	de la	Universitatea Tehnică ”Gheorghe Asachi” din Iași
Referent	Profesor dr. ing. Ștefan-Gheorghe Pentiuc	de la	Universitatea ”Ștefan cel Mare” din Suceava
Referent	Profesor dr. ing. Nirvana Popescu	de la	Universitatea POLITEHNICA din București

București 2018

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Florica Moldoveanu for accepting me for this PhD and giving me the opportunity to work on diverse projects. I am grateful for her patience and understanding regarding career and personal events that had impact on the PhD timeline. Her guidance helped me in all the time of research and writing of this thesis and it has been an honor to meet her and to be her PhD student.

My sincere thanks also go to Prof. Alin Moldoveanu, who provided me the opportunity to join the TRAVEE team and who gave access to the research facilities. I truly appreciate his contribution of time and ideas to make my PhD experience stimulating.

I would like to thank Dr. Victor Asavei, Dr. Anca Morar and Oana Ferche, for their help, collaboration or useful insights during my time at the University POLITEHNICA of Bucharest. Also, I would like to thank Mrs. Catalina Daraban from the doctoral school office which is always helpful with information and paperwork.

Also, I want to thank Mr. Ionut Toma and Ms. Laura Barbulescu for being such great friends. They landed me the Kinect device that I used in the experiments described in this thesis and supported me during stressful times.

Last but not the least, I would like to thank my family. To my husband Radu to which I cannot say in perfect words how much I appreciate his help. He stayed by my side during all these years, motivated me and proofread my articles (sometimes at 4 AM). This PHD changed our lives in the last 4 years and he supported me all this time. Also, I would like to thank my mother for her guidance. She always told me how important the education is and that is the way to a better life. Growing up in a low-income family in one of the poorest towns of Romania this advice made me learn more and to have a life better that I have imagined.

This work was partially funded by TRAVEE grant of the Romanian executive agency for higher education, research, development and innovation funding - UEFISCDI, joint applied research projects program, 1/2014(PN-II-PT-PCCA-2013-4-1580) and the Sectoral Operational Program Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU 187/1.5/S/155420.

Rezumat

Această lucrare prezintă soluții bazate pe Realitate Virtuală (VR) și Augmentată (AR) în domeniul medical, iar accentul este pus pe două domenii de interes: reabilitarea neuromusculară a pacienților care au suferit un accident vascular cerebral și educația medicală.

In prima parte a tezei sunt prezentate informații despre tehnologiile curente folosite in aplicațiile bazate pe realitatea virtuală şi augmentată. In partea a 2-a sunt prezentate contribuțiile autorului la o soluție ce are ca scop recuperarea neuromotorie prin folosirea realității virtuale şi a feedback-ului augmentat. In partea a 3-a tezei este propusă o soluție inovativă pentru domeniul educației medicale, mai exact în studiul biomecanicii, făcându-se o evaluare a oportunității folosirii atât a realității virtuale cât şi a celei augmentate. Se prezintă o evaluare a rezultatelor obținute pe baza unui set de chestionare completate de utilizatorii care au testat aplicațiile dezvoltate. Scopul proiectului a fost de a construi o soluție cu costuri reduse, cu o experiență adecvată a utilizatorilor, care să poată fi ușor distribuită și adoptată de un număr mare de persoane.

Conținutul acestei teze este bazat în principal pe elemente practice și conține mai multe rezultate experimentale obținute în timpul testelor efectuate folosind diverse tehnologii. S-a urmărit dezvoltarea unor soluții competitive pe cele mai recente tehnologii, acolo unde a fost posibil.

Abstract

This thesis presents solutions based on Virtual and Augmented Reality (VR and AR) in healthcare, analyzing two areas of interest: neuromuscular rehabilitation of stroke survivors and medical education.

The first part of the thesis presents information related to the current technologies used in applications based on virtual and augmented reality. In the second part, the author's contributions to a neuromotor rehabilitation system that aimed the usage of virtual reality and augmented feedback are detailed. The third part of the thesis is focused on the design and implementation of a novel solution that uses both virtual and augmented reality in medical education, and more specifically the biomechanics study, along with the assessment of the results obtained after it was tested with a few users. The goal of the project was to build a low-cost solution with an appropriate user experience that can be easily distributed and adopted by a large number of persons.

The content of this thesis is predominantly focused on practical elements and it contains several experimental results obtained while using various technologies. The goal was to use the latest technology (where it was possible) to be able to provide competitive solutions.

TABLE OF CONTENTS

LIST OF FIGURES 9
LIST OF TABLES 12
INTRODUCTION 13
  1.1 MOTIVATION 13
  1.2 CONTEXT 14
  1.3 GOALS OF THE RESEARCH 16
  1.4 SCIENTIFIC PUBLICATIONS IN CONNECTION WITH THE THESIS 17
  1.5 STRUCTURE OF THE THESIS 18
CURRENT TECHNOLOGIES USED IN APPLICATIONS BASED ON VIRTUAL AND AUGMENTED REALITY 20
  2.1 BACKGROUND 20
  2.2 VIRTUAL AND AUGMENTED REALITY DEVICES 24
  2.3 HUMAN BODY MOTION TRACKING SENSORS 26
    2.3.1 Leap Motion 26
    2.3.2 Kinect 28
    2.3.3 VicoVR 29
  2.4 CONCLUSIONS 30
ICT SOLUTIONS FOR NEUROMOTOR REHABILITATION 32
  3.1 RELATED WORK 32
    3.1.1 Rehabilitation Devices 32
    3.1.2 3D Visualization Solutions 35
  3.2 CONTRIBUTIONS 37
    3.2.1 Avatar Personalization 37
    3.2.2 Virtual Reality Display 42
    3.2.3 Motion Tracking 44
  3.3 CONCLUSIONS 48
AUGMENTED AND VIRTUAL REALITY IN MEDICAL EDUCATION 50
  4.1 RELATED WORK 50
  4.2 CONTRIBUTIONS 53

4.2.1 Approach 54
4.2.2 3D Models 57
4.2.3 Tests with different VR/AR technologies 71
4.2.4 Interactive Biomechanics Lessons (IBL) project 82
4.2.5 Implementation Details 83
4.2.6 Performance Analysis 108
4.2.7 Users Questionnaires Results 115
4.3 CONCLUSIONS 126
CONCLUSIONS AND FUTURE WORK 129
5.1 THE ORIGINAL CONTRIBUTIONS OF THIS THESIS 129
5.2 CONCLUSIONS 130
5.3 FUTURE PERSPECTIVES 131
Acronyms 132
Bibliography 133
Appendices 141

LIST OF FIGURES

Figure 2.1 – Real - Virtual environment transition inspired from Virtuality Continuum schema [AV16b]: A. Real image, B. Leap Motion Image Hands application, C. Leap Motion Demo application *22*
Figure 2.2 – Oculus Rift device *24*
Figure 2.3 – Oculus Rift and EEG cap used in Rehabilitation *24*
Figure 2.4 – Leap Motion Controller *26*
Figure 2.5 – Leap Motion Cameras *26*
Figure 2.6 – Leap Motion Interaction Area *26*
Figure 2.7 – Hand bones *27*
Figure 2.8 – Kinect V1 sensor *29*
Figure 2.9 – Kinect V1 Skeleton position and Bones hierarchy *29*
Figure 2.10 – VicoVR – tracking sensor in VR setup *30*
Figure 2.11 – VicoVR –tracking joints in VR setup *30*
Figure 3.1 – TRAVEE workflow that includes the Avatar Personalization *37*
Figure 3.2 – Avatar Personalization Interface *38*
Figure 3.3 – Textured 3D Models variation for the 5 body types categories: XS [0], S [1], M [2], L [3], XL [4]. *40*
Figure 3.4 – Non-textured 3D Model for Height minimum value. *41*
Figure 3.5 – Non-textured 3D Model for Height maximum value. *41*
Figure 3.6 – TRAVEE VR system setup *42*
Figure 3.7 – Scene example on Oculus Rift *44*
Figure 3.8 – Examples of a TVM and PVM scene configuration. *44*
Figure 3.9 – Kinect and Leap Motion cover areas *46*
Figure 3.10 – Hand bones of the 3D model *46*
Figure 3.11 – Basic hand movements tracked in real-time with Leap Motion Controller *47*
Figure 3.12 – Patient and therapist models animated based on Kinect sensor body tracking *48*
Figure 4.1 – Schematic view of the reviewed scientific papers topics *51*
Figure 4.2 – Initial proposed system overview – AR based *54*
Figure 4.3 – Proposed application divided in subsystems *55*
Figure 4.4 – System overview to support VR and AR *57*
Figure 4.5 – OSIRIX samples: OBELIX [A, B], PETCENIX [C, D], MELANIX [E, F, G] *60*
Figure 4.6 – 3D model preview of the muscles and skeleton as obtained from the HTLL dataset *61*
Figure 4.7 – HTLL dataset - initial image *62*
Figure 4.8 – HTLL dataset – improved image to emphasize the bones tissue *62*
Figure 4.9 – LL dataset – masks (orange for skin, blue for muscles) *63*
Figure 4.10 – ULH dataset skin model correction *64*
Figure 4.11 – Complete skin model *64*
Figure 4.12 – Bones meshes import *65*
Figure 4.13 – Model's correction *65*
Figure 4.14 – Final model *65*
Figure 4.15 – Merging meshes *65*
Figure 4.16 – Make Human basic skeleton *66*
Figure 4.17 – Make Human basic skeleton superimposed on bones model. *66*
Figure 4.18 – Bones model animation error *66*
Figure 4.19 – Rigging and skinning of obtained models using Biped structure from 3DS Max (Left – Bones/Skeleton, Middle – Skin and Right – Muscles model) *66*
Figure 4.20 – 3D Model Generation workflow *67*

Figure 4.21 – A version of the obtained 3D models imported in Unity 68
Figure 4.22 – Bones model in T-pose 68
Figure 4.23 – Static bones model of human anatomy 70
Figure 4.24 – Static muscles model of human anatomy 70
Figure 4.25 – Rigged skin model 70
Figure 4.26 – Stereoscopic rendering on mobile 71
Figure 4.27 – Setting the device in the viewer 71
Figure 4.28 – Cardboard headset 71
Figure 4.29 – Graphical User Interface Reticle used for VR applications developed for Cardboard viewer 72
Figure 4.30 – Samsung S6 device connected to a Gear VR headset 73
Figure 4.31 – Gear VR headset buttons and USB port 73
Figure 4.32– HoloLens test– Unity Editor scene........75
Figure 4.33 – HoloLens test – Emulator (Visual Studio Solution) 75
Figure 4.34 – Lighting conditions effects on test AR scene 76
Figure 4.35 – AR scene example - without tracking........76
Figure 4.36 – Combinations of tracking sensors and mobile devices........77
Figure 4.37 – Motion tracking using additional sensors 77
Figure 4.38 – Unity scene settings to enable Kinect functionality 78
Figure 4.39 – Motion tracking using a Kinect sensor 79
Figure 4.40 – Motion tracking using a Kinect sensor with the mirror movement corrected using the bones generated model 79
Figure 4.41 – Model position changed at runtime on Laptop. 81
Figure 4.42 – Model position changes at runtime and was centered to face detection rectangles on Laptop. 81
Figure 4.43 – Model position and scale changed on Nvidia Shield tablet. 81
Figure 4.44 – Motion tracking of an observed user in AR using a Kinect sensor and OpenCV 81
Figure 4.45 – Opaque model........82
Figure 4.46 – Semitransparent bones model 82
Figure 4.47 – Semitransparent bones and muscles models 82
Figure 4.48 – VR classroom application – Initial screen........84
Figure 4.49 – VR classroom application structure from Unity 85
Figure 4.50 – VR classroom application - Imported models presentation menu structure........86
Figure 4.51 – VR classroom application - Imported models' presentation – Skin........86
Figure 4.52 – VR classroom application – Rotation option (yellow rectangle) 87
Figure 4.53 – VR classroom application - Imported models' presentation – Muscles 87
Figure 4.54 – VR classroom application - Imported models' presentation – Bones........88
Figure 4.55 – VR classroom application – Anatomy Notions – Humerus........89
Figure 4.56 – VR classroom application – Anatomy Notions – Bones........89
Figure 4.57 – VR classroom application – Anatomy Notions – Muscles 89
Figure 4.58 – VR classroom application – Transverse........90
Figure 4.59 - VR classroom application – Coronal Plane 90
Figure 4.60 – VR classroom application – Sagittal Plane........90
Figure 4.61 – VR classroom application – Reference Points – Example A........91
Figure 4.62 – VR classroom application – Reference Points – Example B........92
Figure 4.63 – Movements animations states........93
Figure 4.64 – VR classroom application – Flexion/Extension movement example 94
Figure 4.65 – VR classroom application – Adduction/Abduction movement example 94
Figure 4.66 – VR classroom application – Pronation/Supination movement example........95
Figure 4.67 – VR BlueSky – Anatomy notions - Bones........96
Figure 4.68 – VR BlueSky – Anatomy notions - Muscles 96

Figure 4.69 – VR BlueSky – Reference points 96
Figure 4.70 – VR BlueSky – Movements - Flexion 96
Figure 4.71 – VR BlueSky – Movements - Abduction 96
Figure 4.72 – Marker-based AR – Target image detection (Screenshot from the device on left). 97
Figure 4.73 – Lessons in marker-based AR application – Bones Model (Left) and Muscles Model (Right)..........98
Figure 4.74 – Lessons in marker-based AR application – Anatomy Notions - Bones (Left) and Muscles (Right) 99
Figure 4.75 – Lessons in marker-based AR application – Planes/Axes - Transverse (Left) and Coronal (Right) 100
Figure 4.76 – Lessons in marker-based AR application – Sagittal Plane (Left) and Reference Points (Right)............101
Figure 4.77 – Lessons in marker-based AR application – Movements – Flexion (Left) and Extension (Right) 102
Figure 4.78 – Lessons in marker-based AR application – Movements – Abduction (Left) and Adduction (Right) 103
Figure 4.79 – Marker-based AR application –Extended tracking feature 104
Figure 4.80 – Markerless AR application – Muscles and bones models visible, scaled and positioned based on face detection (Laptop) 105
Figure 4.81 – Markerless AR application – Only muscles model is visible 106
Figure 4.82 – Markerless AR application – Only bones model is visible as displayed on the mobile device..............106
Figure 4.83 – Markerless AR application – Muscles and bones models visible – Muscle highlighted (Laptop).........107
Figure 4.84 – Markerless AR application – Muscles and bones models visible – Bone highlighted (Mobile device) .107
Figure 4.85 – Performance metrics for VR with classroom background scenario 109
Figure 4.86 – Performance metrics for VR with no background scenario....................................110
Figure 4.87 – Performance metrics for AR marker-based scenario 110
Figure 4.88 – Performance metrics for AR markerless on mobile device (no Kinect data) - Haar classifier 111
Figure 4.89 – AR markerless performance metrics – OpenCV, no Kinect, on Windows workstation........................112
Figure 4.90 – AR markerless application performance metrics – OpenCV face detection on mobile device 113
Figure 4.91 – AR markerless application performance metrics – OpenCV face detection on Windows workstation 113
Figure 4.92 – AR markerless application performance metrics – Kinect(idle) and OpenCV motion tracking on Windows workstation 113
Figure 4.93 – AR markerless application performance metrics – Kinect skeletal tracking and OpenCV face detection on Windows workstation using models generated from medical images 114
Figure 4.94 – AR markerless application performance metrics – Kinect skeletal tracking and OpenCV face detection on Windows workstation using the imported models. Blue arrow indicated where the motion tracking has started. 114
Figure 4.95 – Results assessment approach 117
Figure 4.96 – Simulator Sickness Questionnaire - Symptoms variation....................................118
Figure 4.97 – Simulator Sickness Questionnaire – Nausea Symptoms 119
Figure 4.98 – Simulator Sickness Questionnaire – Oculomotor Symptoms 119
Figure 4.99 – Simulator Sickness Questionnaire– Disorientation Symptoms....................................120
Figure 4.100 – Simulator Sickness Questionnaire Symptoms Overall 120
Figure 4.101 – Presence Questionnaires - Realism subscale....................................121
Figure 4.102 – Presence Questionnaires - Affordance to act subscale 122
Figure 4.103 – Presence Questionnaires - Interface Quality subscale 122
Figure 4.104 – Presence Questionnaires - Affordance to examine subscale....................................123
Figure 4.105 – Presence Questionnaires - Self-Evaluation of Performance subscale 123
Figure 4.106 – Presence Questionnaires - Overall Score....................................124
Figure 4.107 – Users' feedback regarding the lack of interruptions during VR applications testing 126

LIST OF TABLES

Table 2.1 – Characteristics of top VR and AR devices 25
Table 3 1 – Age, Height, Weight values intervals for the selected models 39
Table 3.2 – Naming correspondence for the RiggedHand script 46
Table 4.1 – ScanIP hardware requirements 60
Table 4.2 – Key performance metrics highlights on the mobile device (Samsung S6) 109
Table 4.3 – Key performance metrics highlights on AR markerless application 112

CHAPTER 1

INTRODUCTION

This research focuses on solutions based on virtual (VR) and augmented reality (AR) and complex sensors in physical rehabilitation and medical education. VR and AR are known subjects for some time and in the recent years they had popularity surges. The market is filled with low-cost, performant hardware solutions that enable the users to see a different type of content even in their home's commodity. This can be seen as an opportunity on multiple domains as VR and AR can be successfully applied on medical, military, manufacturing, entertainment and games, robotics, education, marketing, tourism and many other fields [MM14]. Our research tackles their applicability in the healthcare field: the first part of the research is focused on stroke survivors' rehabilitation using virtual reality and the second part proposes a novel solution based on VR and AR in medical education, more exactly for biomechanics study.

1.1 MOTIVATION

The academic background of the author is in computer science and medical engineering. Also, the author has extensive professional experience with 3D rendering used in mobile games and in other complex systems. The thesis is a sum of the previous experience, as the solutions proposed have a strong connection with the mobile environment and its usability in healthcare. The research is multidisciplinary and aims to add novelty in the medical education field and rehabilitation, based on the previous experience and studies since many rehabilitation and learning solutions use game-like visualization to attract the users' attention and to maximize the results.

Looking at VR and AR, the hardware market is expected to continuously expand 20-fold in the 2015-2021 period where the most productive one should be 2021 with an estimate of 82.5 million headsets shipped. Goldman Sachs expects high surges in a few key sectors regarding the software market for VR and AR, such as: videogames, live events, video entertainment, healthcare, real estate, retail, education, engineering and defense where it forecasts that until 2025 the biggest two markets will be entertainment and healthcare[1]. This proves the high interest into developing performant user friendly applications that can reach to a high number of persons with a moderate cost. Investing now in research in this domain can boost the market revenue in a few years if consistent applications are delivered to the public.

---

[1] http://cdn.instantmagazine.com/upload/4666/bom_vrar_2017reportpdf.68ec9bc00f1c.pdf

There still are predictions made for the impact of AR versus VR. If initially it was thought that VR will be the one leading the market, the recent experience had proved us that there is a slight shift to AR[2]. A part of this research targets to gather more information regarding the impact of each technology based on the users' feedback on a medical education solution developed both in VR and AR. Since VR has still a few unresolved problems such as cybersickness, the experiments were designed so that they take these factors into account.

The research is composed mostly of practice based experimental results. The solutions covered into this thesis aim to be easily adopted by the reader to reproduce the expected behavior targeting new technology and recent state of the art data.

The outcome is the contribution to the field knowledge with advantages and disadvantages of the chosen approaches. The results presented in this thesis were disseminated at various international conferences and the approached methods had a general positive feedback. The solutions built are complex as they have additional features to enhance the users' immersion into the applications developed by using real-time motion tracking or realistic 3D models.

1.2 CONTEXT

The research is divided in two main parts: the first one is related to the author's contributions to the TRAVEE (Virtual Therapist Through Augmented Feedback) project and the second one presents a solution based on both virtual and augmented reality for improving the learning process of biomechanics study. Both parts have similarities such as working with virtual reality, motion tracking devices and with accent on the human motion biomechanics.

The first part of the research focuses on a novel rehabilitation solution named TRAVEE that aimed to aid the stroke survivors with neuromotor deficiencies. Every two seconds someone somewhere in the world is having a stroke and every 10 seconds a life is claimed where 80% of all the people that suffered a stroke are from low and mid-income countries[3]. Stroke survivors often remain incapacitated due to the lack of oxygen and nutrients for the affected brain area. An obstruction that lasts even a few minutes can damage the neurons and therefore they die. The functions that were handled by these neurons are affected and the neuromotor disabilities have the biggest incidence. Thanks to the neuroplasticity of the brain the functions that were executed by the affected

---

[2] https://www.digi-capital.com/news/2017/01/after-mixed-year-mobile-ar-to-drive-108-billion-vrar-market-by-2021/#.WhQ8nzdx2Uk

[3] http://www.worldstrokecampaign.org/learn/facts-and-figures.html

neurons can be relearned and their function can be taken over by other healthy neurons from vicinity [OF15].

Neuromotor disabilities can significantly affect a person's life, especially the activities of daily living (ADL) like eating or washing. This can downgrade significantly one's quality of life as the rehabilitation process is focused on long term kinesiotherapy. The kinetotherapeutic support (classical therapy) is limited due two facts: the long-time sessions (up to 5 hours per day) and the growing number of affected persons [AV15c]. The number of specialized personnel is not growing with the same pace and as a result fewer rehabilitation sessions can be applied to each patient.

The *Simulation* hypothesis indicates that to relearn a particular movement one has to visualize the movement either on its own or as an observation due to the strong connection between the motor and cognitive brain mechanisms [AV15b]. Basically, the patient can start the rehabilitation very early even if he or she just observes certain movements as an example to someone else. Since in the early days the patients stay mostly lied in the hospital bed, it would be very difficult to see the movements executed by the kinesiotherapist. A novel solution that aimed to be an alternative to other rehabilitation devices and an adjuvant for the classical therapy is represented by TRAVEE and each of these parts will be detailed accordingly in the next chapters.

The second part of the research brings into light a new idea regarding the opportunity of utilizing VR and AR applications in medical education. Their main advantage is that they can simulate various realistic scenarios effectuated in a safe environment as the students can be trained on various procedures. The approached subject targets to better understand the processes behind the human movement biomechanics. The initial idea was to develop a system that aims to improve medical education learning techniques using AR or VR and biomechanics seemed a good fit. Both are useful tools for developing a system that enhances the visual feedback with additional information. In the first concept the project was focused only on an AR based solution that tracked the movements of an observed user in order to display an animated 3D model according with the tracked data. Medical education contains many anatomical notions and this interactive solution aimed to enhance the user's attention into the learning process. Fortunately, with the current software tools available on the market it was considered an opportunity to develop a solution that targets both AR and VR using low-cost, mobile technology available for a large number of persons. The goal is to test the developed applications on various users and based on their feedback to assess the impact of each technological system.

1.3 GOALS OF THE RESEARCH

The goals of this research were to create solutions based on virtual and augmented reality for the healthcare system. The solutions provided are complex and include real-time motion tracking and realistic 3D models obtained through image processing of medical images. The research topics are education and rehabilitation and both of them imply a certain level of biomechanics notions knowledge and one can be considered an intermediary step for the another.

The first part of the research was aimed at rehabilitation, while being part of the TRAVEE project. A patient will wear an HMD (Head Mounted Display) to visualize the virtual rehabilitation sessions, as set by the therapists, and to see its own progress enhanced. Since the stroke rehabilitation has more impact in the first stages of recovery, the patient will be able to see the movements through the HMD. The enhancement is very important because it aids the patient not to give up at the rehabilitation sessions if no progress is immediately obvious. The project included modern technologies such as: VR, robotics, BCI (Brain Computer Interface) and FES (Functional Electrical Stimulation). The contributions of the thesis' author to this project were in the first part of the development. The areas of contribution were:

a. *Assessment of the available rehabilitation devices* - This part details a few rehabilitation devices that can be linked with the solution offered by TRAVEE.
b. *Avatar Personalization* - A first stage implementation of the patient virtual model personalization based on different conditions such as: weight, height, skin and hair color.
c. *Virtual reality setup* –The initial setup was done using an Oculus Rift DK1 device.
d. *Motion Tracking Integration* – The purpose was to display the movements that a user is making in the virtual reality environment. Two types of technologies were used, detailed in the second chapter.

The second part of the research is the most diverse one and it is focused on an innovative solution for biomechanics study. The project's name is Interactive Biomechanics Lessons (IBL) and has had as initial concept a solution that targeted the use of augmented reality as a training platform to see the changes that are occurring during the movement in real-time with a 3D model super imposed over an observed user's image. Later it was considered to extend the usage to both AR and VR since with current development tools this would have been a great opportunity to test both technologies on similar scenarios. The goal was to assess the best technological approaches as learning solutions and the users' feedback for them. The point of interest is assessing the importance of immersion and presence versus the minimization of disturbance factors. The experiment design covered 4 cases (2 for VR and 2 for AR):

a. The user wears an HMD and is isolated from external factors. The VR environment is set into a virtual classroom.
b. The user wears an HMD, is isolated from external factors and the VR environment is removed leaving an open space. This scenario was added due to the observed cybersickness.
c. The user uses a marker-based AR application. The virtual lessons have similar structure with the ones provided in the VR setup.
d. The user uses a markerless AR application. This is more at an experimental stage where the user can see the bones' 3D model animated over its image based on the tracked motion.

The first three scenarios do not include motion tracking, and they are based on mobile friendly, cost effective learning solutions. These scenarios were part of an in-depth testing and the results are discussed in the last part of the thesis. This project was developed from the start till the end. It is detailed in the fourth chapter.

1.4 SCIENTIFIC PUBLICATIONS IN CONNECTION WITH THE THESIS

A significant part of the work involved in this thesis was published in the following scientific papers (sorted by the publication year):

2018

1. Alexandra Voinea and Florica Moldoveanu, "*A Novel Solution Based on Virtual and Augmented Reality for Biomechanics Study*" in Scientific Bulletin of UPB, Series C. vol. 80, no.2/2018, ISSN 2286-3540, pp.29-40. WOS:000434342000003.

2017

2. Alexandra Voinea, Florica Moldoveanu and Alin Moldoveanu, "*3D Model Generation and Rendering of Human Musculoskeletal System Based on Image Processing*" in Proceedings of **the 21st International Conference on Control Systems and Computer Science**, Bucharest, Romania, pg. 263-270, DOI: 10.1109/CSCS.2017.43, May 2017. (**IEEE**)

2016

3. Alexandra Voinea, Alin Moldoveanu and Florica Moldoveanu, "*Bringing the Augmented Reality Benefits to Biomechanics Study*" in Proceedings of **the 2016 Workshop on Multimodal Virtual and Augmented Reality** (MVAR 2016), pg. 8757-8764, Tokyo, Japan, DOI: 10.1145/3001959.3001969, ISBN: 978-1-4503-4559-0, November 2016. WOS:000392302900009.

4. Alexandra Voinea, Alin Moldoveanu and Florica Moldoveanu, "*Efficient Learning Technique in Medical Education Based on Virtual and Augmented Reality*" in Proceedings of

9th Annual International Conference of Education, Research and Innovation, Seville, Spain, pg. 8757-8764, DOI: 10.21125/iceri.2016.0975, ISBN: 978-84-617-5895-1, November 2016. WOS:000417330208102.

2015

5. Alexandra Voinea, Alin Moldoveanu, Florica Moldoveanu and Oana Ferche, "*Motion Detection and Rendering for Upper Limb Post-Stroke Rehabilitation*" in Proceedings of **the 5th International Conference on e-Health and Bioengineering** – EHB 2015, Iasi, Romania, pg. 811-814, DOI:10.1109/EHB.2015.7391471, ISBN: 978-1-4673-7544-3, November 2015. WOS:000380397900124

6. Alexandra Voinea, Alin Moldoveanu and Florica Moldoveanu "*3D Visualization in IT Systems Used for Post Stroke Recovery: Rehabilitation Based on Virtual Reality*" in Proceedings of **CSCS20: The 20th International Conference on Control Systems and Computer Science**, Bucharest, Romania, pg. 856-862, 10.1109/CSCS.2015.123, ISBN: 978-1-4799-1779-2, May 2015. WOS:000380375200125

7. Alexandra Voinea, Alin Moldoveanu, Florica Moldoveanu and Oana Ferche," ICT *Supported Learning for Neuromotor Rehabilitation - Achievements, Issues and Trend*s" at **The International Scientific Conference eLearning and Software for Education**, Bucharest, Romania, April 2015, Issue 1, pg. 594-601. WOS:000384469000086

8. Oana Maria Ferche, Alin Moldoveanu, Florica Moldoveanu, Alexandra Voinea, Victor Asavei and Ionut Negoi, "*Challenges and issues for successfully applying virtual reality in medical rehabilitation*" at **The International Scientific Conference eLearning and Software for Education**, Bucharest, Romania, April 2015, Issue 1, pg. 494-501. WOS:000384469000073

9. Oana Ferche, Alin Moldoveanu, Delia Cinteza, Corneliu Toader, Florica Moldoveanu, Alexandra Voinea, Cristian Taslitchi, "*From Neuromotor Command to Feedback: A survey of techniques for rehabilitation through altered perception*" at **E-Health and Bioengineering Conference (EHB)**, Iasi, Romania, November 2015, pg. 1-4. WOS:00038039790010

1.5 STRUCTURE OF THE THESIS

Chapter 2 contains details regarding the current technologies used in VR and AR applications. This section contains background information regarding their definitions and technical details. The chapter continues with a short presentation of a few display devices that were of the most interest at the time of the research and continues with a list of motion tracking devices that were considered to obtain skeletal tracking of an observed user.

Chapter 3 contains the author's contributions to the TRAVEE project. The research focused on solutions used in neuromotor rehabilitations. The literature review of existing solutions contains two subjects: rehabilitation devices and 3D visualization methods. Afterwards, implementation details of three areas are provided: avatar personalization, virtual reality setup and motion tracking.

Chapter 4 contains the details of a novel educational solution named Interactive Biomechanics Lessons that uses virtual and augmented reality for enhancing the learning process for biomechanics study of human motion. This section contains the architecture of the system including devices, sensors and software solutions required for the implementation. A part of the research was focused on obtaining realistic 3D models of human bones and muscular systems. Implementations details are provided along with a few experimental tests on various technologies. Also, this chapter contains the performance metrics of the developed applications and the results obtained based on user's feedback.

Chapter 5 contains the conclusions of the research and summarizes the personal contributions. A few directions for future research that require additional time, work capacity or funding are mentioned as well.

CHAPTER 2

CURRENT TECHNOLOGIES USED IN APPLICATIONS BASED ON VIRTUAL AND AUGMENTED REALITY

This chapter addresses the basic information about virtual and augmented reality and continues with their applicability in medical applications. Details about both technological systems are presented followed by a short revision for some of the most interesting available devices since the complete list is exhaustive at the moment.

VR and AR are known topics in the scientific community for many years and they had a significant improvement in the recent years sustained by the media enthusiasm based on the advancements of both technologies. Nowadays, developing a solution based on VR and AR is in the best of times as the number of available hardware and software solutions are growing in a fast pace [AV16b]. The developers, and with them the scientific community, have many tools that are helping them build better solutions compared with the ones proposed a few years ago and to overcome a part of the known drawbacks (E.g. hardware and design changes to minimize the cybersickness). Fortunately, now are available in high numbers low cost, performant hardware solutions and many system engines that provide consistent VR and AR support.

Healthcare is one of the beneficiaries of these technological advancements. The new level of interaction available through these technologies is a good fit for this research's topics although there are a few challenges and issues for successfully applying virtual reality in medical rehabilitation [OMF15]. Two approaches were considered within this thesis for using VR and AR in healthcare:

a. *In the rehabilitation part,* where the possibility of executing a large number of exercises (compared with limited hours of kinesiotherapy) is speeding the recovering process for the stroke survivors.
b. *In the medical education part*, the beneficiaries will be especially youth individuals that are learning biomechanics notions

2.1 BACKGROUND

Virtual and augmented reality are both technological systems created with software and their goal is to immerse the users into each specific environment. VR displays a fully artificial environment where the user should believe that is the real world[4]. Also, "*VR is a realistic, real-time, 3-dimensional (stereoscopic) computer simulation of physical objects and space*"[5]. These are the technical definitions of virtual reality and our aim is to use it into medical applications. According to [JVWR14], along the years there was no clear consensus about the meaning of VR in medicine. There we find that some[6] shared the same vision of VR in their reviews as they saw the VR as a "*collection of technologies that allow people to interact efficiently with 3D computerized databases in real time using their natural senses and skills*". Others[7], described the VR correlated with the human experience: "*a real or simulated environment in which a perceiver experiences telepresence*". Schultheis (2001) mentions that VR is "*an advanced form of human-computer interface that allows the user to interact with and become immersed in a computer-generated environment in a naturalistic fashion*", while Bellani and Fornasari (2011) view VR "*as only a simulation of the real world based on computer graphics*". The definitions mentioned above are underlining two approaches of VR in healthcare: *VR as a simulation tool* and *VR as an interaction tool* [JVWR14].

On the other side, augmented reality blends real environment and virtual elements while virtual reality is targeting to display only the virtual environment. AR is adding virtual elements over the real-world display [IACG15] and according to [RTA97] AR is based on techniques developed in VR and interacts not only with a virtual world but has a degree of interdependence with the real world. Reference [OB05] mentions the fact that AR systems have three major components:

1. Tracking and registration;
2. Display technology;
3. Real-time rendering.

The main motivation of the usage of medical augmented reality lies in the "*need of visualizing medical data and the patient within the same physical space*" [MM14].

The advantage of AR is the fact that the users are feeling comfortable because the presence is highly achieved. This is a consequence of the fact that the users are still present in the real environment while the virtual elements are superimposed on top of it. On the other side, in VR the medium (environment) can be fully controlled as opposed to AR and this factor is important when it comes to minimizing the external disturbing factors. For example, a user with an HMD and some sound proof headsets can be

---

[4] http://whatis.techtarget.com/definition/virtual-reality

[5] http://www.businessdictionary.com/definition/virtual-reality-VR.html

[6] Rubino (2002), McCloy and Stone (2001), Szekely and Satava (1999)

[7] Riva (2003); Steuer (1992)

approximately fully isolated by outside factors that can disrupt the attention from the operations executed in the virtual environment, while this scenario is unachievable in AR. Both VR and AR are considered reliable methods to simulate realistic experiences and this makes both technological systems a good fit to reproduce real situations for training or educational purposes, into a safe environment.

There are two basic AR software implementation types: marker-based and markerless [MCFM14a]. Results showed that a higher sense of presence was shown for AR invisible marker system compared with the visible marker one [IACG15]. A marker-based application solves the problem using visual markers detectable with computer vision methods (e.g. 2D barcodes) [SS12].

Depth perception issues were noticed with medical AR systems while using semitransparent structure overlay onto visible surfaces [ZY14]. In AR the virtual elements are drawn on top of the real environment and besides that, factors like the lighting can affect the output image and as a solution "*seven depth cues are evaluated with rendering using depth-dependent color and the use of aerial perspective shown to give the best cues.*" [ZY14].

Besides VR and AR there is a third notion named Mixed Reality (MR) and often this one is confused with AR. While AR refers to a system in which an enhanced version of the real world is available for the user, MR refers to a system that combines real and virtual objects and information. The enhancement elements are virtual and can include object and information[8]. Basically, in Mixed Reality the physical and digital objects co-exist and interact in real-time[9]. Figure 2.1 displays the real and virtual environments and the blend between them.

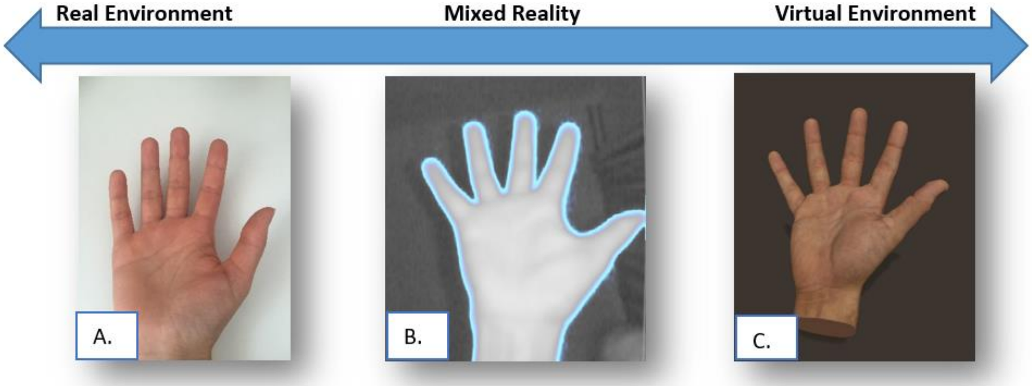


Figure 2.1 – Real - Virtual environment transition inspired from Virtuality Continuum schema [AV16b]: A. Real image, B. Leap Motion Image Hands application, C. Leap Motion Demo application

---

[8] http://courses.cs.vt.edu/cs5754/lectures/AR-MR.pdf

[9] https://www.foundry.com/industries/virtual-reality/vr-mr-ar-confused

Another important aspect regarding the AR is the fact that it shouldn't be limited only to the graphics while it targets the real environment augmentation. Indeed, the graphical augmentation is our focus in this thesis thus in a larger scale the augmented reality can be extended to a multimodal approach [HS16].

After many psychological studies that investigated the adverse consequences of the incorrectly rendered focus cues in stereoscopic displays it was found that the these might contribute to the commonly recognized issues, such as: distorted depth perception, diplopic vision, visual discomfort and fatigue and degradation in oculomotor response [HH14].

Cybersickness is an important topic while developing for VR and it was noticed that some design changes can reduce cybersickness as this is not only related with the hardware specifications. In the following lines a few examples are provided:

1. In virtual environment, the developer should use ramps instead of stairs (based on a survey from Oculus Rift Tuscany Demo) [JD14].
2. A reduction of cybersickness was noticed if the HUD (Heads-up Display) elements are blended in the 3D scene, instead of using the classic 2D elements. This was strongly observed while developing our applications and the positive impact of this design.
3. FOV and Focus are key elements and they should be properly calibrated. For example, the human eye is automatically focusing for near or distance. The best approach will be using an eye tracking device to see where the user is looking into the scene.

There are two VR software solutions that were considered in the same line with the topics of this research. First one is the **People | Be fearless** VR app developed by Samsung and was available with Samsung Gear VR. It tacked phobias, such as: fear of public speaking[10] and fear of heights[11]. The users had available a few relevant scenarios to interact with the fear stimuli. These apps aimed to improve the resistance in real life for these stimuli to be able to overcome the fear. Another app of interest is **Fusion Tech 3D**. This is a project made within the Stanford University that was later acquired by Luminate Health Systems[12] . It is an application available on multiple platforms (mobile, desktop, VR) that is able to visualize 3D data inside the human body. The devices that supported this app were iOS, Android, Oculus and zSpace.

---

[10] https://www.oculus.com/experiences/gear-vr/942681562482500/

[11] https://www.oculus.com/experiences/gear-vr/821606624632569/

[12] https://www.luminatehs.com/

2.2 VIRTUAL AND AUGMENTED REALITY DEVICES

Along the years, the technology had a fast advancement that offered the opportunity to have a wide range of devices available for development. Depending on their price and accessibility some of them had a higher success rate compared with others. In the following are mentioned a few devices that captured our attention and that proved to have a significant impact for these technologies.

For displaying the VR applications, CAVE (Cave Automatic Virtual Environment) and HMD devices proved their efficiency in the past years [FT14]. CAVE is a room-sized virtual reality system and it consists in a series of projectors that are places on the walls of a room. On the other side, HMD is a device worn on the head where the users cannot (shouldn't) see outside the headset while displaying VR. There are a few key differences between these systems and one of them is related with the fact that with the CAVE system the user can still see and perceive its body as normal, while using an HMD this is not available as the user can see most of the time only its hands. In consequence, the immersion of the user in the virtual environment is total while using CAVE compared with the usage of HMDs[13] thus on other hand the prices and mobility of the HMDs are better. Figures 2.2 and 2.3. contain images of a well-known HMD – Oculus Rift.

Figure 2.2 – Oculus Rift device [14]

Figure 2.3 – Oculus Rift and EEG cap used in Rehabilitation[15]

Regarding HMDs, there is a subset of devices named OHMD (Optical Head Mounted Display) that are allowing the users to see through them. We can mention here well-known devices, such as: Google Glass or Microsoft HoloLens. This special category of devices is suited for displaying augmented reality. A consistent number of authors mentioned in their scientific papers the usage of HMD for visualizing AR applications

---

[13] https://www.vrs.org.uk/virtual-reality-environments/cave.html

[14] Image source - http://www.cgmagonline.com/2015/05/08/oculus-rift-release-date-announced/

[15] Image source - https://www.theverge.com/2016/8/11/12443026/virtual-reality-exoskeleton-paraplegic-oculus-rift

[IACG15] [MM14] [CK14] [FC14] [MCFM14a]. However, any device can be utilized for displaying augmented reality if it respects the following rules:

1. The possibility to combine real with virtual;
2. To be interactive in real time;
3. To be registered in 3D space [MM14].

In AR the virtual information is superimposed over the real world and the registration can be interpreted as the accuracy of spatially aligning the virtual elements in the real world. The coordinate system of the real world where the virtual elements are projected should be resolved regardless the environment changes or given time[16]. A camera is needed to be able to get the real environment information to combine it with the virtual elements. There is a wide list of devices that are a good fit for AR where we can mention: mobile devices (smartphones or tablets), desktop/monitors (used with external web cameras) or HMDs. As stated before, OHMD are a perfect fit for AR systems but their costs are often substantial [AV16b]. Table 2.1 contains a list with devices of interest to display VR or AR along with some basic information about their capabilities and prices [AV16b].

Table 2.1 – Characteristics of top VR and AR devices

Device Name	**Reality Type**	**Refresh Rate [Hz]**	**FOV [degrees]**	**Resolution [pixels]**	**Processing Source**	**Price [USD]**
Oculus Rift	VR	90	110	1080x1200	Computer	599
Samsung Gear VR	VR	Depends on the smartphone (~*60*)	96	Depends on the smartphone (*e.g. Samsung Galaxy S7 -2560 x1440*)	Smartphone	99+[17]
HTC Vive	VR	90	110	1080x1200	Computer	799
Sony VR	VR	120	100	960x1080	Game Console	399
Atheer AiR	AR	NA[18]	50	1280x720	Built-in	3950
Microsoft HoloLens	AR & VR	60	120	1268x720	Built-in	3000

---

[16] http://www.cs.bham.ac.uk/~rjh/courses/ResearchTopicsInHCI/2014-15/Submissions/yan--yan.pdf

[17] At the VR headset price is added the smartphone price. This is only an accessory for the compatible smartphones.

[18] The refresh rate for this device is not available. However, the available information in this direction is that the device is based on NVIDIA Tegra K1 processor.

2.3 HUMAN BODY MOTION TRACKING SENSORS

In this subchapter we are discussing about real-time human body motion tracking, more exactly skeleton tracking, as this feature was present on both projects included in this research while developing for AR and VR. Following is provided an overview of the skeletal tracking sensors that were considered during development. The skeleton movements tracked by the sensors were used to animate a 3D virtual human avatar.

In the next sections are presented the technical details of 3 sensors that can provide skeletal tracking in VR and AR based applications: Leap Motion Controller, Kinect and VicoVR. While Leap Motion controller and Kinect were used in our research, VicoVR sensor is a recently released device that provides performant skeletal tracking suited for mobile development.

2.3.1 Leap Motion

Leap Motion Controller has incorporated two cameras and three infrared lights (Fig.2.5). The infrared light is outside the visible light spectrum with a wavelength of 850 nanometers. Its viewing range was roughly 60 cm (2 feet) above the device using the initial version of the software but with their new software (Orion beta) this was extended to 80 cm (2.6 feet). Fig.2.6 displays the device's interaction area. The device is connected to a workstation (PC) via an USB controller (Fig.2.4). After the sensor data is read, resolution adjustments are performed, if necessary. Data that takes form of a grayscale stereo image, separated into left and right cameras, is streamed via USB to the tracking software. As opposed to other solutions, the controller doesn't generate a depth map but instead applies advanced algorithms to the raw data provided by the sensor. The obtained images are analyzed to reconstruct a 3D representation of what is seen and then the tracking algorithms interpret the 3D data while the position of the occluded objects is inferred. On top of that, filtering techniques are applied[19].

Figure 2.4 – Leap Motion Controller[20]

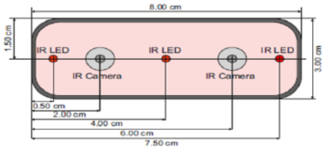


Figure 2.5 – Leap Motion Cameras

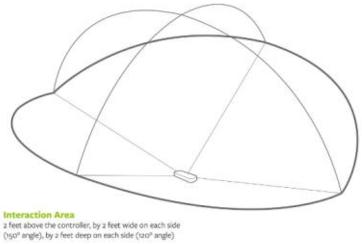


Figure 2.6 – Leap Motion Interaction Area[21]

---

[19] http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work

[20] Image source : http://www.robotshop.com/blog/en/explore-virtual-reality-with-leap-motion-3d-motion-controller-16806

[21] http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

Leap Motion is capable of skeletal tracking only for the user's hands as Leap Motion has *Bone API* that will extract data from the tracked hands based on the anatomy information. The onscreen rigged hands have properties such as: joint positions, bone lengths and individual bones bases as they mirror the behavior of real hands. The virtual hands can appear and interact as physical objects in physics engines. There are a few key aspects regarding the bones references. For example, the human hand has 5 fingers, each has four bones with an exception, the thumb that has three. To overcome this, *Bone API* assumes a zero-length Metacarpal bone (Fig.2.7).

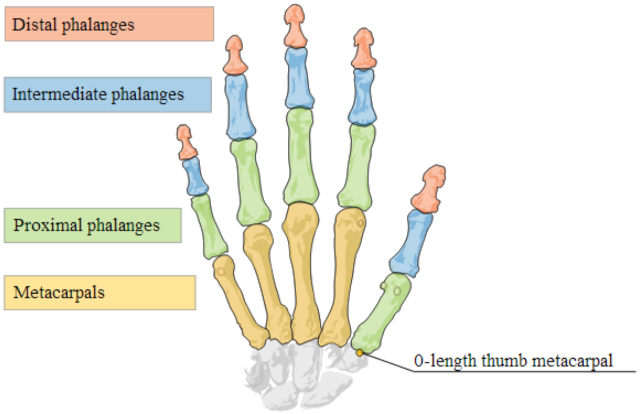


Figure 2.7 – Hand bones [22]

Bones are ordered from proximal to distal where the wrist is considered the base. Taking this into account, their indexes will be from 0 to 3 in the set order or they can be referenced by their anatomical names. Leap Motion Controller will enhance developers to rig meshes of hand and fingers based on the tracked data[23].

Leap Motion Controller is used in a wide range of applications. From games, to desktop applications for seamless presentations till scientific applications. There are some applications of interest for our research topics, where we can mention: *Cyber Science Motion* – to explore, dissect and assemble a human skull and *HandEye* – an experimental app for eye-hand coordination rehabilitation. Also, there are a few scientific papers available that mention hand rehabilitation using this device [MK14] [AK16].

---

[22] Image source: http://blog.leapmotion.com/skeletal-tracking-101-getting-started-with-the-bone-api-and-rigged-hands/

[23] http://blog.leapmotion.com/skeletal-tracking-101-getting-started-with-the-bone-api-and-rigged-hands/

2.3.2 Kinect

Kinect sensor, developed by Microsoft, is a device with depth sensing technology that has a built-in color camera, an IR (infrared emitter) and a microphone array (Fig.2.8). This device can sense the location and movements of people as well as their voices[24] and can be connected to an Xbox console, PC or tablet (Window OS). There were two versions of this sensor available on the market V1 and V2 at the time of the research. The initial one, V1, had lower specs regarding the color camera resolution (640x480 vs 1920x1080). Also, it had a smaller Horizontal and Vertical FOV (57 vs 70 degrees and 43 vs 60 degrees) and number of skeleton joints defines (20 vs 26 joints)[25]. While Kinect V1 could have tracked 2 full skeletons, V2 was able to recognize 6 people and track two[26].

Kinect sensor can perform whole-body skeletal tracking of an observed user. It can locate the joints and track their movements in real-time using the IR camera. One of the advantages is the fact that no specific pose or calibration is necessary for a user to be tracked. Although Kinect sensor can record the hands movements the quality of the movements is lower since there is only one bone assigned for the hand (V1). As opposed with Leap Motion Controller that had a corresponding virtual bone to the anatomical ones. Indeed, Kinect V2 had better results to track the hands but still didn't offer the same quality as the Leap Motion Controller. For the research work of this thesis the most used sensor was Kinect V1 as it was added in both projects. Fig.2.9 illustrates the tracked joints while using a Kinect V1 sensor, as well as their hierarchy versus the parent joint (*Hip Center*). Until recently, Kinect was the only viable solution available that could perform skeletal tracking, even with some downsides regarding the tracking errors. A few back-up solutions were improvised such as: using two Kinect sensors [MCFM15] or using it with OpenCV [BK15]. This device is by far the most used sensor for skeletal tracking that was mentioned in the studied scientific papers [DL15] [ZG15] [MCFM14a] [CK14] [TM14].

---

[24] https://developer.microsoft.com/en-us/windows/kinect/hardware

[25] http://zugara.com/how-does-the-kinect-2-compare-to-the-kinect-1

[26] https://msdn.microsoft.com/en-us/library/hh973074.aspx

Figure 2.8 – Kinect V1 sensor[27]

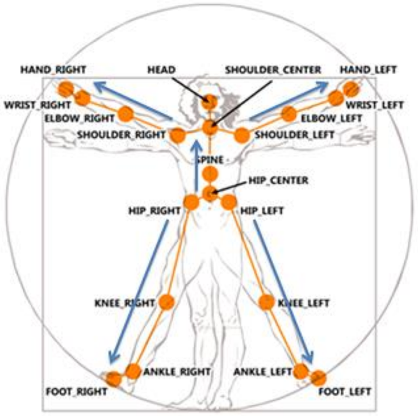


Figure 2.9 – Kinect V1 Skeleton position and Bones hierarchy[28]

2.3.3 VicoVR

This device is one of the newest technology available for full body tracking. This is a viable alternative for Kinect sensors and as a plus it is used with mobile platforms. While the other two devices needed to be connected to a PC in order to work this one is a Bluetooth accessory that can provide wireless skeletal tracking to Android and IOS devices. The device can be added to a VR setup based on a Cardboard like viewer, a smartphone and the tracking sensor (Fig.2.10 and Fig.2.11). This device would have been a perfect fit for our mobile solution for Interactive Biomechanics Lessons but unfortunately it was available for purchasing too late to fit in our research timeline. However, because of its features we considered that it is worth mentioning.

This device was into our attention after their INDIEGOGO campaign. This was marketed as the "*world's first Full Body Controller (Gaming System) for Mobile Virtual Reality*"[29] as it offers precise 3D coordinates of 19 body joints. The depth data of the scene is processed by its internal processor as outputs body tracking data without requiring additional processing power from the VR display device. The transfer is wireless to any Android or IOS based HMD. The SDK is available on well-known games systems such as: Unity 3D and Unreal Engine 4.

---

[27] Image source: https://www.generationrobots.com/en/401430-microsoft-kinect-sensor.html
[28] Image source: https://msdn.microsoft.com/en-us/library/jj131025.aspx
[29] https://www.indiegogo.com/projects/vicovr-full-motion-gaming-in-virtual-reality-vr-technology#/

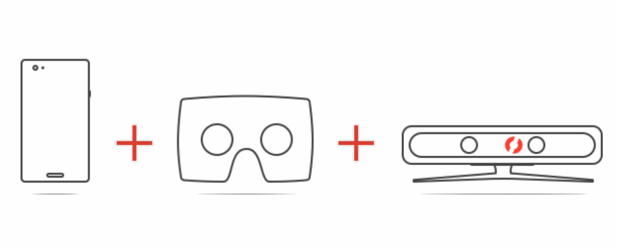


Figure 2.10 – VicoVR – tracking sensor in VR setup[30]

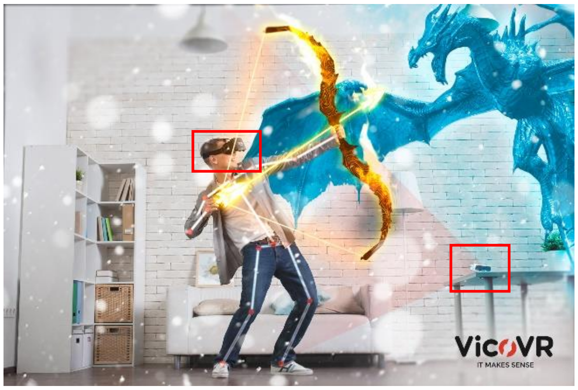


Figure 2.11 – VicoVR –tracking joints in VR setup [31]

2.4 CONCLUSIONS

In this chapter we showcased basic information about the VR and AR systems and their definitions and usability with accent on medical applications. Also, a special attention was accorded to the available devices used for displaying the VR and AR content. The total number of the available devices is high, and we tried to focus on the ones of interest for this research.

For VR, Oculus Rift is a device often mentioned. This is due the fact that this device was one of the key factors of the VR expansion registered in the recent years. Some problems were encountered when trying to develop VR for Oculus using a Laptop (CPU: Intel i7 – 4710HQ at 2.5GHz, GPU: Nvidia GeForce GTX 850M and 8GB of Memory). This is correlated with the fact that the display resolution and frequency in VR is bigger compared with classic applications/games. As a backup a simple viewer such as Google Cardboard was used coupled with a smartphone (Samsung S6). This device's total resolution is 2560 x1440 and is divided for each eye. Fortunately, the mobile platforms market was in continuous development in the past years and the devices performances and specs raised each year. This is considered an opportunity even for VR applications that weren't designed to be the principal beneficiaries. Another point to sustain this is the fact that Facebook (which acquisitioned Oculus a few years ago) recently mentioned the release of a standalone device. This makes a strong case for the necessity of the market to expand to simpler, mobile setup that can be used on the go. In fact, this was our motivation when we started the research and development of the Interactive Biomechanics Lessons (IBL)project.

---

[30] Image source: https://vicovr.com/

[31] Image source: https://www.virtualreality-news.net/news/2016/may/20/vico-vr-crowdfunding-bring-affordable-positional-and-body-tracking-mobile-vr/

For AR we initially focused on two types of devices: OHMD (HoloLens) and mobile devices. The aim was to develop applications to cut edge technology such as Microsoft HoloLens thus we found some challenges regarding the acquisition of this device and as a result the focus was shifted on tablets, smartphones and laptops as they met the requirements to develop AR applications on them. Although HoloLens is unlocking a new level of realism using holograms the second solution will be available for a much larger number of users and this will enhance a new level of testing since the targeted pool of potential users is considerably high. For example, in 2015 there were ~ 1.4 billion Android users and the market is continuously growing.

The manufacturers improved their devices capabilities to give to their users the best experiences. The devices have different components and in consequence their prices may have high variations (as seen in Table 2.1). The scope is to fully immerse the user into the virtual world, not only with high quality graphics but also without motion sickness. This is one of the major concerns for VR development nowadays. This can be improved using a series of special design approaches, but devices capabilities have an important role, too [ZG15].

This chapter also contains details regarding a few skeletal tracking devices suited for integration into VR and AR applications. In this research we used two types of sensors: one for the whole body and another one that tracked only the hand: for the whole body was used a Kinect sensor and for the hand a Leap Motion controller. Although, the Kinect sensor can be considered "old" technology it is extremely used and present in many scientific papers. Recently new devices started to appear on the market with similar capabilities as some were built especially for VR systems. In this context VicoVR sensor was briefly presented which is a promising solution compatible with mobile technology, although it tracks fewer joints compared even with the first version of the Kinect sensor. The device has its own processor source that does all the computations to avoid the transfer of the computational burden on the VR display device (a smartphone in this case).

CHAPTER 3

ICT SOLUTIONS FOR NEUROMOTOR REHABILITATION

This chapter contains the details of the first part of the research that was focused on solutions used in neuromotor rehabilitation of stroke survivors. The following contains the author's contributions to TRAVEE project and it covers multiple areas such as assessment of existing rehabilitation devices, patient avatar's personalization, virtual reality setup and motion tracking integration.

3.1 RELATED WORK

This section is divided in two parts, where at start a list of rehabilitation devices is presented and it's focused more on the hardware part. The mentioned devices are related with the product TRAVEE. They contain complex technologies such as FES (Functional Electrical Stimulation) and robotics. The review is continued with an analysis of the software solutions used in rehabilitation.

3.1.1 Rehabilitation Devices

In rehabilitation, there are a significant number of eLearning solutions that can be used complementary with the classical therapy, based on kinesiotherapy [AV15c]. The solutions contain complex technologies such as FES, NMES (Neuromuscular Electrical Stimulation), EMG (Electromyography), BCI and robotics. Products that target the patient's rehabilitation are available on the market and [PM14] has a detailed list with more than 100 devices used for upper limb rehabilitation. They are classified based on the following criteria:

a. *The joint system they support*.
b. *The device's DOF (Degree of Freedom).* This is represented by a sum of all independent movements performed by the joints of the device.
c. *The supported movements types*, such as: abduction, flexion/extension, pronation/supination, grip and release, horizontal and vertical displacement, etc. Also, the movements can be active or passive (with or without external help to execute a certain movement).
d. *The patient's health condition* as neuromotor rehabilitation solutions can be used for certain conditions such as: stroke, cerebral palsy, essential tremor, multiple sclerosis, primal cord injuries and traumatic brain injury.

In the following, a few devices of interest, that are linked to TRAVEE project, are shortly described.

Music Glove

Music Glove[32] is a rehabilitation device that aims to improve the hand function for stroke survivors and other neuromotor disabilities. The device is connected to a VR module that displays a music environment similar with the games. In the respective application the subjects should follow the musical notes that appear on the screen making certain moves. These types of exercises are helping the subject improve the hand function after a neuromotor disability. The benefits of using *MusicGlove* device are supported by a clinical study [NF14] with patients which suffered a stroke. In that study participated 12 patients which were diagnosed with mild chronic hemiparesis. They were randomly selected to use the *MusicGlove* device in the same time with conventional therapy. Each selected patient used the device for 6 sessions of one-hour length, three times per week for 2 weeks. At the end of the study was shown that the object grip movement of the selected subjects improved with a higher rate than traditional therapy.

MIT-Manus and InMotion ARM

InMotion ARM[33] device represents a clinical version of the *MIT-Manus*[34] robot. The clinicians can establish an efficient, personalized therapy for patients with neuromotor disabilities because the device is based on intelligent, interactive technology which is able to adapt itself to each patient's capacity. *InMotion* covers multiple rehabilitation solutions, such as:

a. For upper limbs: *InMotion ARM Therapy System*, *InMotion WRIST Interactive Therapy System* and *InMotion HAND*.
b. For lower limbs: *Anklebot InMotion ANKLE Exoskeletal Robot* and *InMotion Exoscheletal Arm Robot*.

The improvements obtained using the therapy robot were noticed using a controlled, randomized study [MLA97].

Armeo

This solution uses a VR scenario incorporated with a gravitational compensation system for the therapy for upper extremities with self-initiated and functional treatment [35]. The included exercises are provided into a game-like setup which helps the patients to improve their motor abilities and real-time performance through the augmented performance feedback. In a study [SJH09] that involved stroke patients with mild to

---

[32] https://www.medgadget.com/2014/10/musicglove-hand-rehabilitation-system-now-available-video.html

[33] http://bionikusa.com/healthcarereform/upper-extremity-rehabilitiation/inmotion2-arm/

[34] http://news.mit.edu/2000/manus-0607

[35] https://static.hocoma.com/wp-content/uploads/2016/09/bro_Armeo_160211_en_08_WEB.pdf?x82600

severe hemiparesis, the patients manifested their preference for this rehabilitation solution versus the classical therapy. The gravity-supported arm exercises can improve arm movement ability with a brief 1:1 assistance from a therapist (~ 4 minute per session). The improvements of rehabilitation based on this solution that contains a 3-dimensional weight support, instant visual movement feedback and simple VR software, were noticed even at the 6 months follow up.

Four distinct products are included in the *Armeo* therapy concept: *Armeo Power*, *Armeo Spring*, *Armeo Spring Pediatric* and *Armeo Boom* and each of them are specially conceived for a certain stage of the recovery process, with one exception the *Armeo Spring Pediatric* which is designed to cover children rehabilitation cases.

Bi-Manu-Track

Bi-Manu-Track[36] is a robotic device with 2 DOF designed for wrist and forearm region and works on the principle of bilateral training. According to [ECL11] the device permits a DOF for the pronation and supination of the forearm. When is utilized in vertical position the device permits a DOF for dorsiflexion/volar flexion of wrist. The device is connected to a visual display that shows the number of effectuated cycles and a computer that collects the data and controls the motors. *Bi-Manu-Track* can be used in 3 modes:

a. *Passive*, where the robot assists both upper limbs.
b. *Active-Passive*, where the movements are effectuated in mirror mode initiated by the less affected limb.
c. *Active-Active,* where both upper limbs initiate the movement.

Bi-Manu-Track impact was measured in a study [SH05] where it was found that the greater number of repetitions and the bilateral approach could have impacted positively the upper limb motor control and power compared with other techniques based on ES (Electrical Stimulation).

PowerGrip

The *PowerGrip*[37] device is an EPPO (Electric Powered Prehension Orthosis) that is helpful for picking up, grasping, holding and manipulating objects. *PowerGrip* uses switches or sEMG (surface electromyography) signals to control the input of the device [PM14]. In the newer versions are used myoelectric sensors that are placed on one or two functional muscles.

---

[36] http://www.reha-stim.de/cms/index.php?id=60

[37] http://www.broadenedhorizons.com/powergrip

This part of the literature review was in connection with TRAVEE project contribution. TRAVEE has similar characteristics with the mentioned solutions along with its original elements, such as:

a. *The usage of an interactive virtual environment*, as seen in the *MusicGlove* and *Armeo* case. However, in these cases the VR environment is displayed on a computer monitor instead of an HMD.
b. *The existence of personalized treatment* related with patient's health condition for an appropriate recovery plan.
c. *The possibility to effectuate a high number of repetitions* for a faster recovery.

3.1.2 3D Visualization Solutions

If in the previous section the focus was on the rehabilitation devices, now our attention will be on software solutions and more exactly the 3D visualization methods. Even tough, some of previously mentioned devices contained a VR module the information presented in this section is more fit to our research goals. The review contains information from two solutions considered representative for neuromotor rehabilitation. They are not functional only based on the rendering part as it is important to review the compatibility of the module, dependent tools and libraries.

Rehabilitation Simulator

A set of rehabilitation applications based on virtual reality and physical-haptic procedures are proposed by [LDLL14]. The applications have specific tasks to be performed by patients that suffered a stroke. The applications' solution contains a haptic device (Sensable Phantom) that generates force feedback which indicates the interaction between the patient and a virtual object. It is integrated in the system with Open Haptics API. The 3D visualization is implemented with OGRE (Object-Oriented Graphics Rendering Engine) as this software enabled the visualization of virtual elements in real-time at a high quality [AV15b]. NVIDIA PhysX, a physical engine, was incorporated complementary with the graphical engine for an accurate physical simulation. 3D modeling tools such as Blender and 3D Studio were used to create the 3D models used in this set of applications. They offer two categories of exercises that target the rehabilitation of upper limbs:

a. Cooking tasks - The displayed scene is a virtual kitchen with various related elements such as: a pan, potatoes, skaters, a table and a shelf. The scene complexity is low and based on the rendered models it can be assumed that there are approximately 10-15 simple 3D models [LDLL14]. The scene has a first-person perspective and the user can see the virtual model of one arm as this model is used to interact with other elements from the virtual kitchen simulating basic daily living tasks [AV15b].

b. *Motor activity of grasping a glass* – Another set of tasks that simulate usual daily activities such as drinking water. This scene is fairly simple containing the virtual models of a glass, a table, a coaster and the trained arm. Initially the objective is to grasp the glass, followed by the simulation of the activity of drinking water and as a final target the user should place the glass on top of a coaster model.

Personalized Rehabilitation Gaming System

A VR based system named Rehabilitation Gaming System (RGS) is proposed by [MSC10]. The attention is drawn on a certain rehabilitation scenario called *Spheroids*. Promising results were reported with a consistent transfer of movements kinematics between the physical and virtual tasks during trials that involved 21 acute/subacute stroke patients and 20 controls using a Personalized Training Module (PTM).

The virtual scene is simple and contains a green landscape populated with trees and with a mountain range background. Along with these elements a virtual model of a human torso and arms are added in a first-person perspective. A motion tracking system maps the user's physical movements into the virtual reality scenarios. The user's task is to intercept the spheres that move toward the user and with each successful interception the user obtains a number of points. The difficulty is set by different parameters such as: speed, spheres' appearance interval and the horizontal range of dispersion in the FOV (Field of View). RGS implements training protocols for neurorehabilitation that allows a gradual and individualized treatment of deficits of the upper extremities after stroke [AV15b].

Both solutions have similarities with our contributions and one that is worth mentioning is the fact that on both projects a rendering engine was used to display the virtual scenes. Nowadays the well-known rendering engines (e.g. Unity, Unreal) are integrated into a complete game system and permit to integrate the functionality of additional components, such as motion tracking. We opted for Unity game system while the other solution uses OGRE rendering engine. The available rendering solutions were reviewed multiple times during this research and in each case, it was decided to continue the usage of Unity. The main advantage was the extended support of various technologies for motion tracking and display for VR and AR. The other reviewed solutions were Unreal Engine and CryEngine and we considered that Unity had a wider support for dependent technologies compared with the other two.

3.2 CONTRIBUTIONS

In this section are detailed the author's contributions at TRAVEE project. At the time of the implementation the project was still in early stages of development and unfortunately no performance or user feedback data is provided within this chapter.

3.2.1 Avatar Personalization

The role of this module was to offer to the user the possibility of making a personalized PVM (Patient Virtual Model). The patient will be able to see himself in the virtual world, similar with the mirror therapy[38], and the similarities of the virtual character with the user should make him or her more accustomed with the simulated environment feeling present and immersed in it.

The patient virtual models used in TRAVEE were obtained using *Make Human*[39] software. This open source solution is enabling the possibility of generating human models based on different characteristics such as age, color skin, hair type or weight. Although this solution is great to be used for generating the models we needed to find a solution to personalize the avatar and save the changes within TRAVEE workflow. The solution offered here was the initial prototype.

This module functionality was to set a personalized avatar of the patient and to communicate with other TRAVEE modules. The figure bellow is displaying the workflow.

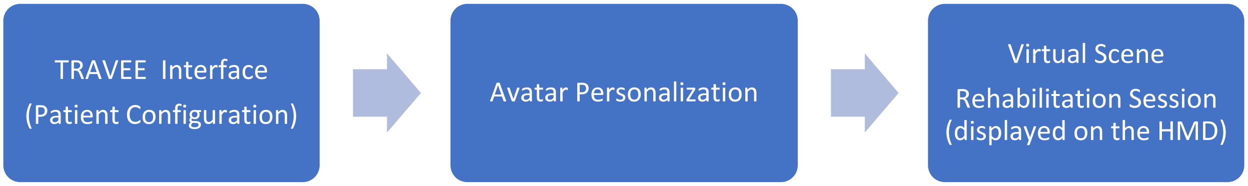


Figure 3.1 – TRAVEE workflow that includes the Avatar Personalization

The *Patient Configuration* step is where the therapist is adding details about the health condition of the patient, the status of the rehabilitation and the body characteristics. The avatar personalization is an intermediary step of the session setup and based on the set parameters the user will see a 3D model appropriate with its body configuration. With this data, the patient will enter in the virtual session where he or she will wear an HMD and will see the personalized exercises as set up by the therapist, depending on the health condition or rehabilitation progress. The patients should be able to see the model set in the *Avatar Personalization* step in the virtual reality environment.

---

[38] http://www.physio-pedia.com/Mirror_Therapy
[39] http://www.makehuman.org/

The module was developed with Unity engine and the results are exported to WebGL so that it could have been easily connected with the interface application that was a web-based solution. As input data, we have the following information about the patient: *age*, *sex*, *height*, *weight*, *skin color* and *hair color*. The first 4 values should be obtained from the TRAVEE Interface (http://app-travee.osf-demo.com/). At the time of this implementation (2015), the connection between TRAVE Interface and the Avatar Personalization module wasn't ready, and the details of interest were added into the *Avatar Personalization* module to demonstrate its capabilities. The values required as input data are: *age*, *sex*, *height* and *weight*. They are included as independent dropdown controls (Figure 3.2 – left side). The other two values: *skin* and *hair color* are set via slider controls (Figure 3.2 – right side).

Female Sex

50-60 Age

181-190 Height

81-90 Weight

Skin Color

Hair Color

Figure 3.2 – Avatar Personalization Interface

Output data consists in one file named *SavePrefs* which is located in the project's root folder. This setup is related with Unity's division in multiple projects, to avoid conflict at data submission on SVN (version control solution). Another implementation option is to utilize the Unity's *PlayerPrefs* feature, where it can directly be saved on user's settings[40].

---

[40] http://docs.unity3d.com/ScriptReference/PlayerPrefs.html

There were 60 models used for this solution. The models were obtained from *Make Human* application. Although, it would have been more suited to obtain only a small range of models and create in this module the functionality for their customization for various body types, a faster solution was used to obtain personalized avatars of the patients.

To obtain the 3D models various settings from Make Human were used to obtain a large variation of body types. The avatars were generated based on different age, sex, weight and height. All the models were selected to have Caucasian body type, since TRAVEE was aimed to offer a cost-effective rehabilitation solution for Romanian patients. The utilization of other body type, such as Asian or African, could have been implemented in a future update. The first clear differentiation was made between women and men (3D models for 30 women and 30 men). For age, height and weight were used values intervals because for closer values the differences between models were imperceptible. The available values for these 3 categories are available in Table 3.1.

Table 3 1 – Age, Height, Weight values intervals for the selected models

AGE [YEARS]		HEIGHT [CM]	WEIGHT [KG]
VALUES	CODIFICATION		
30-40	0	151-160	41-50
40-50	1	161-170	51-60
50-60	2	171-180	61-70
60-70	3	181-190	71-80
70-80	4	191-200	81-90
		201-210	91-100
			101-110
			111-120
			121-130

A selection formula was implemented to be able to choose the appropriate body type. It was noticed that when modifying the height (in Make Human application), the model's changes consisted in a simple scale operation on all axes which will be imperceptible in the virtual scene, as this can be affected by the distance of the model from the camera and its FOV. For this reason, the following solution was implemented: for each values interval (height and weight) the BMI (Body Mass Index) was calculated. The obtained values ranged from 10 till 43 (with approximation). These values were divided in 5 categories: XS [0], S [1], M [2], L [3] and XL [4] (XS – extra slim, S - slim, M – medium, L – large, XL – extra-large). In Figure 3.3 is available a preview of the selected categories and the impact of *Muscles* mass controller, as it affects the physical appearance, as well. In Figure 3.4 and 3.5 are available the default model's structure for minimum and maximum values of the *Height* controller.

Category 0 - XS

Category 1 - S

Category 2 - M

Category 3 - L

Category 4 – XL

Figure 3.3 – Textured 3D Models variation for the 5 body types categories: XS [0], S [1], M [2], L [3], XL [4].

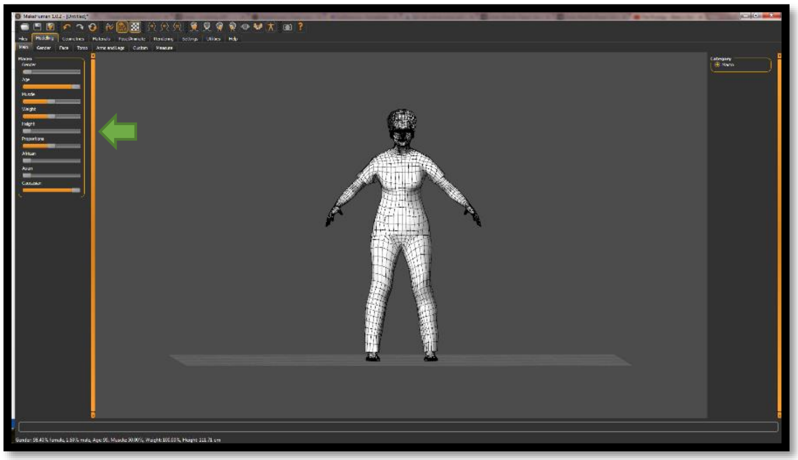


Figure 3.4 – Non-textured 3D Model for Height **minimum** value.

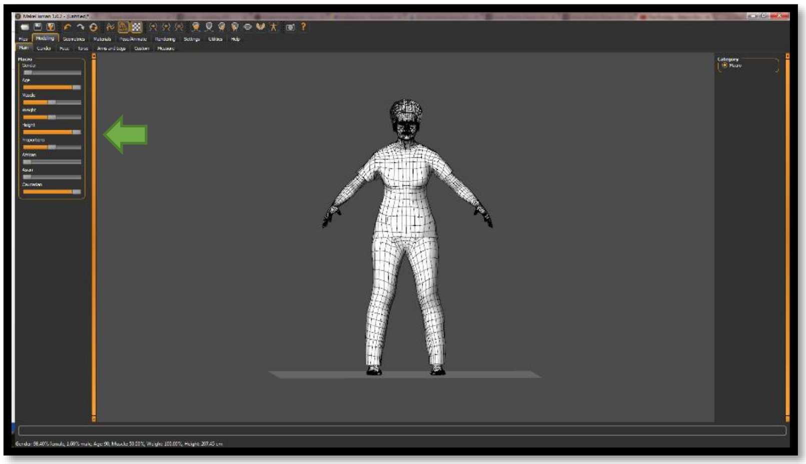


Figure 3.5 – Non-textured 3D Model for Height **maximum** value.

To implement this approach as facile as possible, a codification was assigned on each category. If the virtual model is for a woman, it will receive value 0 and 1 if it is for a man. After this, the age codification is added, as it can be noticed in the second column from Table 3.1. The third element is the category index based on BMI calculation as it can be noticed in Figure 3.3. Following these rules, the names of the 3D models' files are in this format: [0/1]- [0-5]- [0-4].

Storage requirements

The disk space necessary for this approach is 3GB, where 200MB represent the build files and 50 MB the release files (the ones that should be uploaded on the server). All 3D models occupy 1.2 GB, and the rest of data up to 3GB represent the generated temporary files.

Implementation Details

a. SKIN COLOR PERSONALIZATION

To implement the color skin selection and visualization at runtime, an image that contained a variation of different skin shades was obtained. This image is noticeable in the Fig. 3.2, upper right corner, under the skin color slider. To determine the RGB values for these shades, the image was analyzed with a Color Code tool[41]. The obtained values were in hexadecimal and they were transformed to decimal after that, ranging in the [0, 255] interval. Each color channel (Red/Green/Blue) transmitted to the shader should have been a floating-point number in the [0,1] interval. For debugging purposes, the final values were obtained by dividing the previous values to 255 (with corresponding conversions).

b. HAIR COLOR PERSONALIZATION

The hair color personalization has a similar implementation with the skin color. The main difference is related with the fact that the default models that were exported from *Make Human* were changed to have a white color texture for the hair, to be able to properly apply the color changes at runtime. The models were initially exported with a black hair texture and all the changes applied on the models were not visible since the RGB values for black are (0, 0, 0) and any multiplication with another color would have had the same result.

3.2.2 Virtual Reality Display

Oculus Rift was used for the visualization system that displayed the rehabilitation scene in TRAVEE project. As it was already mentioned in the first part of the thesis, TRAVEE aimed to improve the neuromotor rehabilitation process of stroke survivors. The patient uses an HMD, Oculus Rift in this case, to visualize the rehabilitation exercises made by a virtual therapist. The usage of an HMD has the benefits that the patient can see the rehabilitation sessions even when is lied in the hospital bed permitting her/him to start the recovery very early. Fig. 3.6 showcases the system's setup that displays the rehabilitation virtual scene.

Figure 3.6 – TRAVEE VR system setup

Oculus device is responsible only with the visualization and it must be connected all the time to a computer to work as all the processing is made on the workstation. To

---

[41] http://html-color-codes.info/colors-from-image/

communicate with it, the device has an USB and a HDMI port. Recently (late 2017) Facebook announced the release of *Oculus Go*, a device with its own processing source which can be used without cables or connected to a workstation in order to run.

The TRAVEE virtual module was developed using development kits version 1 and 2 (DK1 & DK2) as the retail version wasn't available at that time. The graphical engine used for developing the VR module was Unity. During the implementation on Oculus Rift DK1, Unity *Version 5.2.2f1 Personal Edition* was used. It was developed on a workstation that had Microsoft Windows 7 OS and it was compatible with *Oculus SDK*. Using Unity, any 3D scene could be easily ported to *Oculus* and at that moment (early 2015) we used the package *OculusUnityIntegration* to achieve this. This was a simple integration since the OVR (Oculus VR) resources and dependent plugins were imported into this project ready to be used. The most important resource, *OVRPlayerController*, was localized in *Assets/OVR/Prefab* folder and it was responsible with enabling stereoscopic visualization mode and the input detection (this was registered based on the user's head movements). After the Visual Studio solution was built, 2 executable files were generated: one for testing in Windows and another one to be deployed to run on the Oculus Rift Development Kit.

Figure 3.7 shows an image obtained through the emulation of the virtual scene on the workstation. The patient is using an Oculus Rift device to see the PVM (Patient Virtual Model) and TVM (Therapist Virtual Model) into a virtual scene. The patient should follow the example of the virtual therapist in executing the rehabilitation exercises. There were two approaches that were tested at that moment:

a. The TVM is seen in a window in the upper left corner of the scene (Fig. 3.7 and 3.8A). The rendering techniques used in that case was "render to texture" and the displayed element was 2D;
b. The TVM is seen in the same area with the patients' virtual model (Fig. 3.8B).

After both approaches were tested we realized that the first approach wasn't a good fit although it was successful applied on classical visualization systems (e.g. streamers channels). As we found out meanwhile in VR the rendered elements should be blended in the 3D scene instead of the classic 2D UI interface.

These tests were completed at the start of TRAVEE project (late 2014 – early 2015). In the newer versions of Unity there is no need for additional third-party packages and the support for Oculus Rift is already integrated.

HEALTH & SAFETY WARNING

NO HMD DETECTED

Read and follow all warnings and instructions included with the Headset before use. Headset should be calibrated for each user. Not for use by children under 13. Stop use if you experience any discomfort or health reactions.

More: www.oculus.com/warnings

Press any key to acknowledge

HEALTH & SAFETY WARNING

NO HMD DETECTED

Read and follow all warnings and instructions included with the Headset before use. Headset should be calibrated for each user. Not for use by children under 13. Stop use if you experience any discomfort or health reactions.

More: www.oculus.com/warnings

Press any key to acknowledge

Figure 3.7 – Scene example on Oculus Rift

Figure 3.8 – Examples of a TVM and PVM scene configuration.

3.2.3 Motion Tracking

Overview

For TRAVEE project we needed to track the movements of a patient during the rehabilitation sessions to be able to display them augmented in the virtual environment and to measure the level of completion for the targeted exercises as set by the therapists. This was achieved with a Kinect sensor to track the body's skeleton and a Leap Motion device to track the hands and fingers movements. The fingers movements weren't properly tracked by the Kinect DK1 sensor and Leap Motion was a low-cost and easy to use solution. This sensor is compatible with the usage in VR as it had a special case that could have been mounted on an HMD to have a better position to track the hands and to display their movements in the virtual environment.

Results

The author's contribution was part of the kinematic module of the TRAVEE system prototype. This prototype offered the possibility to set a few simple exercises for training the upper limbs (shoulders, arms and hands). The initial focus was on the upper part of the body because the impairment affects the most the quality of life of a stroke survivor and the rehabilitation will have benefits visible in daily activities [AV15c]. The system had two types of users: the therapist and the patient. Each of them had a different approach for the motion tracking usage:

1. *Therapist* needed to record the exercises offline, to save the data, to allocate to each exercise a unique ID and to use them in a rehabilitation session as a resource for the virtual therapist movements.
2. *Patient's* movements were tracked in real-time to be displayed in the virtual environment. They must execute the exercises similar with the ones previously recorded (therapist movements). This data is also saved to be later analyzed by the

specialized personnel for observations regarding the progress and execution quality.

During the author's contribution to the project, the therapist and the patient were executing movements in the virtual environment based on a single motion tracking sensor and the avatars had the same animations at runtime (e.g. Fig. 3.12).

The movements are replicated in the virtual environment using 3D models (patient and therapist). The patient's virtual model was obtained using *MakeHuman* v1.0.2 and the therapist virtual model was acquisitioned for a small sum from *TurboSquid*. The models were animated at runtime based on the 3D coordinates of the tracked joints. For motion tracking is important to be a correlation between the 3D model's skeleton joints and the joints tracked by the sensor. For example, there are some optimized skeletons versions (that were available in *Make Human*), but they didn't have the necessary number of bones for the hands. Other skeletons that had a higher number of bones were redundant for the developed scenario as we couldn't properly track them, and that could potentially affect the runtime performance of the application as well. For the mentioned reasons, *basic.json* skeleton was chosen as it was the best fit for the developed project's purposes. This *basic* skeleton had a total of 73 bones, where the most optimized one had 19 bones and the most detailed one had 105 bones.

The development environment was Unity v5.1.2f1 game engine. It had support for virtual reality module and it had compatible third-party packages for motion tracking. Since the target for this prototype was the upper limbs rehabilitation, the system was composed from two sensors for motion tracking: Leap Motion Controller and Kinect V1 sensor. Leap Motion was used to track the hand movements, while Kinect was used for the rest of the body. Even though the rehabilitation exercises were focused on the upper limbs, the motion tacking was acquired for the whole body to be displayed realistically in the virtual environment. Fig. 3.9 displays the skeleton bones of the used 3D models and their cover area from each sensor for runtime animation. *LeapMotion* and *LeapAvatarHands* packages from Unity were used to add functionality for hands movements tracking in TRAVEE project. A new component named *LeapController* had an *IKLeapController* script attached to it. Using this script, the arms of the human 3D model were animated via *Inverse Kinematics* [AV15a]. Fig. 3.10 displays which bones from the virtual hand are involved in the animation process while using a Leap Motion device.

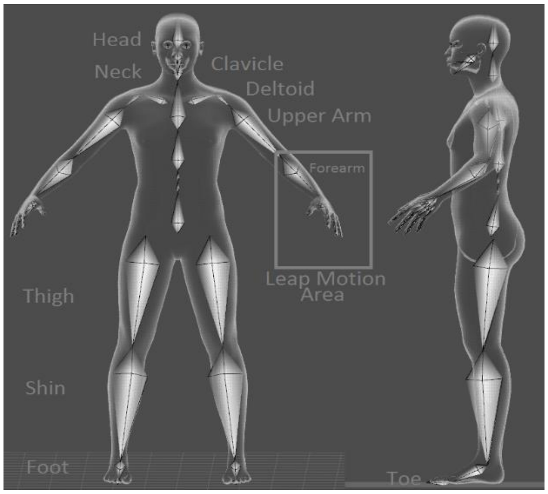


Figure 3.9 – Kinect and Leap Motion cover areas

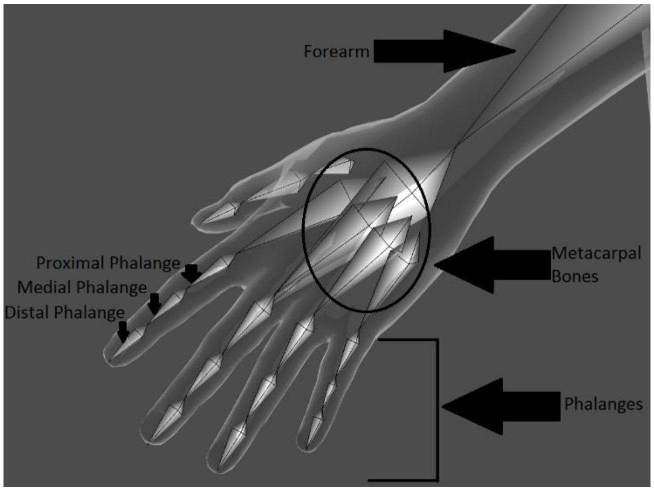


Figure 3.10 – Hand bones of the 3D model

Since the human body is symmetrical, the bones from each part had the naming convention **<*boneName*>_L** and **<*boneName*>_R** for left and right side of the body. To have functionality we attached to left and right hand the *RiggedHand* script that was available in the *LeapMotion* package. After that, references were added to each finger, palm and forearm. Also, there were available wrist and elbow joints, but they were optional as the focus was on the finger bones and in Table 3.2 it is available the correlation between the model's bones names and their identifiers in the *RiggedHand* script. To each of the bones displayed in the Table 3.2 was attached a *RiggedFinger* script available as well in *LeapMotion* package. As mentioned before, there are 4 metacarpal bones for each finger with one exception for the thumb as this is applied for each finger bones indexes, where the thumb has a bone in minus.

Table 3.2 – Naming correspondence for the RiggedHand script

3D Model Bones	**RiggedHand script identifier**
Thumb_01_L	Element 0
Palm_Index_L	Element 1
Palm_Middle_L	Element 2
Palm_Ring_L	Element 3
Palm_Pinky_L	Element 4

The prototype had the option to record the movements and, to obtain this, *RecordingControls* script was attached to the *LeapController* component. This feature was enabled to be able to prerecord the rehabilitation exercises as set by the therapist and to offer the possibility for the specialized personnel to verify the quality of the movements after the session was finished.

Fig. 3.11 displays a list with a few basic hand movements and their visualization in the virtual environment, as they were tracked using a Leap Motion Controller.

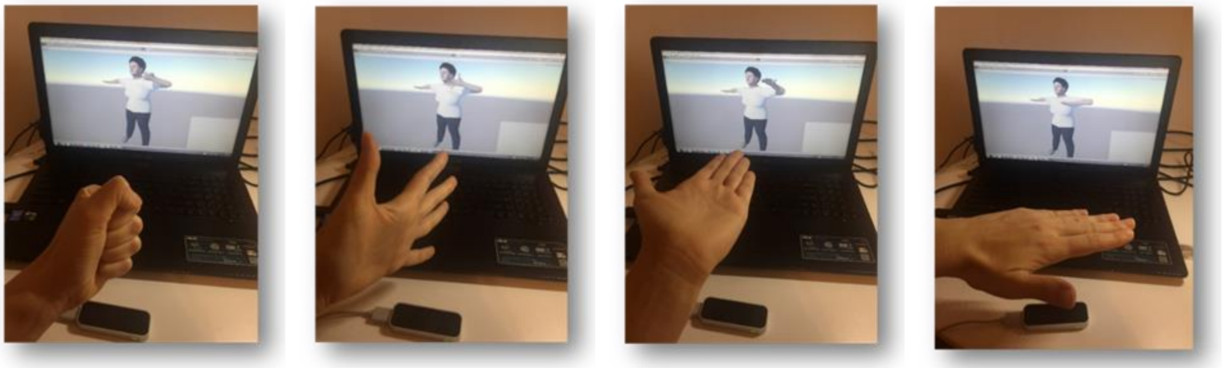


Figure 3.11 – Basic hand movements tracked in real-time with Leap Motion Controller

Besides a Leap Motion device, the kinematics module from TRAVEE project has integrated the functionality for motion tracking for the whole body using a Kinect sensor. At that moment we used the *Kinect with MS-SDK* unity package for development. Similar with Oculus Rift, we needed to install the development libraries (*Kinect SDK* and *Kinect Runtime*) on the workstation. After making these changes, an *AvatarControllerClassic* script was attached to the root element of imported 3D model in Unity. Each element of the script, which represents the joints tracked by Kinect, will have a reference to an element from the model. Since the model's skeleton has 73 bones, and Kinect V1 is tracking only 20 joints, it is obvious that some of them won't have a correspondence into the script. Also, it is possible not to track certain body parts with Kinect sensor (e.g. the LeapMotion cover area) by not selecting a reference from the 3D model for the involved joints. To each tracked joint was attached *GetJointPositionDemo* script from the imported package. Fig. 3.12 displays a few basic movements tracked with Kinect sensor and their visualization in the virtual environment for the patient and therapist virtual models. A red square area can be noticed on the patient virtual model shoulder area that looks odd into the scene. This is related with the fact that the models generated using *MakeHuman* program are in A-pose (e.g. Fig. 3.9) and when using a Kinect sensor with the correspondent unity package the models need to be in T-pose. This can be changed in 3D modelling software solution such as *3DS Max* or *Blender*. This issue is not present on the therapist model that was already in T-pose.

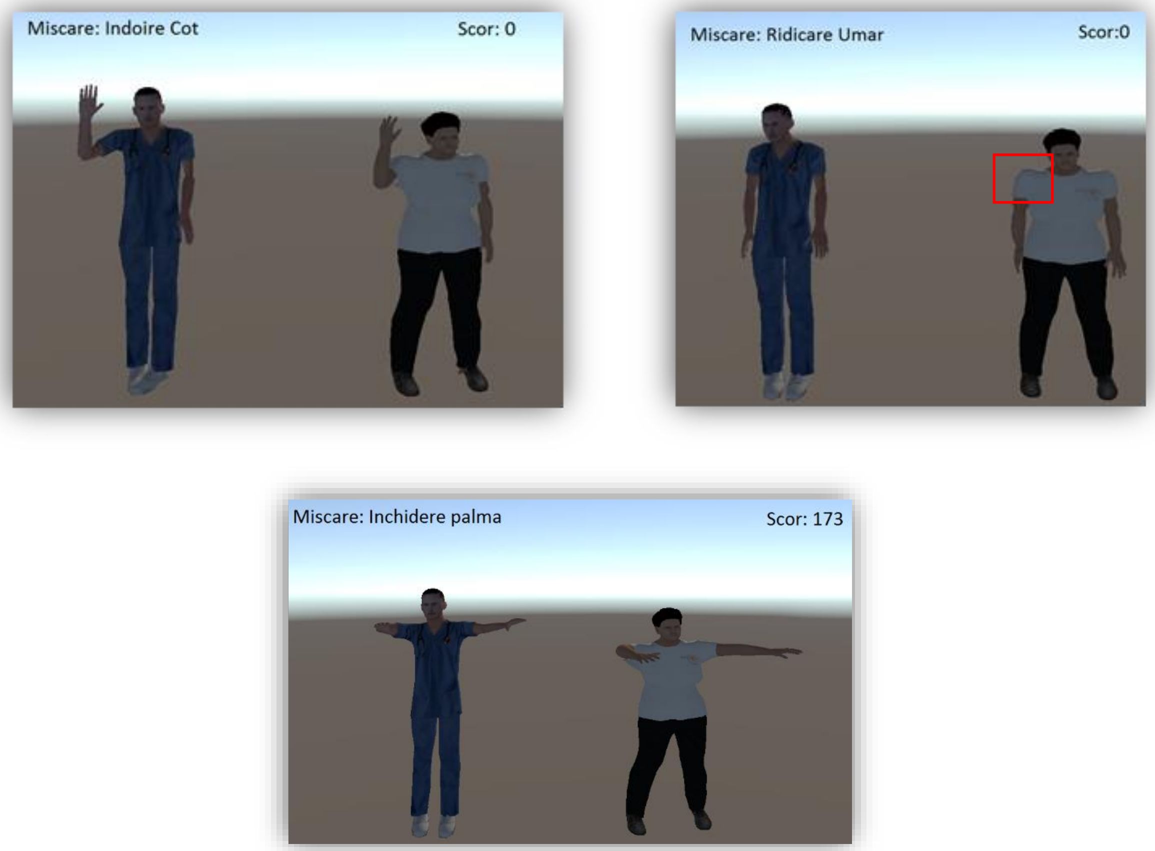


Figure 3.12 – Patient and therapist models animated based on Kinect sensor body tracking

3.3 CONCLUSIONS

This chapter contains the main contributions of the author at the TRAVEE project that aimed to be an effective IT low-cost rehabilitation solution. TRAVEE was based on complex technologies such as VR, BCI, FES and robotics and was a competition for other existing rehabilitation devices since it targeted early recovery and personalized rehabilitation features. For the visualization part we opted to use existing rendering engines and tools that proved they efficiency in many other cases. We could observe in the second part of the 4.1 section that this practice was encountered in other cases, as well.

The contributions were on 3 major areas: avatar personalization, VR module setup and initial display settings and motion tracking. The avatar personalization module was rudimentary but enabled the possibility to make changes according with the patient’s physiognomy at least at a basic level. TRAVEE’s VR module that was detailed in section 3.2.2 created the base for the backend part that later incorporated the motion tracking features. The third-party packages that were used in Unity had some tracking errors, but they were a real help into developing faster these features.

These contributions can be considered an intermediary step for the research as the next project (Interactive Biomechanics Lessons) used common elements such as VR and motion tracking to provide a novel IT educational solution for biomechanics study.

CHAPTER 4

AUGMENTED AND VIRTUAL REALITY IN MEDICAL EDUCATION

This chapter contains the second part of the research and the most extensive one. Our aim was to continue the research of VR and AR usability in Healthcare. We considered an opportunity to develop applications that contain a few interactive biomechanics lessons. The related work section contains a brief literature review for existing AR and VR solutions used in Healthcare. It was considered an opportunity to focus on a novel approach that aimed the usage of interactive technologies in medical education.

4.1 RELATED WORK

This subchapter contains an overview of the areas of interest regarding the usage of virtual and augmented reality in healthcare. The research's goal was to comprehend the state of the art and was completed in early 2016. The target was to focus especially on applications and scientific papers published after 01.01.2014 to have a clear picture of the newest technologies and approaches. We are aware that a few additional solutions and research topics that could be of interest might have been published meanwhile. In any case, the ideas of our own approach were tested through audience feedback during presentations at various conferences. The Interactive Biomechanics Lessons project relates to the data presented in this section and its idea was built based on the information extracted from the literature review and the available solutions.

The initial batch of reviewed scientific papers contained an extensive period to make sure we weren't missing interesting topics that might be implemented differently or better with the current technology as opposed with what was available a few years ago, although the aim was to look mainly at newer research topics. We gathered a total of 77 scientific papers, where 47 are published starting with 2014 and the other 30 were published in the between 1999-2013. More details are available in Figure 4.1 regarding the topics related with this literature review.

From the categories and subcategories mentioned in the Figure 4.1, we considered to detail the ones that contained the most important information for our topic. Also, a part of the data extracted from the *General/Overview* category and *Cybersickness* is mentioned in the second chapter as they include some general insights regarding VR and AR. The ones excluded were considered unfit for our research topic. However, from all the papers reviewed there was overlapping information between them. Note that the rehabilitation related solutions were excluded from the search.

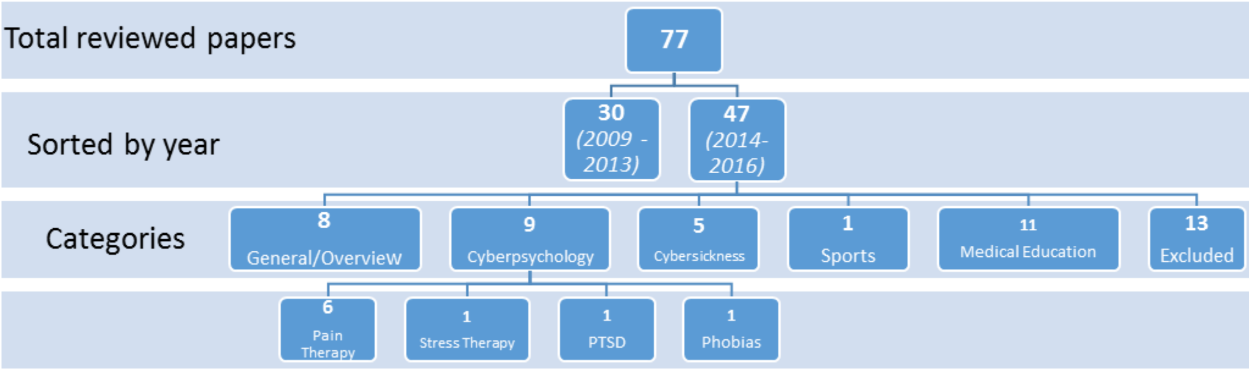


Figure 4.1 – Schematic view of the reviewed scientific papers topics

A fair number of interesting subjects was available in the old references as well. The usage of VR and AR was effective in various areas, such as: smokers quit therapy [BG09], pain therapy [CB13] [ES03], acrophobia [YC01], fear of spiders (arachnophobia)[DM10], public speaking [JL02], urology [RK01], anxiety disorders [AG08] and left hemineglect therapy [RM00], etc.

From the total of 77 studied papers 34 contained relevant information where the principal category of interest is medical education and most of the papers included data about medical imagistics. The second topic of interest, based on the number of found scientific papers, is cyberpsychology where most of the data was related with pain therapy. Below are detailed the studied solutions from the medical education area. They are divided by the technological system used for display and VR is the first one assessed.

The first VR based solution [GS15] is a surgical simulator (Gen2-VR) developed to train the surgeons in skills laboratories. The aim of the research is the assessment of the impact of a realistic simulator and the influence of the disturbance factors. Three scenarios were tested: Case I: "*user interacts with a simulation scenario presented on a computer monitor*", targeted as traditional VR, Case II: "*the user is interacting with the simulation scenario within a HMD, but without distractions and interruptions* and Case III: "*the user interacts with the simulation scenario within a HMD with distractions and interruptions*". The last 2 cases are using the Gen2-VR solution. The next solution [KK14] contains a stereoscopic viewer of the results obtained from vessel segmentation based on 3D magnetic resonance angiography images. The results are 3D models of the vessels as extracted from the medical images. The solution is using Unity game system for development, Leap Motion Controller for tracking the hands movements of the user and Oculus Rift to display the virtual environment. Another paper focused on VR [YL14] targets image guided deep brain simulation neurosurgery to treat patients suffering from neurological disorders such as Parkinson disease, essential tremor and dystonia. In this paper are considered potential applications of VR based technologies for deep brain stimulation with brain magnetic resonance imagining data. The last paper [SL15] mentions the usage of 3D technologies and stereoscopic visualization in medical endoscopic teleoperation.

The AR related papers are in majority related with medical imagistics. Reference [CK14] contains three examples of AR solutions used in medical training:

a. *Visualizing human anatomical structure*. The mentioned solution augments CT data onto the body of a user. The project is named *magic mirror "Miracle"* and displays the training sessions on a TV while the user movements are tracked using a Kinect device.
b. *Visualizing human 3D lung dynamics*. The solution is based on a system that allows real-time visualization of the lung dynamics superimposed on a patient in the OR (Operating Room).
c. *Laparoscopy skills training*. The laparoscopy environments based on augmented reality offer realistic haptic feedback crucial for the development of the necessary skills.

A video see-through solution [FC14] uses the AR HMD in the medical procedures, such as: maxillofacial surgical clinical study, orthopedic surgical clinical study and AR magnetic guidance of endovascular device. Another system that contains the usage of a HMD [SY14] uses a vison finger tracking technique applied to medical education. Reference [ZY15] contains a review of AR in OR applications, more precisely the augmented visualization in surgical navigation. Based on the medical imagistics modality, 4 situations of augmented guidance were approached:

a. Augmented X-ray guidance.
b. Augmented ultrasound guidance.
c. Augmented video and SPECT (Single-photon emission computer tomography) guidance.
d. Augmented endoscopic video guidance.

The last 3 scientific papers [MCFM14a] [MCFM14b] [MCFM15] refer to different areas of development for an AR based solution for in-situ visualization of the craniofacial region. The augmentation is done using medical images such as MRI (Magnetic Resonance Imagistics) or CT (Computed Tomography). Regarding the augmentation tracking, initially a semi-automatic markerless augmented reality approach was considered [MCFM14a] and later a markerless AR environment was implemented [MCFM15]. The markerless live tracking was completed based on the registration between a 3D reference model and the 3D model captured with the tracking sensor.

Besides the initial assessment, during our research we discovered another application that served as example. Reference [BP14] contains details regarding a solution dedicated for biomechanics study by visualizing the lower limbs muscles activity in real time. The presented material targets a small set of lower limb movements such as knee flexion and extension. It showcases in real-time the muscles activations on a 3D avatar based on the movements of an observed user. The muscles activation data was acquired separately (preliminary) using a *Biopac MP150* device with non-invasive EMG and was added into a database while assigning the tracked movements type. This operation was effectuated separately to improve the runtime performance by minimizing the required computations. The movement of the observed user was registered using a Kinect sensor, *NiTE* and *OpenNi SDK*[42]. Fifteen joints are tracked by this system while the 3D models of the lower limbs are imported from *MyCorporisFabrica*[43]. C++ was used for framework development and the main application runs on a Laptop PC while the feedback is provided on the computer's screen. All the tests were completed in the same room to preserve similar lighting conditions. As mentioned by the authors, the AR visual feedback needed improvements as it shown separately the animated muscles models.

If we look at the resources available for learning, we can observe that they are diverse. The basic selection criteria, when it comes to educational applications, is based on the quality of the information, the setup difficulty and the cost. Based on the scientific papers studied we considered an opportunity to develop a solution that targets biomechanics study of the human movements. It has similarities with the previous topic (TRAVEE project) hence the thesis could maintain the same research note. The project is complex, so practices found in the related literature were applied during its development. For example, a part of the 3D models used in the applications are obtained from medical images. Two approaches for augmented reality display were considered and 4 different scenarios were developed for both AR and VR [AV18].

4.2 CONTRIBUTIONS

In this subchapter we present the results of our research. After making an in-depth analysis of the existing solutions based on virtual and augmented reality in healthcare we observed the opportunity of developing a solution that targets medical education and more exactly to improve the learning process of biomechanics study. We want to demonstrate that the usage of VR and AR in medical education can be a plus. We consider that today's technology can aid the educational process as we have the opportunity to unlock a new visualization method that can add details on the fly to the observed environment.

---

[42] NiTE and Open NI online page: http://openni.ru/files/nite/

[43] MyCorporisFabrica: http://www.mycorporisfabrica.org/.

4.2.1 Approach

Interactive Biomechanics Lessons aimed to provide a novel solution for biomechanics study. The initial idea was to develop an AR application that superimposed a 3D model of the human anatomy over an observed user's image. The 3D model would be animated in real time based on the registered movements of the tracked user. Also, it aimed to contain the soft tissue deformation to showcase the muscles deformation during a certain movement. Figure 4.2 displays the application's system overview of the initial concept. The proposed solution had several design changes based on the results obtained on various technologies and test scenarios that are documented in the *Tests* section.

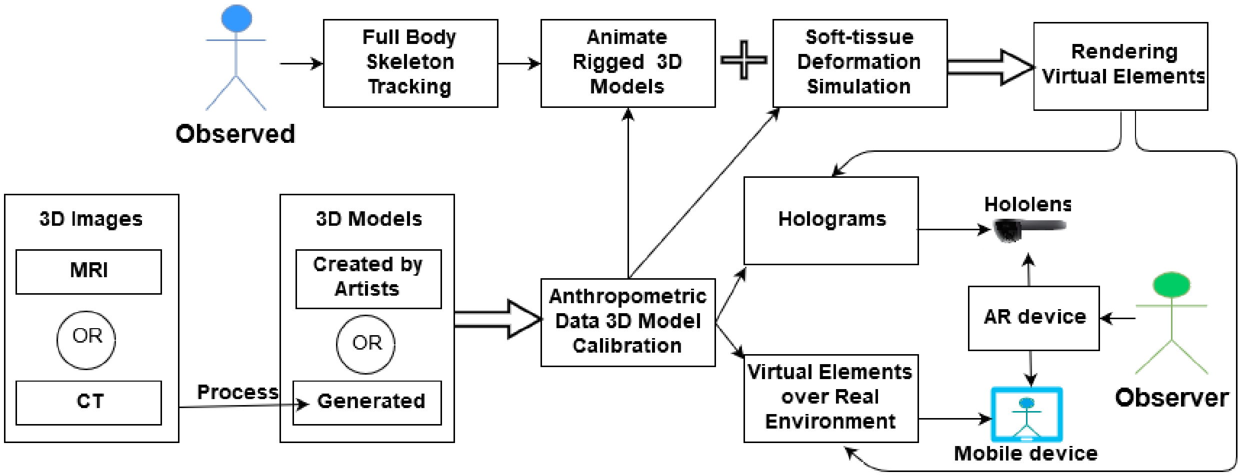


Figure 4.2 – Initial proposed system overview – AR based

The observer should be able to see as an overlay the virtual models that are animated in real-time based on the observed user's movements as the system was designed to include motion tracking. The display device that seemed the best fit for this approach was HoloLens. However, due to its price and shipping region (was available only for Canada and US in 2016) we had to consider alternative options. As mentioned in the second chapter there are multiple types of devices that can display AR content as they have to respect 3 rules: to combine real with virtual, to render in real-time and to be registered in 3D space. Mobile devices or a PC with a camera attached can be a viable alternative solution to display AR.

As initially designed, the system had two users: an observed user, that had its whole-body motion tracked, and an observer that used the AR display device and could see the combination of real environment and virtual elements based on the tracked movements and obtained models. The observer would have been able to see the position changes of the bones and, on top of that, as an extra information, the deformation of the soft tissues (bones are rigid). Since the work volume for implementing this system was a consistent one for the available development capacity, the simulation of soft-tissue wasn't covered by this research. This is a nice to have feature for this project as the main focus was on developing an educational solution using augmented reality and firstly had to obtain a positive result for the proof of concept of the base idea.

This application was divided in subsystems to modularize the necessary work and to be able to use parts of it in various combinations (Fig. 4.3). The development was divided in:

a. Display – that covered the display method and the targeted devices. More details are offered in the 4.2.3.1 section that covers the visualization during multiple tests effectuated with different AR and VR technologies.
b. 3D models – that aimed to obtain realistic rigged models of muscles and bones. In the section 4.2.2 are presented the methods we applied to obtain the 3D models used in the developed applications.
c. Movement – which targeted the real-time motion tracking. Similar with the display part, more details are offered in the *Test* section (4.2.3.2).
d. 3D models' animations to animate the rigged 3D models based on a given input.
e. Final Rendering – that combined the real environment with virtual elements.

The last two items are detailed per application and more information is available in section 4.2.5.

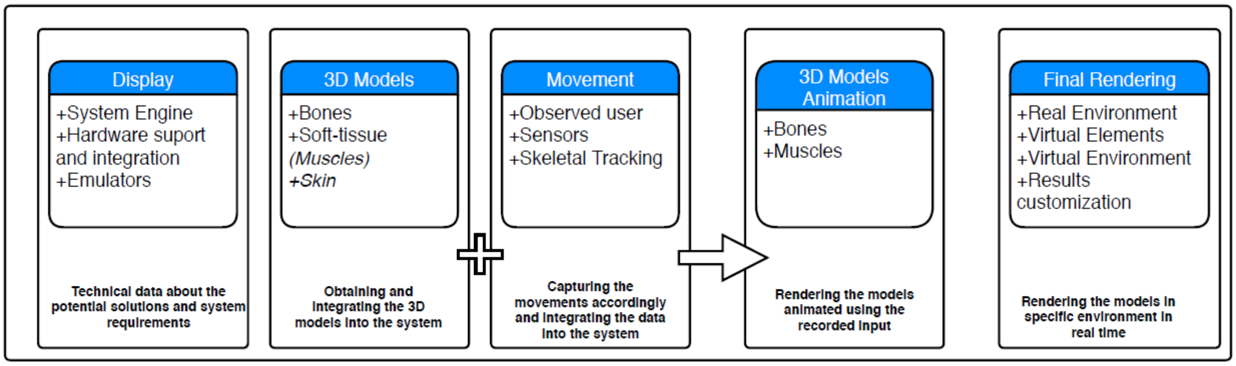


Figure 4.3 – Proposed application divided in subsystems

The final prototype of the implemented applications is not based exactly on the initial setup as additional features, that were considered more appropriate, were added while others that seemed to be unfit at that moment were dropped. Even though the initial ideas seemed a good fit, after having a part of the implementation tested, we observed a few characteristics that seemed not to bring the expected benefits to the final product or others that seemed to be a new opportunity. For example, taking into account the large support offered by the current game systems for VR we considered it would be a great opportunity to develop an interactive solution that uses both technologies (AR and VR) to learn biomechanics notions as we could test the users' responsiveness or preferences for one of them. Even though they have different properties, versatile game systems such as Unity are providing flexibility into developing these types of applications. Another strong motivation behind this addition is the fact that we could test the benefits of minimizing the external factors impact such as external noises to obtain better results in a shorter period.

During the implementation we realized that the application testing is highly impacted by the fact that the development team consisted of only one person as the system was designed to include two actors: the observer and the observed person. We considered two actors for the initial system because we believed that an interactive system will have more benefits and will be easier to be adopted by the users. In fact, our assumptions are strengthened by a newly released app that uses AR and 2 users to learn biology[44]. However, the presented solution is less complex compared with our initial proposal as it uses a marker-based AR on a T-shirt and without skeletal tracking.

The motion tracking feature was reanalyzed after testing the implemented solution (markerless AR). Since the tracking was obtained using a Kinect V1 sensor that had to be connected with wires to a workstation, the data should have been transmitted over the network to the mobile display device. The realism of the displayed movements would be affected by the synchronization issues, tracking errors, or affected by the applications' performance at runtime (the performance section contains more data regarding the obtained results in various scenarios of the implemented solution). At least for VR, these types of issues would seriously affect the user's immersion into the simulated environment. To ensure that the efficiency of our solution is not impacted by this module alone, we considered having a set of predefined biomechanics lessons that will be presented in an interactive manner. The introductory biomechanics notions that were chosen were part of the course with the same name from the Medical Engineering master program from University POLITEHNICA of Bucharest.

The presented solution is developed for AR and VR environments as each setup contains specific approaches. The preregistered lessons are used in the VR and AR marker-based setup, while the motion tracking feature is part of the markerless AR based scenario. The implementation details of each scenario are presented into 4.2.5 section and Fig. 4.4 showcases an overview of the whole solution that targets the usage of VR and AR.

---

[44] https://www.kickstarter.com/projects/curiscope/virtualitee

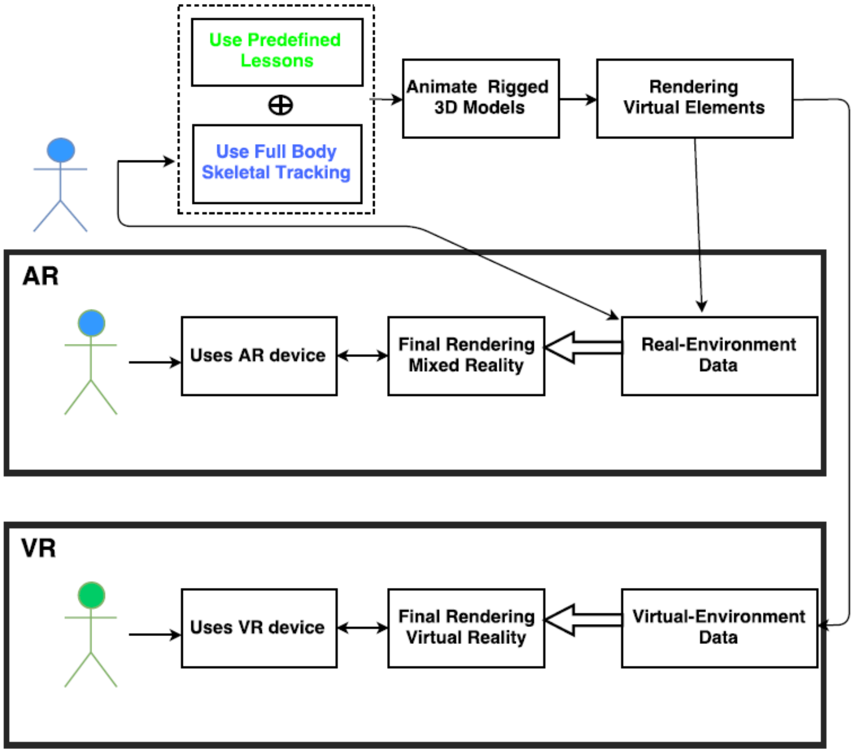


Figure 4.4 – System overview to support VR and AR

4.2.2 3D Models

A part of the contributions is represented by the methods we used to obtain realistic 3D models that were integrated into the applications. These models serve as avatars for the designed applications and they are characters in the VR and AR scenes. A cost-effective solution should be implemented since the level of realism needs to be at an acceptable standard and since perfect faithfulness is impossible [JVWR14].

This subchapter showcases how we obtained the 3D models of the human musculoskeletal system and skin. Two options were considered to obtain these models:

a. To purchase them from an online 3D models store.
b. To generate them, since the models that have a high degree of realism and details usually have a high cost.

The cost of these types of models varied from 500-600$ for a rigged model of bones and muscles and went up to 2000$ for the high-quality ones. However, these models weren't perfect as there were reviews that mentioned the clear differences between them and the biological data or the fact that the abdominal muscles meshes were a simple texture [AV16a]. Some solutions that targeted medical education used raw 3D images that were rendered as virtual elements in an AR environment [MCFM15] [TB12] [CK14]. Others, such as the one from [KK14] used 3D models of blood vessels generated from 3D magnetic resonance angiography images. However, the level of complexity of those models is much smaller compared with the ones presented in this section.

The first set of models were obtained from medical images for two reasons: realism and cost efficiency. For specialists that are not graduated in art and 3D modelling, creating a human musculoskeletal system from nothing is a very challenging task. The methods we applied for obtaining 3D models from medical images are detailed in the next section. Due to some implementation limitations we continued searching for relevant 3D models suited for our research topic. We managed to find alternative options that became available during the implementation period.

4.2.2.1 3D Models Obtained from Medical Images

MRI (Magnetic Resonance Imaging) or CT (Computed Tomography) datasets can be used to generate the 3D models. To obtain the needed data a professional software such as *SimpleWare Scan IP*[45] or *3D Doctor*[46] can be used. Both solutions aren't free, but we managed to obtain trial access to *SimpleWare Scan IP* software for a limited period in which we could proceed fast to obtain our data. This means that additional issues that were discovered after the access was removed couldn't have been modified but the quality of this solution was far superior to others. These models will be animated at runtime and they need to have a bone hierarchy attached and to be skinned. The initial results are static meshes that cannot be animated at runtime without additional changes regarding rigging and skinning.

Although the focus is on the human musculoskeletal system, we generated the skin layer as well to be able to test an alternative scenario regarding the immersion of the users as this will make the users see initially a realistic human body at the first interaction with the application.

As previously mentioned, the quality of the models and their fidelity with the biological data is important for the developed application. However, these types of models tend to be very complex, with a large number of vertices that can overload the application and can impact the users' experience. To be able to improve the quality of the resulted models a significant additional time was necessary, hence a balance between the realism, cost effectiveness and development time had to be chosen.

---

[45] https://www.simpleware.com/software/scanip/

[46] http://www.ablesw.com/3d-doctor/

Input Data

3D medical images were used as input data for the model generation. They were imported from the *OSIRIX* viewer samples[47]. There were 55 datasets available, acquired with different techniques (*CT – Computed Tomography, MRI – Magnetic Resonance Imaging, PET – Positron Emission Tomography, MRA – Magnetic Resonance Angiography*). The datasets are in DICOM (Digital Imaging and Communications in Medicine) format which is a standard for storing and transmitting medical images, creating the possibility of using the same scan in different medical facilities. The scans represent persons with certain medical conditions and their identity is anonymous, in consequence an alias is given for each dataset. Moving forward, we'll proceed using the alias to identify them. The scans were taken from different body regions and only three of them (out of 55) were tagged as whole-body scans: OBELIX, MELANIX and PETCENIX. Our interest was to generate models of the human musculoskeletal system and it was necessary that the images covered the whole-body. All three datasets that were tagged as whole-body were acquired using CT modality.

Further investigation was needed to select a dataset from the ones available. The data was analyzed using the *3D Slicer*[48] software platform. This is a free solution that is able to visualize DICOM files. We analyzed the datasets and observed that all three of them have the same number of rows and columns (512x512) and the differences were on the Z-axis, that corresponds with the slice thickness[49]. OBELIX dataset dimension is 512x512x1558, MELANIX is 512x512x1708 and PETCENIX is 512x512x291 where OBELIX and MELANIX have the best image resolution. Another important aspect is the fact that MELANIX is composed from two separate scans (upper limbs and the lower part), therefore the mentioned value (1708) represents the sum of these scans resolution on the Z axis, although the covered regions are overlapping. After a visual assessment of the 3D images per slice, we observed that the two datasets that were marked as whole body didn't contained the forearms. MELANIX was the only one that contained this information, but indeed it was in a separate set of images. Fig. 4.5 displays captures from the mentioned datasets and the body parts that they covered.

Based on the reasons exposed above, MELANIX dataset was selected as input data for the model generation pipeline as it has the best image quality, covers the whole-body (even if it was in two separate sets) and it offers the necessary biological information about the musculoskeletal system of a person.

---

[47] http://www.osirix-viewer.com/resources/dicom-image-library/

[48] https://www.slicer.org/

[49] http://tech.snmjournals.org/content/35/3/115.full

Figure 4.5 – OSIRIX samples: OBELIX [A, B], PETCENIX [C, D], MELANIX [E, F, G]

Data Preprocessing

The selected dataset was composed of two parts: upper limbs (Fig. 4.5 F) and the lower part (Fig. 4.5 E, G). An alias is assigned to each part and moving forward these scans are identified by their newly given aliases. The first one, displayed in Fig. 4.5F, is named ULH (Upper Limbs and Head) while the second one is named HTLL (Head, Torso and Lower Limbs). The scans are available in OSIRIX data set in separate folders. ULH has 512x512x506 voxels, while HTLL had 512x512x1202 voxels with overlapping regions, such as the head area.

Both scans were imported and processed using *ScanIP* program. From these two, HTLL had the biggest performance issues during processing and model export phase. This was difficult especially for the most complex models/layers such as muscles and skin, while the application froze or sent a memory exhaustion error in 90% of the cases. The workstation used for processing had better specification compared with the minimal requirements but didn’t met all the recommended ones. More details about the hardware specifications are available in Table 4.1.

Table 4.1 – ScanIP hardware requirements

	Minimum	**Recommended**	**Used**
Processor	Intel Core i3 or equivalent	Intel Core i7 or equivalent	Intel Core i7-4710HQ
Memory[50]	4GB	16GB	8GB
Screen Resolution	1024x768	1920x1080	1920x1080

---

[50] The actual memory requirements depend on the size of the images used for processing. Source: https://www.simpleware.com/software/

The HTLL scan was divided in two parts to overcome the mentioned performance problems: Head and Torso (HT) and Lower Limbs (LL).

The bones models were easily obtained from the initial scans without further preprocessing. The bones were obtained from the ULH and HTLL subsets and the muscles and skin from ULH, HT and LL subsets. Bones, muscles and skin layers were each processed and exported separately. Fig. 4.6 displays the 3D model previews of bones and muscles from the HTLL datasets (before division) as obtained with Scan IP software.

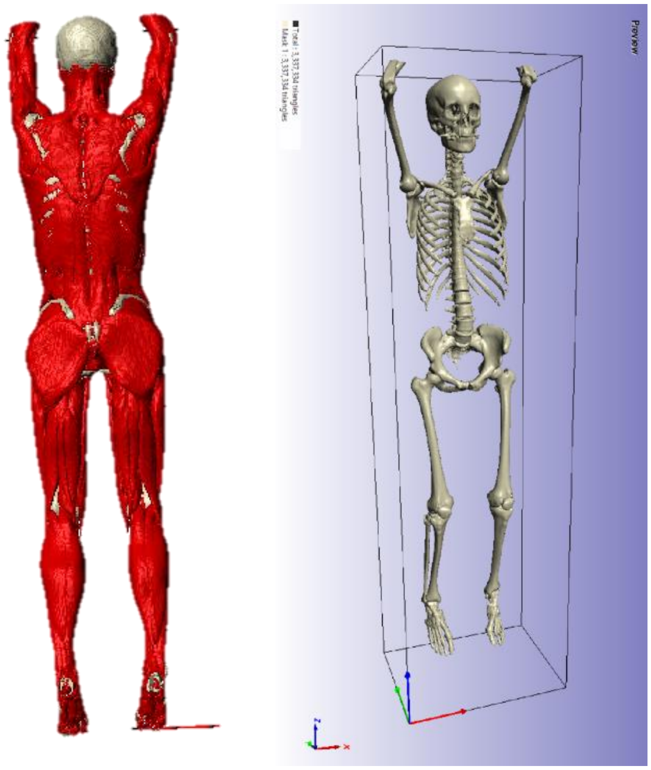


Figure 4.6 – 3D model preview of the muscles and skeleton as obtained from the HTLL dataset

Processing Methods

Each data subset is separately treated since different challenges were encountered for them. Input data consists of grayscale images and to obtain 3D models based on these images masks are created and modified based on the pixels values. Masks are images that have the same dimension as the input dataset and have assigned a single color for certain pixels. The masks can be edited per *Selection*, *Active Slice* and *All Slices*. For example, ULH subset has 512 slices on X-axis, 512 on Y-axis and 506 on Z-axis. Each slice contains a greyscale 2D image as they represent the loaded DICOM files. The masks can be edited manually or with various automatic methods (*E.g. Multilevel Otsu Segmentation*).

In *ScanIP* software the datasets have models associated besides the masks. These are created and updated based on the corresponding masks values. Each mask can generate a 3D model and one configuration can have maximum 255 masks. Surface and FE (Finite Element) mesh types could be obtained with this solution. Surface type was selected in this case and the format of the resulted files was STL (STereoLithography).

The technique used to obtain the bones was by far the easiest one as the contrast between the bones and the rest of the image is prominent. The pixel values that displayed the bone tissue were clearly higher compared with the other regions. We could identify all the elements (biological data) without many issues. Figure 4.7 displays a part of the lower limbs and figure 4.8 displays the dataset that had its contrast improved to distinguish better the bones.

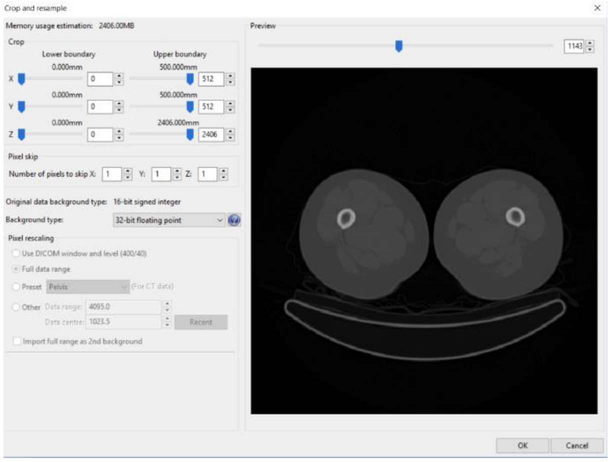

Crop and resample
Memory usage estimation: 2406.00MB
Crop
Lower boundary / Upper boundary
X: 0.000mm, 0 / 500.000mm, 512
Y: 0.000mm, 0 / 500.000mm, 512
Z: 0.000mm, 0 / 2406.000mm, 2406
Pixel skip
Number of pixels to skip X: 1 Y: 1 Z: 1
Original data background type: 16-bit signed integer
Background type: 32-bit floating point
Pixel rescaling
○ Use DICOM window and level (400/40)
● Full data range
○ Preset Pelvis (For CT data)
○ Other Data range: 4095.0 Data centre: 1023.5
☐ Import full range as 2nd background
Preview: 1143

Figure 4.7 – HTLL dataset - initial image

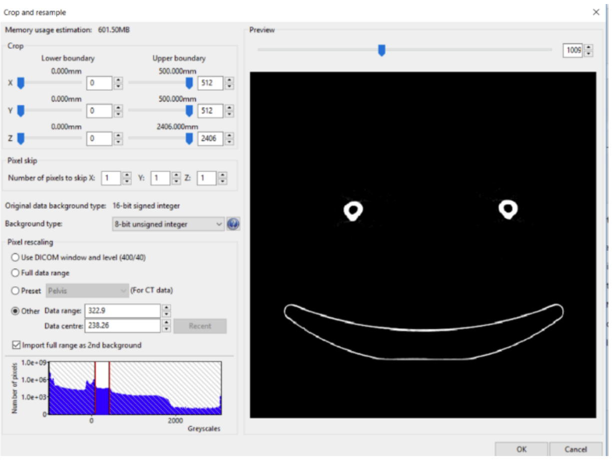

Crop and resample
Memory usage estimation: 601.50MB
Crop
Lower boundary / Upper boundary
X: 0.000mm, 0 / 500.000mm, 512
Y: 0.000mm, 0 / 500.000mm, 512
Z: 0.000mm, 0 / 2406.000mm, 2406
Pixel skip
Number of pixels to skip X: 1 Y: 1 Z: 1
Original data background type: 16-bit signed integer
Background type: 8-bit unsigned integer
Pixel rescaling
○ Use DICOM window and level (400/40)
○ Full data range
○ Preset Pelvis (For CT data)
● Other Data range: 322.9 Data centre: 238.26
☑ Import full range as 2nd background
Preview: 1009

Figure 4.8 – HTLL dataset – improved image to emphasize the bones tissue

Similar operations were effectuated for skin and muscles and after that, masks were created for each of them and as contained the pixels that had they greyscale values that included the targeted regions. For example, figure 4.9 showcases how the masks looked superimposed to greyscale images and how the generated meshes looked in preview (bottom right). The resulted models were obtained in the STL format.

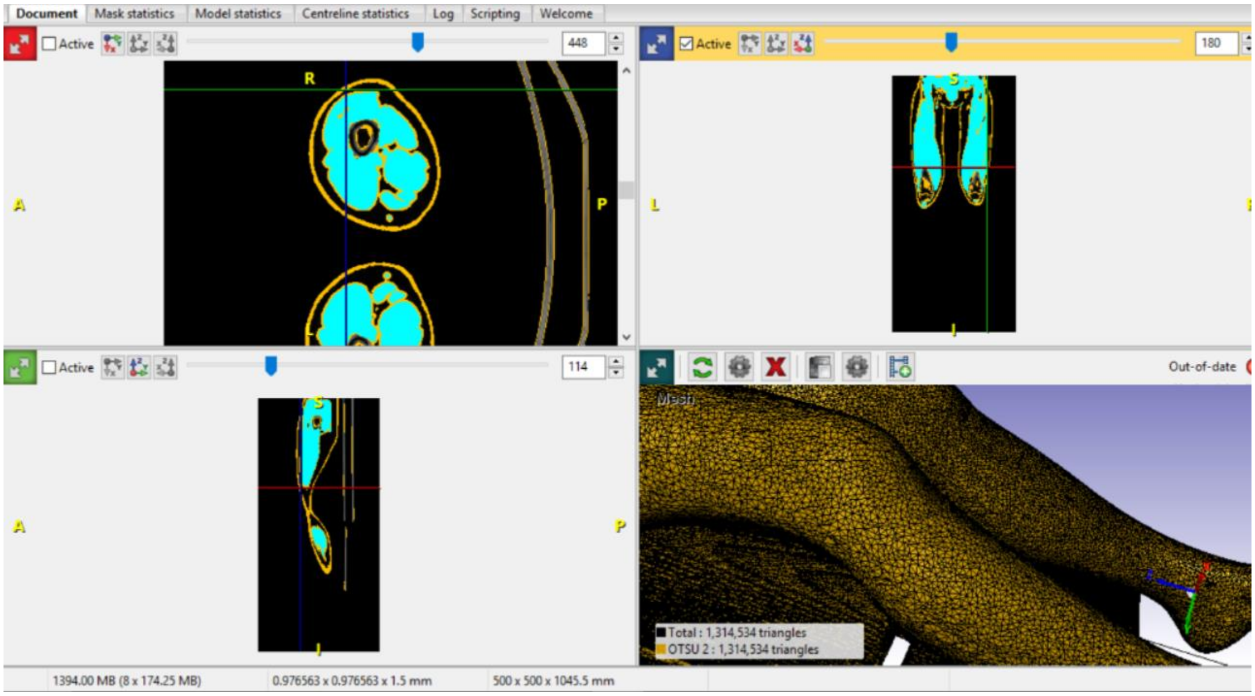


Figure 4.9 – LL dataset – masks (orange for skin, blue for muscles)

Post Processing

The data is divided into subsets and therefore models from multiple regions were obtained instead of three individual 3D models that represent the bones, muscles and skin. In this step these subsets are merged back together, and the models have a few corrections applied. *Blender*[51], *3DS Max*[52] and *3D Builder*[53] applications were used for this phase. Initially, *Blender* was considered a viable option because it's free and open source. However, some performance issues were noticed during the usage of this program with the obtained models and *3DS Max* software was considered as well. *3D Builder* was used as well because it had the best performance during the merging operations. Unfortunately, it has a limited functionality and the optimization algorithms provide weaker results compared with *3DS Max*.

At first, the models were part of a cleaning process where the unneeded geometries were removed (e.g. clothes, the board from the CT device on which the patients were sat during the scan, etc.). It was easier to perform these corrections on the obtained models since the potential errors that could have been generated would be observed easier. Fig. 4.10 displays the corrections performed on the skin model of the ULH dataset. The selected vertices (with red) were removed from the model and the corrected models were

---

[51] https://www.blender.org/
[52] https://www.autodesk.com/products/3ds-max/overview
[53] https://www.microsoft.com/ro-ro/store/p/3d-builder/9wzdncrfj3t6

merged together (Fig. 4.11). Figures 4.12, 4.13, 4.14 and 4.15 show the post processing phase of the bones models step by step in *3D Builder*.

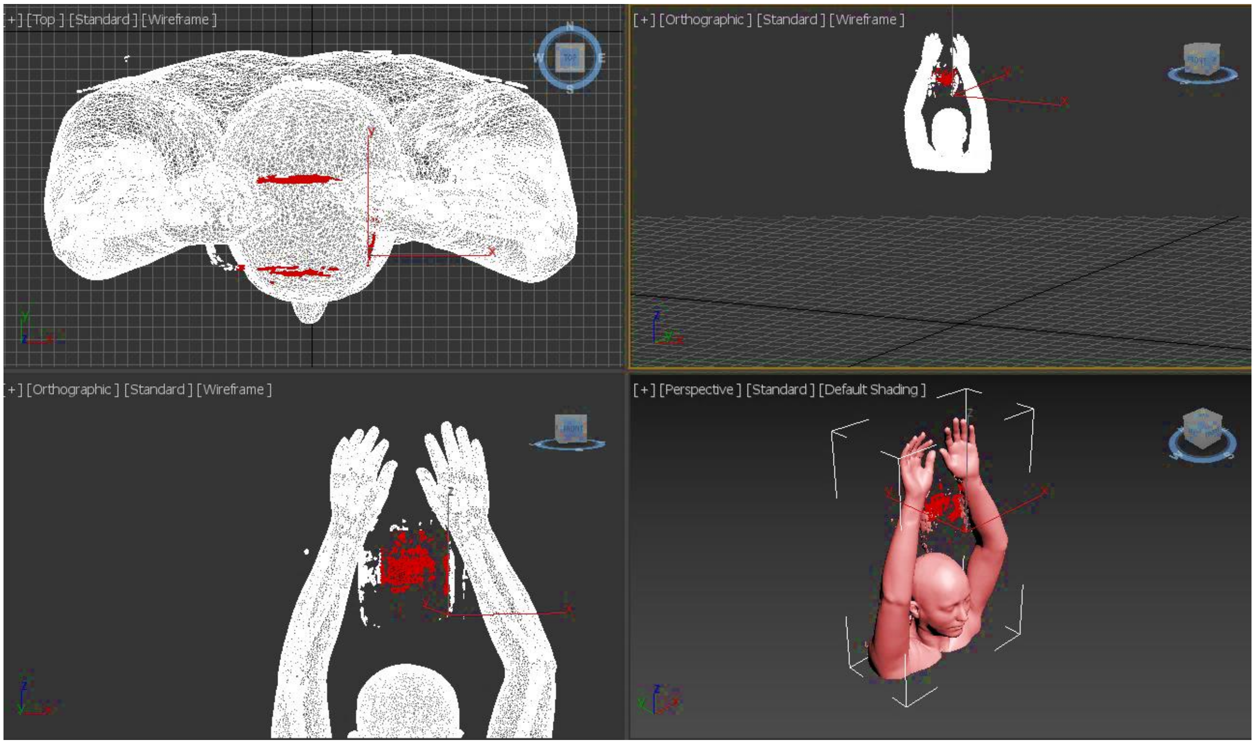


Figure 4.10 – ULH dataset skin model correction

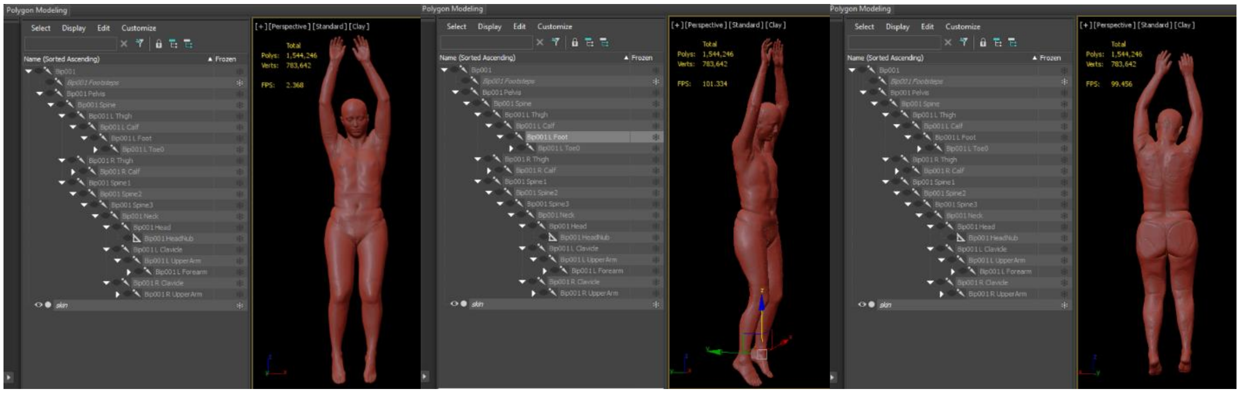


Figure 4.11 – Complete skin model

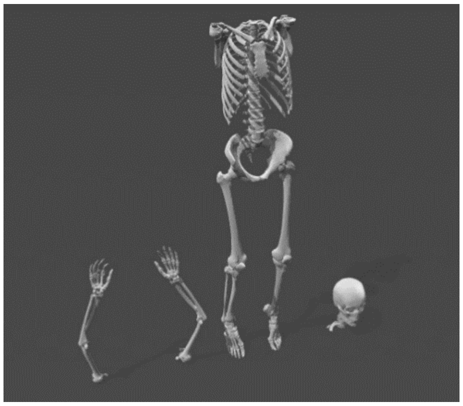


Figure 4.12 – Bones meshes import

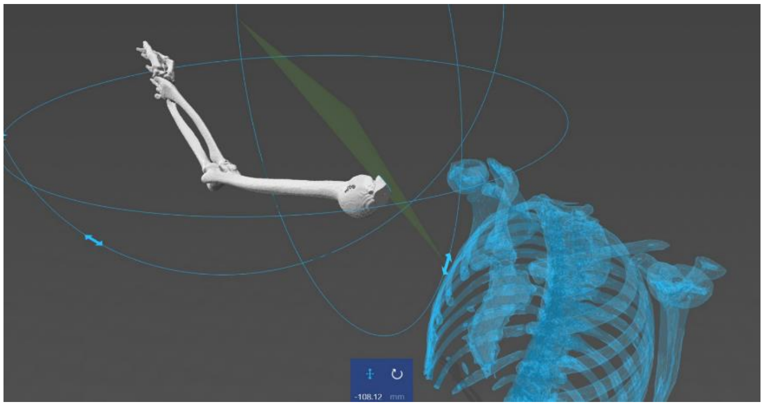


Figure 4.13 – Model's correction

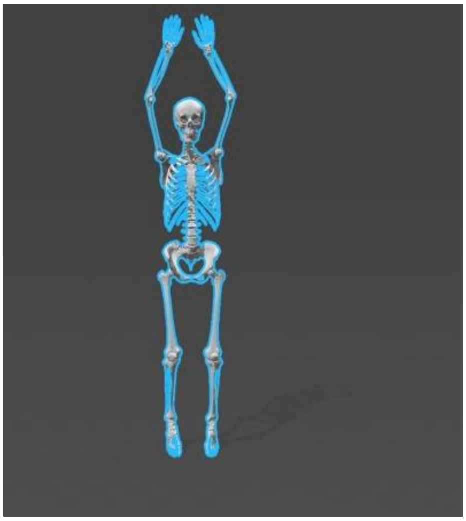


Figure 4.14 – Final model

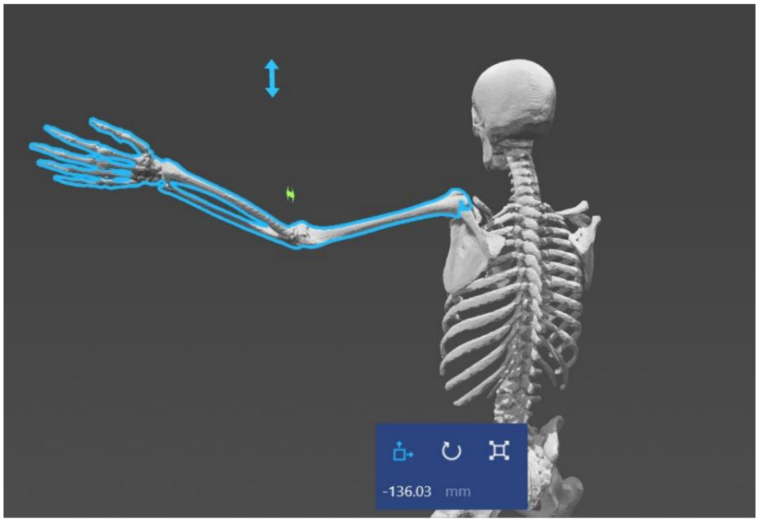


Figure 4.15 – Merging meshes

Rigging and Skinning

The models obtained in the previous step must be rigged and skinned to be able to animate them as they are just static meshes without a bones structure attached (skeleton). Initially, the *Mixamo* application was chosen as it should permit to rig and animate characters in just a few minutes. Unfortunately, this application didn't work in this complex case and the bones model was the only one that could be uploaded while for others the upload process froze the application. For bones, the resulted animations were clearly broken, and a backup solution was considered where the default skeleton from *Make Human* was imported into the *3DS Max* scene that contained the bones static model. We removed the mesh of the *Make Human* model that had the skeleton attached and made it fit over our own model (Fig. 4.17) since its initial stance was in A-pose (Fig. 4.16). Unfortunately, on test phase this approach had issues (Fig. 4.18).

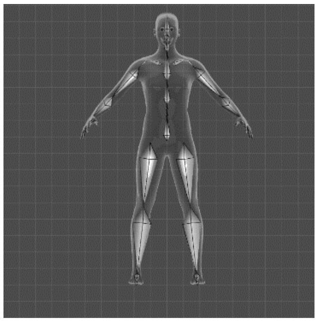


Figure 4.16 – Make Human basic skeleton

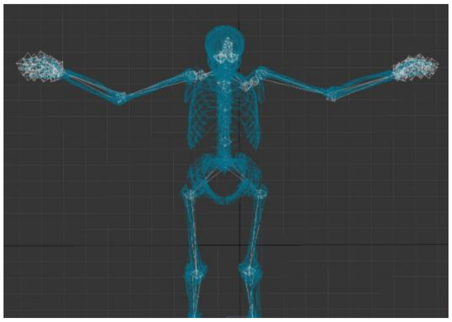


Figure 4.17 – Make Human basic skeleton superimposed on bones model.

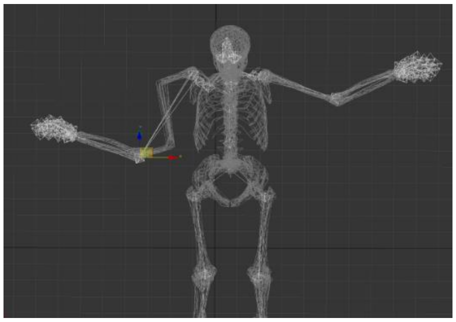


Figure 4.18 – Bones model animation error

We could have corrected them manually, but this would have been a very long process as this should have been done for almost every bone. After an in-depth analysis was performed, we concluded that the best approach was to use the *Biped* structure from *3DS Max* as we noticed that this one produced the best results. Using *Biped,* the rigging and skinning was most of the time straight forward, at least for the bones model. Corrections to skin envelopes were necessary as well but their incidence was much smaller. The biggest issues that were encountered were at the body parts that were too close one another as our models weren't in A-pose or T-pose while executing this operation. The skin envelopes of problematic bones were manually readjusted to fit better over the model's geometry, as seen in Fig. 4.19.

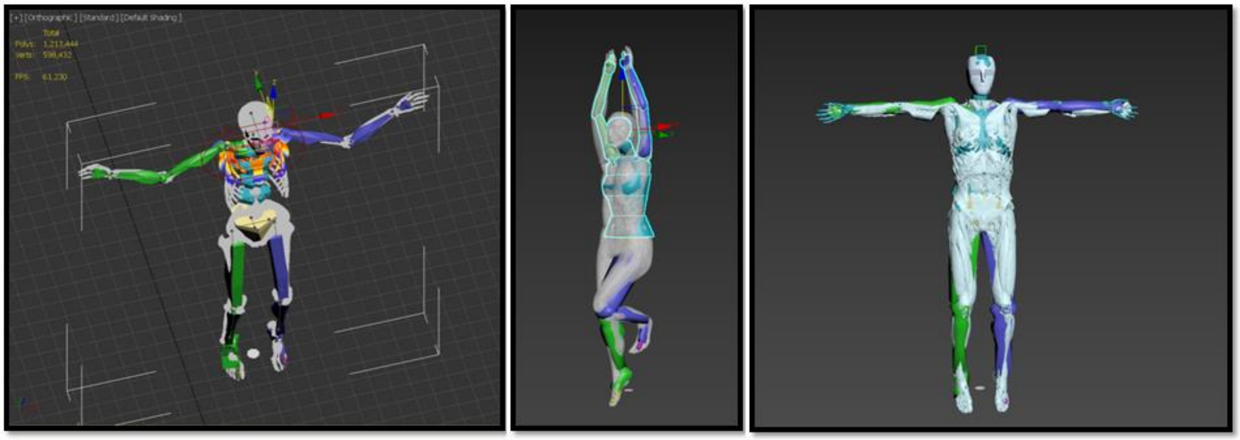


Figure 4.19 – Rigging and skinning of obtained models using Biped structure from 3DS Max (Left – Bones/Skeleton, Middle – Skin and Right – Muscles model)

With this final step, the models should be ready for integration. However, after integrating them into our application, some errors were observed along the way. Several issues were corrected, and the models were reimported in Unity for testing.

Workflow

This section presents the workflow used to obtain realistic 3D models of human bones, muscles and skin based on medical images. Performance issues were encountered on each step and backup solutions had to be put in place. For example, the HTLL datasets had problems with muscles and skin meshes export. This was overcome by dividing the scan in two separate images and processing them individually thus they required additional work in the post-processing phase. Fig. 4.20 displays the main data workflow used for generating realistic 3D models. The left part shows the files' format as obtained from the tools used: DCM (the DICOM files), STL (STereoLitography) and FBX (Filmbox).

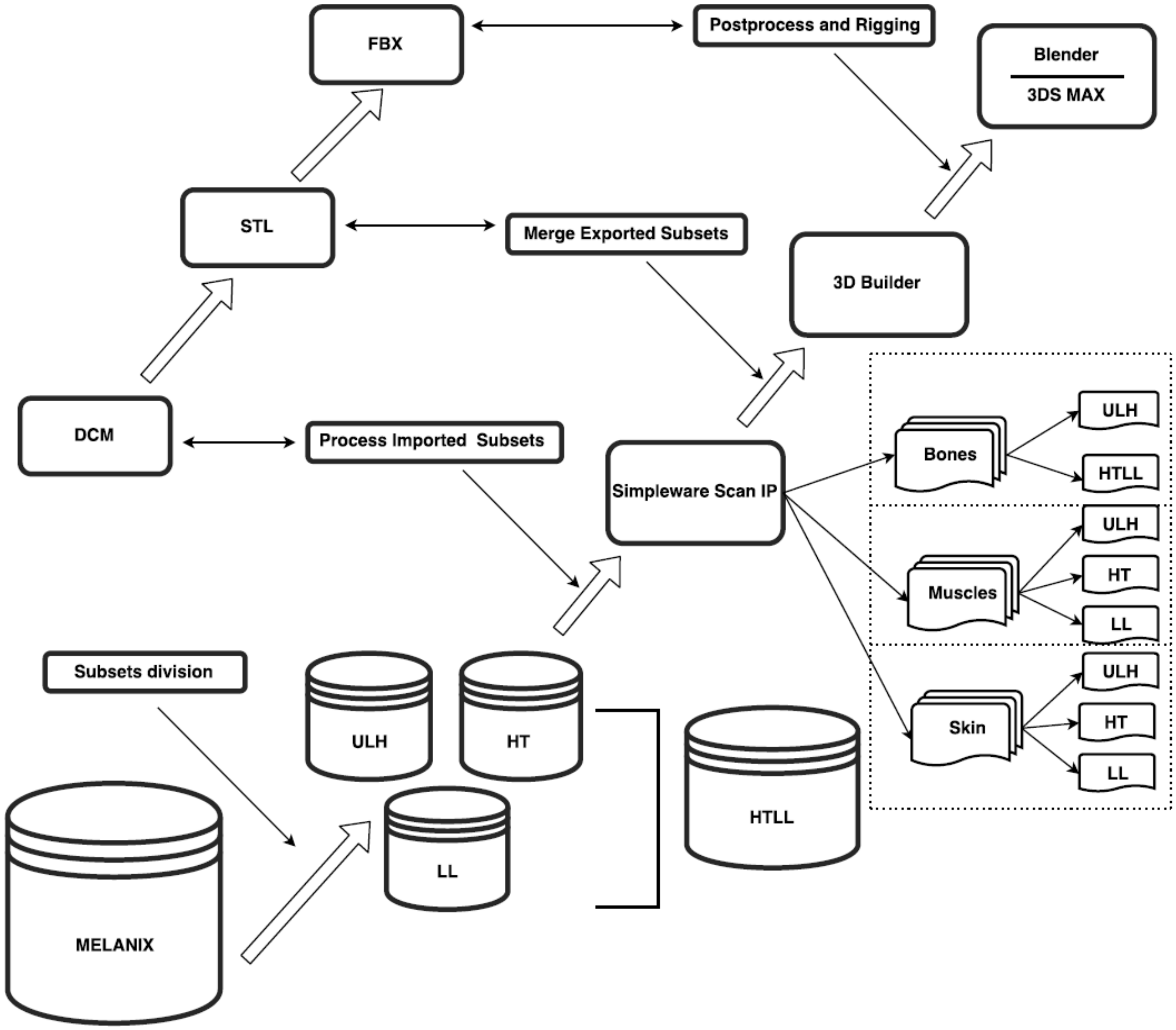


Figure 4.20 – 3D Model Generation workflow

Results

Unity game engine was used for developing all the solutions for this research, so the next step was to import the resulted models in the working projects. In Fig. 4.21 is displayed a set of models of bones, muscles and skin that were imported in a test Unity scene. A few errors were visible only after this step was effectuated and in consequence there were cases where the models needed further adjustments. For example, we had issues with the correctness of models' animation during motion tracking. We observed that the problems were generated as the model wasn't in a correct T-pose position and the models' limbs positions at runtime didn't correspond with the motion of the observed user. These were solved with a correct stance (the upper arm and forearm are at 180 degrees). Fig. 4.22 displays the bones model in the first approach that generated issues (left side) and the corrected version (right side).

On top of that, special shading materials were created for the models as they were exported only with the geometry information, no texturing or illumination information attached to them. We performed these operations in Unity using low cost shaders to make sure that the performance of the resulted applications won't be affected. These changes are observable in the figures bellow.

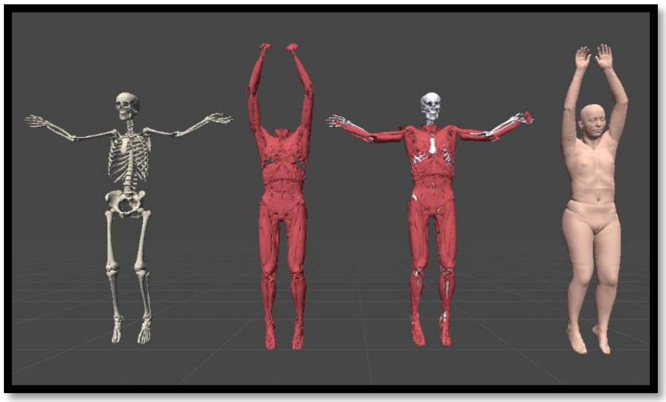


Figure 4.21 – A version of the obtained 3D models imported in Unity

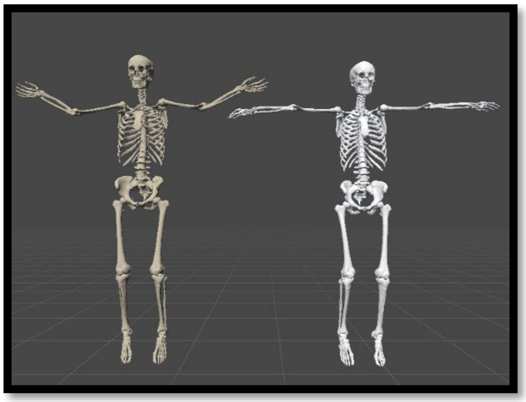


Figure 4.22 – Bones model in T-pose

One of the main advantages of the chosen solution is its cost. We managed to obtain realistic 3D models of human body that are anatomically correct for a zero cost. Recently *OSIRIX* samples are available only with premium membership although at the time this solution was developed they were free for academic use. *Scan IP* was used for image processing and even though the program has a high price for a license, we managed to get a free trial for 2 months. *3D Builder* and *Blender* are free as well and *3DS Max* was available with the student version.

As a disadvantage, the solution was extremely complex, requiring many working hours to make the models functional and to make the necessary corrections. The models' complexity is very high and performance issues were encountered almost all the time for various executed steps, as mentioned in the previous subsections. Also, skin and muscles models need further processing to improve their appearances, but unfortunately, we didn't have enough time to make these changes and we aim to improve them in the future.

Obtaining 3D models from medical images is not a new method and examples are present in the current literature [JT05]. Most of the research refers only to a small part of the body, such as bones, liver, face, etc. The novelty of our approach is the fact that we managed to obtain whole-body models even if this was achieved manually. If we would have proceeded to obtain results only for certain body parts, the used methods would have been much simpler and the restrictions less inconvenient. Since the whole-body was very complex we often managed to improve the visualization for some body parts but affected the others. Also, the computational costs necessary for this pipeline were significant. There are details that need to be improved but the current state of the models was sufficient for the initial implementation of the IBL project. Looking back at the process, we consider that it would have been better to export all three layers into the same models (different meshes per layer) to diminish the effort and errors that occurred at the merging phase (Fig. 4.21 – the model with bones and muscles).

4.2.2.2 Imported 3D Models

Late in our research we had the opportunity to find a set of anatomically correct models of bones muscles and skin that are suited for simulation that were free[54]. Although that point we had already implemented the applications, we decided to improve our results with these models that looked more suited for our research. Figures 4.23 – 4.24 (screenshots from *3DS Max*) and Fig. 4.25 (screenshot from *Mixamo*) showcase the imported models suited for simulation of bones, muscles and skin.

One important aspect is the fact that the models are not rigged, and we had to solve this by adding a bones hierarchy and skin the models. *Mixamo* software gave us the best results, even though the models weren't perfectly skinned the results were acceptable. Figure 4.25 displays the skin model test animation as obtained with *Mixamo*. Similar operations were effectuated with bones and muscles models.

Another plus of these models was the fact that each anatomical part was separated in individual meshes as opposed with the models obtained from medical images. This will

---

[54] https://www.turbosquid.com/3d-models/free-human-simulation-3d-model/1118599

help with applications' interactivity as the individual parts can be highlighted properly instead of using various workarounds.

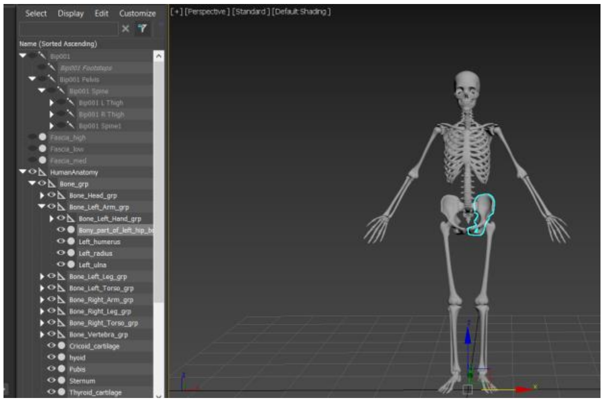


Figure 4.23 – Static bones model of human anatomy

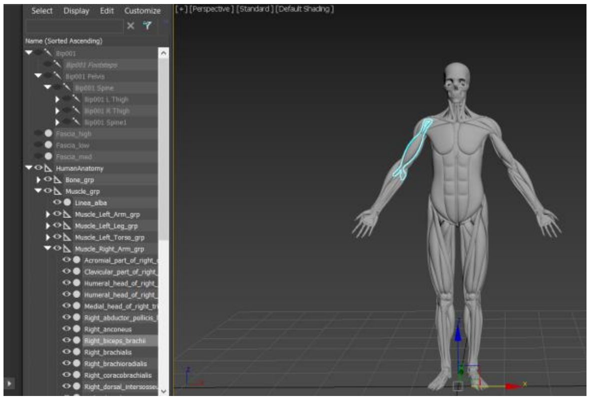
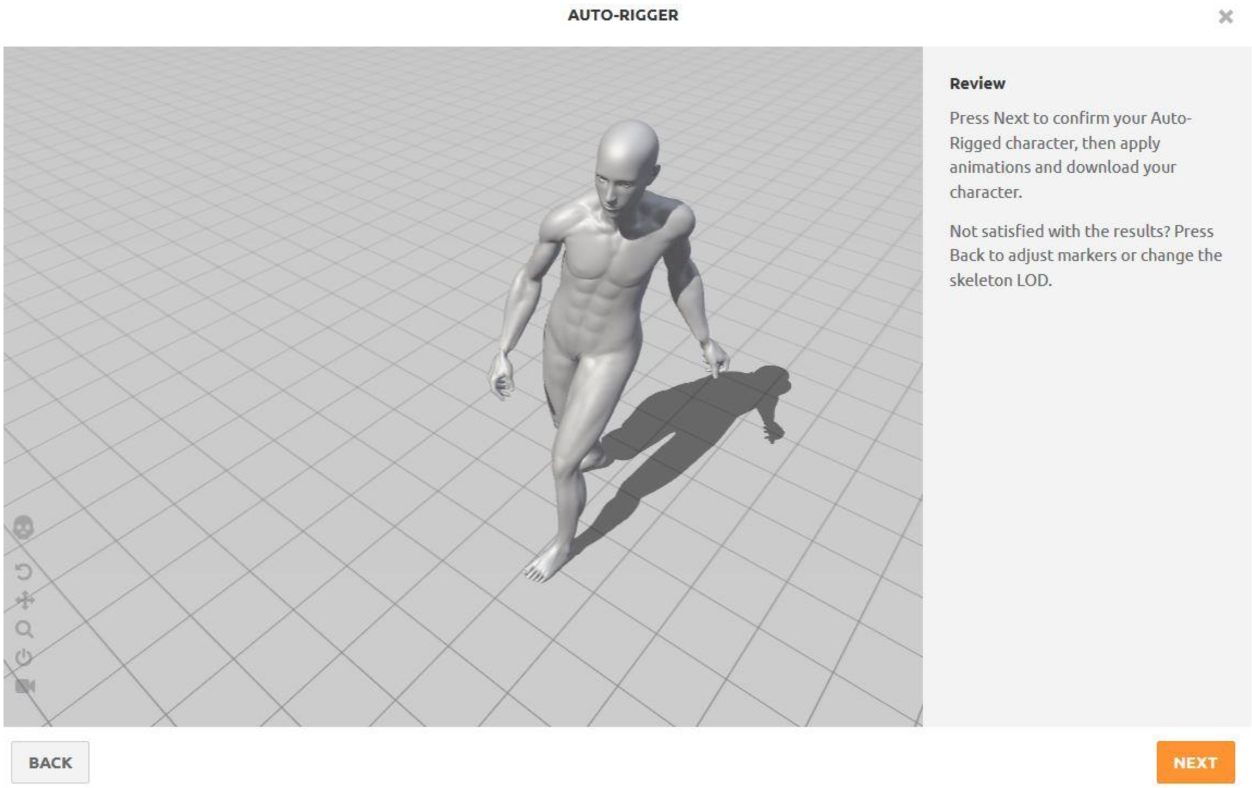


Figure 4.24 – Static muscles model of human anatomy



Review

Press Next to confirm your Auto-Rigged character, then apply animations and download your character.

Not satisfied with the results? Press Back to adjust markers or change the skeleton LOD.

Figure 4.25 – Rigged skin model

4.2.3 Tests with different VR/AR technologies

This section contains the results we obtained on various technologies while developing Interactive Biomechanics Lessons. These technologies were used for VR and AR display and motion tracking.

4.2.3.1 Visualization

Google Cardboard

Cardboard is a low-cost viewer that can be used with a smartphone (Android or iOS) to display VR content. The device can be manufactured at home or purchased as existing viewers are available at different prices that vary between 7USD and 70 USD[55]. No separate controllers are needed since the users can interact with apps through the viewer's trigger input (Fig. 4.28 red rectangle). The Cardboard viewer was used with a Samsung S6 device and the development on this setup was facile.

To display the VR content, we need to use stereoscopic rendering and the device will display on its screen the images for both eyes (Fig. 4.26 as example). The user will see the content properly thanks to the two lenses that will each capture the images for each eye. They enable the users to focus their vision on the smartphone screen and allow the device to be placed to a short focal distance (2-5cm)[56]. Figures 4.26, 4.27 and 4.28 illustrate the Google Cardboard setup used for Interactive Biomechanics Lessons development.

Figure 4.26 – Stereoscopic rendering on mobile

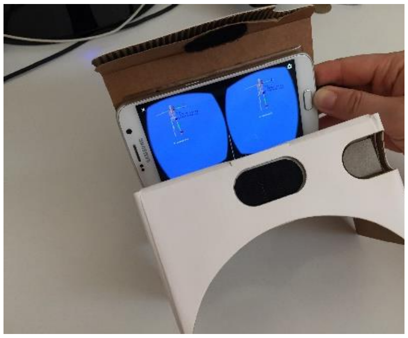


Figure 4.27 – Setting the device in the viewer

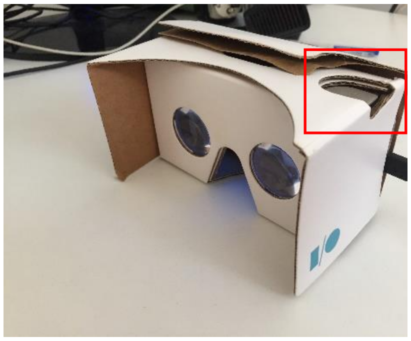


Figure 4.28 – Cardboard headset

Samsung S6 device used to display the VR content has a total resolution of 2560x1440 pixels with approximately 577 ppi density. The CPU chipset is Exynos 7420 Octa core (4x2.1GHz Cortex-A57 & 4x1.5 GHz Cortex-A53) and the GPU is Mali-760MP8. During the first VR implementation (only bones from the generated models) no framerate issues were encountered. However, while adding more complex models (skin and muscles) a

---

[55] https://vr.google.com/cardboard/get-cardboard/index.html

[56] https://static.googleusercontent.com/media/vr.google.com/ro//cardboard/downloads/manufacturing-guidelines.pdf

performance drop was observed. More details regarding runtime performance values are showcased in section 4.2.6.

Since we used as an HMD a Google Cardboard viewer with a smartphone device attached to it, the methods of interaction with the applications were restrictive. The display device receives input via the viewer's generic button mechanism that taps the device's screen when the button is pressed. The interaction with the virtual environment was based on a reticle (the red point in figure 4.29 highlighted by the blue rectangle) placed in the center of the viewport to be able to select various 3D elements within the scene. When hovering the reticle over an interactive element it acts as a select button. These interactive 3D elements that are blended into the virtual world aim to replace the classical 2D UI approach that is not suited for VR. The navigation through the predefined lessons uses additionally the viewer's button. The navigation options (e.g. "Next", "Previous") consists of 3D Text elements that are zoomed when the reticle is displayed over the text. The select method was designed in this manner to ensure a fast and prompt feedback during each biomechanics lesson and to make sure that the users didn't advance to the next lesson by mistake (if they just directed the reticle over the "Next" option). Figure 4.29 displays highlighted in the blue rectangle the reticle used in the VR implementation for Cardboard. Same functionality is available in Unity editor and it was implemented using a "*FakeCamera*" movement to simulate the same behavior as using an HMD for visualizing the 3D scene (by changing the main camera direction based on the mouse input).

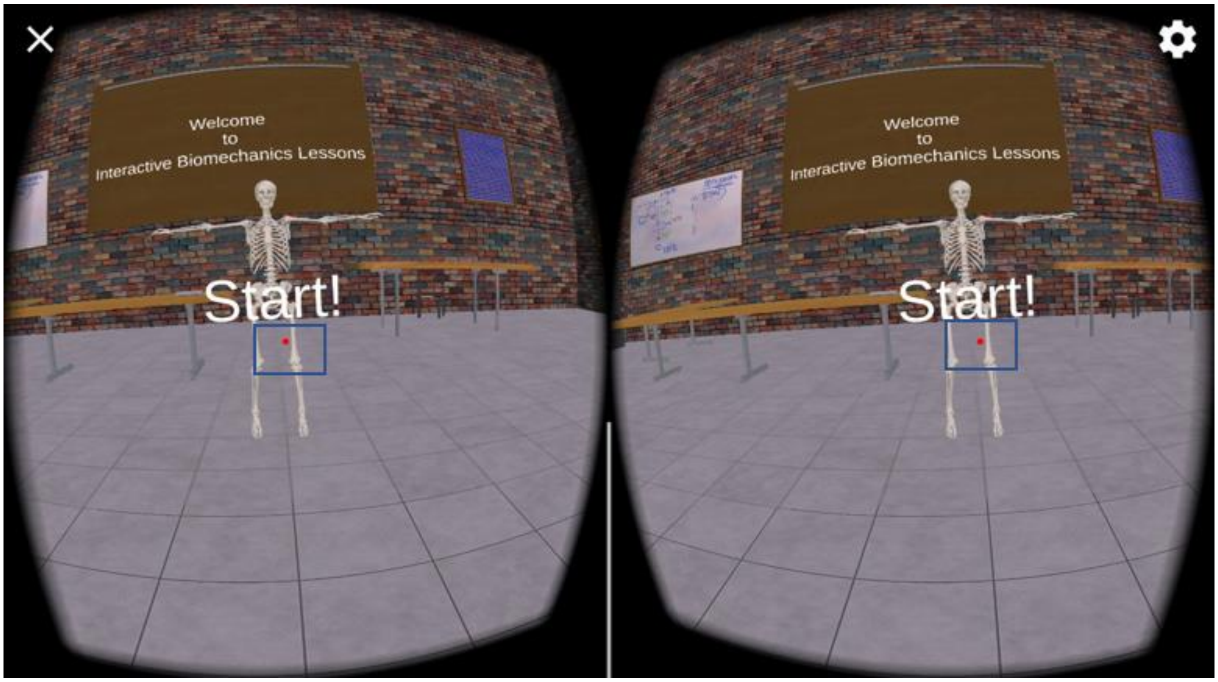


Figure 4.29 – Graphical User Interface Reticle used for VR applications developed for Cardboard viewer

Gear VR

In the second part of the project we used a Gear VR headset. This was compatible with the Samsung S6 device that was previously used with the Cardboard headset. Gear VR is compatible with the latest Samsung S series devices and proves to be a real alternative to expensive VR headsets. Besides the viewer it also has a controller for a better user interaction in the VR environment. In our applications we used only a button (the Android back button) that was placed on the headset to interact with the virtual environment as we kept the parity with the initial Cardboard implementation. This is showcased in Fig. 4.31 in the red rectangle.

There are a few similarities with the Cardboard viewer as both work with mobile devices and simply offer the possibility to render stereoscopically the virtual scene. While the Cardboard supports a wider range of phones, Gear VR is compatible only with Samsung smartphones. Another difference is related with the fact that the smartphone needs to be connected at the Gear VR's USB port (Fig. 4.31 yellow rectangle) to be able to run the applications. This proved to be a downside since it was harder to debug and profile application on the device. For certain cases we used the cardboard SDK to be able to see the VR content and interact with it.

We used *VR Samples* package to add input in our projects using Gear VR headset[57]. The main camera game object was replaced with the imported package prefab one. Moving forward we used the GUI (Graphical User Interface) reticle available in *VR Samples* resources. Figure 4.30 showcases the Samsung S6 device attached to the Gear VR headset.

Figure 4.30 – Samsung S6 device connected to a Gear VR headset

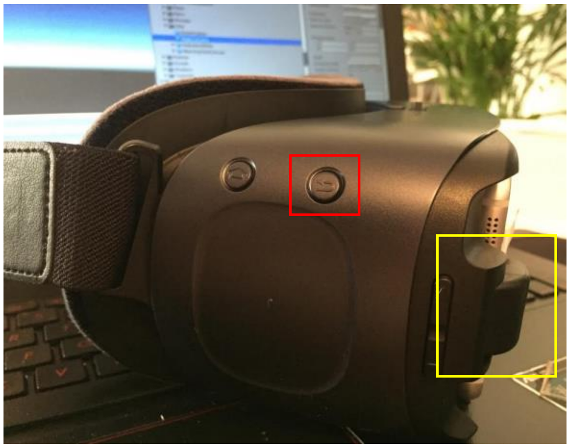


Figure 4.31 – Gear VR headset buttons and USB port

---

[57] https://unity3d.com/learn/tutorials/topics/virtual-reality/vr-overview

HoloLens

HoloLens is a Mixed Reality device developed by Microsoft. "*HoloLens lets you create holograms, objects made of light and sound that appear to be in the world around you, just as they are real objects. Holograms respond to gaze, gestures and voice commands, and can interact with real-world surfaces around you*"[58]. The initial proposal was to use a HoloLens device to display the Augmented Reality, but unfortunately, we couldn't manage to obtain a device for development and taking into account the high cost of a device this couldn't be acquisitioned. However, before renouncing at the idea of using a HoloLens device we started to test the possibility of developing on it. The work mentioned here is a very brief proof of concept and it can be extended in case the opportunity of developing on this device is available in the future.

These tests were conducted in mid-2016 period. The development was targeting both HoloLens and other mobile platforms (tablets) for displaying the holograms/AR content. At that point a special version of Unity was necessary for HoloLens development (5.4.0B22-HTP), along with the official releases (5.3.5.5f1) that were used for other mobile platforms. During the platform testing we found out that HoloLens is not providing skeletal or point cloud data, although it has a depth camera incorporated as the motion tracking feature is one of the interest points of this research. The HoloLens device has an Intel 32-bit architecture and can be used with Windows 10. There are two possibilities to run the applications: directly on the device or with an Emulator. We tested the possibility of emulating the HoloLens device and some very basic elements were added in those tests as seen in Fig. 4.32 and 4.33. Unfortunately, the Emulator cannot replace an actual device but can offer the opportunity to develop a large part of the work involved using it. Another option recently available was the device's simulation in Unity Editor, offered by the newer versions. A great feature of HoloLens is the fact that the applications are compatible with Universal Windows Platform. Microsoft enhanced the possibility to write common code for all their platforms improving the development standards.

---

[58] https://developer.microsoft.com/en-us/windows/mixed-reality/Hologram

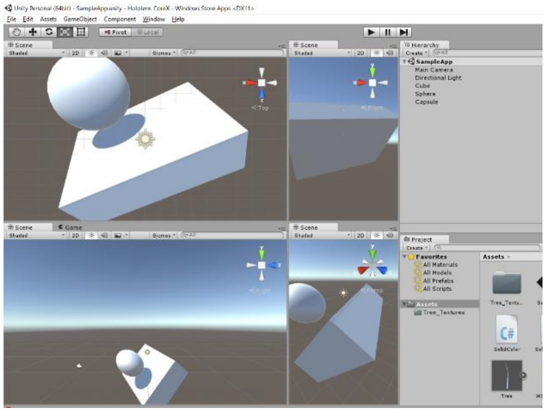


Figure 4.32– HoloLens test– Unity Editor scene

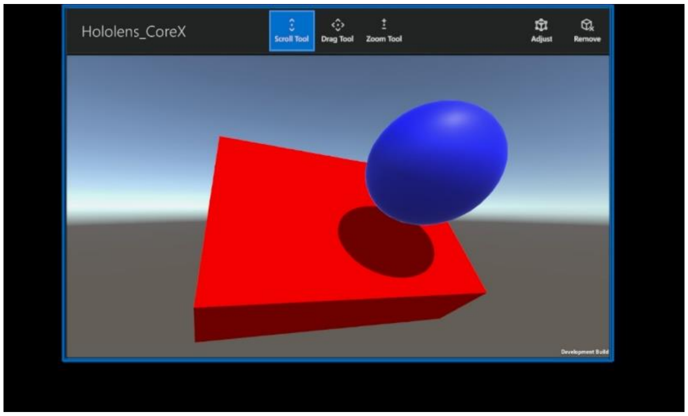


Figure 4.33 – HoloLens test – Emulator (Visual Studio Solution)

Vuforia

Vuforia is an augmented reality platform that supports a majority of smartphones, tablets, notebooks, digital eyewear, AR glasses and VR viewers[59]. We used it for testing various AR scenarios. This powerful platform was used to create an overlay with virtual elements on top of the real environment. It was used for multiple devices: a smartphone, a tablet and a laptop. Vuforia has multiples AR tracking features such as: objects recognition, cylinder targets, image targets, user defined targets and VuMarks targets[60].

Similar with the previous cases, Vuforia had support in Unity and the progress from VR setup could be ported and adapted to this platform. The support for Vuforia was available in Unity for some time but as a separate unity package. However, starting with Unity version 2017.2 this became integrated into Unity, making the development process easier. The developed application is built using a single Unity scene and it contains an *AR camera* that is specific to Vuforia while the default *Main Camera* is removed. This special camera has a *Vuforia Behavior* script attached. Two directional lights are added, one for the camera and another one for the scene. In the first tests the lights settings weren’t properly set, and the environment lighting conditions affected the resulted image as well (Fig. 4.34).

Vuforia was used to implement the AR marker-based scenario of IBL project and examples of incipient results are available in Fig.4.34 and Fig.4.35. Additional implementation details of the marker-based AR predefined lessons are provided in section 4.2.5. For the markerless AR scenario we decided to renounce to Vuforia since we needed additionally to detect at runtime the face of the observed user and continued with OpenCV to solve this aspect.

---

[59] https://www.vuforia.com/devices.html

[60] https://www.vuforia.com/features.html

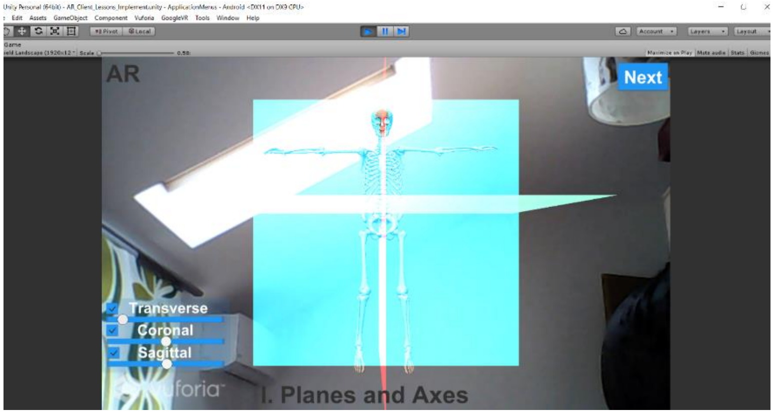


Figure 4.34 – Lighting conditions effects on test AR scene

Figure 4.35 – AR scene example - without tracking

4.2.3.2 Motion Tracking

We aimed to add a new level of interaction for the users when learning biomechanics lesson by animating the 3D models based on the tracked movements of an observed user. The tests results shown into this section present a big part of the AR markerless application of Interactive Biomechanics Lessons as we developed it using motion tracking from an additional sensor and computer vision.

The project target is to be a mobile friendly solution and a few options were analyzed to bring this idea to reality. Fig. 4.36 presents a short list with devices and compatible sensors sorted by their mobility category.

As mentioned in the previous section, the initial idea was to use a HoloLens device and unfortunately, even if the device had a depth camera incorporated it didn’t return any data necessary for the skeletal tracking. VicoVR was another interesting option for skeletal tracking as it was created especially for mobile environments, but their product release date was delayed and interfered with our own project timeline. We continued to work with Kinect V1 sensor from Microsoft as we already knew the technology and could rely on its capabilities, since the other potential technologies weren’t at that point released (mid 2016).

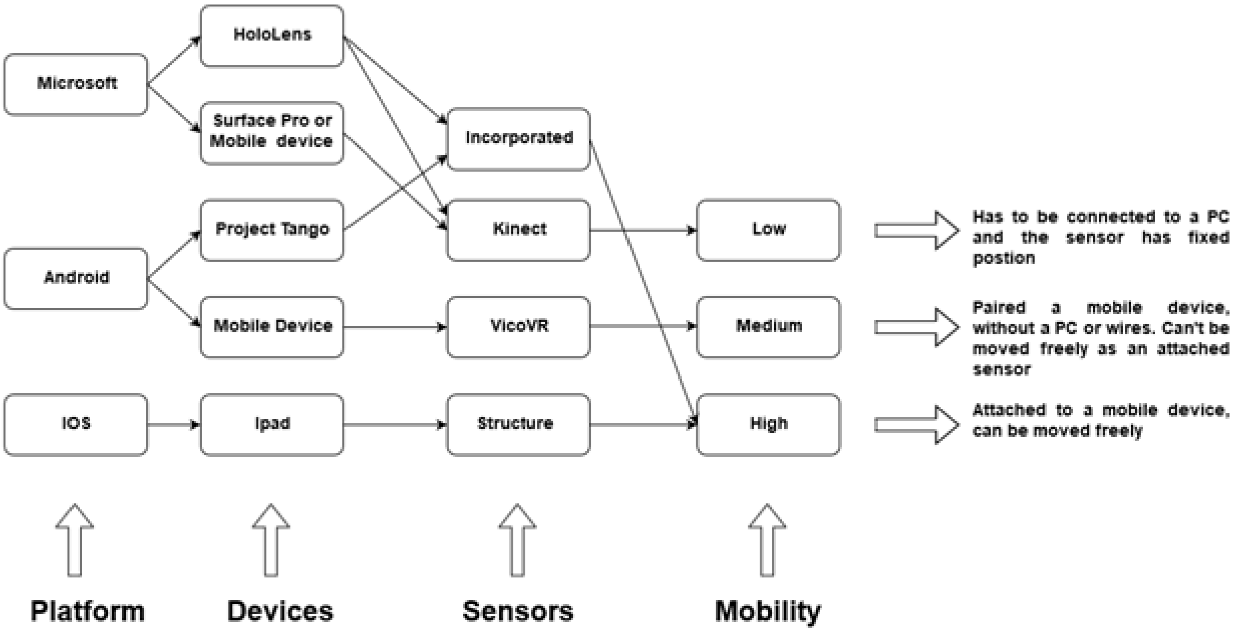


Figure 4.36 – Combinations of tracking sensors and mobile devices

The sensors that were considered in the setup of our applications were Kinect and VicoVR. One needs to be connected to a PC in order to work while the other can be connected directly with a mobile device via Bluetooth. Fig. 4.37 illustrates a brief scheme of an AR solution that includes motion tracking using additional sensors.

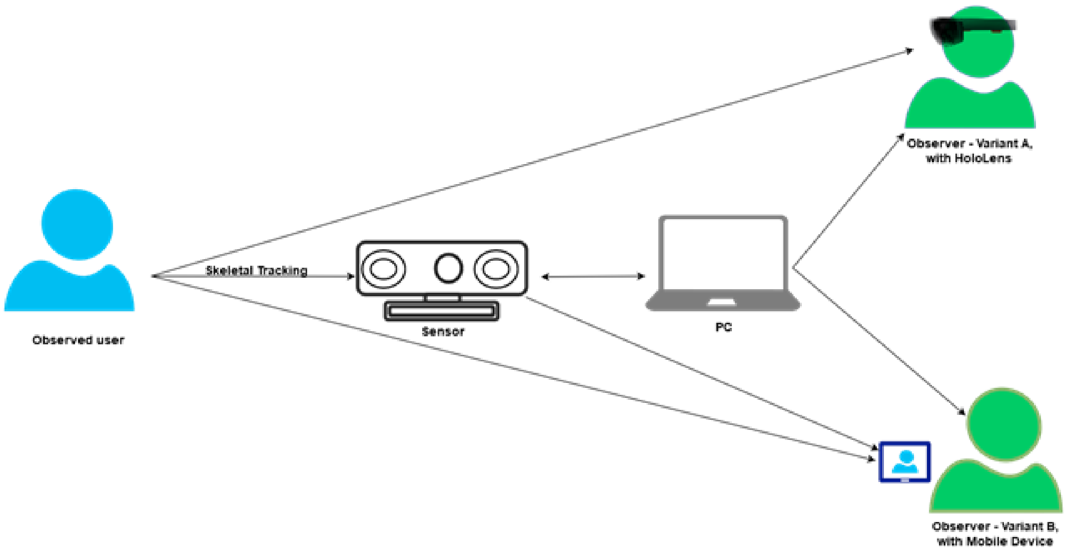
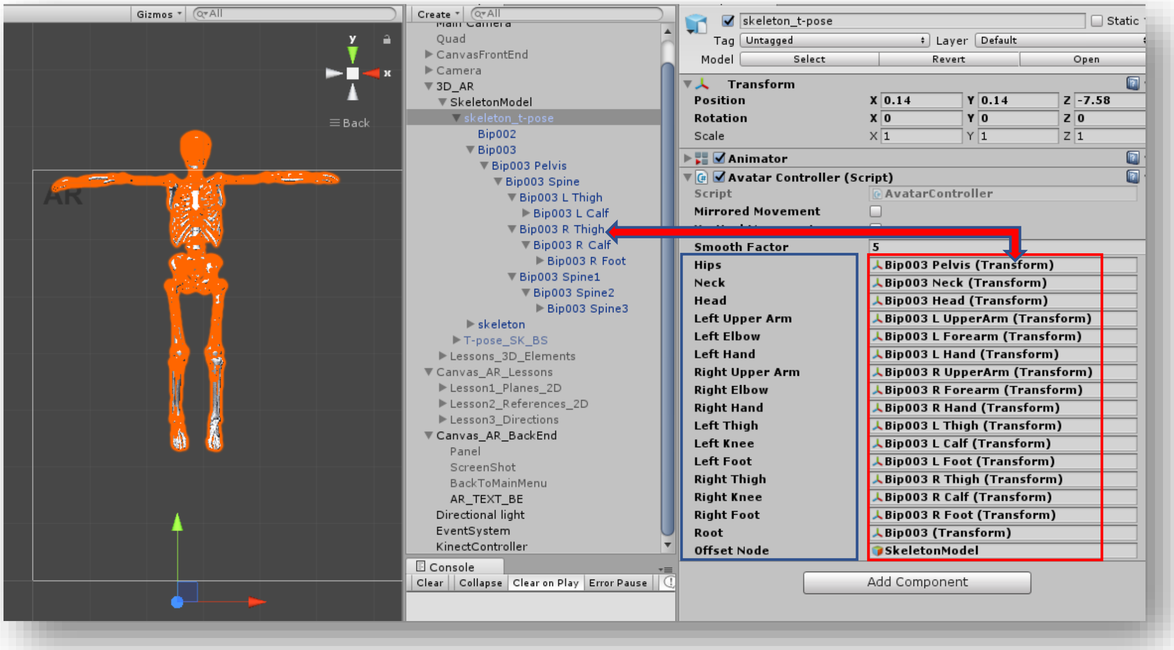


Figure 4.37 – Motion tracking using additional sensors

Two motion tracking scenarios are displayed in the previous figure:

a. One using a wired sensor that needs to be connected to a computer, like Kinect, where the skeletal data should be shared to the display device over network, leading to potentially synchronization issues.
b. One where the skeletal tracking is sent to mobile device (Android or iOS) through a wireless communication, like VicoVR.

In our tests we used a Kinect sensor to animate the human body skeletal system based on the detected movements. If the implementation for TRAVEE project's kinematics module was implemented in mid-late 2015, these tests were started somewhere late 2016. In this time, the dependent unity package (*Kinect with MS SDK*) wasn't available anymore because it became deprecated. Therefore, we found and imported into our Unity solution *Kinect with OpenNI2* package and the scripts were very similar with the previous package. The used 3D model was named *skeleton_t-pose* to which we attached an *AvatarController* script from the mentioned unity package. Each bone of interest that is tracked by the Kinect sensor is in the blue rectangle from the Fig. 4.38. Their references from the 3D model, that are animated accordingly, are set in the red rectangle area. The joints set as references to Kinect script (*AvatarController*) variables are available in the model's skeleton structure visible on the middle side of the figure. The model is composed of two parts: the skeleton and the meshes.



☑ skeleton_t-pose ☐ Static

Tag Untagged Layer Default

Transform

Position	X 0.14	Y 0.14	Z -7.58
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

☑ **Animator**

☑ **Avatar Controller (Script)**

Script	AvatarController
Mirrored Movement	☐
Smooth Factor	5
Hips	Bip003 Pelvis (Transform)
Neck	Bip003 Neck (Transform)
Head	Bip003 Head (Transform)
Left Upper Arm	Bip003 L UpperArm (Transform)
Left Elbow	Bip003 L Forearm (Transform)
Left Hand	Bip003 L Hand (Transform)
Right Upper Arm	Bip003 R UpperArm (Transform)
Right Elbow	Bip003 R Forearm (Transform)
Right Hand	Bip003 R Hand (Transform)
Left Thigh	Bip003 L Thigh (Transform)
Left Knee	Bip003 L Calf (Transform)
Left Foot	Bip003 L Foot (Transform)
Right Thigh	Bip003 R Thigh (Transform)
Right Knee	Bip003 R Calf (Transform)
Right Foot	Bip003 R Foot (Transform)
Root	Bip003 (Transform)
Offset Node	SkeletonModel

Figure 4.38 – Unity scene settings to enable Kinect functionality

After the Kinect scripts variables were set, the animation outcome was tested. Issues were seen, and several adjustments had to be made to the scripts and the models. One of them was the fact that we considered the sensor all the time in *Near Mode* because of the small distance available between the user and the device. Also, the possibility to move the model in space was disabled for Kinect tracking since this generated a large number of graphical issues. The target is to display the model superimposed over the face and the body of a user on the screen. Another issue was the fact that the skeleton model wasn't set in a perfect T-pose stance from the start and we needed to correct this aspect using 3DS Max program. Fig. 4.38 displays in the left side the 3D model that stands in T-pose in the unity scene. Fig. 4.39 and Fig. 4.40 showcase the visual output, while tracking the body movements using a Kinect sensor in an AR setup.

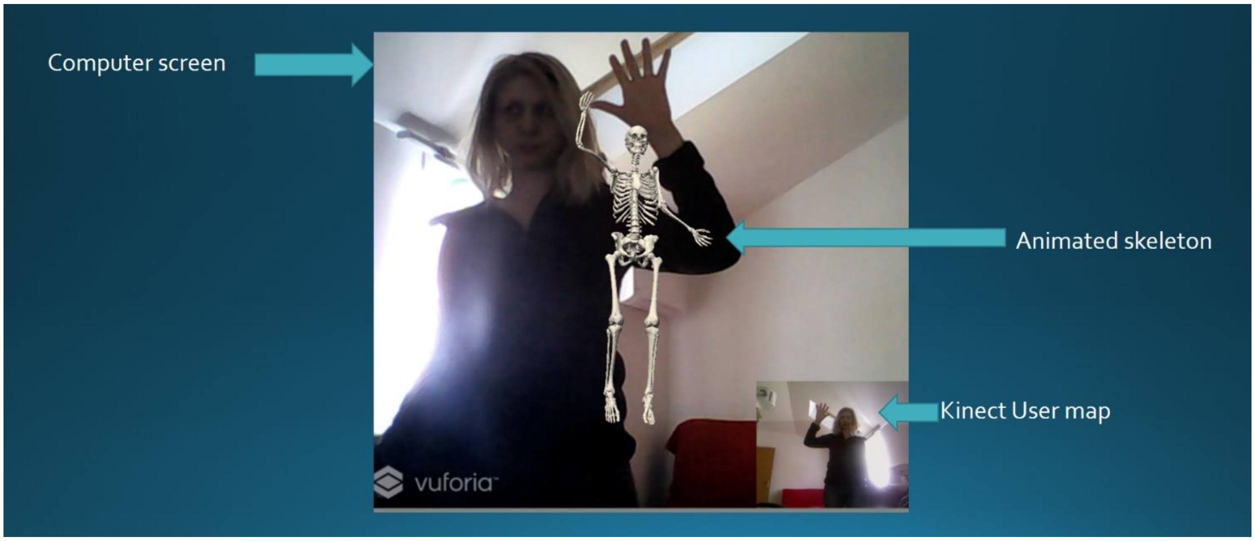


Figure 4.39 – Motion tracking using a Kinect sensor

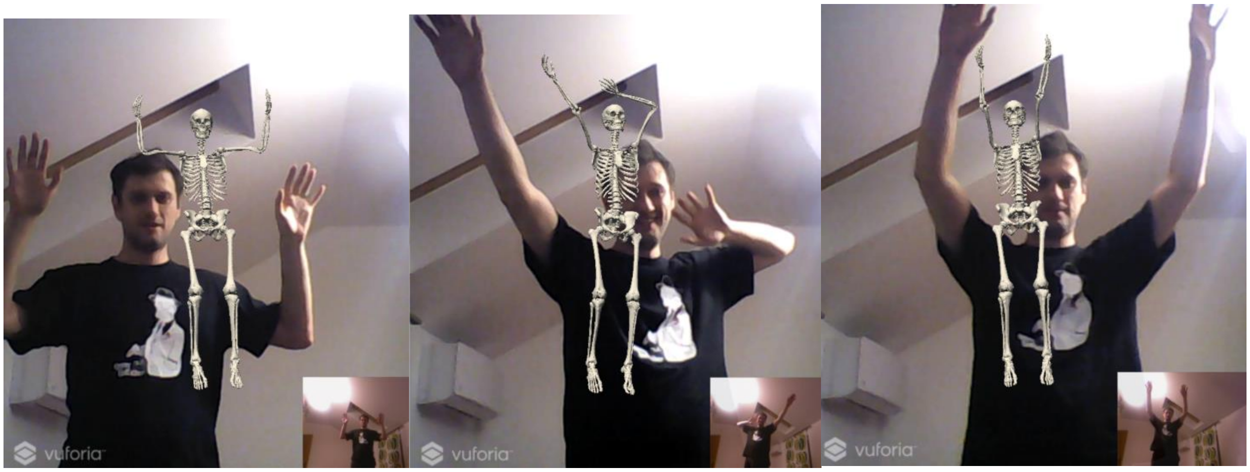


Figure 4.40 – Motion tracking using a Kinect sensor with the mirror movement corrected using the bones generated model

Complementary Tracking Solution Using OpenCV

OpenCV (Open Source Computer Vision Library) was designed for computational efficiency with a strong focus on real-time applications, it is cross-platform and should work on almost any commercial system. The library has more than 2500 algorithms and can be used to detect and recognize faces, identify objects, classify human actions in videos, track moving objects, produce 3D point clouds from stereo camera, follow eye movements and establish markers to overlay it with Augmented Reality, etc.[61].

Although we used Vuforia and Kinect V1 sensor, we later realized that we can use a face detection feature to easily calibrate the virtual models on top of the user's real image. As seen in the previous figures this wasn't achieved in the first tests and we needed something more appropriate. Since the application was developed using Unity, OpenCV functionality needed to be brought in. Unfortunately, as opposed with the other technologies, OpenCV wasn't included or offered via a free third-party package. We found the package *OpenCV for Unity* and decided to acquisition it to speed up the development process and to focus on the application's functionality as Unity was already used for VR and AR development and real-time body tracking. Vice-versa, the solution that included the classical OpenCV support[62] would have slowed down our progress regarding the support and functionality for the other components. *OpenCV for Unity* package is a clone of *OpenCV Java* version 3.2.0 and has support for Android, iOS, WebGL, Windows Store Apps 8.1, Windows10 Universal Windows Platform and supports preview in Unity Editor.

After the *OpenCV for Unity* package was imported, *WebCamTextureDetectFaceExample* Unity scene files were used as a starting point. We imported the obtained 3D models into the scene (Fig. 4.41) and changed their position at runtime to fit in the center of the boundaries generated by the face detection algorithms (Fig. 4.42). Following this, the model was scaled at runtime based on the detected bounds (Fig. 4.43). This was accomplished using a decoy model of the head that was substantially optimized to get to a very small number of vertices (~440). The scaling was set as a ratio between the face detection boundaries and the head model bounding box. As a next step, the Kinect sensor functionality was added to this scene to animate the 3D models joints according with the tracked data. The obtained visual results are available in Fig. 4.44.

---

[61] https://opencv.org/about.html

[62] https://opencv.org/releases.html

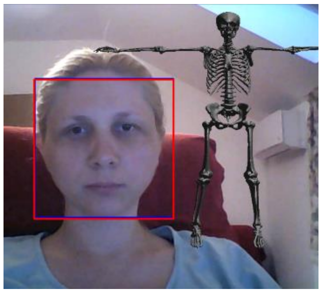


Figure 4.41 – Model position changed at runtime on Laptop.

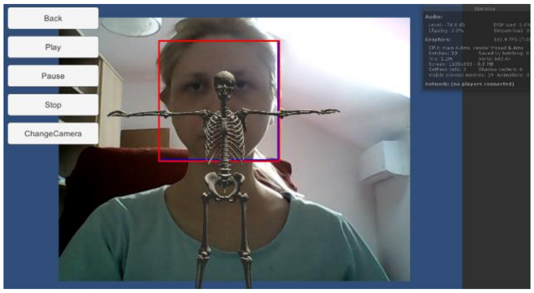


Figure 4.42 – Model position changes at runtime and was centered to face detection rectangles on Laptop.

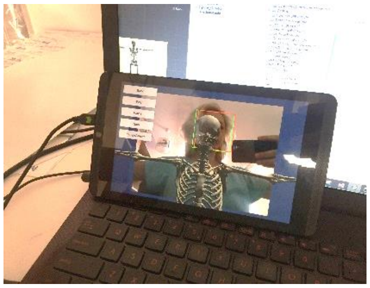


Figure 4.43 – Model position and scale changed on Nvidia Shield tablet.

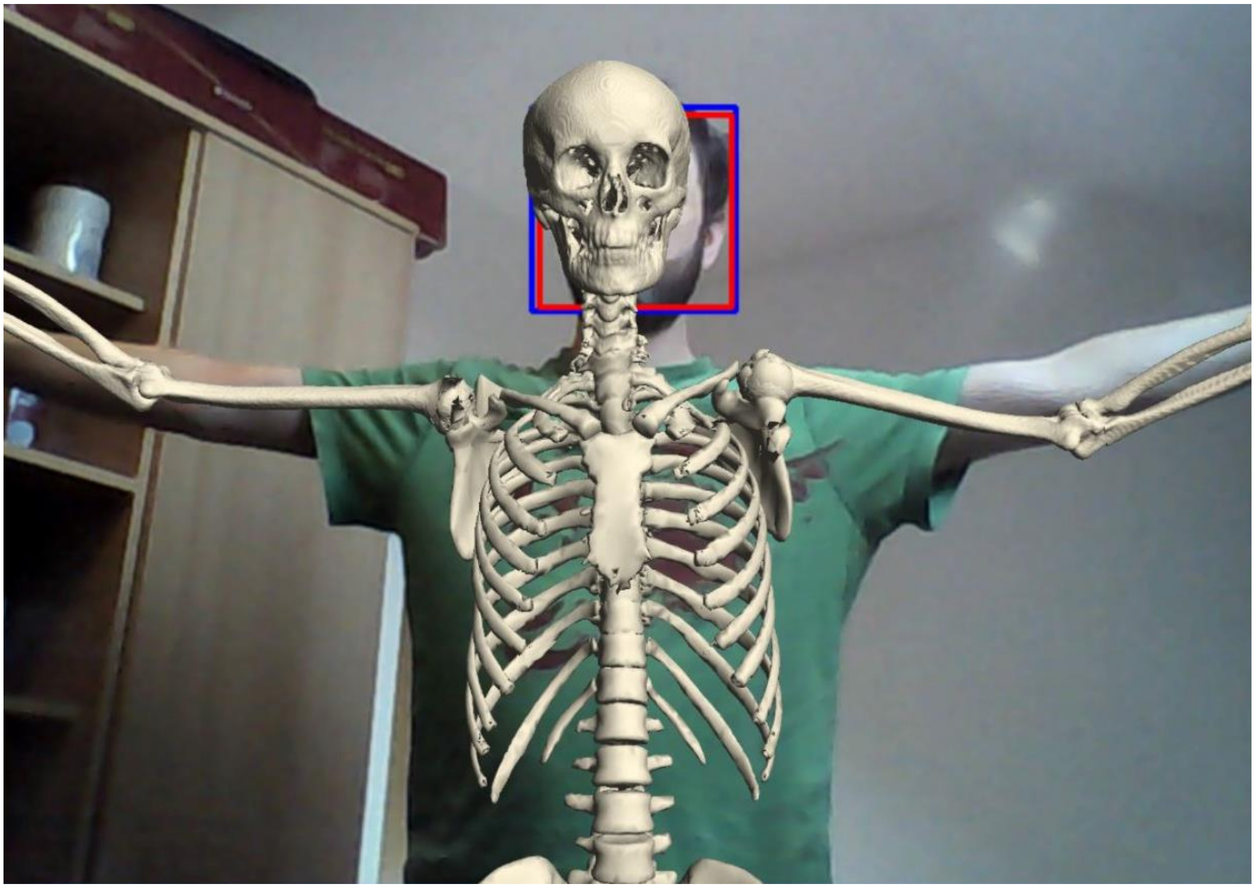


Figure 4.44 – Motion tracking of an observed user in AR using a Kinect sensor and OpenCV

Various tests were performed with a more appropriate shading technique for the loaded models to offer a semitransparent overlay on the video stream. Figures 4.45-4.47 showcase the visualization differences between the initial setup and the semi semitransparent overlay of the bones and muscles models.

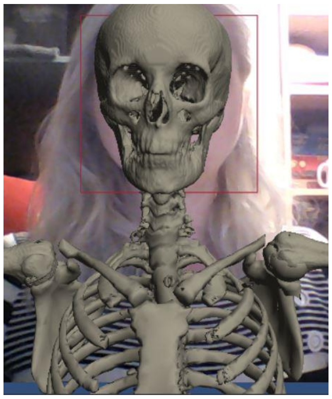


Figure 4.45 – Opaque model

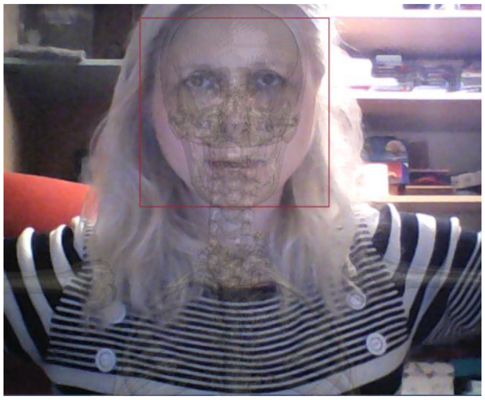


Figure 4.46 – Semitransparent bones model

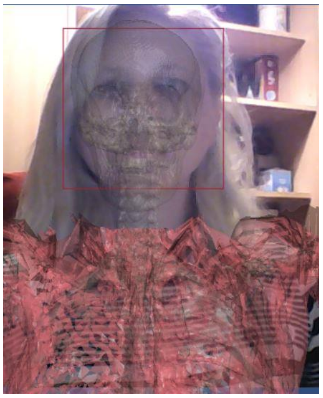


Figure 4.47 – Semitransparent bones and muscles models

To detect the objects in the video stream, OpenCV uses cascade classifiers; two classifiers are available for use: *Haar* and LBP (Local Binary Patterns). OpenCV already contains pre-trained classifiers for face detection which are stored in XML files that are loaded at the initialization step of the application. LBP and Haar detection quality depend on the quality of the training dataset and the training parameters. It's possible to train an LBP-based classifier that will provide almost the same quality as Haar-based one[63] although LBP shows better performance in terms of detection time and is more suited to mobile platforms as it performs better under limited resources [SG15]. We used both Haar and LBP classifiers and observed that in different lighting conditions LBP had some tracking issues but it was much faster, with the same training data for both classifiers. We tracked the performance data from both cases using the Unity profiler and we observed that the CPU usage was at 6-7ms for LBP while for Haar it stayed somewhere at 11-12ms. OpenCV offered us the opportunity to display the virtual models on top of the user's image more realistically as this wasn't achieved using only motion tracking sensors. However, after the application was tested, we realized that the setup wasn't as user friendly as we initially predicted and the benefits of using the motion tracking as the main AR setup wouldn't give the best results for the moment.

4.2.4 Interactive Biomechanics Lessons (IBL) project

The IBL project consists of 4 separate applications:

a. *VR with a classroom background* that uses predefined biomechanics lessons;
b. *VR without a closed environment (no classroom background)* to overcome the cybersickness – as from experimental tests it was noticed that this element improved the simulator sickness symptoms;
c. *AR marker-based* that uses the same lessons from VR into an augmented reality environment;

---

[63] https://docs.opencv.org/3.1.0/dc/d88/tutorial_traincascade.html

d. *AR markerless* that uses motion tracking sensors to animate in real time the virtual 3D models superimposed over the user' image.

The VR applications were displayed on a low-cost HMD composed of a Samsung S6 device, model SM-G920F, with a Google Cardboard viewer, initially, and later with a Gear VR viewer. The built scenes were developed using Unity version 5.6.0b.10. The cardboard viewer functionality was included in that specific Unity version and no additional packages were required to add the functionality for VR display with it. We later added *VR Samples* package while making the applications compatible with Gear VR.

The AR applications had separate setups, as the markerless one included additional tracking sensors while the marker-based one had a simpler structure, similar with the VR ones. Vuforia was used for the AR marker-based specific features and, in order to include its functionality without additional support packages, we used Unity version 2017.2. The AR markerless application was built using Unity version 5.6.0b.10. The setup contains a Kinect V1 sensor for motion tracking that is connected to a Laptop with this configuration: CPU: Intel Core i7 4710HQ @ 2.50 GHz, GPU: Nvidia GeForce 850M, Memory: 8GB with Windows 10 Pro OS. The tracking is complemented with OpenCV face detection functionality to improve the superimposing of the 3D models over the user's image. The functionality is brought into this project using *OpenCV with Unity* package while the Kinect functionality is imported from *Kinect with OpenNi2* unity package. The final prototype of this scenario was functional on the mentioned laptop while several tests were done with OpenCV on mobile devices such as a smartphone and a tablet. The smartphone is the one used in the VR setup while the tablet is NVIDIA Shield tablet with a screen resolution of 1200x1920 pixels, 2.2GHz quad-core and Nvidia Tegra K1 graphics card running with Android 6.0.1 OS. These setup details are correlated with the performance data presented in section 4.2.6.

4.2.5 Implementation Details

This section presents the implementation details of the developed applications. As we already mentioned, the AR marker-based one and the VR ones have a similar setup and are based on predefined biomechanics lessons while the AR markerless was kept more at a prototype status due to the intermediary results obtained. Since the AR markerless setup was consistently described in a previous section (4.2.3.2) in the following lines we'll focus on the implementation details of the other three scenarios which have a common background.

The pre-defined lessons can be considered a proof of concept for the targeted subject and further additions can be applied. The main applications contain 4 simple biomechanics lessons that cover different areas:

- Lesson I contains information about the human anatomy.

- Lesson II displays interactively the hypothetical anatomical planes that are used to transect the human body.
- Lesson III displays 6 reference points situated on wrist, elbow, shoulder, hips, knee and foot joints and their position versus their neighbors relative to the center of the body.
- Lesson IV presents basic movements such as: *flexion/extension, adduction/abduction* and *pronation/supination* and highlights the involved active muscles.

Each lesson corresponds to a screen and, along these 4, there are three additional ones: one for presenting the 3D models of human anatomy, one for the intro and one for the end. The interaction and visualization methods are different in AR and VR, and each of them will be detailed in the following subsections.

4.2.5.1 Virtual Reality

Initially the users are welcomed to the application and using the reticle they can interact with it. The start text as shown in Fig. 4.48 is animated to attract users' attention and when the reticle is overlapping the text the user can navigate to the next screen. This was implemented in this manner for the users to be accustomed with interaction method before the actual lessons begin.

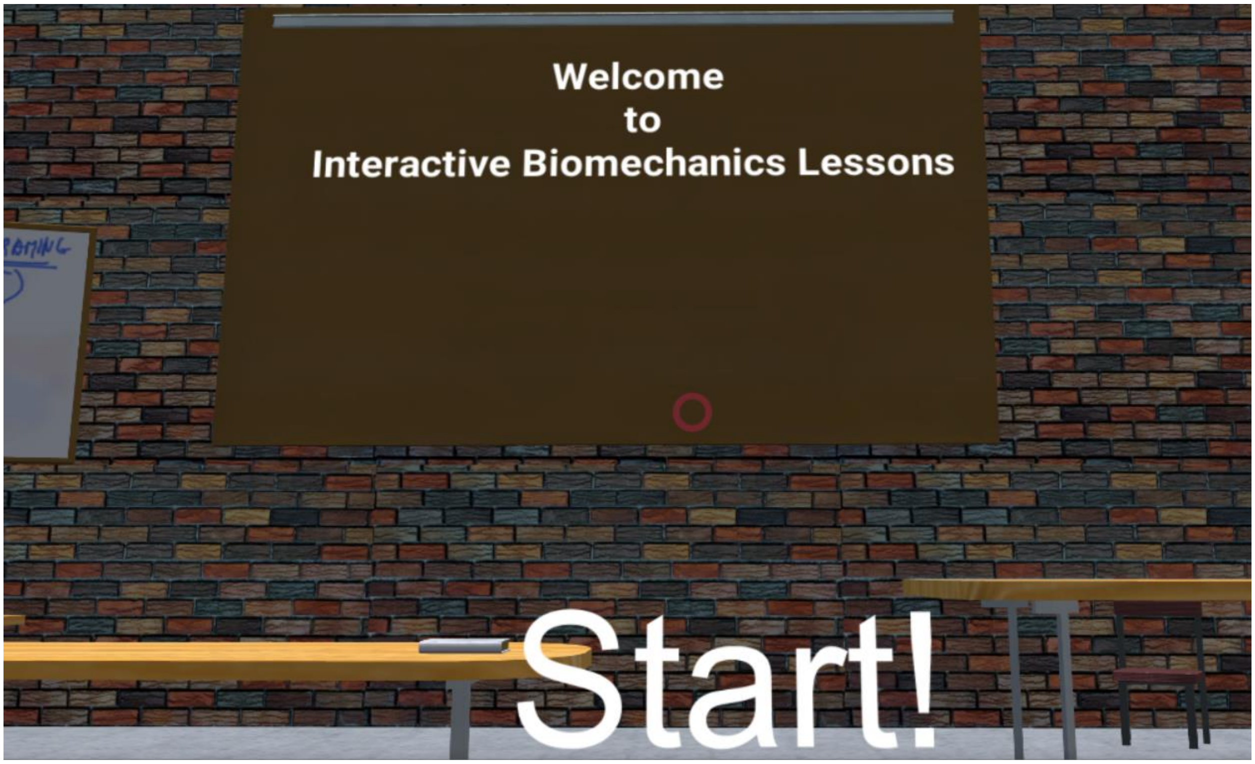


Figure 4.48 – VR classroom application – Initial screen

The VR classroom application contains a few key elements such as: a camera, the 3D models of musculoskeletal system, the reticle component, the 3D elements which compose the classroom environment and the lessons' game objects. Fig. 4.49 showcases the application's structure as seen in Unity. These components are common with the other VR application where the main difference consits in the discard of the 3D elements which create the classroom.

Figure 4.49 – VR classroom application structure from Unity

Models Presentation

The applications contain a preliminary screen where the users get accustomed with the imported 3D models. In VR, the users can switch between bones, muscles and skin models by directing the reticle over the desired option. The text changes its color in real-time when the option is selected and to detect it we used the ray casting method. Each 3D Text that it interacts has a capsule game object around it. The capsules have assigned a special material that has alpha color set to zero so that they are not visible. More details are available in Fig. 4.50. The 3D Text changes its color when the ray hits the corresponding capsule object (Fig. 4.51). Even though the applications were displayed with stereoscopic rendering, the following screenshots presented for VR are taken while running the application with Unity Editor to offer a clearer image with the contained elements.

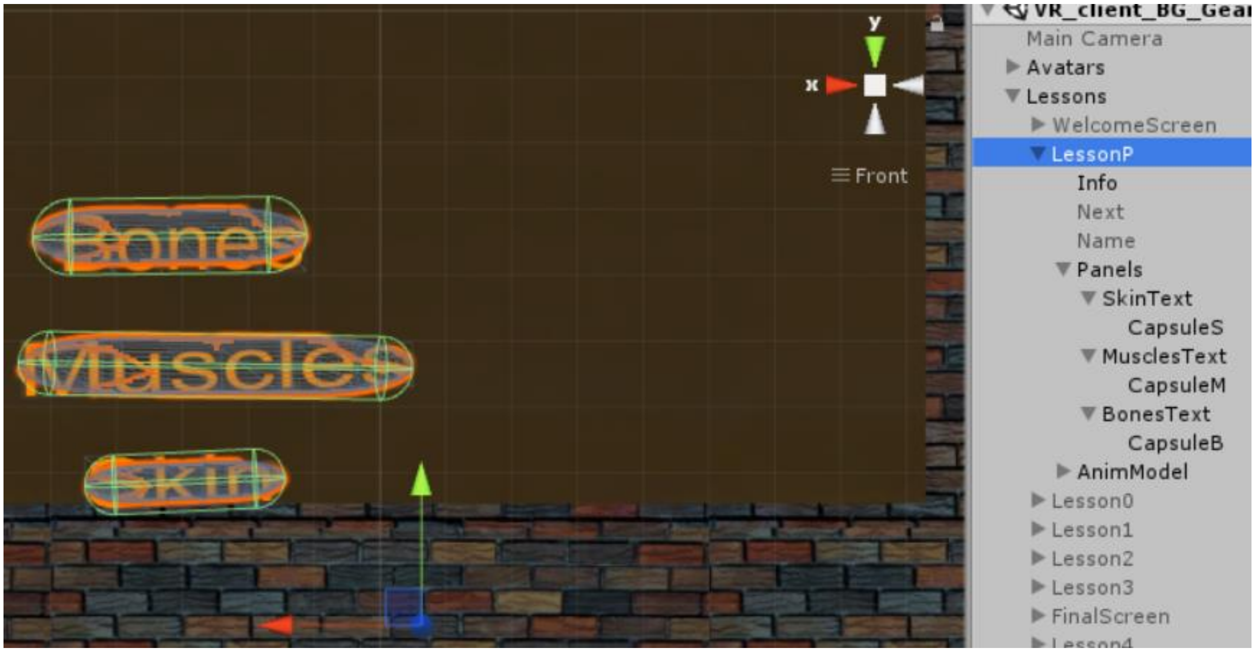


Figure 4.50 – VR classroom application - Imported models presentation menu structure

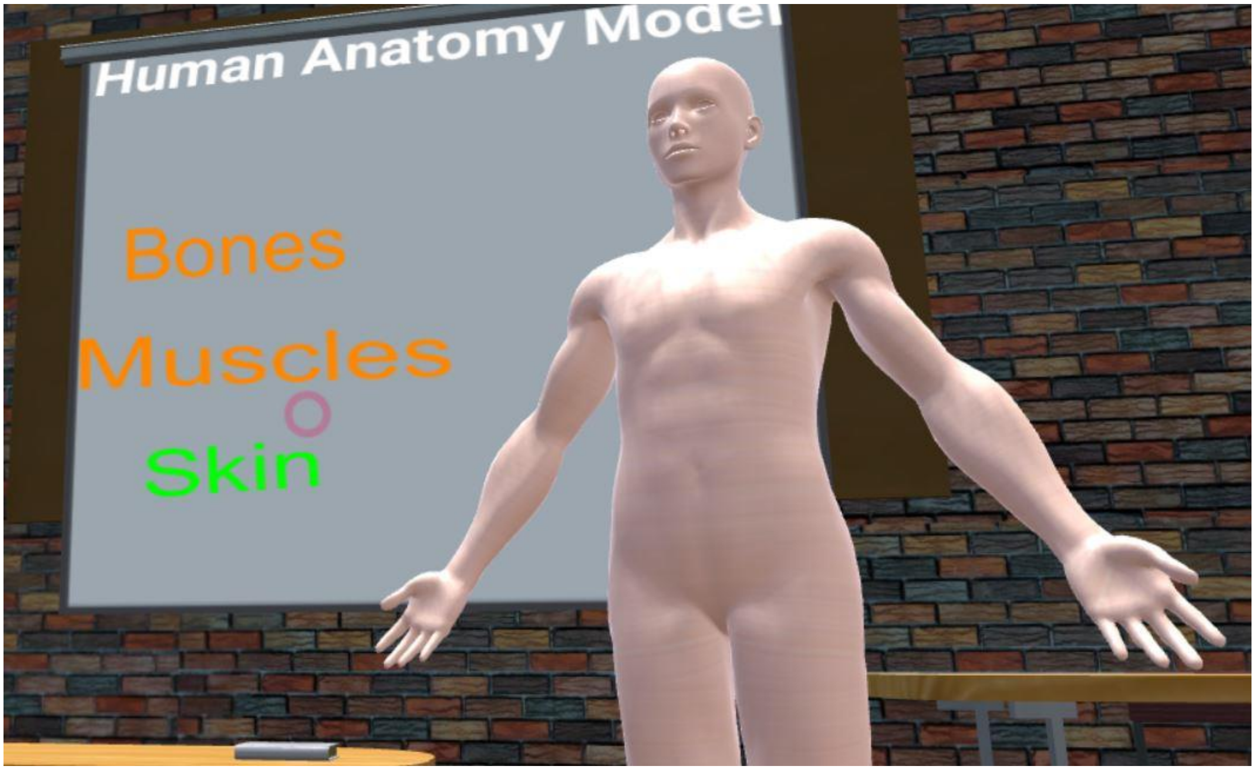


Figure 4.51 – VR classroom application - Imported models' presentation – Skin

The skin model is rendered by default when entering this section and all three models can be observed. Initially, the models were updated when selecting an option by making the current model's game object active (e.g. skin) while making the other ones inactive but we observed some consistent framerate drops during this operation. We solved this problem by making all three models (skin, muscles, bones) active all the time but placing the models that weren't of interest outside the FOV at a remote location. The users could rotate the 3D models to observe them all around by selecting the rotation icon available at the bottom of the model (Fig. 4.52).

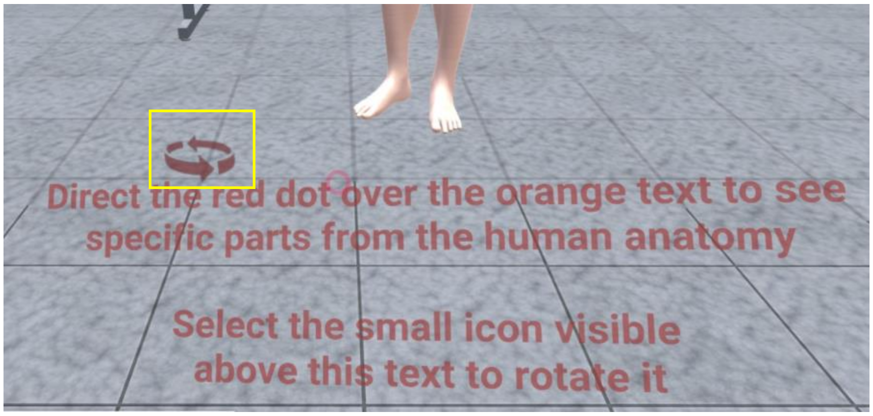


Figure 4.52 – VR classroom application – Rotation option (yellow rectangle)

Figure 4.53 and 4.54 show the muscles and bones models as visible in the VR classroom application. The 3D elements that are displayed in the VR classroom application are imported from *Props for the Classroom*[64] package from Unity Asset Store.

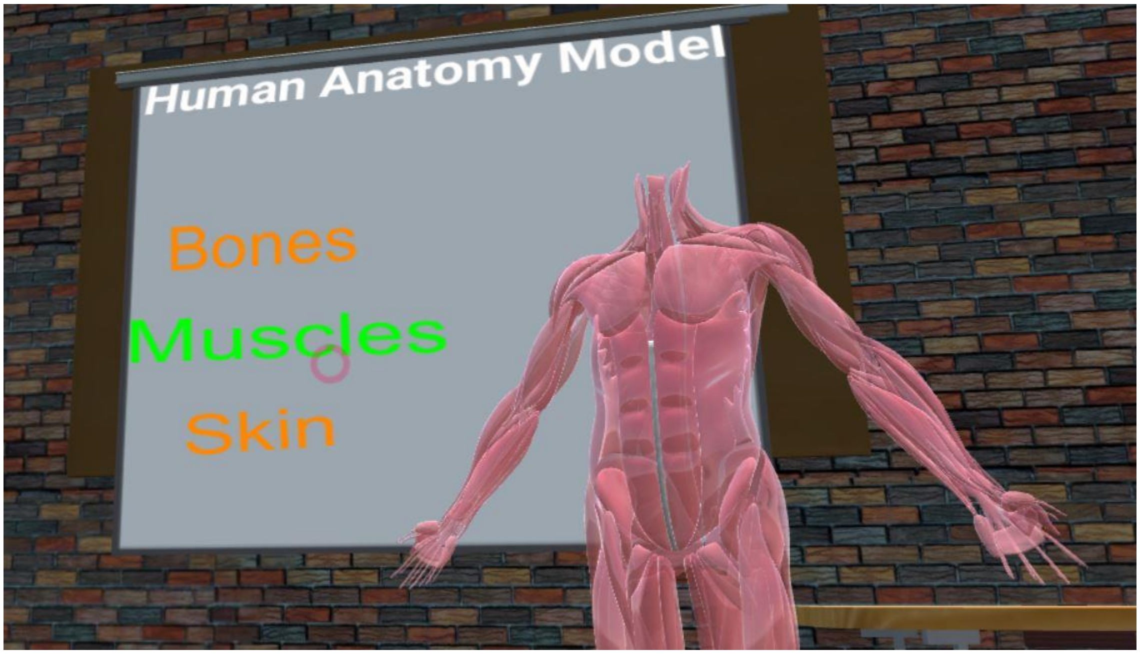


Figure 4.53 – VR classroom application - Imported models' presentation – Muscles

---

[64] https://www.assetstore.unity3d.com/en/#!/content/5977

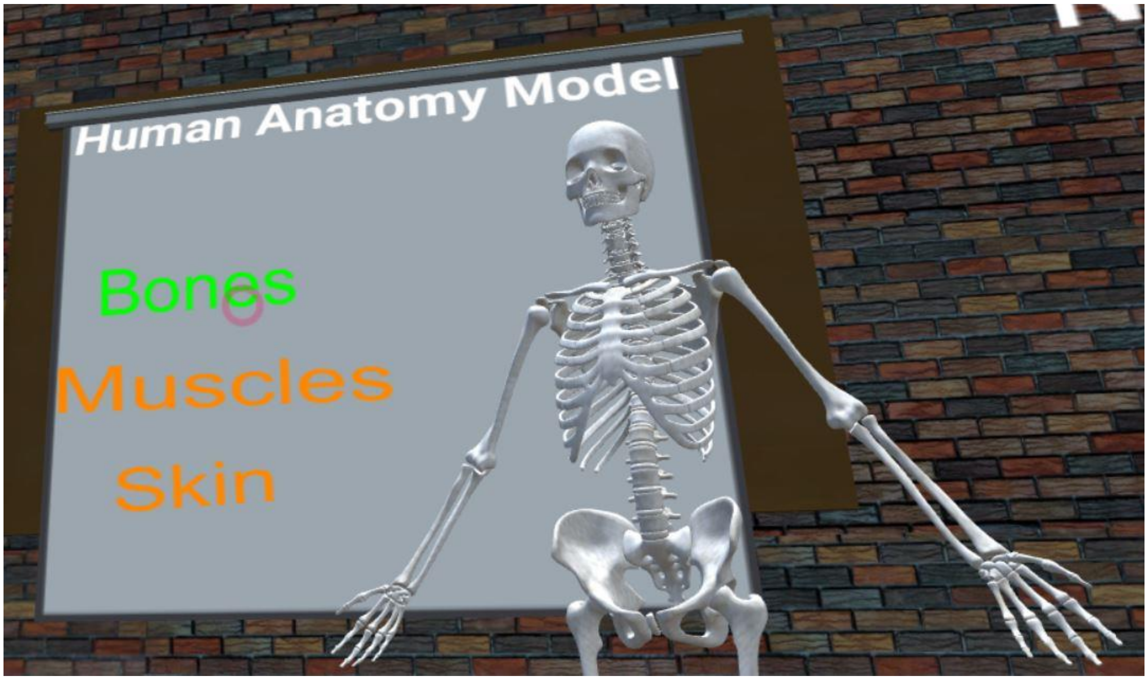


Figure 4.54 – VR classroom application - Imported models' presentation – Bones

Lesson I – Anatomy

The first interactive biomechanics lesson subject is the anatomy of the musculoskeletal system. The user can examine the bones and muscles models to discover their individual elements. Figure 4.55 and 4.56 show an example of information provided for various bones. Similar behavior is available for the muscles as the user can select to view the muscles or bones models by directing the reticle over the bottom icon as displayed in Figure 4.56 and 4.57 (yellow rectangle). This lesson's content was one of the reasons we continued searching new models besides the ones obtained from medical images. Each bone and muscle are currently represented in individual meshes (vs only one for the other models) and we could display better each element of interest. Each mesh is colored with green when the reticle is placed over it and the white board information is updated accordingly as it can be observed in the following images.

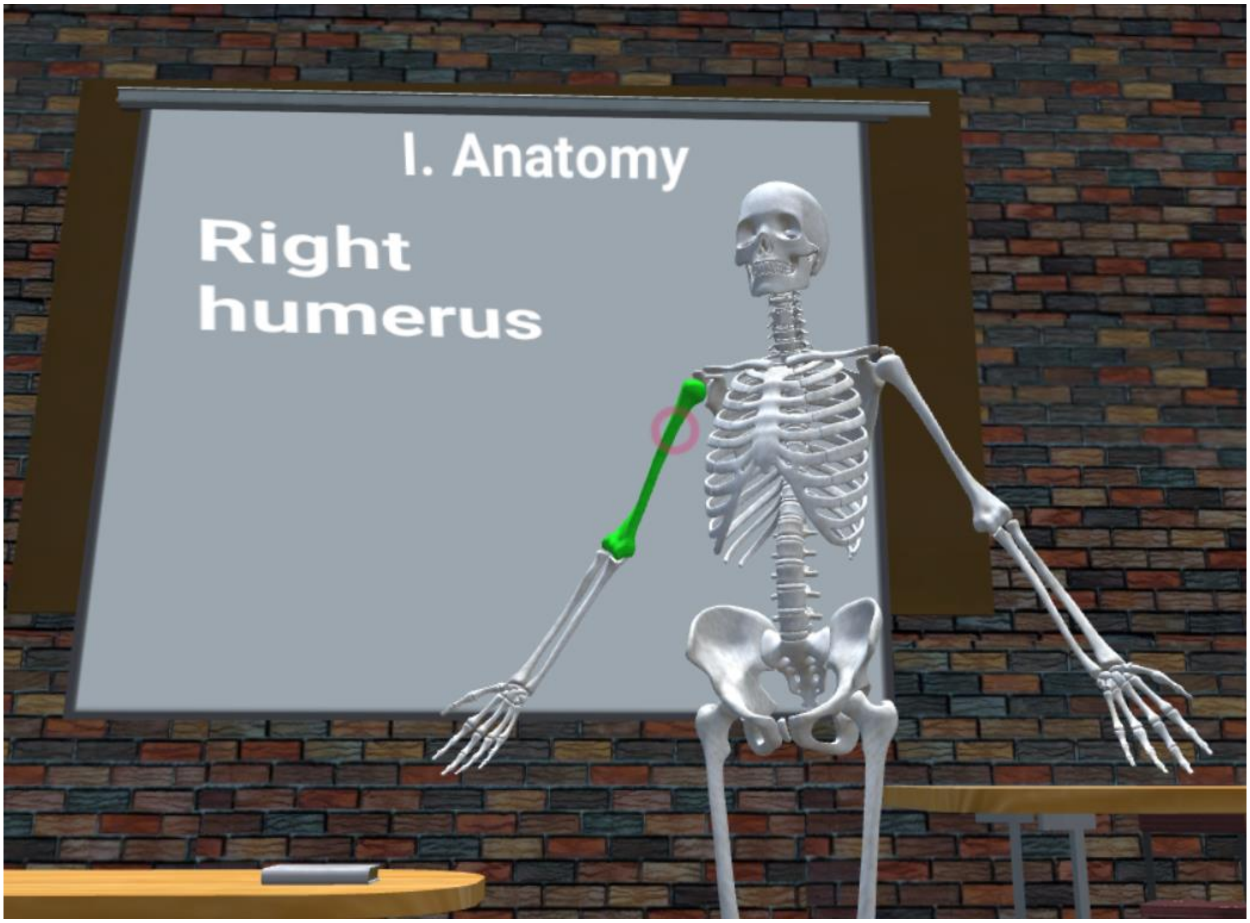


Figure 4.55 – VR classroom application – Anatomy Notions – Humerus

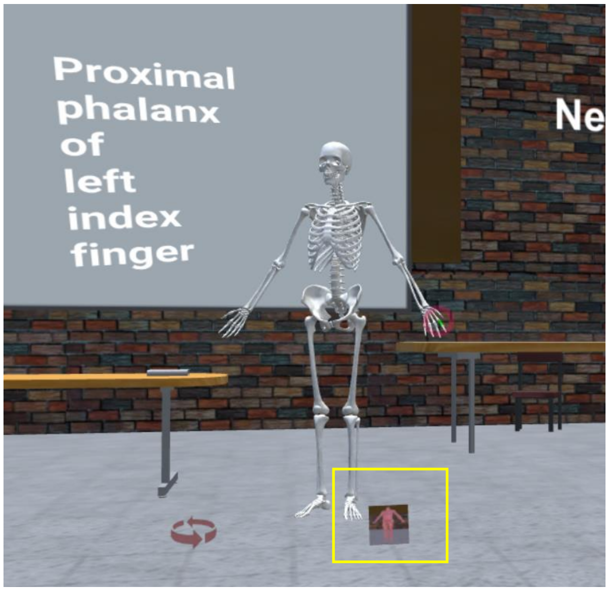


Figure 4.56 – VR classroom application – Anatomy Notions – Bones

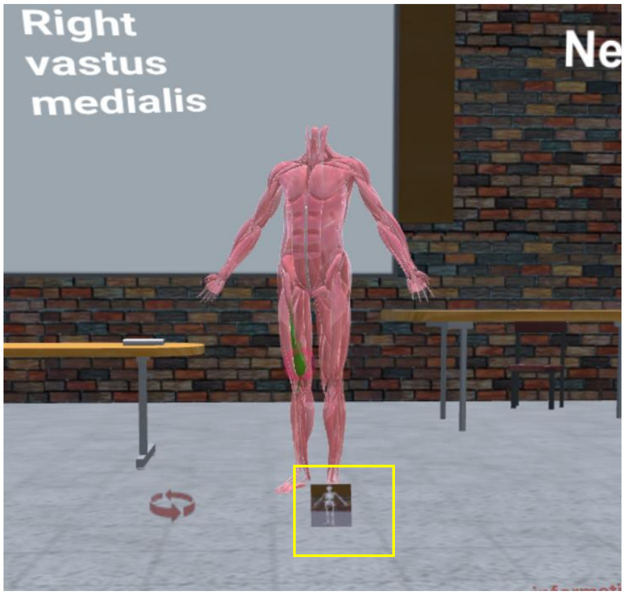


Figure 4.57 – VR classroom application – Anatomy Notions – Muscles

Lesson II – Axes

The second lesson contains information regarding the hypothetical anatomical planes: *traverse*, *sagittal* and *coronal*. Each plane's name is a 3D text that has similar functionality with the models' presentation where an invisible capsule is placed over a 3D element. We have chosen this implementation to determine which plane is selected at runtime when the reticle is directed to it. If one of the planes is selected, then another element placed correspondingly in the 3D space is animated over the displayed model as long as the pointer is placed over the pointed text. The *transverse*, *sagittal* and *coronal* planes are translated in the world's space on its corresponding axis (X, Y or Z). Only one plane can be rendered and animated at a time. Figures 4.58, 4.59 and 4.60 show these anatomical planes as they are rendered in the VR classroom scenario.

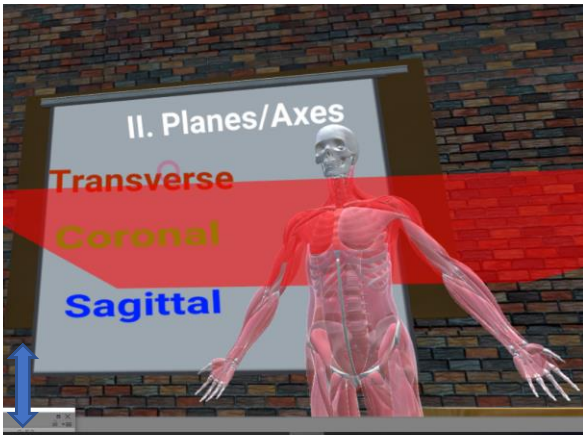


Figure 4.58 – VR classroom application – Transverse

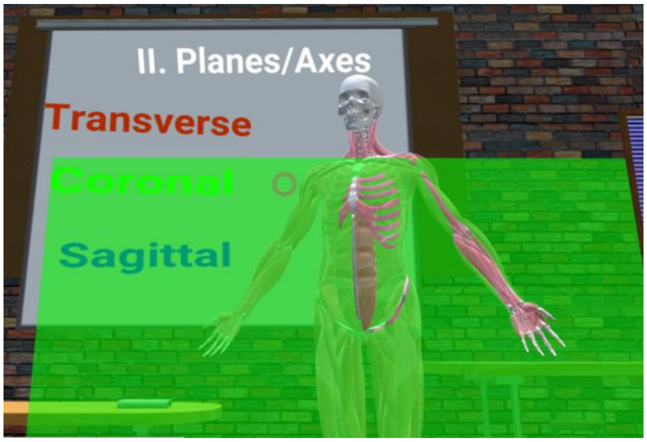


Figure 4.59 - VR classroom application – Coronal Plane

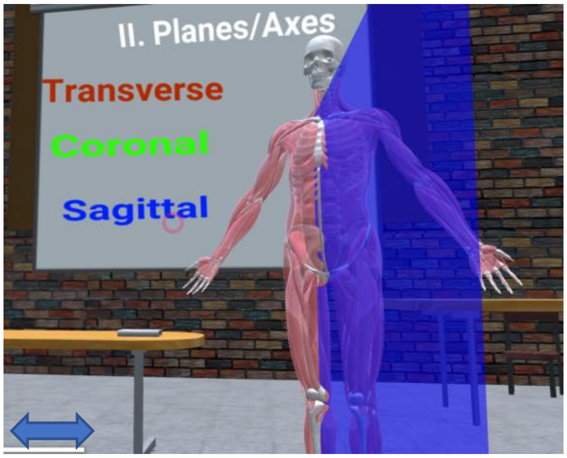
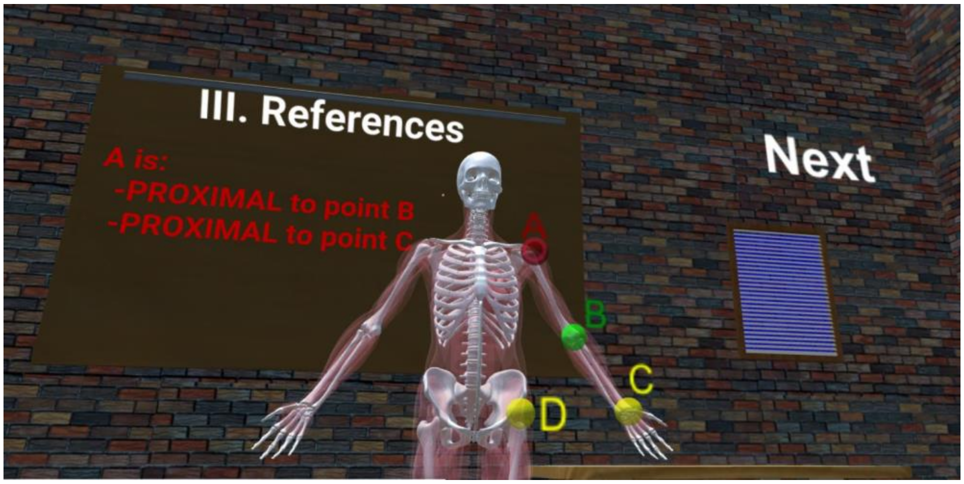


Figure 4.60 – VR classroom application – Sagittal Plane

Lesson III – Reference Points

The third lesson displays the 3D models of the bones and muscles systems with 7 additional elements placed over them. The new elements are composed of a human heart model and six spheres where each sphere represents an individual point in the 3D space. The heart model is a reference to the body's center since the points locations (proximal or distal) are based on it. Ray casting technique is used as well as a method of interaction within the lesson and each time the reticle is superimposed over a certain point, a message with its location is displayed on the board. No invisible capsules were required this time since we used the spheres as ray hit references. The points were positioned on the main joints of the upper and lower limbs: wrist, elbow, shoulder, hips, knee, foot. To be able to see better the points over the bones and muscles of the 3D model, the muscles had assigned a slightly more transparent material compared with previous lessons. Figure 4.61 and 4.62 display two examples of the interaction method and the displayed messages regarding the selected reference points location.



III. References

A is:
-PROXIMAL to point B
-PROXIMAL to point C

Figure 4.61 – VR classroom application – Reference Points – Example A

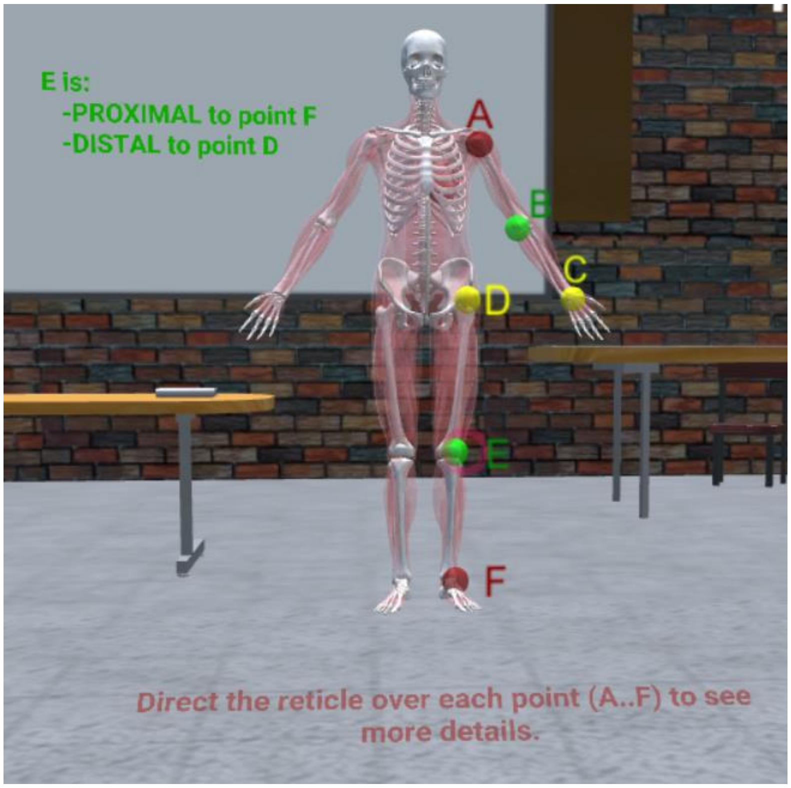


Figure 4.62 – VR classroom application – Reference Points – Example B

Lesson IV – Movements

The fourth lesson exemplifies some simple movements like *flexion/extension*, *adduction/abduction* and *pronation/supination.* They are in a group of two since one is complementary with the other. For example, one movement starts with abduction and ends with adduction (presuming the body is in neutral position). The displayed 3D model is animated differently compared with the previous tests where we used additional tracking sensors as the system received the joints positions and animated the model based on the recorded data. This time, we used **Mecanim** tool from Unity and created a small set of animations and at runtime the active muscles involved in the selected movements are highlighted with different colors for a better observation.

Figure 4.63 displays the animation states for each option as set in Unity. This animator setup was attached to the rigged human anatomy model. For example, by default the model is not moving and for flexions/extension we have 2 separate animations active: one for flexion and the other one for extension and depending on which one is played a different set of muscles are showcased.

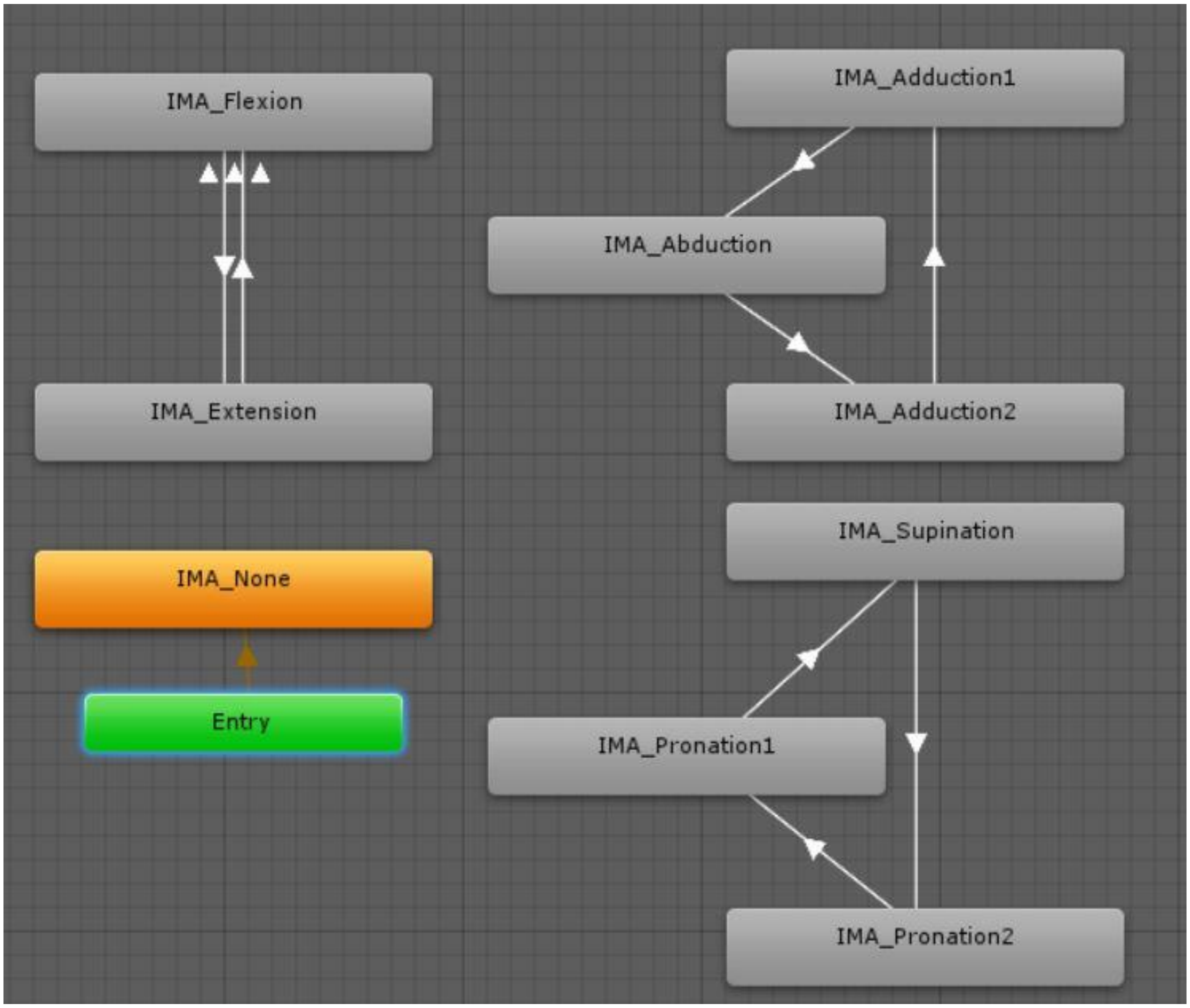


Figure 4.63 – Movements animations states

Adduction/abduction and pronation/supination have 3 animations. The first and the last ones are complements, since the joints are brought into the same state as before the animation started, to avoid any potential issues. This would have been avoided also if the body would have been posed in neutral position to be able to make a complete movement from start. However, considering the content of these biomechanics lessons it was more appropriate to be kept in A-pose for a more facile access to visualize various muscles and bones. Figures 4.64, 4.65 and 4.66 show examples of each movement type.

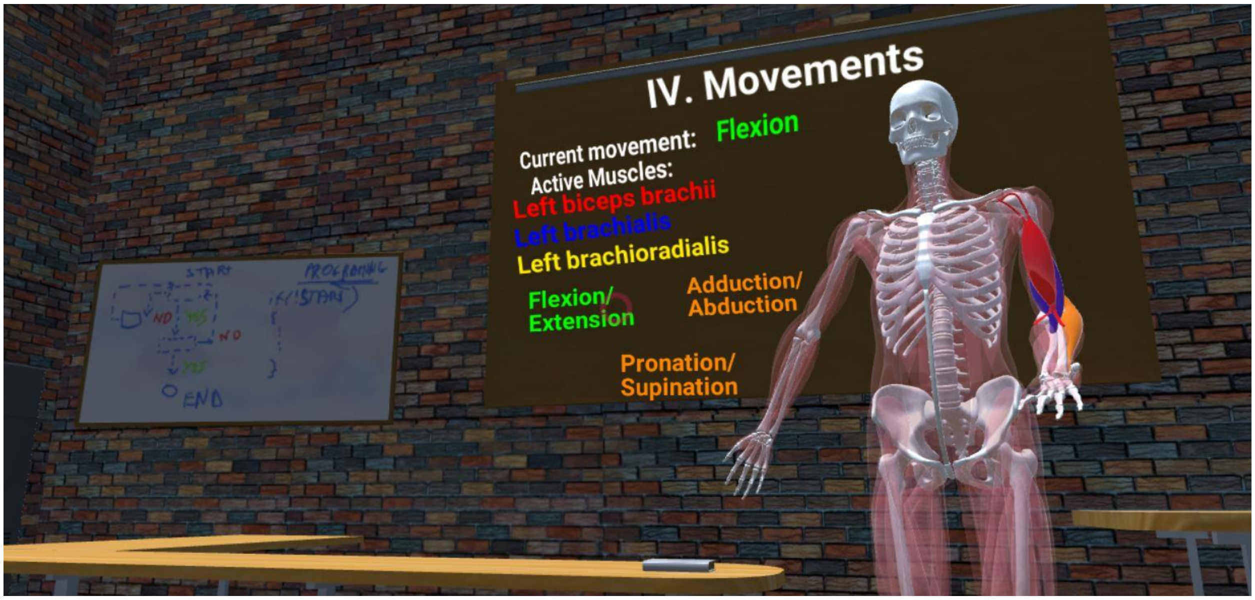


Figure 4.64 – VR classroom application – Flexion/Extension movement example

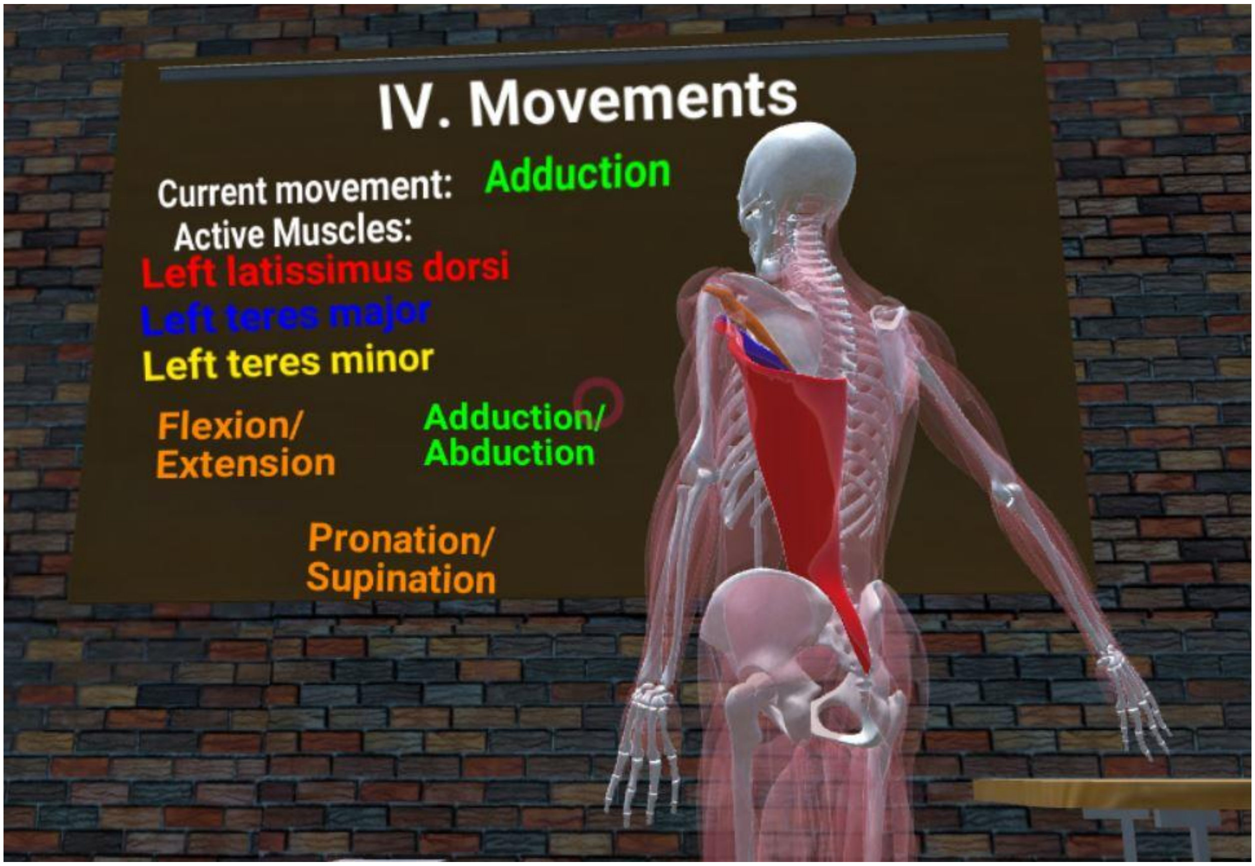


Figure 4.65 – VR classroom application – Adduction/Abduction movement example

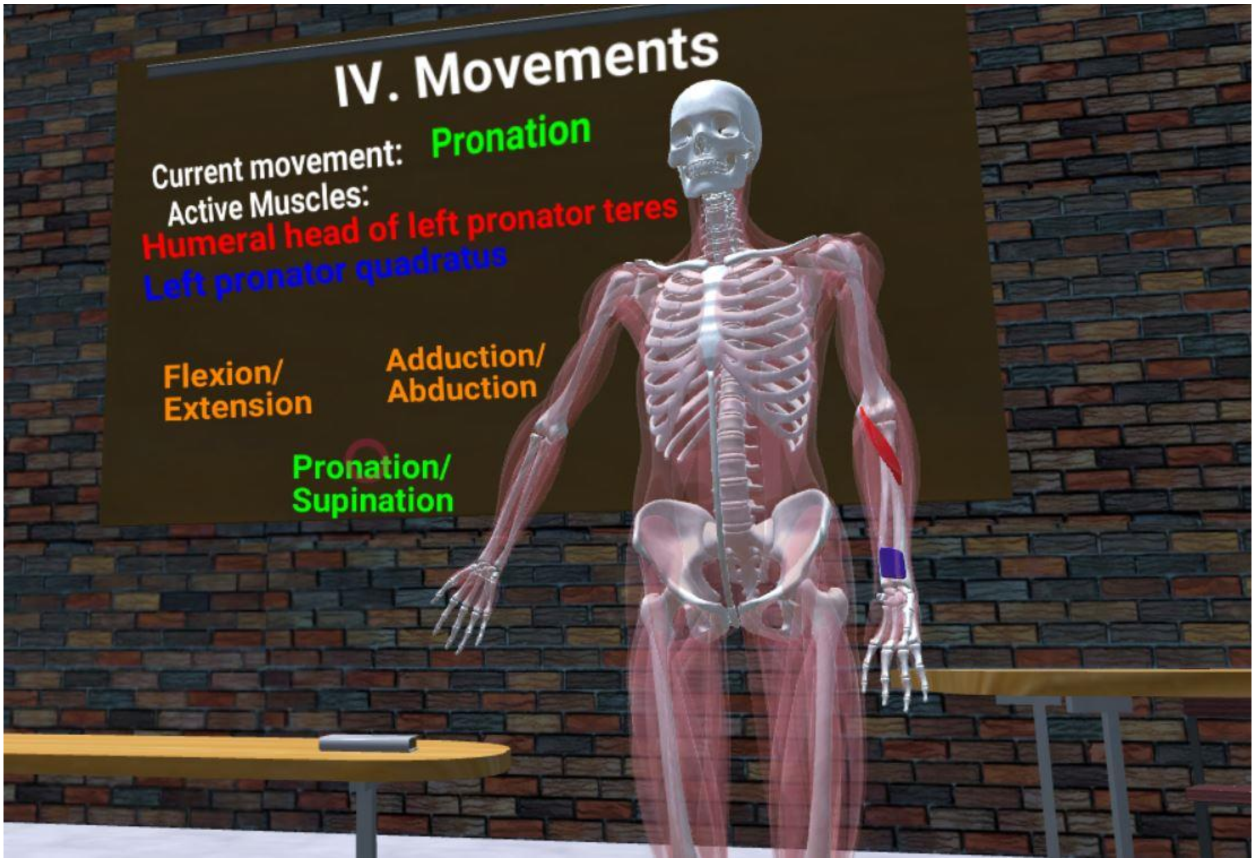


Figure 4.66 – VR classroom application – Pronation/Supination movement example

Improving Cybersickness

During the implementation, the VR classroom application was tested to make sure the 3D elements appeared and behaved as they were designed. During these tests we observed that we had simulator sickness symptoms, like general discomfort, fatigue, eye strain, headache and nausea [SD14]. Initially, the symptoms were obvious during the visualization of some z-fighting issues that were present in the virtual classroom. We noticed that we didn't felt as sick when we tested various VR applications where the view was into an open environment, without closed spaces, and decided to implement a similar scenario. The symptoms were improved after we proceeded to remove the background elements from our VR classroom scene. Even though we fixed the z-fighting related issues we decided to keep in parallel this scenario to assess the cybersickness conditions on both cases as VR experience is very subjective. In the virtual classroom and the one without a closed environment the interactive elements were placed differently, each of them with a suited approach. For example, for the virtual classroom, the data was displayed over the board/projector to resemble as much as possible with a real classroom, while in the other scenario we placed the items to be easy noticeable, at a close distance of the character to minimize the head movement.

In any case, both scenarios used VR specific design options as we integrated the user interface in the virtual 3D scene to improve the presence and immersion of the user in the virtual world. Figures 4.67-4.71 display a few examples from the second VR scenario that was named VR BlueSky due to the background color.

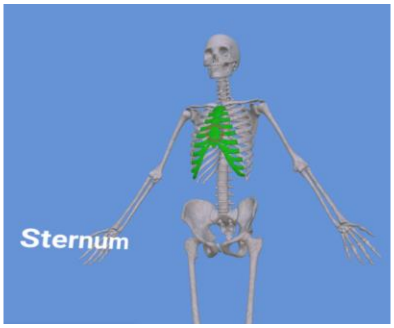


Figure 4.67 – VR BlueSky – Anatomy notions - Bones

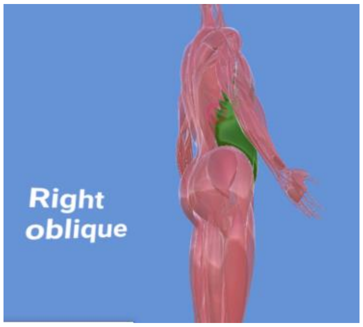


Figure 4.68 – VR BlueSky – Anatomy notions - Muscles

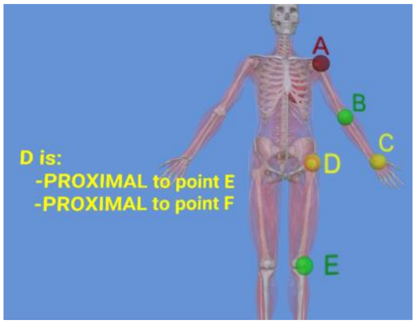


Figure 4.69 – VR BlueSky – Reference points

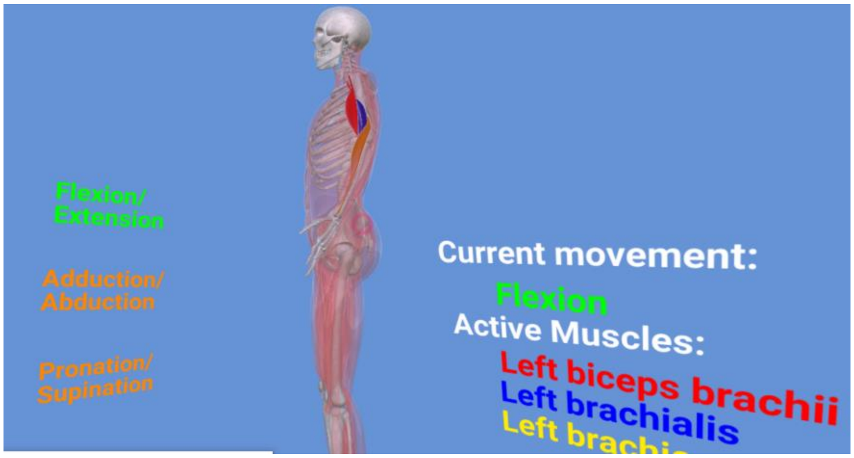


Figure 4.70 – VR BlueSky – Movements - Flexion

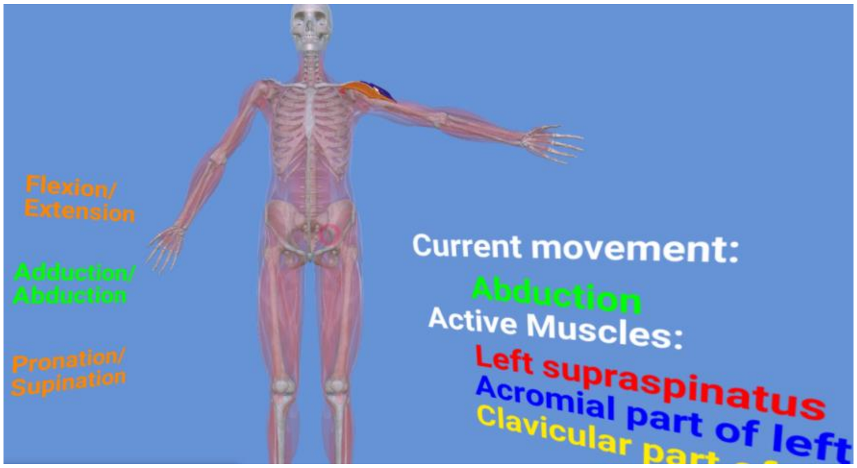


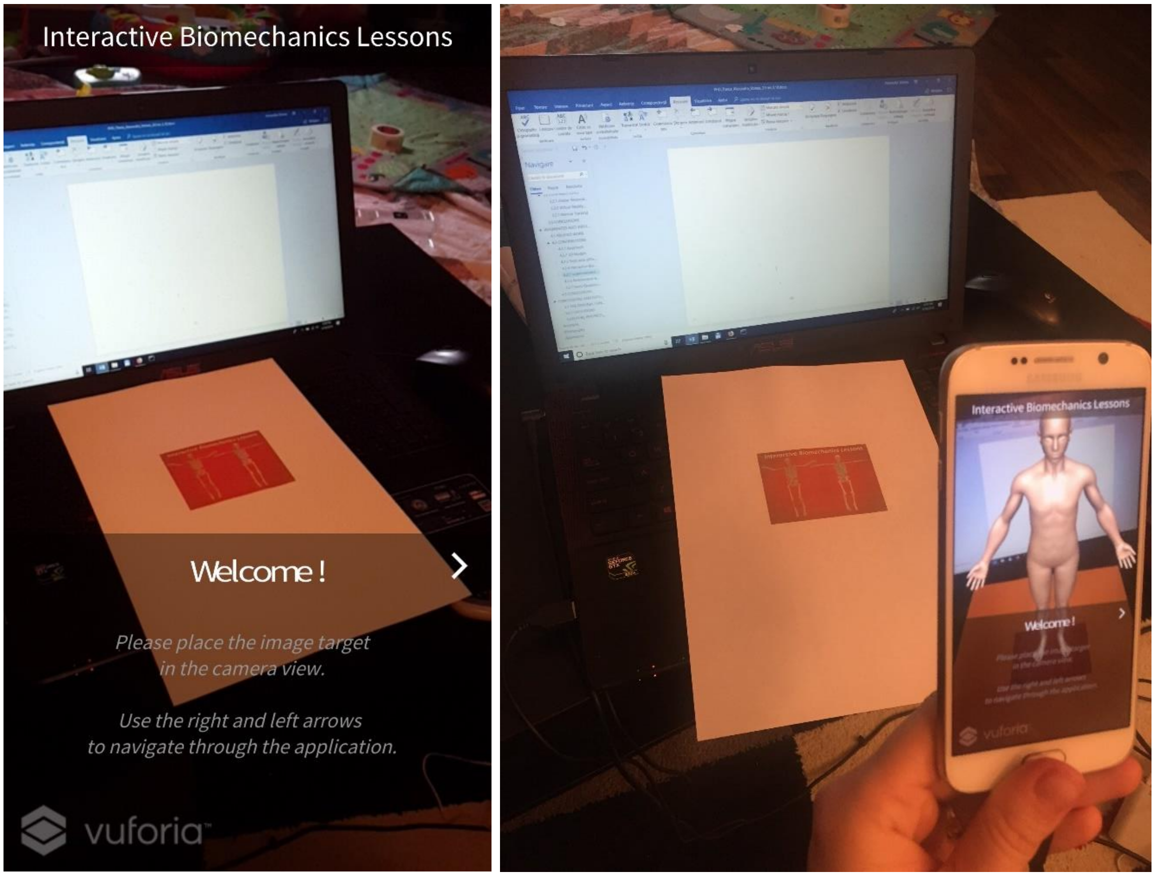
Figure 4.71 – VR BlueSky – Movements - Abduction

4.2.5.2 Marker-based Augmented Reality

Interaction Methods

The methods of interaction in VR and AR are completely different as in VR we had to blend the user interface between the 3D models while in AR we kept the 2D UI approach. In this section we are discussing about the marker-based AR scenario of our solution that uses predefined biomechanics lessons, similar with the VR case. The marker-based solution depends on visual markers detectable with computer vison methods. As already mentioned, we used Vuforia to add support for the marker-based AR scenario. The marker is a target image that we have added into our project's Vuforia database.

The target image was downloaded from the Vuforia developer portal and then imported in the unity project. The image was printed to an A4 sheet to be able to use it as a target while the application was running (The selected target image is available for print in Appendix 1). Figure 4.72 shows the printed target image marker and its detection.



Welcome !

Please place the image target in the camera view.

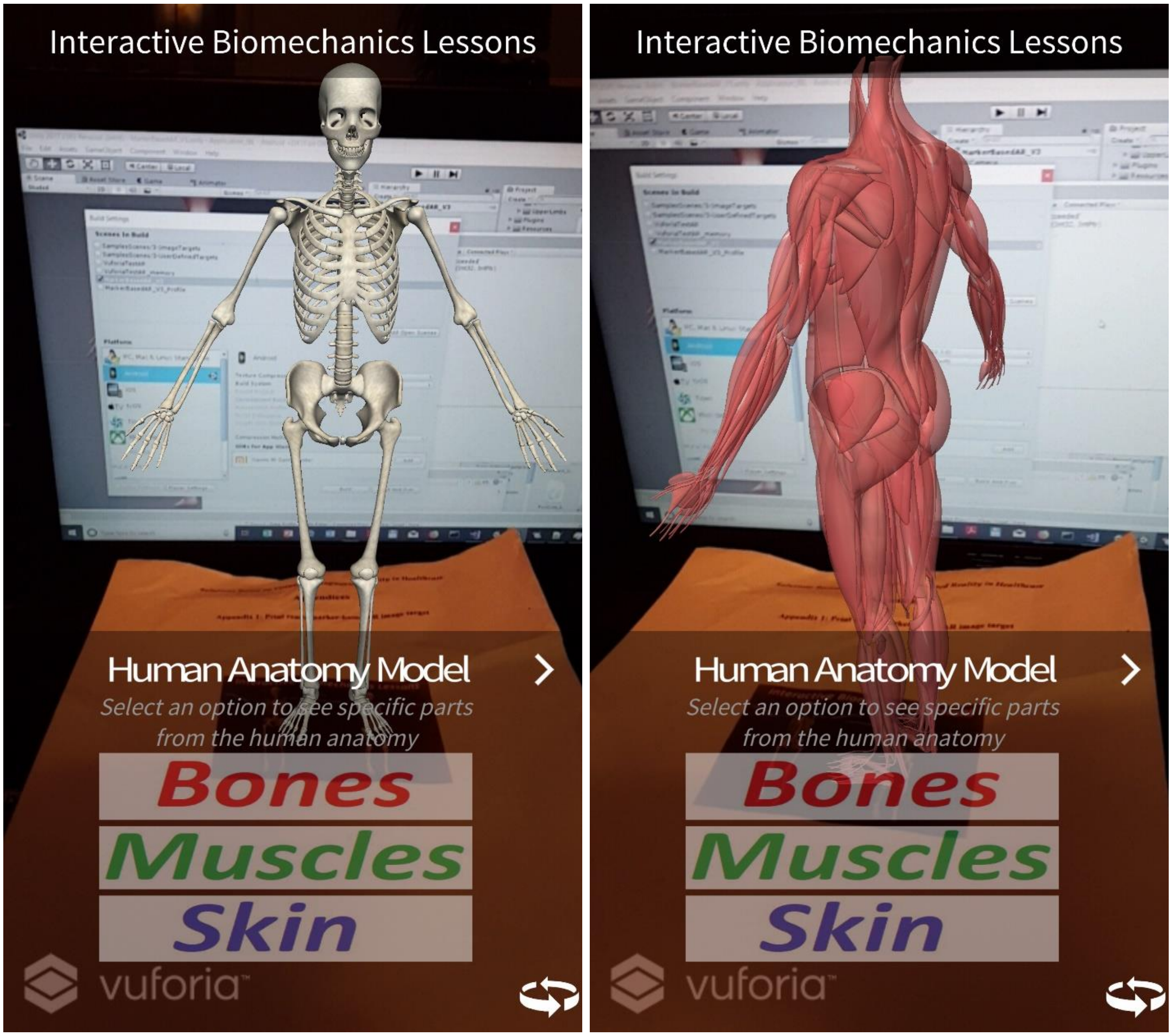
Use the right and left arrows to navigate through the application.

Figure 4.72 – Marker-based AR – Target image detection (Screenshot from the device on left).

The printed A4 page is placed on a flat surface and the user starts the marker-based AR application following the onscreen instructions. The skin model appears after the application detects the project's target image.

The user interface is 2D with a semitransparent overlay as we noticed that opaque elements are impacting the user experience in the AR environment. The application had a similar method of interaction with the 3D elements from the scene as we continued using ray casting technique but with different parameters.

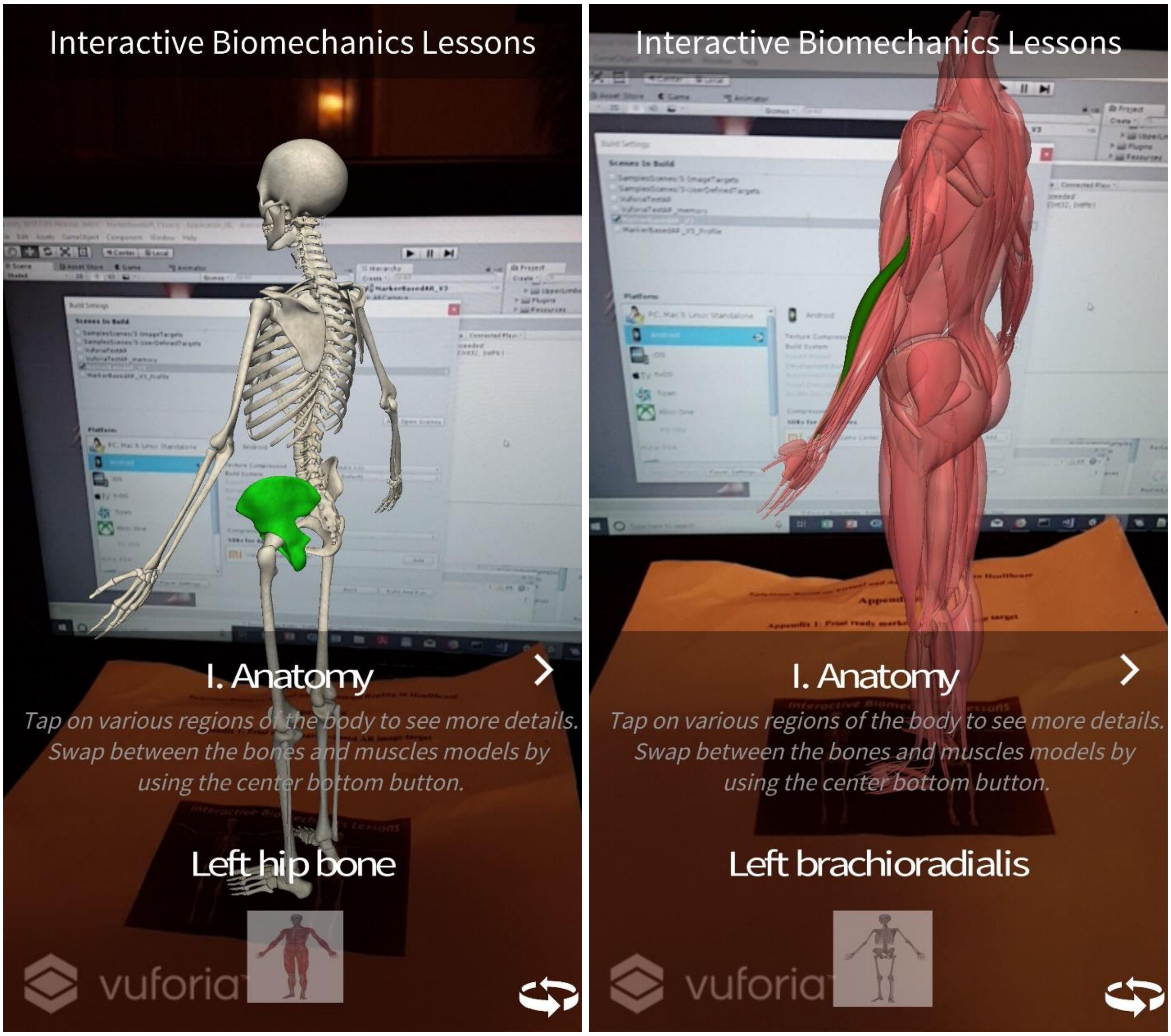
The scene contains the main 3D model represented by the skin, muscles and bones and various 3D elements depending on the selected lesson. Many of the 3D elements were imported from the VR project through unity prefabs. Figures 4.73-77 contain a few examples of the marker-based AR application lessons' implementation and UI.



Select an option to see specific parts from the human anatomy

Select an option to see specific parts from the human anatomy

Figure 4.73 – Lessons in marker-based AR application – Bones Model (Left) and Muscles Model (Right)



Interactive Biomechanics Lessons

I. Anatomy

Tap on various regions of the body to see more details. Swap between the bones and muscles models by using the center bottom button.

Left hip bone

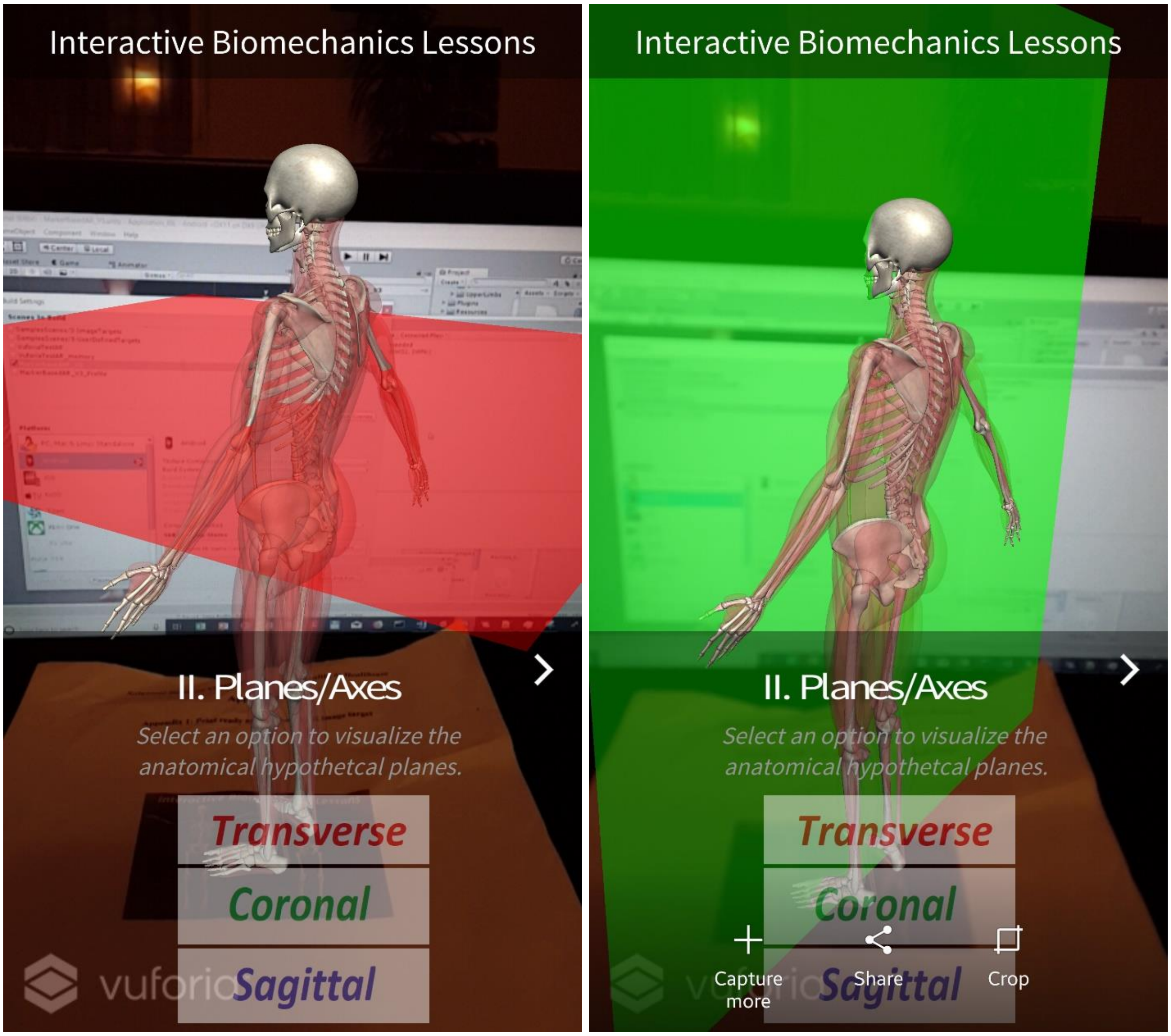
Interactive Biomechanics Lessons

I. Anatomy

Tap on various regions of the body to see more details. Swap between the bones and muscles models by using the center bottom button.

Left brachioradialis

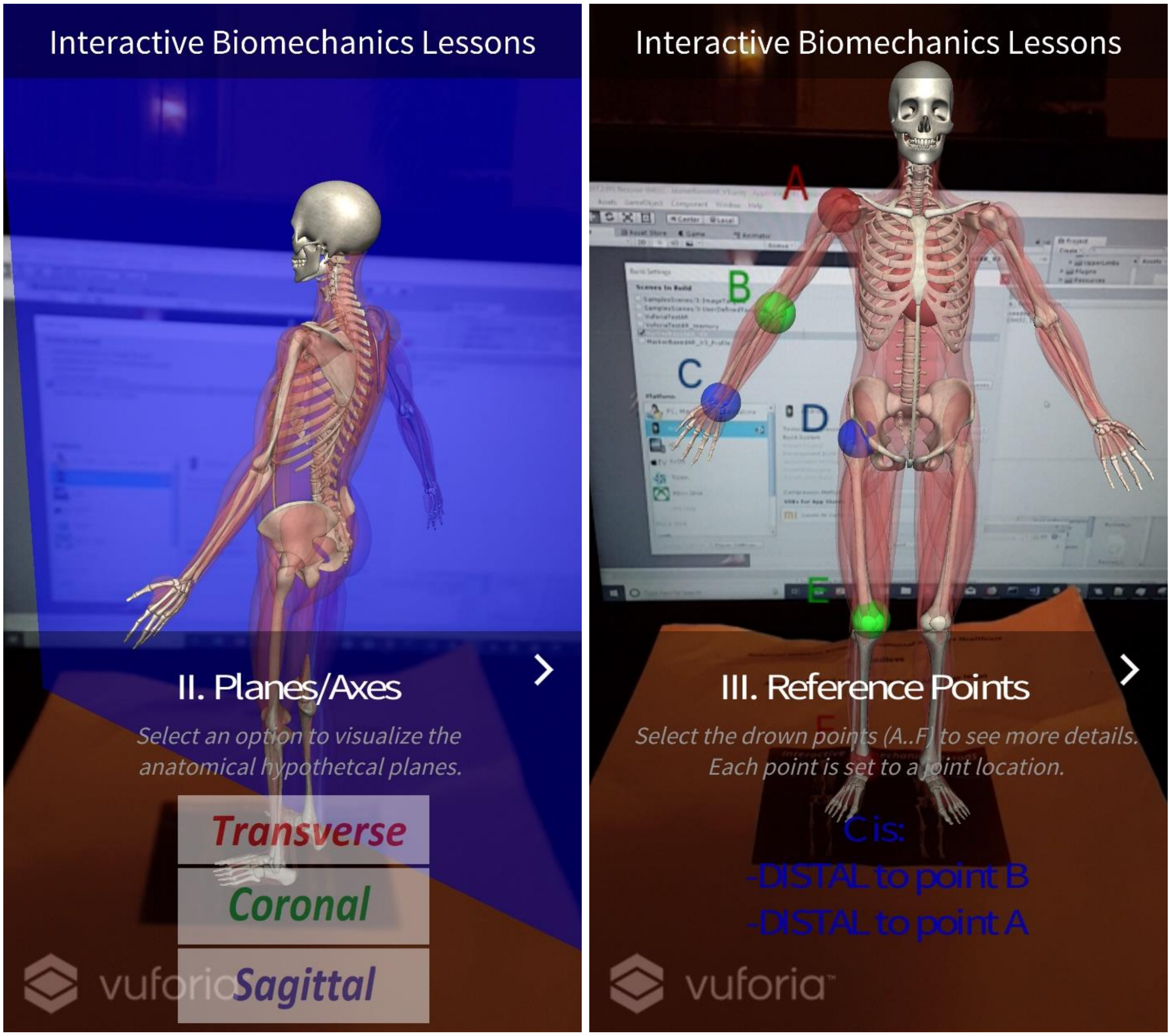
Figure 4.74 – Lessons in marker-based AR application – Anatomy Notions - Bones (Left) and Muscles (Right)



Select an option to visualize the anatomical hypothetcal planes.

Select an option to visualize the anatomical hypothetcal planes.

Figure 4.75 – Lessons in marker-based AR application – Planes/Axes - Transverse (Left) and Coronal (Right)

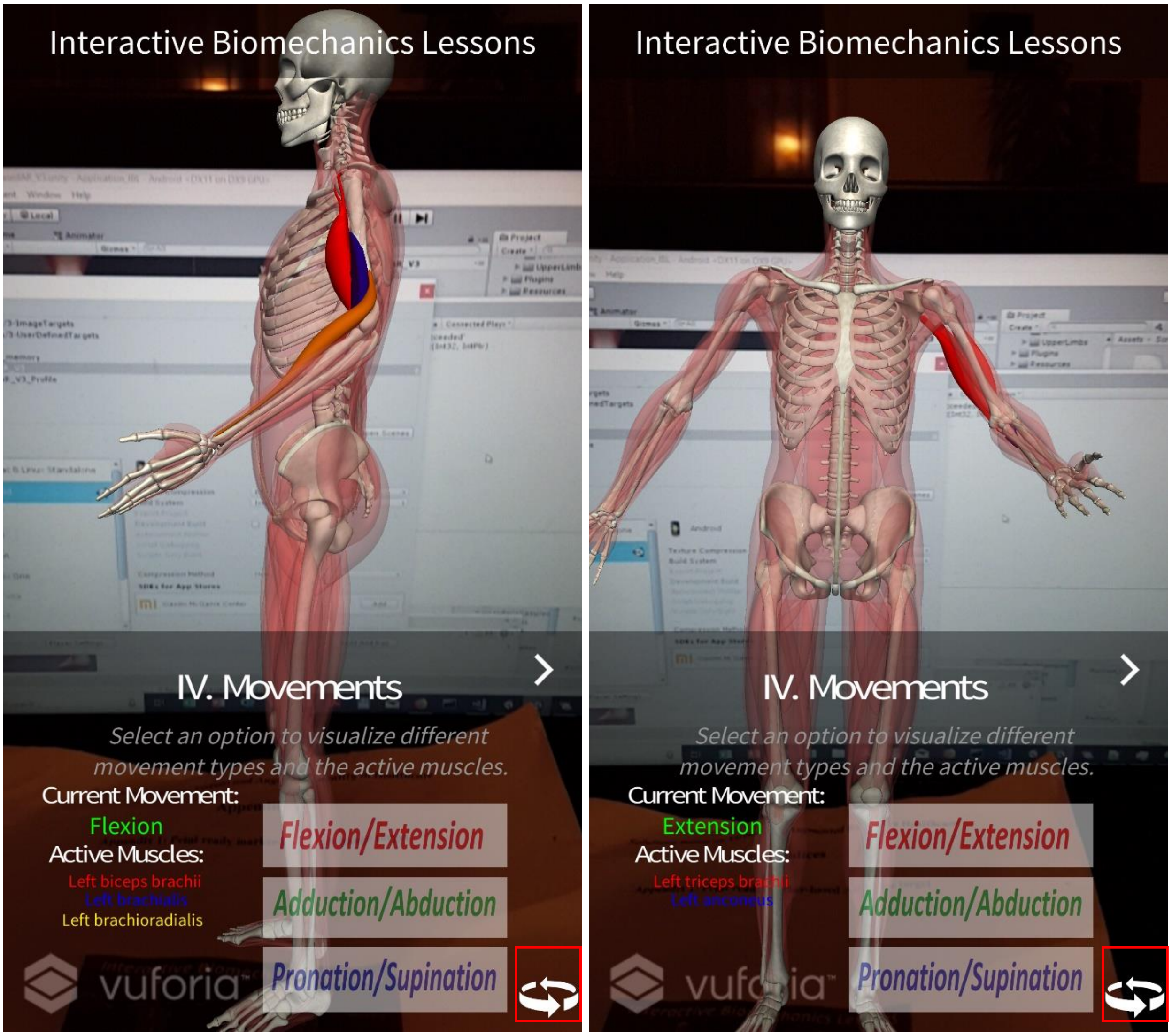


Select an option to visualize the anatomical hypothetcal planes.

Select the drown points (A..F) to see more details. Each point is set to a joint location.

C is:
-DISTAL to point B
-DISTAL to point A

Figure 4.76 – Lessons in marker-based AR application – Sagittal Plane (Left) and Reference Points (Right)

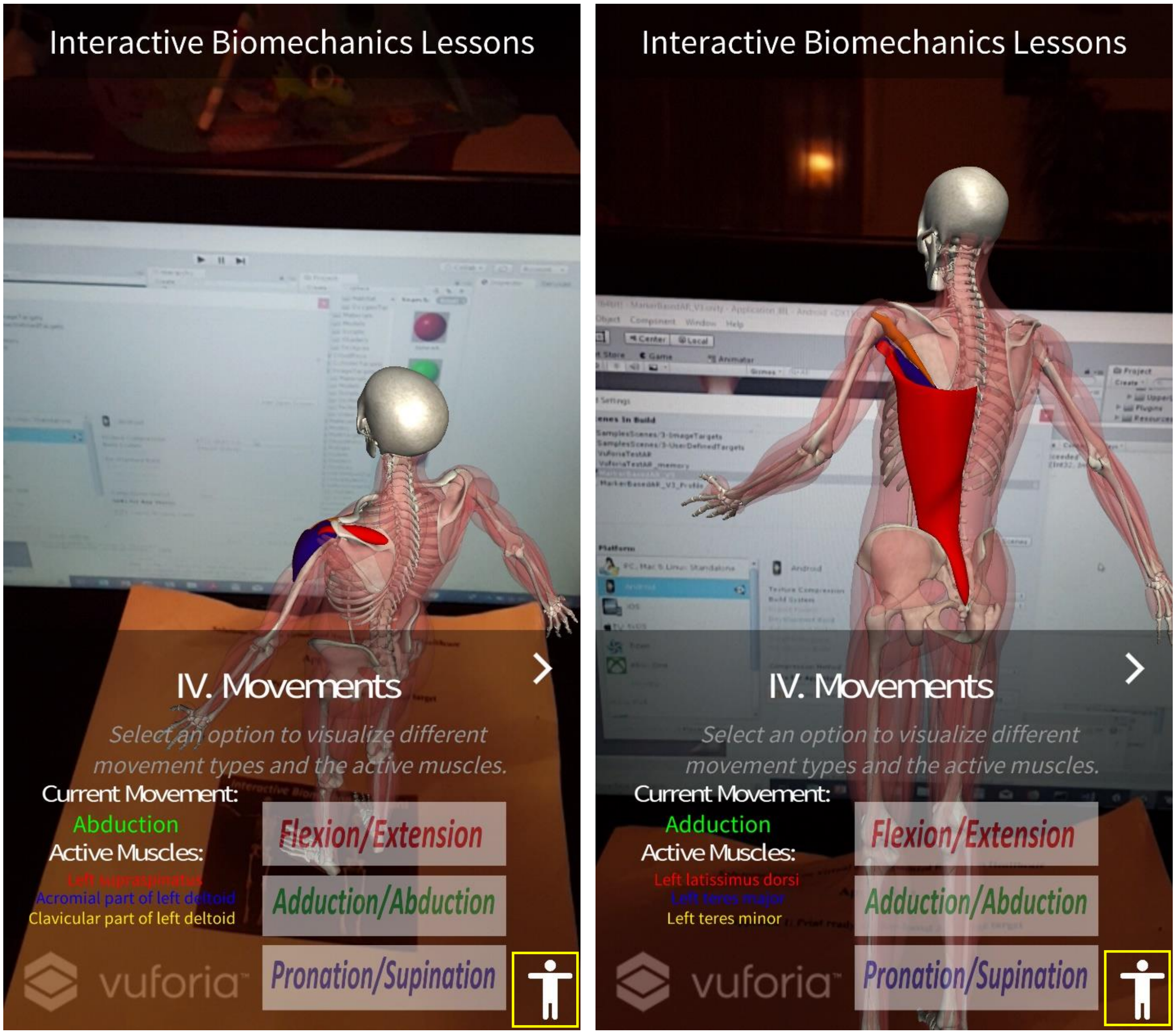


Select an option to visualize different movement types and the active muscles.
Current Movement:
Flexion
Active Muscles:
Left biceps brachii
Left brachialis
Left brachioradialis

Select an option to visualize different movement types and the active muscles.
Current Movement:
Extension
Active Muscles:
Left triceps brachii
Left anconeus

Figure 4.77 – Lessons in marker-based AR application – Movements – Flexion (Left) and Extension (Right)

Similar with VR, the AR lessons menus contain the option to rotate the model around the Y-axis at runtime. Figure 4.77 highlights in red rectangles the rotation option available at runtime. To achieve a correct model rotation in the AR environment, we changed – *World Center Model* – Vuforia behavior setting *from* the default *Camera* option to *First Target*. After the user selects the rotate option the user has the option to reverse it. The "stand" option is highlighted in the yellow rectangles from the images bellow (Fig. 4.78) as it is complementary with rotation and by selecting it the model returns to the original position.



Interactive Biomechanics Lessons

IV. Movements

Select an option to visualize different movement types and the active muscles.

Current Movement:
Abduction
Active Muscles:
Left supraspinatus
Acromial part of left deltoid
Clavicular part of left deltoid

Interactive Biomechanics Lessons

IV. Movements

Select an option to visualize different movement types and the active muscles.

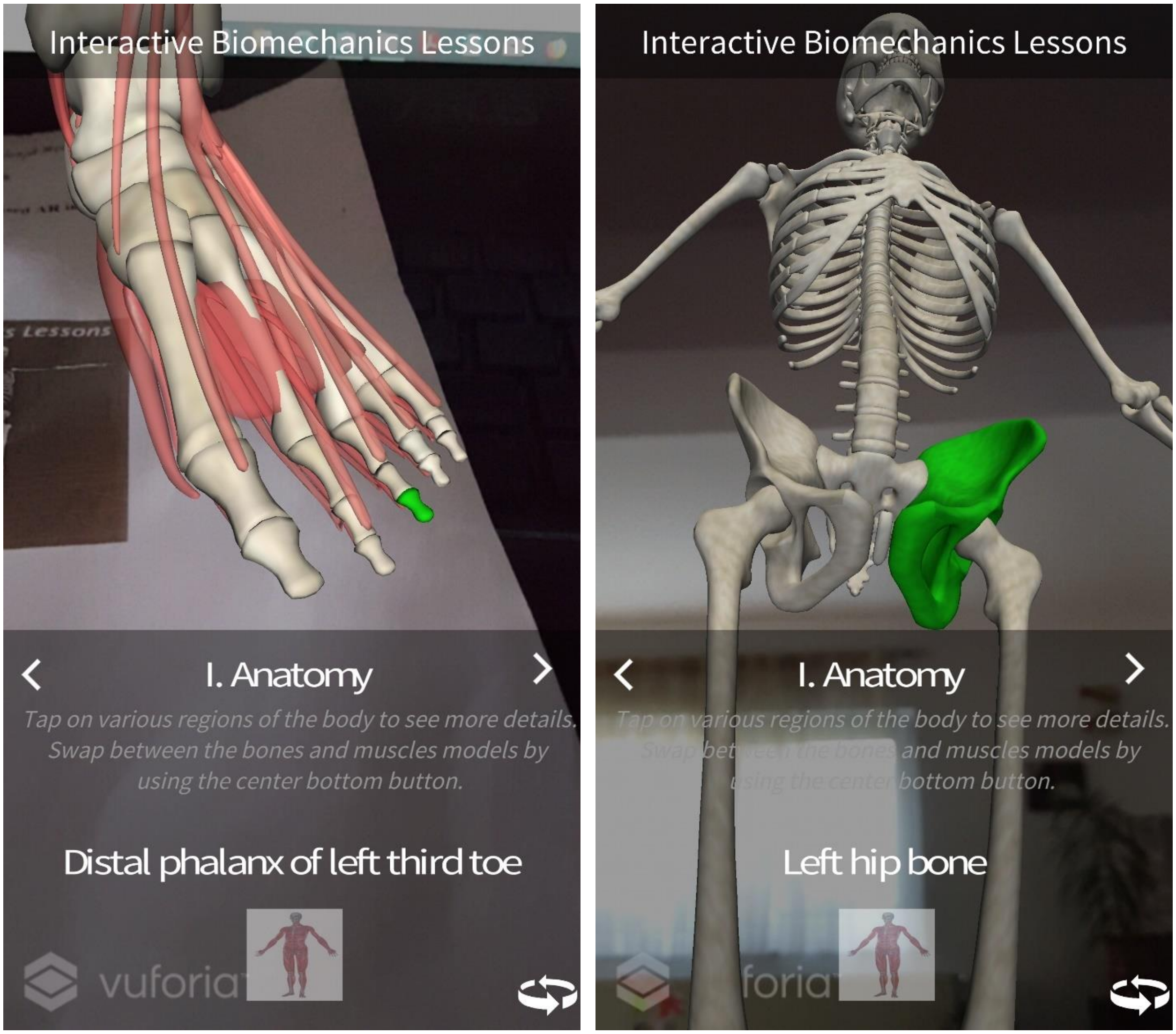
Current Movement:
Adduction
Active Muscles:
Left latissimus dorsi
Left teres major
Left teres minor

Figure 4.78 – Lessons in marker-based AR application – Movements – Abduction (Left) and Adduction (Right)

Extended Tracking

Another Vuforia feature that seems interesting was *extended tracking*. This means that we don't need to have the image target into our camera view to be able to visualize the 3D model, as opposed with the default version. Vuforia is creating target maps at runtime based on the background with the condition that it stays mostly static.

We consider that this feature has a high potential because we were able to see closely the tested 3D model's details and in fact we managed to find a few geometry bugs that weren't previously observed with classical modelling tools (3DS Max and Blender). Figure 4.79 shows how the 3D model is displayed in the extended tracking mode. This model is 4 times bigger compared with the default version and can be adapted to any size depending on the required level of details.



Tap on various regions of the body to see more details. Swap between the bones and muscles models by using the center bottom button.

Distal phalanx of left third toe

Tap on various regions of the body to see more details. Swap between the bones and muscles models by using the center bottom button.

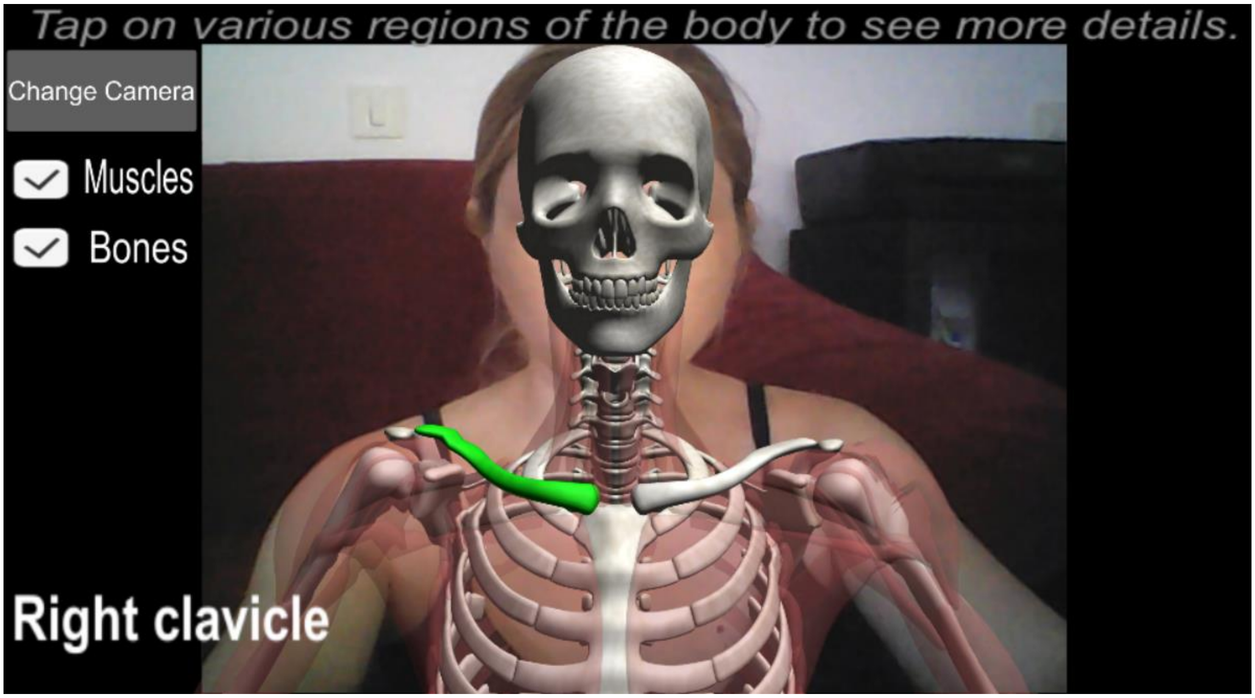
Left hip bone

Figure 4.79 – Marker-based AR application –Extended tracking feature

4.2.5.3 Markerless Augmented Reality

The main markerless AR application is based on a Kinect device to detect the user's movements in real time and the implementation details were already presented in 4.2.3.2 section. That version of the application used the models obtained from medical images as avatars and they contained a single mesh. We used a Laptop PC and a mobile device to display the AR environment. We used the PC with the Kinect device and the face detection while the mobile version is based only on face detection. The Kinect sensor works only connected to a PC as opposed with the newer sensors such as VicoVR that can be connected via Bluetooth to mobile devices. Both versions or the markerless AR application had a consistent lower performance compared with the virtual reality ones or the maker-based AR application that didn't contained motion tracking.

We run the application both on a Laptop PC (Unity Editor) and a mobile device and besides the PC's improved performance at runtime, there was another difference related with the visualization part: The smartphone camera is set on one side when holding it in landscape mode and the camera on the laptop was exactly at middle, directed to the face. The overall performance impacted the user's experience on the mobile device and we observed the fact that because the model was facing the user all the time, this diminished the access and visualization of the back part meshes. Figures 4.80 – 4.84 show the markerless AR application as displayed on a mobile device where we used the imported models of muscles, bones and skin. The models are scaled at runtime based on the face detection boundaries and were repositioned to fit the skull mesh over the face region. Even though the skin model is not visible, it was used to obtain the models size (based on boundary box) on Y axis to properly position the models after the scaling operation was performed. The skin was the only one from the newly used models that contained only one mesh and it was more facile to obtain at runtime its size.

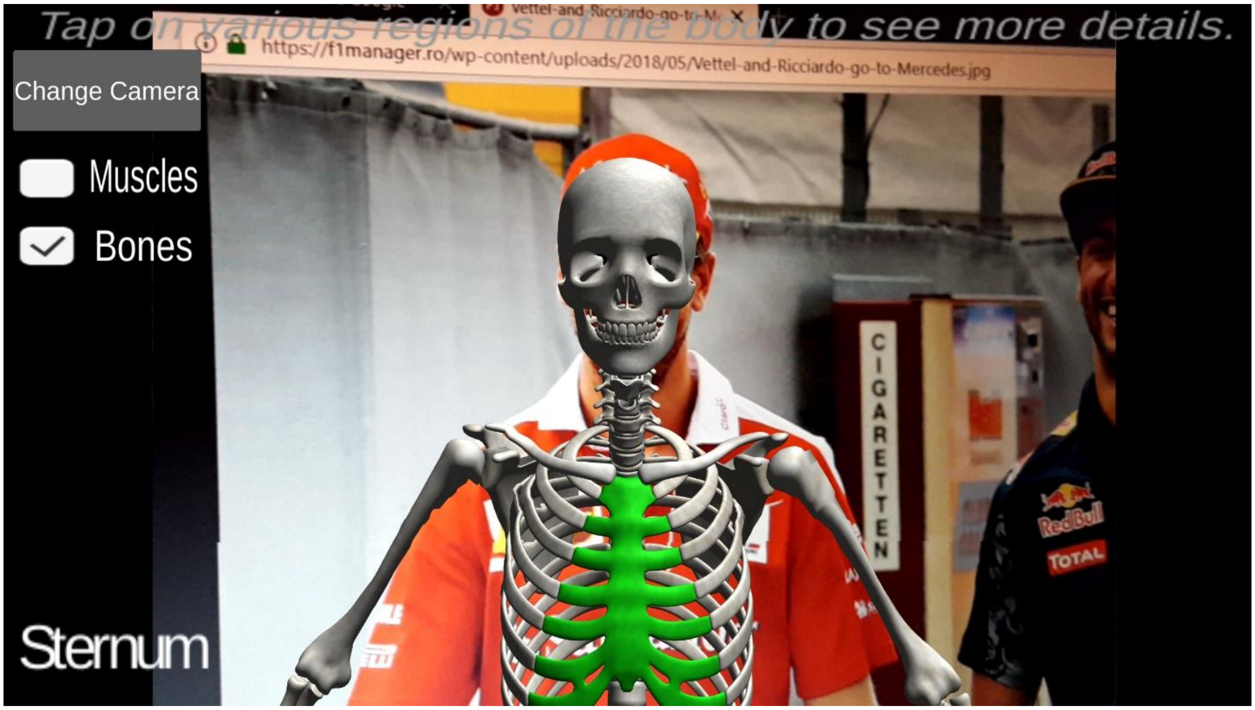


Tap on various regions of the body to see more details.

Figure 4.80 – Markerless AR application – Muscles and bones models visible, scaled and positioned based on face detection (Laptop)

Tap on various regions of the body to see more details.

Figure 4.81 – Markerless AR application – Only muscles model is visible



Tap on various regions of the body to see more details.

Figure 4.82 – Markerless AR application – Only bones model is visible as displayed on the mobile device[65]

---

[65] Background image source: https://f1manager.ro/wp-content/uploads/2018/05/Vettel-and-Ricciardo-go-to-Mercedes.jpg

Figure 4.83 – Markerless AR application – Muscles and bones models visible – Muscle highlighted (Laptop)

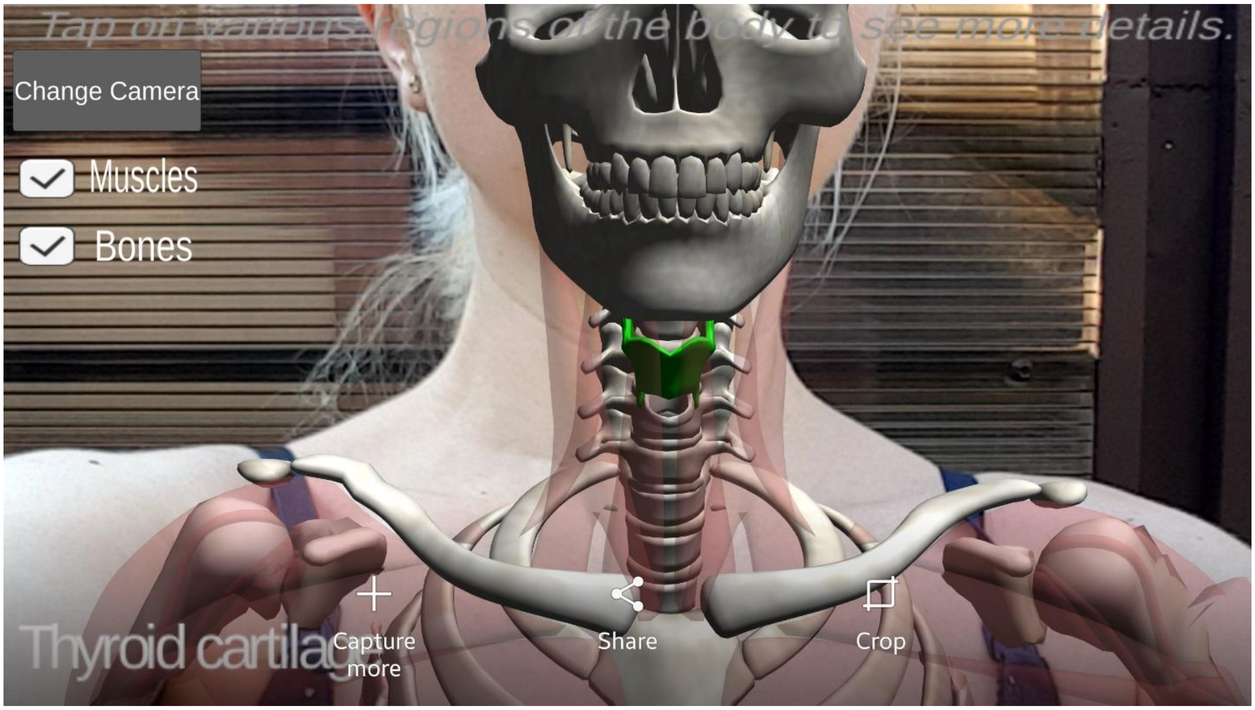


Figure 4.84 – Markerless AR application – Muscles and bones models visible – Bone highlighted (Mobile device)

4.2.6 Performance Analysis

When building an application, either it is for desktop, mobile, VR or AR, the performance metrics are very important. Usual metrics that are tracked in applications development are: framerate, memory, application size or the processor heat. The last one is special for Android devices as it was noticed that during the utilization of heavy computational applications the mobile devices start to heat. This was observed during author's extensive experience in the mobile game development on various titles and Android devices had a higher occurrence of the heating issues. This issue is obvious usually at extensive usage of the application and as a result the application's framerate is impacted due to CPU and GPU throttling. After the processor reached a certain temperature, it starts to function at a smaller frequency and in consequence it affects the processing power. Taking into consideration the fact that the presented applications contain a small number of interactive lessons and the usual session time is maximum 10 minutes, we didn't invest to much in monitoring this metric but considering the complexity of the obtained 3D models this one should be closely supervised if the solution is extended.

Regarding the other metrics, we gathered data for FPS, memory and rendering usage using Unity profiler and Android Monitor (from Android Studio). This data was available only on the development build and we should mention that usually a development build has lower performance results compared with a release build because the code has a lower level of optimization. One can add a proprietary tracking method to gather the performance results with minimal consumption. Due to the available time for this operation we based our information exclusively on the existing tools.

The AR markerless scenario has two tracking methods: one on the mobile device (Samsung S6), only with OpenCV face detection and model scaling functionality on, and one on the PC with all the functionalities enabled. We added the performance data extracted from the mobile device to be able to make a clear comparison between all 4 scenarios.

Table 4.2 shows the high-level results of the tracked metrics on all 4 scenarios and Figures 4.85 – 4.88 display the results, as obtained with Unity profiler.

Table 4.2 – Key performance metrics highlights on the mobile device (Samsung S6)

Scenario	Rendering CPU usage [ms]	Scripts CPU usage [ms]	Total Allocated Memory [MB]	Textures Memory [MB]	Mesh Memory [MB]
VR- virtual classroom	30.2	4.1	234.1	14.6	87.2
VR- no background elements	29.9	4.0	233.9	14.6	87.2
AR – marker-based	14.40	5.35	234.6	37.6	93.9
AR- markerless – mobile	37.3	70.9	264.1	14.5	85.5

Figure 4.85 – Performance metrics for VR with classroom background scenario

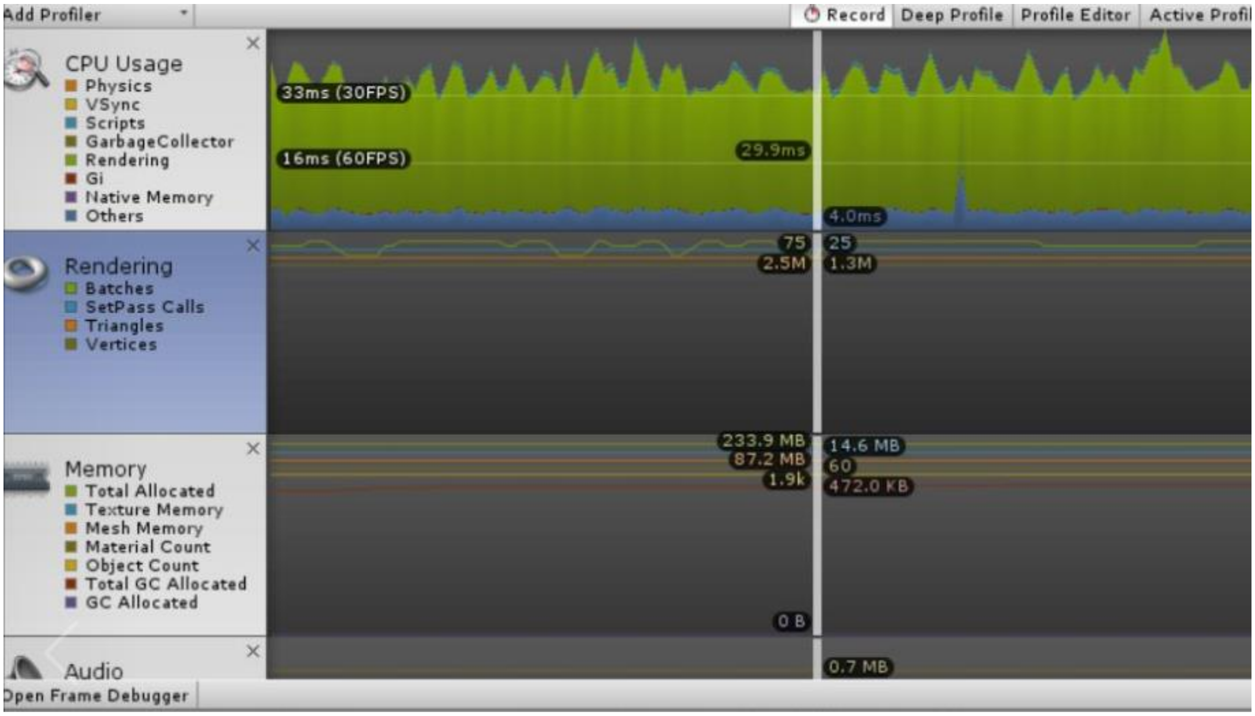


Figure 4.86 – Performance metrics for VR with no background scenario

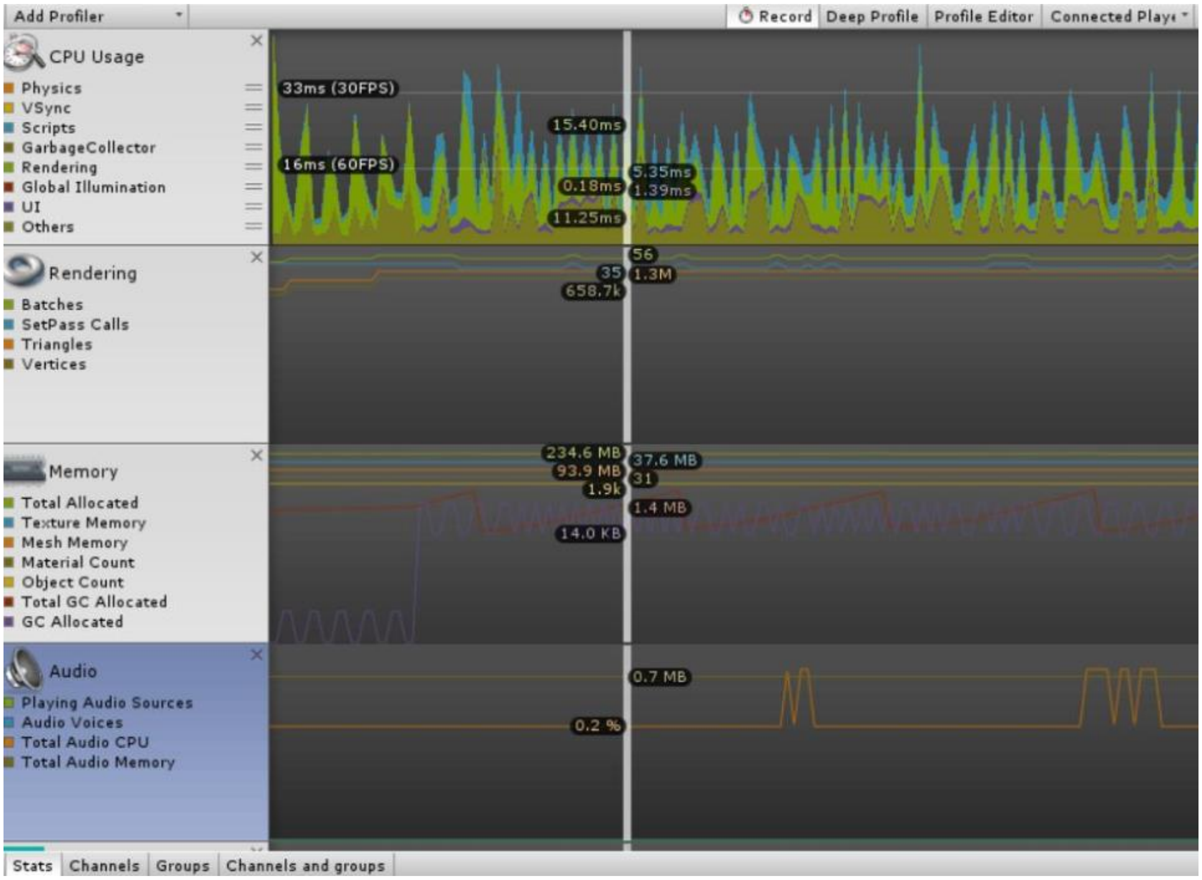


Figure 4.87 – Performance metrics for AR marker-based scenario

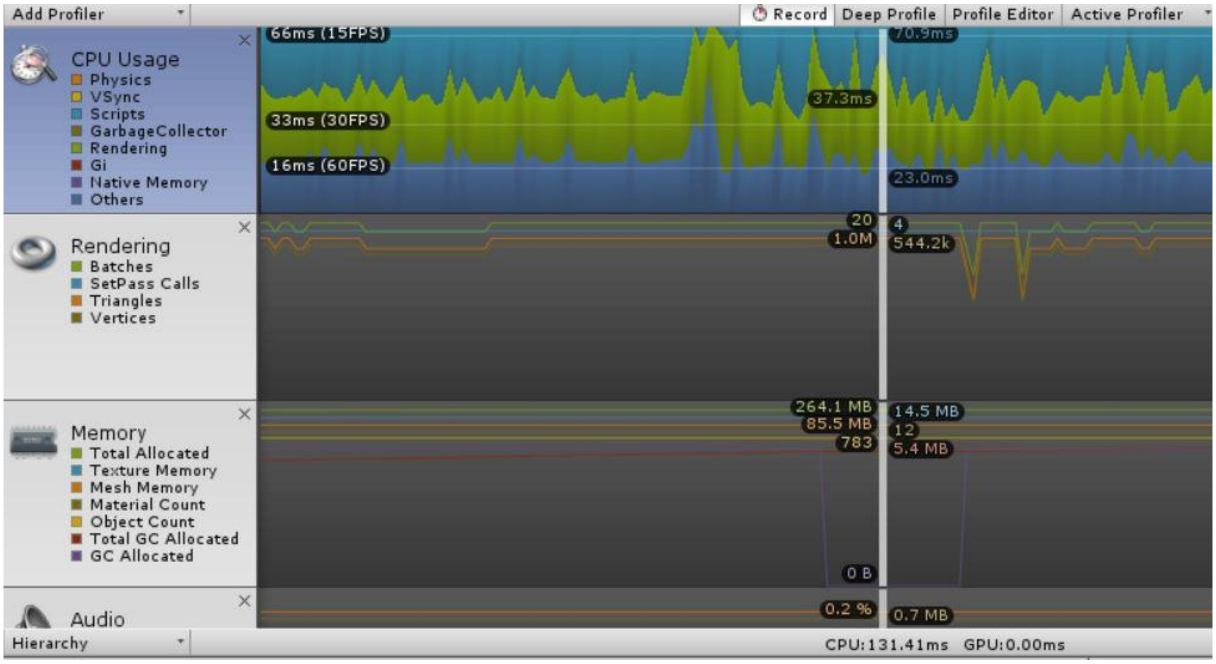


Figure 4.88 – Performance metrics for AR markerless on mobile device (no Kinect data) - Haar classifier

We can notice from the extracted data that the VR scenarios have very similar performance metrics, and this can be related to the fact that the background environment is added into both scenarios' scenes. The only difference is the fact that the game object that contains all the environment's data (virtual classroom) is disabled. We observed the same behavior in the AR marker-based application where we initially consumed ~900MB on totally allocated memory because we kept disabled (and not removed) some Vuforia targets that were used as examples at first.

The AR scenarios have different metrics values compared with each other and with the VR scenarios. Firstly, the marker-based one has the best CPU usage consumption but as seen in the tracing data the values are constantly spiking. For the markerless scenario the CPU usage has by far the highest consumption. The scenario was tested on the mobile device only with OpenCV functionality for face detection and model scaling and positioning at runtime and without Kinect motion tracking. We selected it to observe the performance on the same device even if didn't included Kinect data to be able to have the same reference to make a proper comparison. Even without Kinect, we could observe that the markerless scenario had the highest CPU consumption and the lowest performance.

To have a complete picture we need to add the performance metrics registered with the AR markerless application used on the PC since the full functionality was built on it. We added various utilization scenarios of the application to have enough data for a correct correlation with the data presented previously. Figure 4.89 displays the metrics from the AR markerless application but on Windows workstation instead of a mobile device.

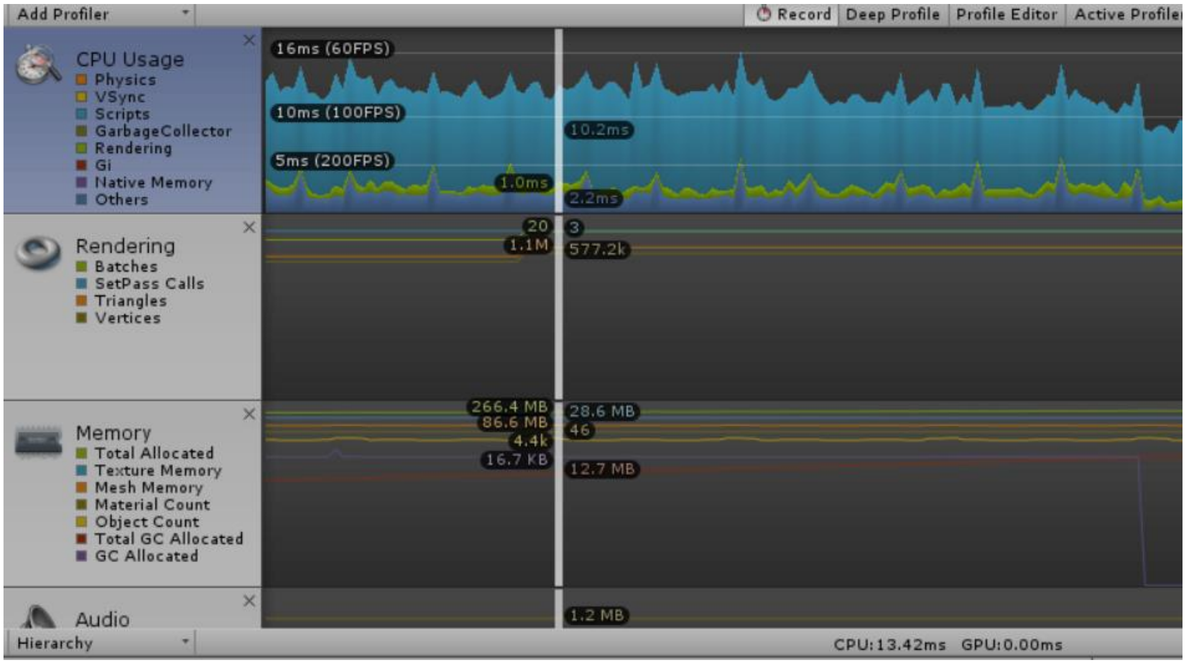


Figure 4.89 – AR markerless performance metrics – OpenCV, no Kinect, on Windows workstation

Table 4.3 displays the CPU usage metrics of several tests/versions of the AR markerless application (using the models generated from medical images) on the workstation since the memory and rendering ones were similar with the previous case as the only seen difference was only for the CPU.

Table 4.3 – Key performance metrics highlights on AR markerless application

Scenario	Rendering CPU usage [ms]	Scripts CPU usage [ms]	Total CPU usage [ms]
OpenCV face detection and model scaling on <u>mobile device</u>	37.3	70.9	131.41
OpenCV face detection and model scaling on <u>workstation</u>	1.0	10.2	13.42
Kinect (idle) and OpenCV motion tracking on workstation	< 1.0	33.2	34.77

Figures 4.90 – 4.92 show the tracing data for CPU usage as registered with Unity profiler for the tracked scenarios mentioned in table 4.3.

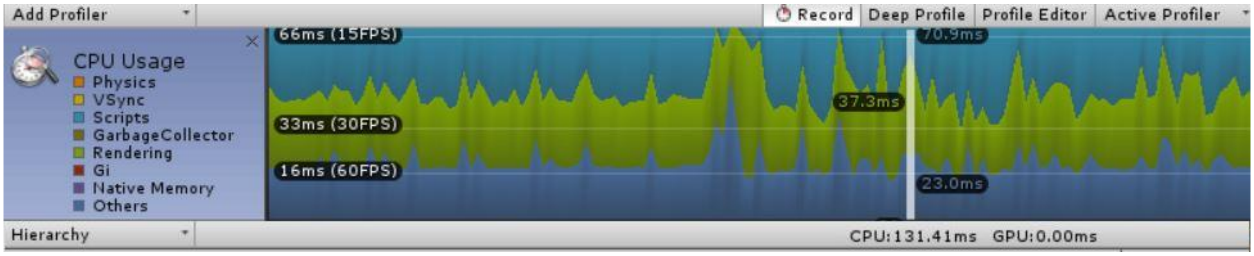


Figure 4.90 – AR markerless application performance metrics – OpenCV face detection on mobile device

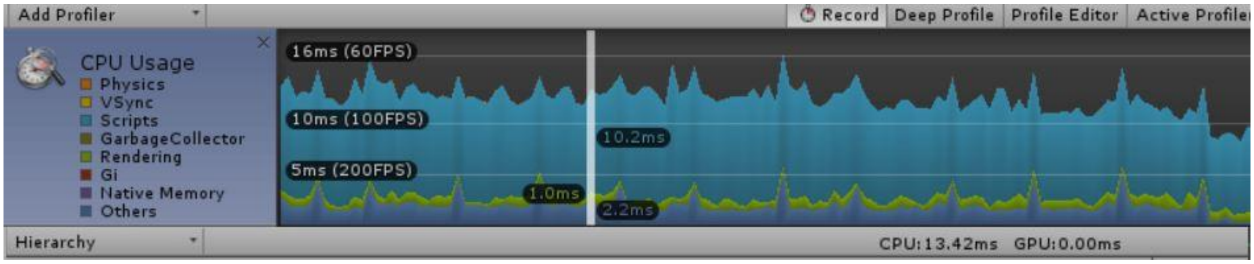


Figure 4.91 – AR markerless application performance metrics – OpenCV face detection on Windows workstation

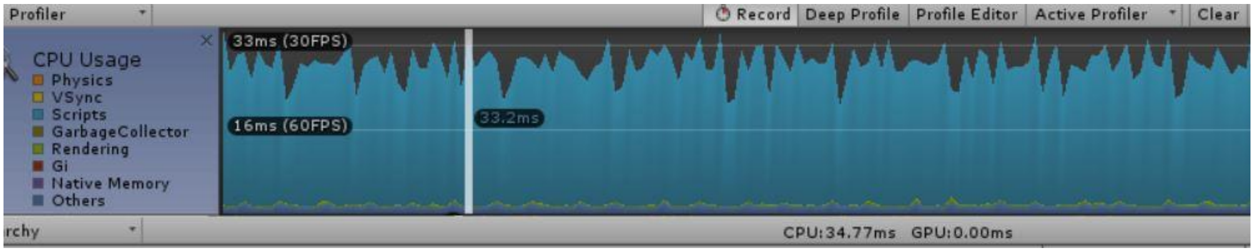


Figure 4.92 – AR markerless application performance metrics – Kinect(idle) and OpenCV motion tracking on Windows workstation

In the figure 4.92, the performance data has those values with *KinectController* and *AvatarController* scripts enabled but without any tracked joints at runtime (the user wasn't in the sensor's FOV). The scenario has all the elements enabled into the scene: Kinect dependent scripts and OpenCV based complementary tracking method based on the face detection for properly positioning and scaling the model. We noticed that when we started the tracking with the Kinect device (moved the arm and legs), and animated the models based on skeletal tracking, the performance of the application had a consistent impact. Figure 4.93 shows how the performance metrics changed after the sensors detected joints movements. Although the performance data obtained in the previous cases showed a low CPU usage on the Windows workstation machine for the AR markerless application, after the Kinect sensors starts to perform skeletal tracking the performance starts to diminish visibly.

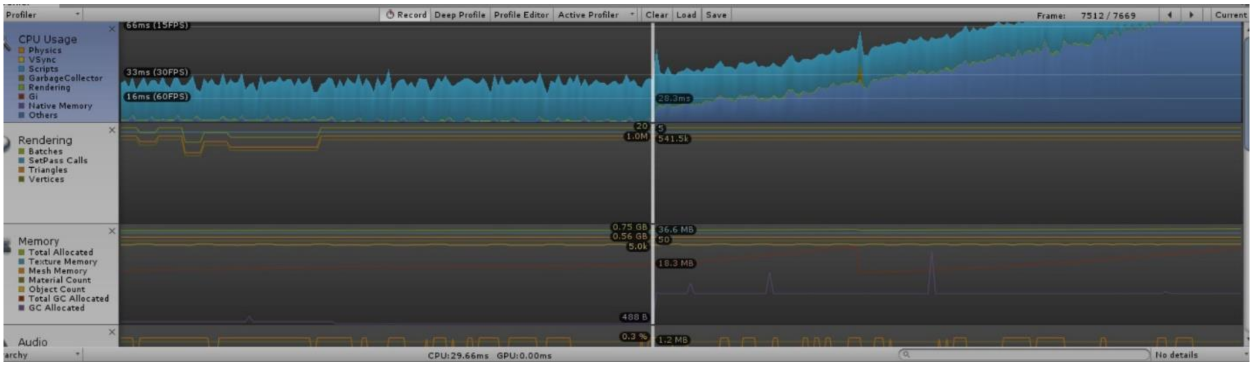


Figure 4.93 – AR markerless application performance metrics – Kinect skeletal tracking and OpenCV face detection on Windows workstation using models generated from medical images

Figure 4.94 shows the application performance when using the imported models of skin, muscles and bones. As it can be observed, the CPU consumption raises instantly when the system detects the user's movements and the application renders the models over the user's figure. Since the performance was highly impacted on the version that used the Kinect sensor, we couldn't have a proper experience with these newer models. Considering the fact that the Kinect devices are out of use we didn't invest more time into optimizing the application that used motion tracking.

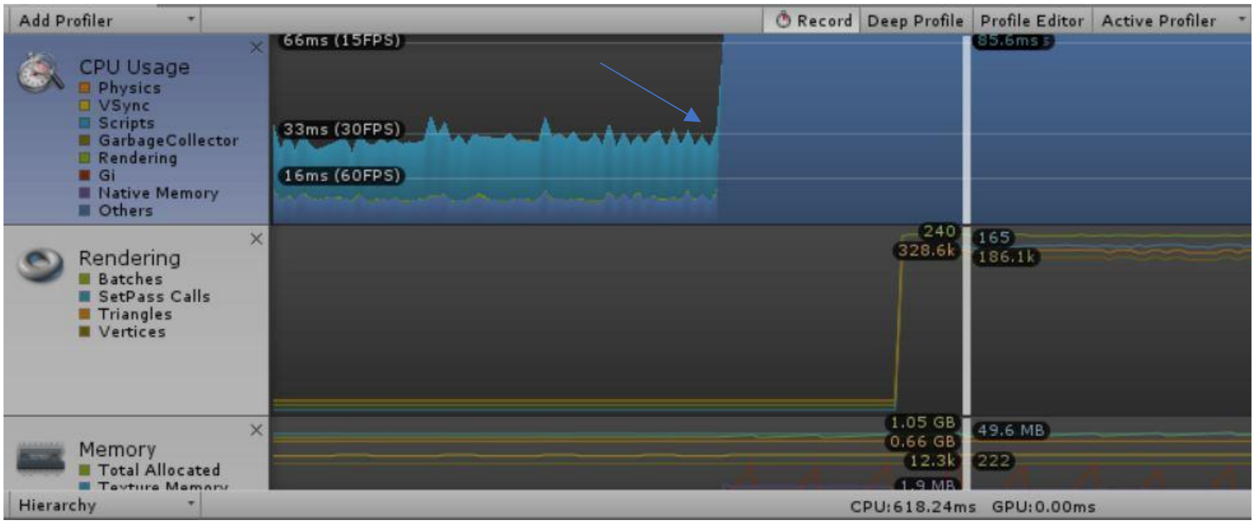


Figure 4.94 – AR markerless application performance metrics – Kinect skeletal tracking and OpenCV face detection on Windows workstation using the imported models. Blue arrow indicated where the motion tracking has started.

4.2.7 Users Questionnaires Results

While developing VR applications, the questionnaire evaluation is a very important step in obtaining improved results as the presence evaluation and correction are critical. Reference [DM10] mentions that for virtual-reality 5 subscales are tracked in the presence questionnaire:

1. **Realism** – Similarity between virtual and real (natural) environment.
2. **Affordance to Act** – Measures the ability to actively explore and manipulate the virtual environment.
3. **Interface Quality** – Refers to the runtime performance of the tested software and hardware. Are there any delays?
4. **Affordance to Examine** – The ability to examine the virtual elements from different angles.
5. **Self-Evaluation of Performance** – Is the user able to perform the required tasks in the displayed Virtual Environment?

Another point of interest is the cybersickness, which will directly affect the previous scales and the actual success of utilizing this method. This is mentioned as Simulator Sickness Questionnaire [DM10] and it targets subscales such as: **Nausea**, **Ocular-Motor Problems** and **Disorientation**. Also, the questionnaire evaluation was used to assess the benefits of VR [IM14]. In our research we used a more complex questionnaire that combines the assessment of the results, based on the tested technologies, and the users' feedback regarding the presence and cybersickness while using different scenarios.

For the results assessment we used the three fully functional applications that contain the predefined biomechanics lessons: the two VR applications and the AR marker-based one. In the first stage all the users received two questionnaires (Q1a and Q1b). Q1a contains the pre-exposure simulator sickness questionnaire (SSQ) to assess the state of the users before using the first VR application. The SSQ was developed by Kennedy and his colleagues in 1993 (Kennedy et al. 1993) as they came up with a list of symptoms which are experienced by users in virtual reality systems[66]. The questionnaire contains 29 symptoms and an example of it is available in Appendix 2. Each question related with sickness symptoms has 4 potential answers: None (0), Slight (1), Moderate (2) and Severe (3). If no answer is given to a symptom then zero value is assigned to it. The total score is the sum obtained by adding the symptoms scores. Also, there are 3 subcategories of simulator sickness: nausea, oculomotor and disorientation. The symptoms that are considered for each subcategory are as well available in Appendix 2. Q1b questionnaire contains some basic questions regarding the users' previous experience with VR and AR applications to assess their predisposition to adapt in a VR/AR application.

---

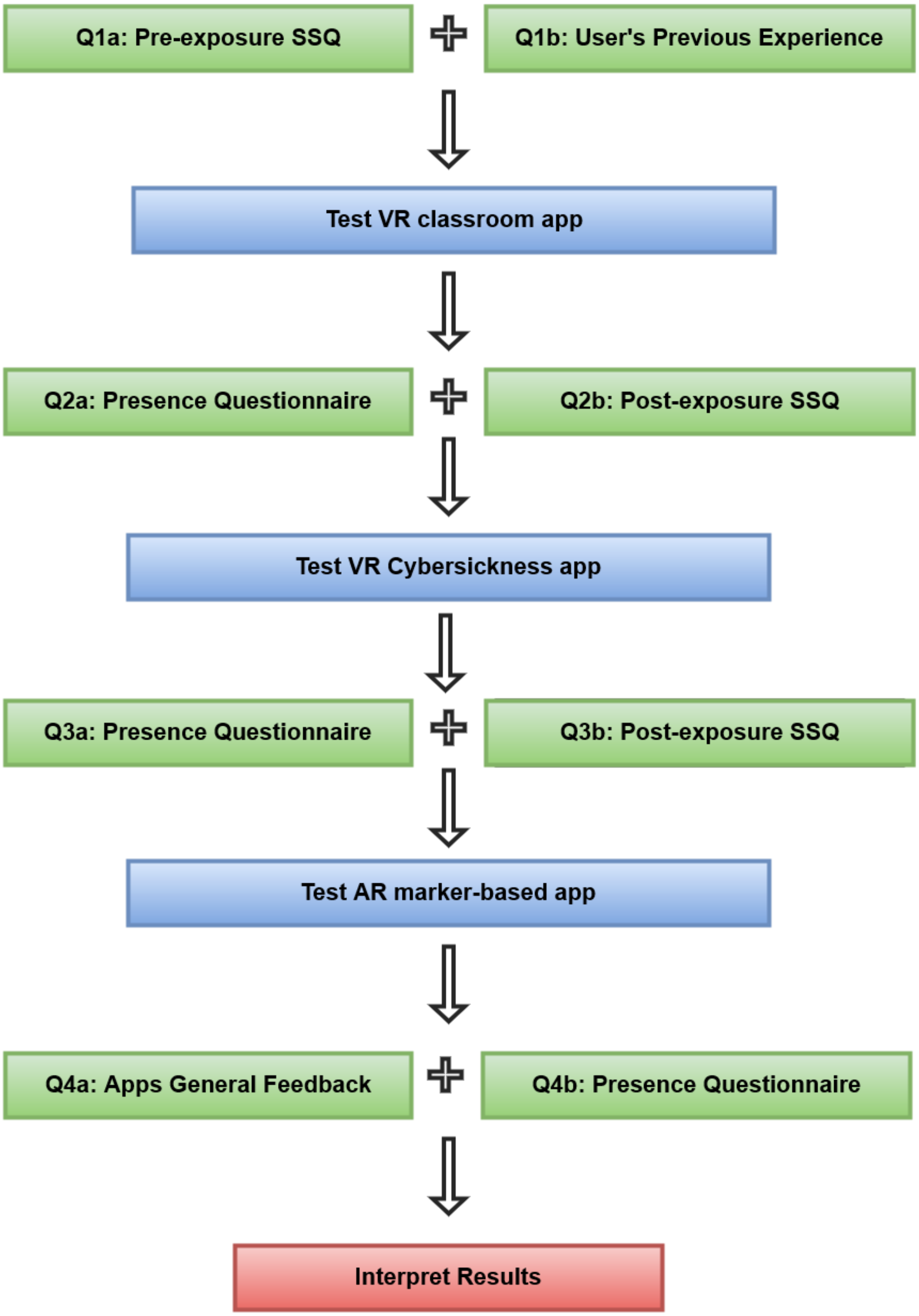
[66] https://www.twentymilliseconds.com/html/ssq-scoring.html

The testing order was the following: VR application with classroom background, VR application improved for cybersickness (BlueSky) and the marker-based AR application. After each VR application utilization, the users had to complete the post-exposure SSQ since each application can give different sensations to the them. The pre-exposure questionnaire is not applied multiple times since the symptoms are the same and we can correlate the data based on multiple post exposure data. The presence questionnaire was inspired from Witmer & Singer presence questionnaire and revised by the University of Quebec and Outaouais (2004)[67] and is applied after each application utilization to assess the user's level of presence. The presence questionnaire is applied after both VR applications and the AR one, as one of the questions of our research is to assess the level of presence felt by the users in a virtual versus an augmented environment. At last, the users were given a feedback questionnaire to assess their preferences for one environment versus another. Figure 4.95 contain the testing approach for all 3 applications and the types of questionnaires applied in each stage.

Initially we considered to use the VR and AR applications in parallel to assess the efficiency of each environment in the learning process thus considering the diverse background of the users and the consistent number of questionnaires related with presence and simulator sickness we decided to simplify the process and to rely on the user's general feedback as a method of assessment of each environment's efficiency.

---

[67] http://w3.uqo.ca/cyberpsy/docs/qaires/pres/PQ_va.pdf



***Q - Questionnaire**

SSQ - Simulator Sickness Questionnaire

Figure 4.95 – Results assessment approach

4.2.7.1 Simulator Sickness Questionnaires

The SSQ questionnaires were applied 3 times: a pre-exposure questionnaire was completed at the start of the experiment and two post-exposure questionnaires were applied after each VR application utilization. The data was gathered from all the users (5) and then analyzed for each one of them along the median values. We opted to assess the cybersickness data on each individual to make sure that the answers of two users won't annul the symptoms occurrences results as we want to see how they progresses through the experiment.

From 29 symptoms available in the SSQ questionnaire we observed a variation on responses for 12 of them though the whole experiment for all the users (Fig. 4.96). The felt symptoms were: *fullness of the head, aware of breathing, difficulty concentrating, difficulty focusing, dizziness with eyes closed, drowsiness, eyestrain, general discomfort, headache, increased appetite, nausea, vertigo.*

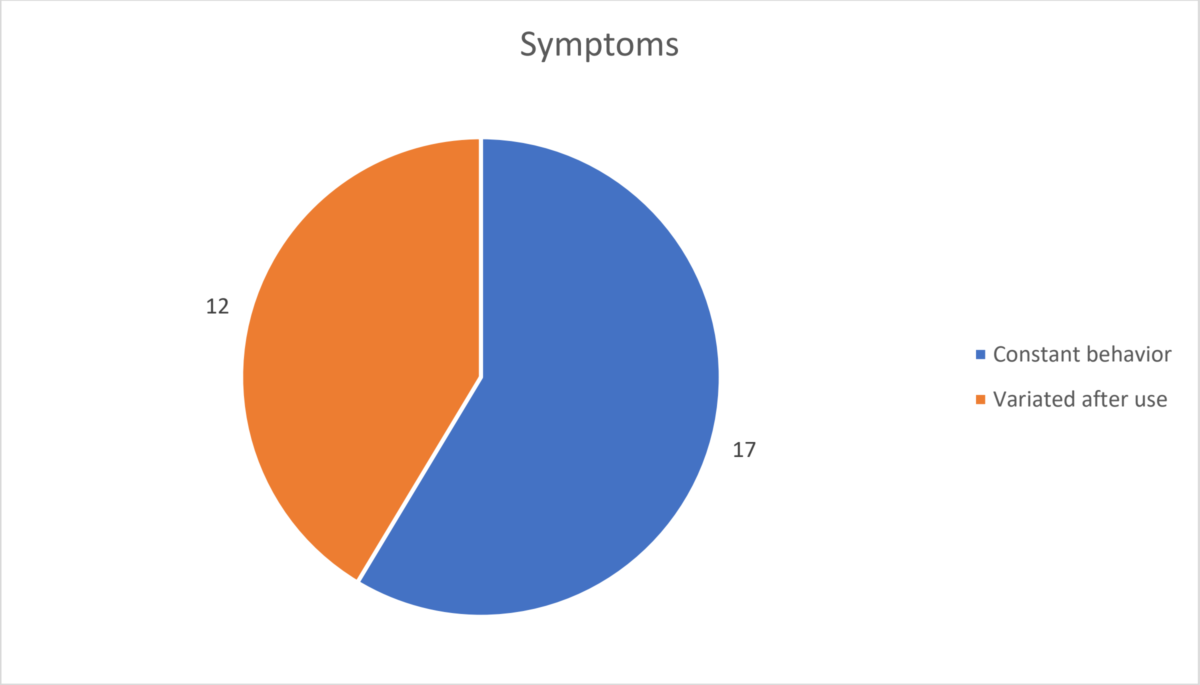


Figure 4.96 – Simulator Sickness Questionnaire - Symptoms variation

Each sickness subcategory (Nausea, Oculomotor and Disorientation) and the overall results were assessed for each user versus the median. We can observe in figures 4.97 – 4.100 the progress of cybersickness symptoms though the experiment by using the VR applications. Even though some symptoms variated along the experiment they weren't very drastic as the present symptoms were felt slightly in most of cases hence the relatively small scores numbers.

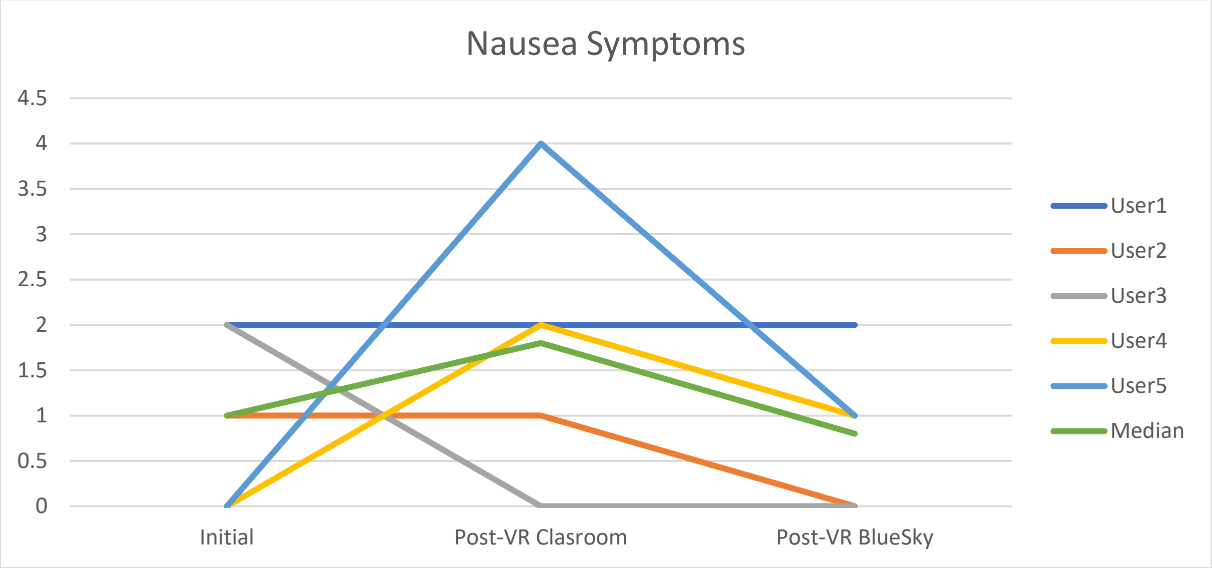


Figure 4.97 – Simulator Sickness Questionnaire – Nausea Symptoms

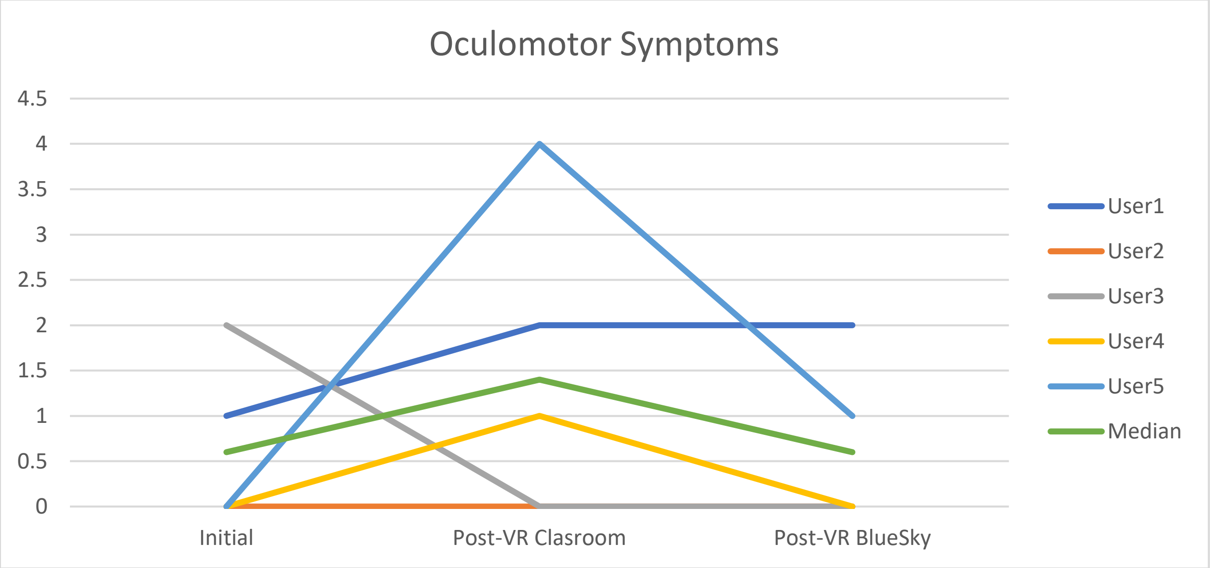


Figure 4.98 – Simulator Sickness Questionnaire – Oculomotor Symptoms

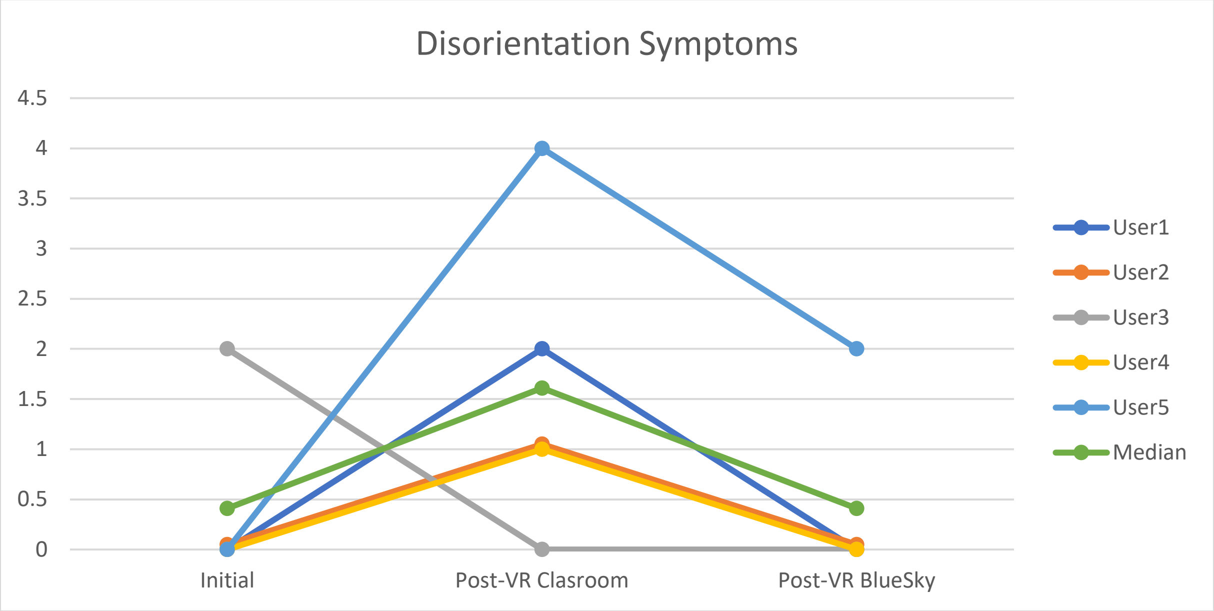


Figure 4.99 – Simulator Sickness Questionnaire– Disorientation Symptoms

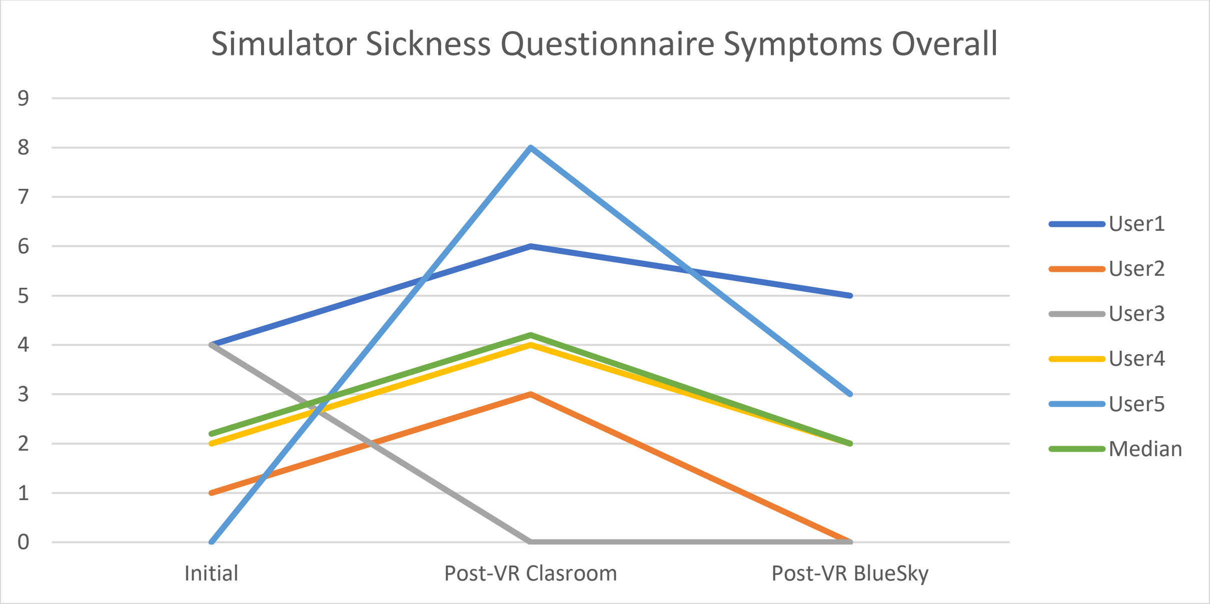


Figure 4.100 – Simulator Sickness Questionnaire Symptoms Overall

Looking at these figures we can observe the fact that VR classroom scenario had increased the sickness symptoms of the users and they were diminished with the second VR application. Also, we can notice the importance of the pre-exposure questionnaire as the users had some cases of sickness before testing the applications. On the other side, there are some cases where the sickness symptoms were diminished while using the VR applications and maybe this is a consequence of the fact that they were distracted in the virtual environment. Another note regarding the collected data is the fact that **User3** had very suspicious results as both post-exposure questionnaires had 2-3 symptoms with no grade selected (None, Slight, Medium or Severe) and we registered all the records with 0 as it is mentioned in the questionnaire source[68].

4.2.7.2 Presence Questionnaires

The presence questionnaire addressed 19 questions (detailed in Appendix 2) to assess the level of presence of the users after utilizing each application. We used a 7-points Likert scale for this questionnaire and we graded each response with values from 0 to 6. The presence questionnaire has 5 subscales: *Realism*, *Affordance to act*, *Interface quality*, *Affordance to examine* and *Self-evaluation of performance*. The questions that entered in each subscale calculus are as well detailed in Appendix 2. We focused on these subscales results instead of looking at individual questions. Figures 4.101 – 4.105 display each user's results per scale and Fig. 4.106 showcases the obtained overall results. All graphics contain the obtained median values.

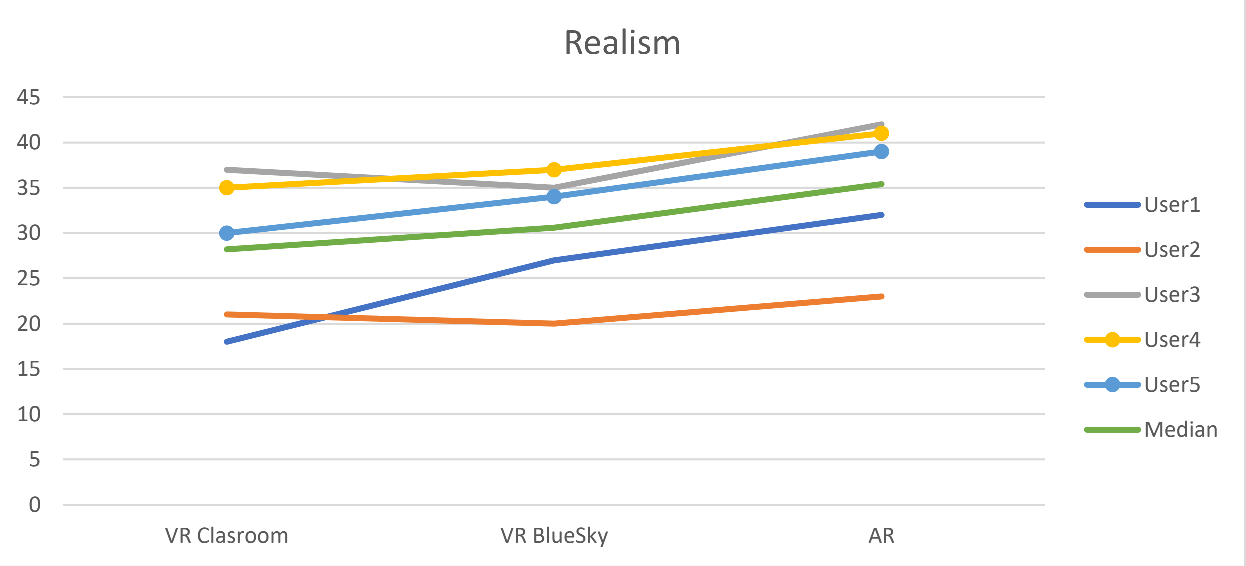


Figure 4.101 – Presence Questionnaires - Realism subscale

If we look at the median values, we can observe that the realism improved with each application although the scores differences are not that dramatic. Some of the users considered that the VR application with no classroom background (VR BlueSky) felt less real and all of them considered that the marker-based AR application had a higher degree of realism compared with the other two applications. In this case it would have been interesting to see the scores obtained by a markerless AR application.

---

[68] https://www.twentymilliseconds.com/html/ssq-scoring.html

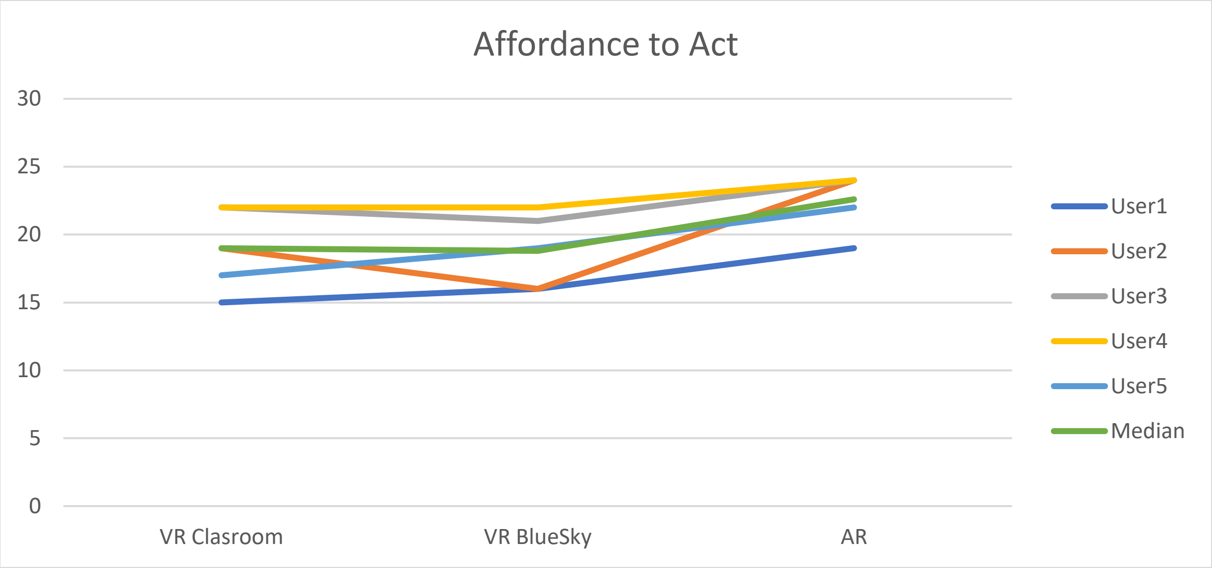


Figure 4.102 – Presence Questionnaires - Affordance to act subscale

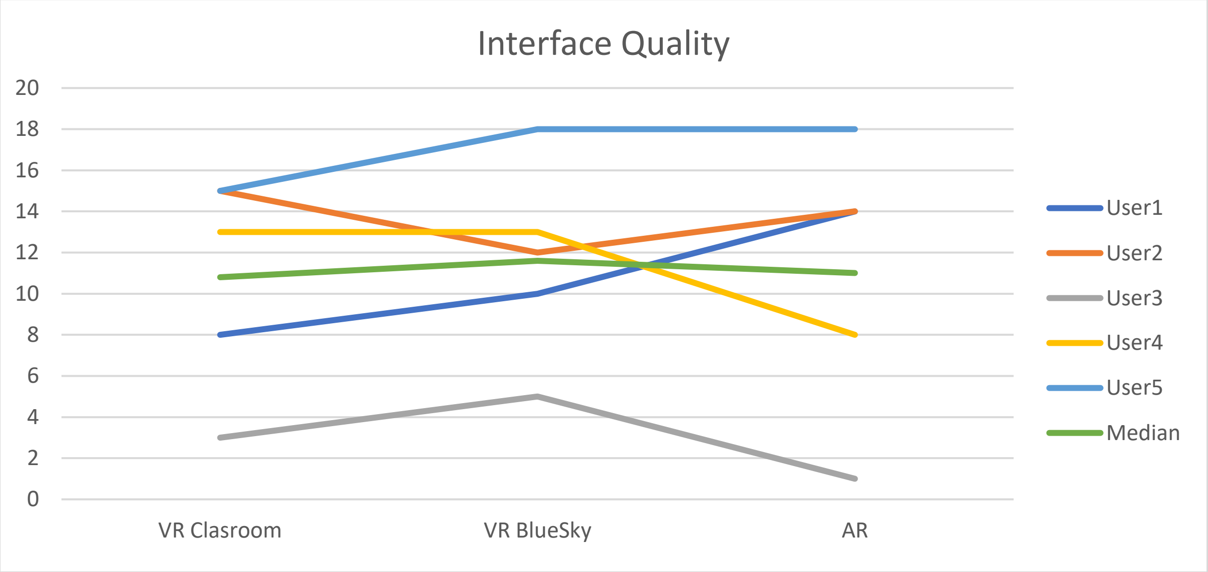


Figure 4.103 – Presence Questionnaires - Interface Quality subscale

A surprising result was the one regarding the *Interface Quality*, as we expected to see significant higher numbers for the AR application compared with the VR ones.

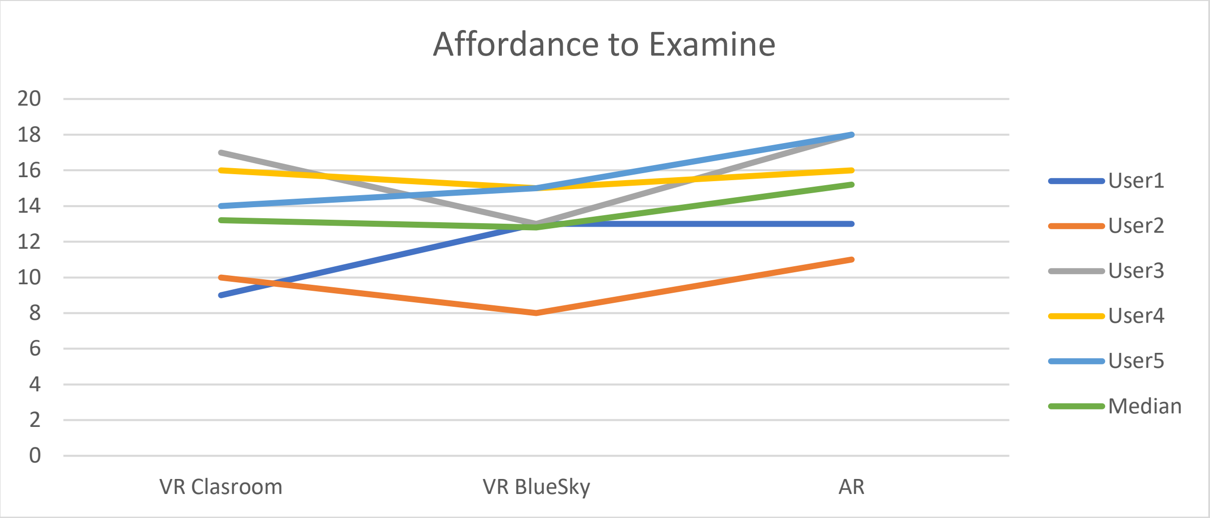


Figure 4.104 – Presence Questionnaires - Affordance to examine subscale

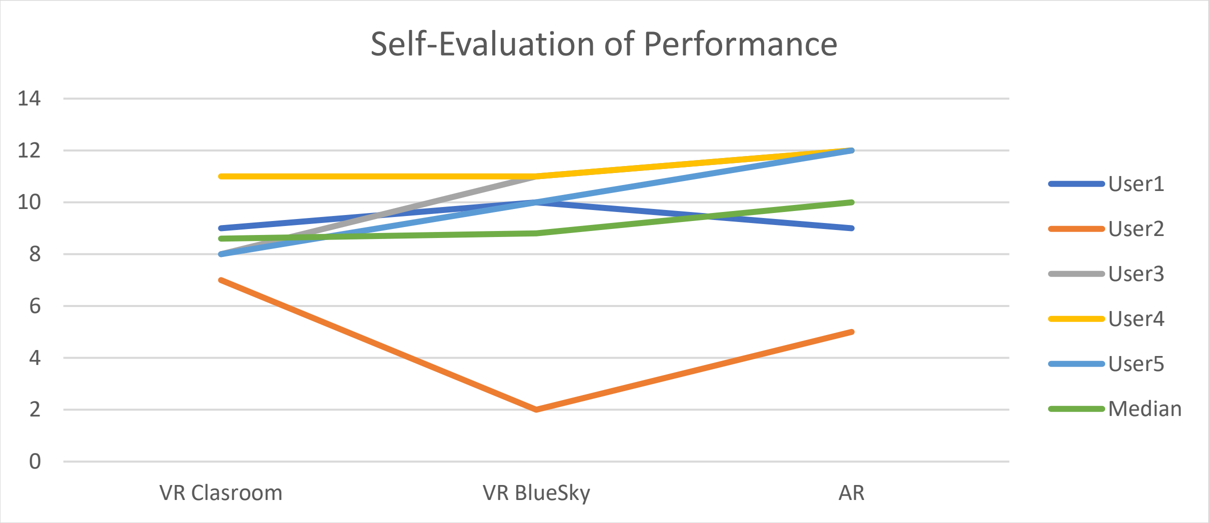


Figure 4.105 – Presence Questionnaires - Self-Evaluation of Performance subscale

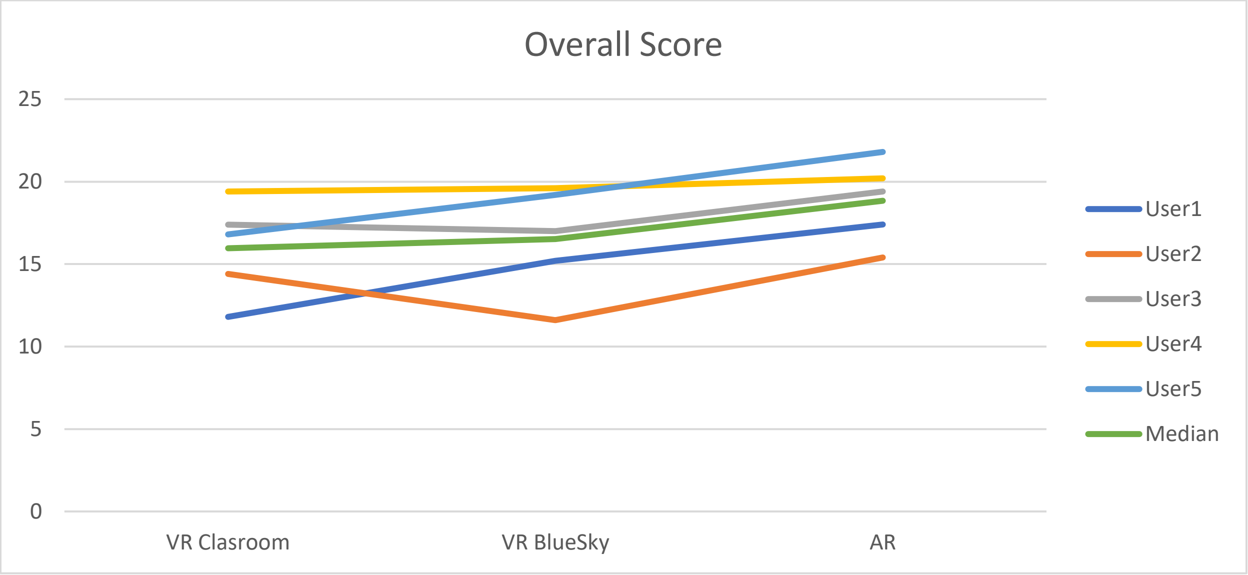


Figure 4.106 – Presence Questionnaires - Overall Score

The overall results confirmed the initial assumptions made at the start of this research that the users will feel more present in the AR environment compared with the VR ones. However, we expected the VR application created for cybersickness improvement (BlueSky) with no classroom background to give some poorer results for the presence questionnaire compared with the VR classroom one since the second one aimed to resemble as much as possible with the real environment. Although, users mentioned the fact that even though the classroom background looked very good, they felt it distracted them from the task they had to perform and that felt more comfortable with the information panels from VR BlueSky application being positioned closer to the main character and minimizing the head movement.

4.2.7.3 Users Data Questionnaires

This last section contains the gathered information that contained users' data extracted from 2 questionnaires:

- **Q1b**: that asked generic details about the respondents and their previous experience and opinion versus the VR and AR environments.
- **Q4a**: that asked general feedback regarding the applications and their preferences for one versus another as well as their fit with the presented content. The questions from both questionnaires are detailed in the Appendix 3 section.

Another important aspect is the fact that while testing the VR applications, the users wore a noise canceling headset (*Bose Quiet Comfort 35*) that played music suited for studying to isolate them as much as possible from the external environment.

Our experiment included 5 users (2 females and 3 males) within 2 age groups: *18-30* and *31-40*. They considered that they have *high* technical skills and *high* and *very high* interest in technology. All of them knew what VR is while only one didn't knew what AR is. Also, 3 out of 5 respondents said that they have previously tried VR applications and that they didn't felt sick after they used them. Two users said that their VR experience was *neither good or bad* while one mentioned that it was a *very good* experience as all 3 of them felt present in the displayed virtual environment. Regarding the previous AR applications usage, 3 users responded that they had a *good* experience while one said that the experience was *neither good or bad*. One user considers both AR and VR interesting while the other 3 said that they prefer the AR applications. For this assessment was excluded the one user that said that he/she didn't knew what AR is.

Regarding their feedback for the applications described in this thesis, two users said that they rate the overall experience *good* while 3 responded that they considered it *very good*.

From the total users, four said that they preferred augmented reality application versus the virtual reality ones while one said that he/she preferred the virtual reality application with no background as no one favored the VR application with the classroom background. During the experiment the users mentioned the fact that the second VR application was more suited for them because the displayed elements were closer to the 3D model and they could see the information better, along the fact that they felt better in the virtual environment because there was no background environment (although one complained about the lighting conditions and the chosen color – blue sky). They mentioned the fact that it felt that the classroom environment was a disturbance factor and couldn't focus as well on the tasks. On the same note, regarding the disturbance factors, three users *agreed* the fact that the lack on interruptions within VR applications testing was beneficial while another one *strongly agreed* with this. Surprisingly, one user *completely disagreed* with this approach (Fig 4.107).

While all of them responded that the app with AR environment is technically more suited for the learning experience, for VR one we didn't saw complementary responses as these questions are opposing themselves in this experiment. Two users said that they consider VR technically more reliable for the learning experience, other two disagreed and one was neutral.

Four users *agreed* and *strongly agreed* with the fact that the applications are easy to use and have a low learning curve while another one *completely disagreed*.

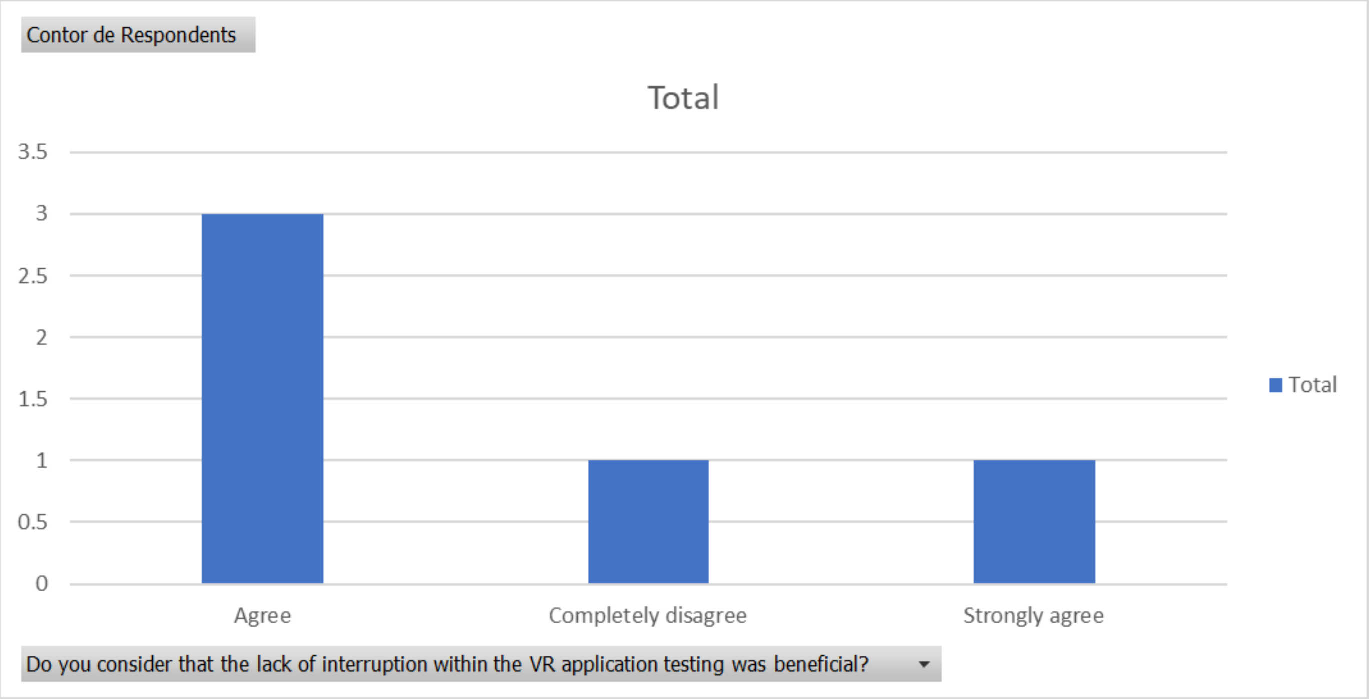


Figure 4.107 – Users' feedback regarding the lack of interruptions during VR applications testing

4.3 CONCLUSIONS

In this chapter we presented technical choices, implementation details and performance analysis of the interactive biomechanics lessons applications. We managed to build a low-cost, mobile friendly solution that has a high potential to reach to a large number of users in case it is extended. The various implementation details applied on different cases are briefly explained as our aim was the assessment of each environment's potential. The proposed solution had various directions changes during the implementation based on the obtained results as we tried to adapt it to obtain the best solution in the available time and budget.

Motion tracking was used on both projects that we detailed in this thesis and for this part a Kinect sensor was used again for recording the motion of an observed user. To obtain more realistic results, by improving the 3D models superimposing over the user's image, a solution based on OpenCV was incorporated in the developed system. A face detection feature was used to scale and transform the rendered model at runtime and a few results were showcased.

The performance results obtained for the markerless AR application are confirming the fact that this approach needs to be implemented on systems with much higher capabilities than a mobile device has. The results were obtained only with the bones models and not with skin or muscles that are much more complex and without the simulation of the soft tissue deformation which is also a computational expensive operation. We could also observe the fact that during Kinect skeletal tracking the application's performance consistently drops. Taking into account the fact that Microsoft recently announced that moving forward they are offering only support for Kinect technology, the usage of a Kinect sensor in this type of applications is not the best option for further development. We consider that new technology such as VicoVR is a much better fit as it has its own processing source that minimizes the display device overhead during the skeletal tracking.

The most interesting point from this chapter is the results assessment based on users' testing and feedback. One of the questions of this research was the fact that in AR applications users should feel more present while the minimization of external disturbance factors will aid the learning experience. Both assumptions were confirmed although we expected a clearer difference. Another interesting point is the fact that while we tried to create a world that resembles as much as possible with the reality in an VR scenario (with the classroom background) we actually managed to introduce disturbance factors as the users felt distracted by its composition. The majority of users said prior the experiment that they preferred AR applications and the same trend was somehow kept at the end of our experiment. One user preferred the VR application with no background created to improve cybersickness. These responses are actually in the same trend with the market forecasts. If initially it was tough that VR would have a great evolution along the years, now AR seems to have the lead between these two thus both technological systems have overall positive feedback.

As a future perspective we can think of algorithms and solution based only on computer vision for full body skeletal tracking thus the current ones are just at the start and it required more research to be applied at large scale even though there are a few approaches that seem to be promising [RD15]. In this direction, one of the newest approaches is to use human pose estimation implemented with TensorFlow[69]. We managed to get the current implementation and to set the appropriate parameters to run acceptable on our machines. This solution is implemented in JavaScript and offers the possibility to obtain human poses in real time in a browser and this is available for PC and mobile devices as well. We added Three.js[70] (JavaScript 3D Library) to be able to render the 3D elements. Even though the visualization and functionality has still a long way to come this seems to be a viable approach to track the movements of an observed user in real time without additional sensors. However, in this case the performance has a tradeoff with the tracking quality as determined by the parameters' settings (e.g. **output**

---

[69] https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5

[70] https://threejs.org/

stride that affects the height and the width of the layers in the neural network. The lower the value of the output stride the higher the accuracy but slower the speed[71]).

---

[71] https://storage.googleapis.com/tfjs-models/demos/posenet/camera.html

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 THE ORIGINAL CONTRIBUTIONS OF THIS THESIS

A large part of this thesis' content is practice based research as it is composed by various experimental results or novel implementations.

Chapter 3 contains the author's contribution to TRAVEE – an informatic solution based on VR for neuromotor rehabilitation. The first part tackled the adaptation of the rendered 3D models based on the patient's body characteristics such as the age, weight, height, skin and hair color. This is followed by the initial work at the VR module where the content was displayed using an Oculus Rift device and the results were published at CSCS15 conference [AV15b]. Another contribution is the development of the kinematics module as the system incorporated two tracking devices: Leap Motion and Kinect. The obtained results were disseminated at EHB2015 conference [AV15a].

Chapter 4 contains the complete details of a solution based on VR and AR in medical education. It is designed and built completely by the author and the obtained results are analyzed based on the visual feedback and users' responses to questionnaires. Not only the efficiency of the implemented methods is considered but also the responses to the presence and simulator sickness related questions. The initial proposal of this solution that aims to improve the learning process in biomechanics study by using VR and AR was published at ICERI2016 conference and received positive feedback [AV16b].

The author implemented a novel solution for obtaining rigged 3D models for bones, skin and muscles of the human body based on medical images. One of the most important aspects is the fact that the 3D model generation was obtained for the whole body not for smaller regions, as the previously available solutions, bringing new content for advanced visualization. The results were published at CSCS17 conference [AV17].

The tests section contains various experimental results obtained while testing the available AR and VR technologies. The results were obtained with Oculus Rift, Google Cardboard, Gear VR, HoloLens Emulator and Vuforia platform. Another contribution is related to the real time motion tracking of a human body. This part provides details regarding the whole-body tracking of an observed user and rendering the 3D models accordingly with the detected movements. The initial assessment of the viable solutions was published into an article published at MVAR workshop from ICMI2016 conference [AV16a].

The author implemented a solution based on a tracking sensor (Kinect) and computer vison (OpenCV) to display the 3D model superimposed over the tracked user's image at the same scale into an AR environment. The initial results of this approach were published at CSCS17 conference [AV17].

The biomechanics lessons implementation and results assessment were included into an article published in the Scientific Bulletin of UPB [AV18]. The most important original contribution of this thesis is related with the results obtained after testing the developed applications. The VR application that was placed into a classroom environment generated an increase of the simulator sickness symptoms of the users and they were diminished with the second VR application that targeted the cybersickness improvement. In some cases, the sickness symptoms were diminished while using VR and this might be a consequence of the fact that they were distracted in the virtual environment.

The results obtained from presence questionnaires confirmed the assumptions of this research that the users will feel more present in the AR environment compared with the VR ones. However, we expected that the VR application created for cybersickness improvement (BlueSky), with no classroom background, to give some poorer results for the presence questionnaire compared with the VR classroom one since the second one aimed to resemble as much as possible with the real environment. The users mentioned the fact that even though the classroom background looked very good, they felt it distracted them from the task they had to perform and that felt more comfortable with the information being positioned closer to the character and minimizing the head movement. Also, they appreciated the fact that the lack of interruptions within VR applications testing was beneficial. Overall, the augmented reality application was preferred in most cases.

5.2 CONCLUSIONS

This thesis is focused on solutions based on virtual and augmented reality in healthcare and two main subjects were discussed: rehabilitation and medical education. The solutions covered in this thesis are complex as they contained realistic 3D models and real-time motion tracking of an observed user in various combinations. We had the opportunity to add motion tracking to both VR and AR. Even though the functionality exists, there are future improvements that could be applied to these implementations. We tried to improve the tracking performance, the encountered issues with some poses and the visual output displayed in the applications. Unfortunately, due to time restrictions we couldn't have applied all the improvements that we wanted, and we aim to do this in the future.

The original contributions of this thesis were validated by various experimental results and the proposal of a novel solution that targets both VR and AR systems. Not all the features were at the desired level of quality as the time was limited or alternative solutions would have required additional costs. The focus was to design and implement a mobile friendly solution since the market forecasts and current indices show a high interest into this area. Also, the cost of the proposed solution was an important aspect as we had to develop it with minimal additional costs. We believe that this novel solution is an excellent candidate for a funded project and in that scenario, we could deliver a better visual and informational content.

5.3 FUTURE PERSPECTIVES

The future perspectives of the current research are based mainly on the registered progress in the Interactive Biomechanics Lessons project. A first area of improvement will be the quality enhancements and skinning adjustments to the 3D models of muscles and skin obtained from medical images. Moving forward, we are considering the extension of the existing biomechanics lessons with a more mathematical approach to the biomechanics applications. The simulation of muscles deformation and its visualization in the VR and AR scene is the most interesting point of the future perspective. This subject was already tacked by various researches and it would be a novelty to be applied to virtual and augmented reality though we are aware of the computational requirements challenges.

Another point of interest will be the thoroughgoing study of the full body motion tracking. An interesting point will be the acquisition of novel tracking sensors that are compatible with the mobile development, but this would require funding. Another solution with a high potential in the future is to build an efficient skeletal tracking solution based only on computer vision which will bring the idea of mobility to a whole new level.

Acronyms

ADL	Activities of Daily Living
AR	Augmented Reality
BCI	Brain Computer Interface
BMI	Body Mass Index
CAVE	Cave Automatic Virtual Environment
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
DK	Development Kit
DOF	Degree of Freedom
EMG	Electromyography
EPPO	Electric Powered Prehension Orthosis
FBX	FilmBox file format
FES	Functional Electrical Stimulation
FOV	Field of View
HMD	Head-Mounted Display
HT	Head and Torso
HTLL	Head, Torso and Lower Limbs
IBL	Interactive Biomechanics Lessons
LOD	Level of Details
LL	Lower Limbs
MRA	Magnetic Resonance Angiography
MRI	Magnetic Resonance Imagistics
NMES	Neuromuscular Electrical Stimulation
OGRE	Object-Oriented Graphics Rendering Engine
OHMD	Optical Head-Mounted Display
OR	Operating Room
PET	Positron Emission Tomography
PTM	Personalized Training Module
PVM	Patient Virtual Model
RGB	Red Green Blue
RGS	Rehabilitation Game System
sEMG	Surface Electromyography
SPECT	Single-photon emission computer tomography
SSQ	Simulator Sickness Questionnaire
STL	STereoLitography file format
TVM	Therapist Virtual Model
UI	User Interface
ULH	Upper Limbs Head
UWP	Universal Windows Platform
VR	Virtual Reality

Bibliography

[AG08] A. Gorini and G. Riva, "*Virtual Reality in Anxiety Disorders: the past and the future*, in **Neurotherapeutics**, vol. 8, no. 2, pp. 215-233, 2008.

[AK16] A. Karashanov, A. Manolova and N. Neshov, "*Application for Hand Rehabilitation Using Leap Motion Sensor based on Gamification Approach*" in **International Journal of Advance Research in Science and Engineering**, Volume 5, Issue 2, ISSN: 2319-8354, February 2016.

[AV15a] A. Voinea, A. Moldoveanu, F. Moldoveanu and O. Ferche, "*Motion Detection and Rendering for Upper Limb Post-Stroke Rehabilitation" at* **E-Health and Bioengineering Conference**, Iasi, Romania, November 19-21, DOI:10.1109/EHB.2015.7391471, ISBN: 978-1-4673-7544-3, pages1-4, 2015.

[AV15b] A. Voinea, A. Moldoveanu and F. Moldoveanu, "*3D Visualization in IT Systems Used for Post Stroke Recovery: Rehabilitation Based on Virtual Reality*" at **CSCS20: The 20th International Conference on Control Systems and Computer Science**, 27-29 May, Bucharest, Romania. 10.1109/CSCS.2015.123, p856-862, ISBN: 978-1-4799-1779-2, 2015.

[AV15c] A. Voinea, A. Moldoveanu, F. Moldoveanu and O. Ferche, "*ICT Supported Learning for Neuromotor Rehabilitation - Achievements, Issues and Trend*s" at **The International Scientific Conference eLearning and Software for Education**, Bucharest, Romania, Issue 1, p594-601. 8p, 2015.

[AV16a] A. Voinea, A. Moldoveanu and F. Moldoveanu, "*Bringing the Augmented Reality Benefits to Biomechanics Study"* in **MVAR '16 Proceedings of the 2016 workshop on Multimodal Virtual and Augmented Reality**, Tokyo, Japan, DOI: 10.1145/3001959.3001969, ISBN: 978-1-4503-4559-0, November 2016.

[AV16b] A. Voinea, A. Moldoveanu and F. Moldoveanu, "*Efficient Learning Technique in Medical Education Based on Virtual and Augmented Reality*" in **ICERI2016 Proceedings from 9th Annual International Conference of Education, Research and Innovation,** Seville, Spain, DOI: 10.21125/iceri.2016.0975, ISBN: 978-84-617-5895-1, November 2016.

[AV17a] A. Voinea, F. Moldoveanu and A. Moldoveanu, "*3D Model Generation and Rendering of Human Musculoskeletal System Based on Image Processing*" at **the 21st**

International Conference on Control Systems and Computer Science, Bucharest, Romania, 2017.

[AV18] Alexandra Voinea and Florica Moldoveanu, "*A Novel Solution Based on Virtual and Augmented Reality for Biomechanics Study*" in **Scientific Bulletin of UPB**, Series C. vol. 80, no.2/2018, ISSN 2286-3540, pp.29-40.

[BG09] B. Girard, V. Turcotte, S. Bouchard and Br. Girard, "*Crushing Virtual Cigarettes Reduces Tobacco Addiction and Treatment Discontinuation*" in **CyberPhychology & Behavior**, vol. 12, no. 5, 2009.

[BK15] B. Khelil and H. Amiri, "*Using Kinect System and OpenCV Library for Digits Recognition*" in **International Journal of Computer Science and Software Engineering**, Volume 4, Issue 12, ISSN: 2409-4285, pp. 315-322, December 2015.

[BP14] A. Bauer, F. Paclet, V. Cahouet, A. Dicko, O. Palombi, F. Faure and J.Trocaz, "*Interactive Visualization of Muscles Activity during Limb Movements*" in **Eurographics Workshop on Visual Computing for Biology and Medicine**, 2014.

[CB13] C. Botella, A. Garcia-Palacios, Y. Vizcaino, R. Herrero, R. M. Banos and Miguel Angel Belmonte, "Virtual *Reality in the Treatment of Fibromyalgia: A Pilot Study*" in **Cyberpsychology, Behavior, and Social Networking**, Volume 16, Number 3, DOI: 10.1089/cyber.2012.1572, 2013.

[CK14] C. Kamphuis, E. Barsom, M. Schijven and N. Cristoph, "*Augmented Reality in medical education?*", in **Perspectives on Medical Education**, vol. 3, 2014, pp.3, pp. 300-311, 2014.

[DL15] D. Lee, G. Baek, Y. Lim and H. Lim, "*Virtual Reality Contents using Oculus Rift and Kinect*", in **Mathematics and Computers in Science and Industry**, ISBN: 978-1-61804-327-6, 2015.

[DM10] D. Michaliszyn, A. Marchand, S. Bouchard, MO. Martel and J. Poirier-Bisson, "*A Randomized, Controlled Clinical Trial of In Vitro and In Vivo Exposure for Spider Phobia*" in **Cyberpsychology, Behavior and Social Networking**, Volume 13, Number 6, DOI: 10.1089/cyber.2009.0277, 2010.

[ECL11] E. Chen Lu, "*Development of an Upper Limb Robotic Device for Stroke Rehabilitation*", thesis for Master of Health Science Graduate Department of **Institute of Biomaterials and Biomedical Engineering in the University of Toronto**, 2011.

[ES03] E. Steele, K. Grimmer, B. Thomas, B. Mulley, I. Fulton and H. Hoffman, "*Virtual Reality as a Pediatric Pain Modulation Technique: A Case Study*", in **CyberPsychology & Behavior**, vol. 6, no. 6, 2003.

[FC14] F. Cutolo, P.D. Parchi and V.Ferarri, "*Video See Trough AR Head-Mounted Display for Medical Procedures*" at **IEEE International Symposium on Mixed and Augmented Reality**, 2014.

[FT14] F. Tecchia, G. Avveduto, R. Brondi, M.Carrozzino, M. Bergamasco and L.Alem, "*I'm in VR! Using your own hands in a fully immersive MR system*" in Proceedings **of the 20th ACM Symposium on Virtual Reality Software and Technology** (VRST 2014), p.73-76, 2014.

[GS15] G. Sankaranarayanan, B. Li, K. Manser, S. Jones, D. Jones, S. Schwaitzberg, C. Cao and S. De, "*Face and construct validation of a next generation virtual reality (Gen2-VR) surgical simulator*", in **Surgical Endoscopy**, June 2015.

[HH14] H. Hua and B. Javidi, "*A 3D Integral Imaging Optical See-Trough Head-Mounted Display*" in **Optics Express**, 22(11), 13484-13491. DOI:10.1364/OE.22.013484, 2014.

[HS16] Ha. Scharffenberger and E. van der Heide, "*Multimodal Augmented Reality – The Norm Rather Than the Exception*" in **Proceedings of the ACM Workshop on Multimodal Virtual and Augmented Reality**, DOI:10.1145/3001959.3001960, 2016.

[IACG15] I. A. Chicchi Gigloli, F. Pallavicini, E. Pedroli, S. Serino and Giuseppe Riva, "*Augmented Reality: A Brand-New Challenge for the Assessment and Treatment of Psychological Disorders*" in **Computational and Mathematical Methods in Medicine, Hindawi Publishing Corporation,** Volume 2015, Article ID 862942, 12p, DOI: http://dx.doi.org/10.1155/2015/862942, 2015.

[IM14] I. Mahalil, M. E. Rusli, A. M. Yusof, M. Z. M. Yusof and A. R. R. Zainudin, "*Virtual Reality-based Technique for Stress Therapy*" at the **4th International Conference on Engineering Technology and Technopreneurship** (ICE2T), Kuala Lumpur, Malaysia, 2014.

[JL02] J. Lee, J. Ku, D. Jang, D. Kim, Y. Choi, I. Kim and S. Kim, "*Virtual Reality Systems for Treatment of the Fear of Public Speaking Using Image-Based Rendering and Moving Pictures*", in **CyberPhychology & Behavior**, vol. 5, no. 3, 2002.

[JT05] J. Teran, E.Sifakis, S. S. Blemker, V. Ng-Thow-Hing, C. Lau and R. Fedkiw, *"Creating and Simulating Skeletal Muscle from the Visible Human Data Set"* in **IEEE Transactions on Visualization and Computer Graphics**, Volume 11, Issue 3, DOI: 10.1109/TVCG.2005.42, May-June 2005.

[JVWR14] *Overview: Virtual Reality in Medicine*, **Journal of Virtual Worlds Research**, ISSN:1941-8477, Lantern Part ½, Volume 7, No. 1, January 2014.

[KK14] K. Karolczak and A. Klepaczk, "*A stereoscopic viewer of the results of vessel segmentation in 3D magnetic resonance angiography images*", **in Proceedings of the International Conference on Signal Processing: Algorithms Architectures Arrangements and Applications**, Sept. 2014.

[LDLL14] L.D. Lledo, S. Ezquerro, F.J. Badesa, R. Morales, N. Garcia -Aracil and J.M. Sabater, "*Implementation of 3D visualization applications based on physical-haptics principles to perform rehabilitation tasks*" at **5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics** (BioRob), Sao Paulo, Brazil, August 2014.

[LDP11] L. De Paolis, G.Aloisio and M. Pulimeno, "*An Augmented Reality Application for the Enhancement of Surgical Decisions*" at **ACHI 2011 – The Fourth International Conference on Advances in Computer-Human Interactions**, ISBN:978-1-61208-117-5, pg.192-196, 2011.

[MCJ06] M. C. Juan, R. Banos, C. Botella, D. Perez, M. Alcaniz and C. Monserrat, "*An augmented reality system for the treatment of acrophobia: The sense of presence using immersive photography*" in **Presence: Teleoperators and Virtual Environments**, vol.15, no.4, pp.393-402, 2006.

[MCJ10] M. C. Juan and D.Prez, "*Using Augmented and Virtual Reality for the development of acrophobic scenarios. Comparison of the levels of presence and anxiety*" in **Computers and Graphics**, vol.34, no.6, pp.756-766, 2010.

[MM14] M. Mekni and A. Lemieux, "*Augmented Reality: Applications, Challenges and Future Trends*" in **Applied Computational Science**, ISBN: 978-960-474-368-1, 2014.

[MCFM14a] M.C.F. Macedo and A. L. Apolinario Jr., "*A Semi-Automatic Markerless Augmented Reality Approach for On-Patient Volumetric Medical Data Visualization*" at **XVI Symposium on Virtual and Augmented Reality**, 978-1-4799-4261-9/14, DOI: 10.1109/SVR.2014.29, 2014.

[MCFM14b] M. Macedo, A. Apolinario, A. Souza and G. Giraldi, "*High-Quality On-Patient Medical Data Visualization in a Markerless Augmented Reality Environment*", in **SBC Journal in Interactive Systems**, vol. 5, no. 3, 2014.

[MCFM15] M. C.F. Macedo, C. S. de B. Almeida, A.C.S. Souza, J. P/ Silva, A. L. Apolinario Jr. and G.A. Giraldi, "*A Markerless Augmented Reality Environment for Medical Data Visualization*" in: **Proceedings of Conference on Computer Graphics, Patterns and Images** (SIBGRAPI 2015), Salvador, Brazil, 2015.

[MK14] M. Khademi, H. Mousavi Hondori, A. McKenzie, L. Dodakian, C. Videira Lopes and S. C. Cramer, "*Free-hand Interaction with Leap Motion Controller for Stroke Rehabilitation*" in **Proceedings of CHI EA '14 – Extended Abstracts of Human Factors in Computing Systems**, ISBN: 978-1-4503-2474-8, DOI: 10.1145/2559206.2581203, 2014.

[MLA97] ML. Aisen, HI. Krebs, F. McDowell and BT. Volpe, "*The effect of robot-assisted therapy and rehabilitative training on motor recovery following stroke*" in **Archives of Neurology**, 54(4):443-6, April 1997.

[MSC10] M.S. Cameirao, S. Bermudez i Badia, E. Duarte Oller and P. FMJ Vershure, "*Neurorehabilitation using the virtual reality based Rehabilitation Gaming System: methodology, design, psychometrics, usability and validation*" in **Journal of NeuroEngineering and Rehabilitation,** 2010.

[NF14] N. Friedman, V. Chan, AN. Reinkensmeyer, A. Beroukhim, GJ. Zambrano, M. Bachman and DJ. Reinkensmeyer, "*Retraining and Assessing Hand Movement After Stroke Using the MusicGlove: comparison with conventional hand therapy and isometric grip training*" in **Journal of NeuroEngineering and Rehabilitation**, DOI: 10.1186/1743-0003-11-76, 2014.

[OB05] O. Bimber, R. Raskar and M. Inami, "*Spatial Augmented Reality*", **AK Peters Wellesley**, 2005.

[OF15] O. Ferche, A. Moldoveanu, D. Cinteza, C. Toader, F. Moldoveanu, A. Voinea, C. Taslitchi, "*From Neuromotor Command to Feedback: A survey of techniques for rehabilitation through altered perception*" at **E-Health and Bioengineering Conference (EHB)**, Iasi, Romania, November 2015.

[OMF15] O. M. Ferche, A. Moldoveanu, F. Moldoveanu, A. Voinea, V. Asavei and I. Negoi, "*Challenges and issues for successfully applying virtual reality in medical rehabilitation*" at **The International Scientific Conference eLearning and Software for Education**, Bucharest, Romania, Issue 1, p494-501, 8p, April 2015.

[PM14] P. Maciejasz, J. Eschweiler, K. Gerlash-Hahn, A. Jansen-Troy and S. Leonhardt, "*A Survey on Robotic Devices for Upper Limb Rehabilitation*" in **Journal of NeuroEngineering and Rehabilitation**, DOI: 10.1186/1743-0003-11-3, 2014.

[RD15] R. Damle, A. Gurjar, A. Joshi and K. Nagre, "*Human Body Skeleton Detection and Tracking*" in **International Journal of Technical Research and Applications**, Volume 3, Issue 6 (November-December 2015), pp.222-225, e-ISSN:2320-8163, 2015.

[RK01] R. Kuo, F. Delvecchio, and G. Preminger, "*Virtual Reality: Current Urologic Applications and Future Developments*", in **Journal of Endourology**, vol. 15, no. 1, Feb. 2001.

[RM00] R. Myers and T. Bierig, "*Virtual Reality and Left Hemineglect: A Technology for Assessment and Therapy*", in **CyberPshychology & Behavior**, vol. 3. No. 3, 2000.

[RTA97] R. T. Azuma et al, "*A survey of Augmented Reality*" in **Presence**, vol.6, no. 4, pp.355-385, 1997.

[SD14] S. Davis, K. Nesbitt and E.Nalivaiko, "*A Systematic Review of Cybersickness*", at **Interactive Entertainment Conference**, Newcastle, Australia, 2014.

[SJH09] SJ. Housman, KM. Scott and DJ. Reinkensmeyer, "*A randomized controlled trial of gravity-supported, computer-enhanced arm exercise for individuals with severe hemiparesis*" in **Neurorehabilitation and Neural Repair**, February 2009.

[SG15] S. Guennouni, A. Ahaitouf and A. Mansouri, "*A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Local Binary Patterns in Several Platforms*", in **Modelling and Simulation in Engineering**, Article ID 948960, 2015.

[SH05] S. Hesse, C. Werner, M. Pohl, S. Rueckriem, J. Mehrholz and ML. Lingnau, "*Computerized arm training improves the motor control of the severely affected arm after stroke: a single-blinded randomized trial in two centers*" in **Stroke**, DOI: 10.1161/01.STR.0000177865. 37334.ce, August 2005.

[SL15] S.Livatino, L. De Paolis, M. D'Agostino, A. Zocco, A. Agrimi, A. De Santis, L. Bruno and M. Lapressa, "*Sterepscopic Visualization and 3-D Technologies in Medical Endoscopic Teleoperation*", in **IEEE Transactions on Industrial Electronics**, vol. 62, no. 1, Jan. 2015.

[SS12] Sanni Siltanen, in book: ***Theory and applications of marker-based augmented reality***, VTT Science 3, ISBN: 978-951-38-7449-0, 2012.

[SY14] S. Yoshida, K. Kihara, H. Takeshita and Y. Fuji, "*Instructive head-mounted display system: pointing device using a vision-based finger tracking technique applied to surgical education*", in **Videosurgery Mininv**, vol. 9, no. 3, 2014.

[TB12] T. Blum, V. Kleeberger, C. Bichlmeier and N. Navab, "*Miracle: An Augmented Reality magic mirror system for anatomy education*" in **IEEE Virtual Reality Workshops (VRW)**, Costa Mesa, CA. DOI: 10.1109/VR.2012.6180909, 2012.

[TM14] T. Mota, M. Mello, L. Nedel, A. Maciel and F. Faria, "*Mobile Simulator for Risk Analysis*", at **XVI Symposium of Virtual and Augmented Reality**, DOI: 10.1109/SVR.2014.52, 2014.

[TO96] T. Ojala, M. Pietikäinen, and D. Harwood, "*A comparative study of texture measures with classification based on feature distributions*" in **Pattern Recognition**, 29, 1996.

[YC01] Y. Choi, D. Jang, J. Ku, M. Shin and S. Kim, "*Short-Term Treatment of Acrophobia with Virtual Reality Therapy (VRT): A Case Report*" in **CyberPsychology & Behavior**, vol. 4, no. 3, 2001.

[YL14] Y. Liu, "*Virtual Neurosurgical Education for Image-guided Deep Brain Stimulation Neurosurgery*", in **2014 International Conference on Audio, Language and Image Processing**, July 2014.

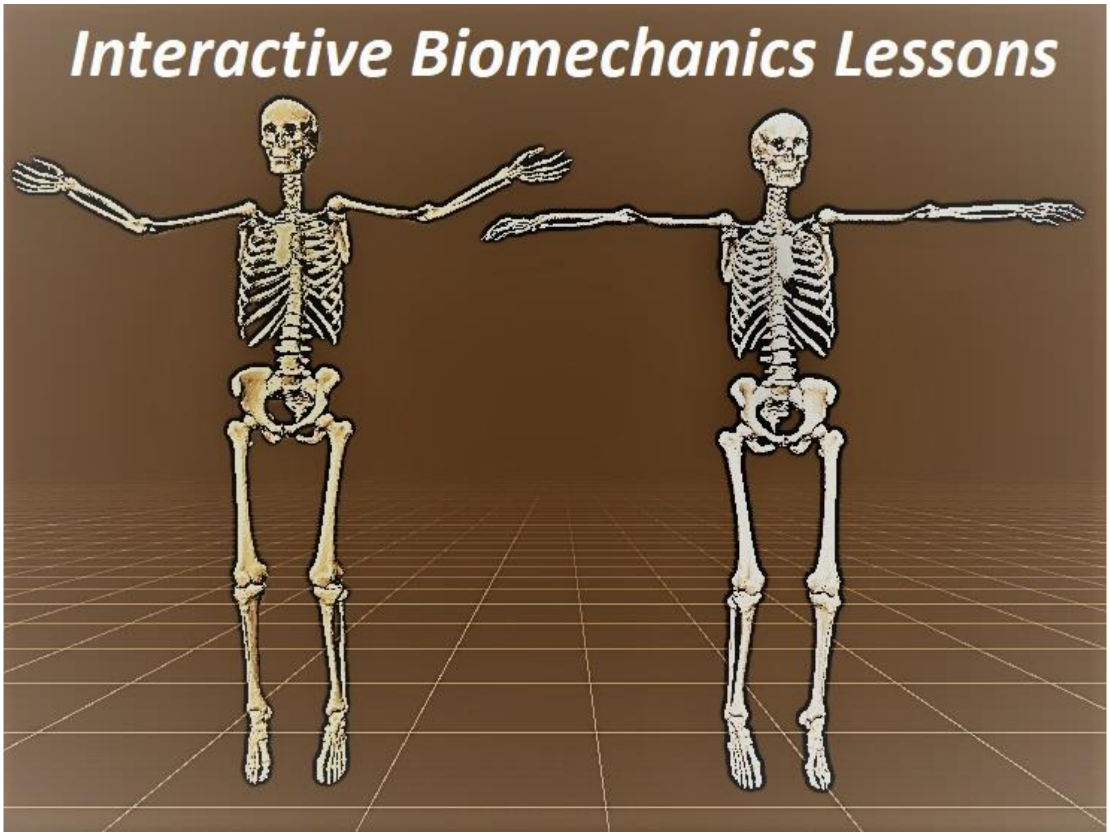
[YR06] Yann Rodriguez, "*Face Detection and Verification using Local Binary Patterns*", PHD Thesis at École Polytechnique Fédérale De Lausanne, 2006.

[ZG15] Z. Geyao, "*Walking in Virtual Reality*", **Design & Technology, Parsons the New School for Design,** 2015.

[ZY15] Ziv Yaniv and Cristian Linte, "*Applications of Augmented Reality in Operating Room*", in book: **Fundamentals of Wearable Computers and Augmented Reality**, second Edition, pp.485-518, DOI: 10.1201/b18703-23, 2015.

Appendices

Appendix 1: Print ready marker-based AR target image



Appendix 2

PRESENCE QUESTIONNAIRE[72]

(Witmer & Singer, Vs. 3.0, Nov. 1994) *
Revised by the UQO Cyberpsychology Lab (2004)

Characterize your experience in the environment, by marking an "X" in the appropriate box of the 7-point scale, in accordance with the question content and descriptive labels. Please consider the entire scale when making your responses, as the intermediate levels may apply. Answer the questions independently in the order that they appear. Do not skip questions or return to a previous question to change your answer.

WITH REGARD TO THE EXPERIENCED ENVIRONMENT

1. How much were you able to control events?
|________|________|________|________|________|________|________|
NOT AT ALL SOMEWHAT COMPLETELY

2. How responsive was the environment to actions that you initiated (or performed)?
|________|________|________|________|________|________|________|
NOT RESPONSIVE MODERATELY COMPLETELY RESPONSIVE

3. How natural did your interactions with the environment seem?
|________|________|________|________|________|________|________|
EXTREMELY BORDERLINE COMPLETELY
ARTIFICIAL NATURAL

4. How much did the visual aspects of the environment involve you?
|________|________|________|________|________|________|________|
NOT AT ALL SOMEWHAT COMPLETELY

5. How natural was the mechanism which controlled movement through the environment?
|________|________|________|________|________|________|________|
EXTREMELY BORDERLINE COMPLETELY
ARTIFICIAL NATURAL

6. How compelling was your sense of objects moving through space?
|________|________|________|________|________|________|________|
NOT AT ALL MODERATELY VERY
COMPELLING COMPELLING

7. How much did your experiences in the virtual environment seem consistent with your real-world experiences?
|________|________|________|________|________|________|________|
NOT MODERATELY VERY
CONSISTENT CONSISTENT CONSISTENT

8. Were you able to anticipate what would happen next in response to the actions that you performed?

---

[72] http://w3.uqo.ca/cyberpsy/docs/qaires/pres/PQ_va.pdf

|________|________|________|________|________|________|________|
NOT AT ALL SOMEWHAT COMPLETELY

9. How completely were you able to actively survey or search the environment using vision?
|________|________|________|________|________|________|________|
NOT AT ALL SOMEWHAT COMPLETELY

10. How compelling was your sense of moving around inside the virtual environment?
|________|________|________|________|________|________|________|
NOT MODERATELY VERY
COMPELLING COMPELLING COMPELLING

11. How closely were you able to examine objects?
|________|________|________|________|________|________|________|
NOT AT ALL PRETTY VERY
CLOSELY CLOSELY

12. How well could you examine objects from multiple viewpoints?
|________|________|________|________|________|________|________|
NOT AT ALL SOMEWHAT EXTENSIVELY

13. How involved were you in the virtual environment experience?
|________|________|________|________|________|________|________|
NOT MILDLY COMPLETELY
INVOLVED INVOLVED INVOLVED

14. How much delay did you experience between your actions and expected outcomes?
|________|________|________|________|________|________|________|
NO DELAYS MODERATE LONG
DELAYS DELAYS

15. How quickly did you adjust to the virtual environment experience?
|________|________|________|________|________|________|________|
NOT AT ALL SLOWLY LESS THAN
ONE MINUTE

16. How proficient in moving and interacting with the virtual environment did you feel at the end of the experience?
|________|________|________|________|________|________|________|
NOT REASONABLY VERY
PROFICIENT PROFICIENT

17. How much did the visual display quality interfere or distract you from performing assigned tasks or required activities?
|________|________|________|________|________|________|________|
NOT AT ALL INTERFERED PREVENTED
SOMEWHAT TASK PERFORMANCE

18. How much did the control devices interfere with the performance of assigned tasks or with other activities?

|________|________|________|________|________|________|________|

NOT AT ALL OMEWHAT INTERFERED INTERFERED GREATLY

19. How well could you concentrate on the assigned tasks or required activities rather than on the mechanisms used to perform those tasks or activities?

|________|________|________|________|________|________|________|

NOT AT ALL SOMEWHAT COMPLETELY

Realism Questions: 3, 4, 5, 6, 7, 10, 13

Affordance to Act Questions: 1, 2, 8, 9

Interface Quality Questions: 14, 17, 18 (all reversed)

Affordance to Examine Questions: 11, 12, 19

Self-Evaluation of Performance Questions: 15, 16

Simulator Sickness Questionnaire[73]

	Symptoms	**0 None**	**1 Slight**	**2 Moderate**	**3 Severe**
1	General Discomfort				
2	Fatigue				
3	Boredom				
4	Drowsiness				
5	Headache				
6	Eyestrain				
7	Difficulty focusing				
8	Salivation increase				
9	Salivation decrease				
10	Sweating				
11	Nausea				
12	Difficulty concentrating				
13	Mental depression				
14	"Fulness of the head"				
15	Blurred vision				
16	Dizziness with eyes open				

---

[73] https://www.twentymilliseconds.com/html/ssq-scoring.html

17	Dizziness with eyes closed				
18	Vertigo				
19	Visual flashbacks				
20	Faintness				
21	Aware of breathing				
22	Stomach awareness				
23	Loss of appetite				
24	Increased appetite				
25	Desire to move bowels				
26	Confusing				
27	Burping				
28	Vomiting				
29	Other				

Simulator sickness subcategories where are considered the following symptoms:

1. **Nausea**: General discomfort, Increased salivation, Sweating, Nausea, Difficulty concentrating, Stomach awareness and Burping.
2. **Oculomotor:** General discomfort, Fatigue, Headache, Eyestrain, Difficulty focusing, Difficulty concentrating and Blurred vision.
3. **Disorientation:** Difficulty focusing, Nausea, Fullness of head, Blurred vision, Dizziness with eyes open and Dizziness with eyes closed.

Appendix 3

Users Data Questionnaires

Q1b: User's Previous Experience

1. What is your gender?
   - A. Female
   - B. Male

2. What is your age?
   - A. <18
   - B. 18-30
   - C. 31-40
   - D. 41-50
   - E. > 50

3. How do you rate your computer skills?
   - A. None
   - B. Low
   - C. Average
   - D. High
   - E. Very High

4. Do you have interest in technology?
   - A. None
   - B. Low
   - C. Average
   - D. High
   - E. Very high

5. Do you know what Virtual Reality (VR) is?
   - A. Yes
   - B. No

6. Did you previously used a VR application?
   - A. Yes
   - B. No

7. **If you responded yes at the previous question**, how would you rate your experience?
   - A. Very Good
   - B. Good
   - C. Neither good or bad
   - D. Bad
   - E. Very bad

8. **If you responded yes at question no.6**. Have you felt present in the displayed virtual environment?
   A. Yes
   B. No

9. **If you responded yes at question no.6**. Did you felt sick while trying the VR application(s)?
   C. Yes
   D. No

10. Do you know what Augmented Reality (AR) is?
   A. Yes
   B. No

11. Did you previously used an AR application?
   C. Yes
   D. No

12. **If you responded yes at the previous question**, how would you rate your experience?
   A. Very Good
   B. Good
   C. Neither good or bad
   D. Bad
   E. Very bad

13. **If you responded yes question no.6 and 10,** which options suits you best?
   A. I prefer VR applications
   B. I prefer AR applications
   C. None of them seem interesting for me
   D. I consider both interesting

Q4a: User's General Feedback

14. How would you rate the overall experience with the tested applications?
   A. Very Good
   B. Good
   C. Neither good or bad
   D. Bad
   E. Very bad

15. Which application did you prefer?
   A. Virtual Reality with classroom background

   B. Virtual Reality without background
   C. Augmented Reality

16. Which application do you think is best suited for the displayed information?
   A. Virtual Reality with classroom background
   B. Virtual Reality without background
   C. Augmented Reality

17. Do you consider that the lack of interruption within the VR application testing was beneficial?
   A. Completely disagree
   B. Disagree
   C. Neutral
   D. Agree
   E. Strongly Agree

18. The app with AR function is technically more reliable for learning experience?
   A. Completely disagree
   B. Disagree
   C. Neutral
   D. Agree
   E. Strongly Agree

19. The app with VR function is technically more reliable for learning experience?
   A. Completely disagree
   B. Disagree
   C. Neutral
   D. Agree
   E. Strongly Agree

20. The apps are easy to use and have a low-learning curve.
   A. Completely disagree
   B. Disagree
   C. Neutral
   D. Agree
   E. Strongly Agree