Development of a Virtual Reality Simulation for ECG Interpretation and Clinical Skill Training in Medical Education

Tuğrul Hasan Furkan Abay

2312468

Link to project board on JIRA
Link to project repository on GITLAB

A thesis submitted for the degree of Master of Science in Computer Games.

Supervisor: Dr. Katerina Bourazeri
School of Computer Science and Electronic Engineering
University of Essex

August 2024

Acknowledgment

I would like to express my sincere gratitude to Dr. Katerina Bourazeri for her valuable support and feedback throughout the project.

I am also grateful to my partner, Ilayda, and family who were with me throughout this process. I would like to express my special thanks to Dr. Emir Can Yolsal who helped me with theoretical knowledge and a better understanding of the medical subject.

Finally, I would like to thank all the creators in the Appendix A section for the game assets I used in the project.

List of Abbreviations

Abbreviation	Definition
2D	Two-Dimension
3D	Three-Dimension
AI	Artificial Intelligence
APK	Android Package Kit
ECG	Electrocardiogram
EKG	Elektro-kardiographie
FBX	Film Box
FOV	Field of view
GPU	Graphic Processing Unit
HMD	Head Mounted Display
Hz	Hertz
NASA	National Aeronautics and Space Administration
RAM	Random Access Memory
UI	User Interface
USB-C	Universal Serial Bus Type-C
VR	Virtual Reality
XP	Experience Point

List of Figures

Figure 1 : Heart Conduction System [9] 12
Figure 2: The lead on the left side is used for chest [11] and the clamps on the right side are used for limb leads [12]. 13
Figure 3: Placement of electrodes on the body 15
Figure 4: Normal ECG waveform 15
Figure 5: Normal and abnormal P waves in case of P pulmonale 16
Figure 6 : ECG with Right Bundle Branch Block [16] 17
Figure 7: The area shown in red is the field of view obtained during the VR experience. (Adopted from [20]) 18
Figure 8: Normal visual field [21] 18
Figure 9: Meta quest 2's HMD and hand controller device [23] 19
Figure 10 : Screenshot from Acadicus ECG Simulator 20
Figure 11 : Screenshot from MedVR-EKG Simulator 21
Figure 12: Screenshot from Leonardo VR Simulator 21
Figure 13: User Persona created on Miro board. 23
Figure 14: Design of level flow created on Miro board. 24
Figure 15: Wireframe design of Theory section using Figma 24
Figure 16 : Reference design for Heart Conduction system 25
Figure 17: Reference design for chest electrodes placement game 25
Figure 18: Reference design for limb electrodes placement 26
Figure 19: Reference design for lead orientation 26
Figure 20: Reference design for ECG waveforms game 27
Figure 21 : Wireframe design of the Quiz section using Figma. 27
Figure 22: Room interior design of the simulation in Unity 28
Figure 23: 3D model created for the Heart Conduction System interaction section 29
Figure 24: The 6 views of the heart in a 6 chest lead model in the game 30
Figure 25 : Limb clamps placement game image. 31
Figure 26 : The image of the Precordial electrodes placement game 32
Figure 27 : The image of the ECG waveforms game 33
Figure 28 : The in-game view of the Case level section 34
Figure 29 : User interface transaction tree 35
Figure 30: Menu UI transactions in the game 36
Figure 31: Menu UI transactions in the game 36
Figure 32 : Quiz section in the game 37
Figure 33: The scene worked on in Unity and its contents 39
Figure 34: Camera component used in the XR Origin game object in the project 39
Figure 35: Directional light values used in the game scene 40
Figure 36 : Transparent material assigned to the human model surrounded by orange lines on the right 41
Figure 37: Learning part 2 screen where lead_scale animation is made 41
Figure 38: Lead_scale animator and animation 42

Figure 39 : Individual accessing Virtual Controller [32] 42
Figure 40: In-game view of hand tracker controllers 43
Figure 41 : An example of communication between managers when a task is completed 43
Figure 42 : Some important methods in UIManager 44
Figure 43: CompleteCurrenTaskCoroutine method in TaskManager 44
Figure 44 : Some important methods in LevelManager 45
Figure 45: Level task object assignments in the Level Manager contain these objects according to the game scene task and level index. 45
Figure 46: Level task object assignments in the Level Manager contain these objects according to the game scene task and level index. 46
Figure 47: Snippet of scripts that check the list of interactable objects in the HeartAnalysis script and finish the task 47
Figure 48: Actions that will occur when interactive objects are selected 47
Figure 49 : Some important methods in the SnapToPair script 48
Figure 50: Precordial electrodes placement game objects and script assignments 48
Figure 51 : Methods for updating UI and finishing sub-game in ECGSnapManager script 49
Figure 52: Game objects in the ECG waveforms game and ECGManager assignments on Unity 49
Figure 53: Methods that create ECG waveform using LineRenderer in ECG Generator script 50
Figure 54: Three different case panels at the case level and their assignments on the inspector. 51
Figure 55 : A section of the results of the survey created in Google form and presented to users 53
Figure 56 : A section of the results of the survey created in Google form and presented to users 54
Figure 57 : A section of the results of the survey created in Google form and presented to users 54
Figure 58 : A section of the results of the survey created in Google form and presented to users 55
Figure 59: A Gantt Chart 56
Figure 60 : A Work Breakdown Structure 56
Figure 61 : A Milestones and deliverables 57
Figure 62 : Screenshot of commits made throughout the project taken from gitlab 57
Figure 63 : Cumulative diagram which is a Kanban board showing to-do in progress and done stages. 57

List of Tables

Table 1: Nomenclature and coloring of electrodes according to AHA/AAMI and IEC 13
Table 2: Badges and degrees obtained with experience points 38

Abstract

With the advancement of Virtual Reality (VR) technology, interactive learning environments have opened new avenues in medical education. VR technology has the potential to create supportive simulations in teaching topics such as interpreting Electrocardiogram (ECG) reports. ECG interpretation is an important skill that requires the ability to quickly and accurately recognize heart rhythms and distinguish abnormal conditions in emergency situations. Existing ECG trainings are traditionally based on heart sound analysis, graphics, and theoretical training in a separate manner. However, these approaches are provided separately from practical applications.

This thesis proposes the development of a VR-based simulation that aims to improve the interpretation skills of ECG reports for medical students and medical professionals. Within the scope of the proposed VR application, users receive the theoretical training content interactively with both audio, visual, and 3D models and are allowed to practice ECG electrode placement and interpretation. By integrating visual and auditory elements, the experience of learning theoretical knowledge and practical skills in the same environment is provided. In the test phase, the application was tested with 5 different users, and a survey was conducted. Although it is difficult to express percentage and numerical improvement due to the lack of data, the subjects evaluated the simulation as more fun and interesting than traditional training, and stated that they would recommend it in terms of educational simulation.

As a result, this application can be used as a supporting material for the development of ECG report interpretation skills and can be a pioneer for future studies.

Keywords: Virtual Reality (VR), Medical Education, Electrocardiogram (ECG) Interpretation, Simulation-Based Learning, Medical Training

Table Of Contents

Acknowledgment 1
List of Abbreviations 2
List of Figures 3
List of Tables 4
Abstract 5
Table Of Contents 6
1. Introduction 8
2. Literature Review 10
2.1. Definition and Importance of the Topic 10
2.1.1. Definition of Electrocardiogram 10
2.1.2. Definition of Virtual Reality 10
2.1.3. The Role of VR in ECG Training 10
2.2. Historical Development 11
2.2.1. History of ECG 11
2.2.2. History of VR 11
2.3. Theoretical Foundation 11
2.3.1. Theoretical Foundation of ECG 11
2.3.1.1. Electrocardiogram (ECG) 11
2.3.1.2. Cardiac Conduction System 12
2.3.1.3. Electrodes and color codes 12
2.3.1.4. ECG electrode placement 14
2.3.1.5. ECG waveforms 15
2.3.1.6. Normal and Abnormal Findings 16
2.3.2. Theoretical Foundation of VR 17
2.3.2.1. Components of VR 18
2.4. Literature Review 19
2.4.1. Significant Studies 19
2.4.2. Related Works 20
2.4.2.1. Industry and Current Technology 20
2.5. Challenges 22
3. Methodology 23

3.1. Design Methodology 23
3.1.1. Planning and wireframe phase 23
3.1.1.1. Content Design 23
3.1.1.2. Level Design 23
3.1.1.2.1. Theory Sections 24
3.1.1.2.2. Interaction Sections 24
3.1.1.2.3. Quiz Sections 27
3.1.2. Designing and Implementation Phase 27
3.1.2.1. Heart Conduction System Game 29
3.1.2.2. Lead orientation Game 30
3.1.2.3. Limb clamps placement game 31
3.1.2.4. Precordial electrodes placement game 32
3.1.2.5. ECG waveforms game 33
3.1.2.6. Case level game 34
3.1.3. UI designs 35
3.1.4. Point System Design 38
3.2. Development Methodology 39
3.2.1. Unity Components 39
3.2.1.1. Scene 39
3.2.1.3. Animations 41
3.2.1.4. Controllers and Input System 42
3.2.1.5. Scripts 43
3.2.1.5.1. Main Managers 43
3.2.1.5.2. Game Managers and Scripts 46
3.2.2. Technologies and Tools 51
4.Testing and Result Analysis 53
4.1. Survey 53
5. Project Management 56
6. Conclusions and Future Work 58
References 59
Appendix 61
Appendix A 61

1. Introduction

According to different studies, when we look at the accuracy rates of ECG reports, it shows results that are worth studying especially for medical students. In a study, it is observed that it is 55.8% in medical students, 68.5% in practicing physicians and 74.9% in cardiologists [1]. In a different study conducted on medical students, we observe that more than 70% of the students who learned from textbooks and web-based materials do not feel confident about normal ECG interpretation (see Table 2 from [2]) and 56.1% of all participants were evaluated as low levels in the tests performed (see Table 3 from [2]) These results emphasize the need for different teaching environments related to ECG in medical education.

With the rapid development of technology, the use of digital and virtual environments by students and instructors in medical education is becoming widespread. Many supporting teaching materials are being developed with Augmented Reality (AR), Virtual Reality (VR) and Mixed Reality (MR) technologies that will contribute to medical education. In this way, students and healthcare professionals support their education with interactive simulations.

According to research, the market size of VR gamified education in 2022 was determined to be more than $13 billion and is estimated to reach $123 billion by 2030 [3]. The fact that the market continues to grow and the correct interpretation of the ECG plays a critical role in the healthcare sector makes the development of such training more meaningful.

ECG records the electrical activity of the heart via electrodes and converts it into a graph. During this formation, it may be exposed to many external factors such as drug effects, external noise, physiological conditions, patient mobility, and incorrect drawings may occur.

ECG interpretations are a skill that requires interpersonal skills such as visual, auditory and theoretical knowledge. Many different case studies are required for the development of these skills, and this creates a learning difficulty. In trainings given with textbooks and presentations, learning is provided with limited visual and auditory elements, and this reveals the deficiency of traditional education. In addition, in order to correctly understand the positions of the human heart and electrodes while interpreting ECG, it is difficult to perceive the position of the heart in three dimensions. When we consider the difficulties of both practical and theoretical skills, the interpretation process is a complex process. Considering all these, the VR environment can provide an opportunity to develop ECG interpretation skills without experiencing these difficulties and can create a realistic simulation for students.

This research and the simulation to be developed were built on the research question “Can a VR-based simulation be effective in improving users' ECG interpretation and clinical skills?” and the goal of the project was determined accordingly.

This medical simulation project has set its goals as providing the opportunity to learn all the environments and information that will help ECG interpretation at the same time and to help students develop these skills. Students can learn the inner workings of the heart with theoretical, auditory and visual educational content, examine ECG waves, interact with the 3D heart and understand the importance of the positions of the electrodes in signal formation. They can also have the chance to place the electrodes. In this way, they can understand the positions of the leads in humans. They can learn and simulate the signal differences between normal and abnormal situations.

In the following sections, a literature review will be conducted on this subject, and previous research and industrial studies in this field will be examined, information will be provided about the game design and

levels related to the developed simulation, information will be provided about the purpose of the project, the development environment and software development process will be discussed, the results obtained from the tests will be analyzed and suggestions will be made for future studies.

2. Literature Review

This literature review aims to evaluate the potential positive effects of the application to be developed on medical professionals and to understand the basic information and background in the literature.

In this section, we will define the topics, touch upon their historical development, provide theoretical background explanations, review the literature and current applications in the “Current Technology” section, and finally discuss the challenges.

As a result, we will obtain information to understand the effects of VR technology on ECG interpretation skills through the literature review.

2.1. Definition and Importance of the Topic

2.1.1. Definition of Electrocardiogram

Electrocardiogram is a graph of the electrical activity of the heart recorded by electrodes attached to the body. This term is derived from the German word "electro-cardiographie". For this reason, the abbreviation ECG and sometimes EKG are used in different countries. ECG is a diagnostic method that has an important effect in investigating the severity of cardiovascular diseases. ECG is also a research tool used in the evaluation of patients before non-cardiac surgery, in the monitoring of patients using antiarrhythmic drugs, and in the monitoring of drugs with known cardiac effects [4].

2.1.2. Definition of Virtual Reality

Virtual Reality (VR) is a three-dimensional virtual environment simulation that users model in 3D with special equipment such as headsets and hand trackers and coded with software. VR completely separates users from the real world and immerses them in a virtual environment, thus changing the users' perception of the real world.

Although virtual reality comes with immersive and entertaining video games, it also has more serious applications in areas such as architecture, sports, medicine and, entertainment [5].

2.1.3. The Role of VR in ECG Training

In order to prevent and diagnose cardiovascular diseases, which are shown as the number one cause of death, healthcare professionals need to improve their ECG interpretation skills. Many healthcare professionals find advanced ECG interpretation complex [4].

As we will discuss in the “Significant Studies” section, there are various studies in the literature examining the role of VR in ECG training. In these studies, it has been observed that the use of VR in ECG training improves the ECG interpretation skills of healthcare professionals and students.

In addition, VR technology allows users to practice in a virtual environment. It makes the learning process effective and facilitates the understanding and interpretation of ECG findings. In this way, users can develop their skills by experiencing ECG scenarios similar to the real world.

2.2. Historical Development

2.2.1. History of ECG

ECG, one of the oldest methods used in heart disease techniques, was first developed by Willem Einthoven in 1901. Efforts to monitor heartbeats have been ongoing since the early 1700s. In 1872, Gabriel Lippmann developed a device. In 1887, A.D.Waller measured the first heart measurement using this device, but could not obtain perfect results. Einthoven managed to fix the errors in the device in 1901 and made this device automatic by developing the string galvanometer. The first modern ECG device measured the electrical activity of the heart by placing electrodes on both sides of the left leg. This arrangement is called the “Einthoven Triangle”. While the device weighed 272 kg when it was developed in 1911, today's modern ECG devices weigh approximately 3.6 kg and are considered the gold standard in heart health diagnosis [6].

2.2.2. History of VR

Virtual reality can be defined as the experience of images produced by computers in order to trick a person's brain into perceiving something that does not exist as real. The first attempts at virtual reality can be given as 360-degree panoramic pictures in the 19th century. Charles Wheatesone's stereoscope in 1838 and Edward Link's flight simulation in 1929 can be seen as the first steps of virtual reality. The term "virtual reality" was first used by Jaron Lanier in 1987. After the 1990s, virtual reality began to be used in NASA and the gaming industry. Today, VR technology is developed by large companies such as Meta, Apple and Microsoft, is used by the masses and continues to develop [7].

2.3. Theoretical Foundation

2.3.1. Theoretical Foundation of ECG

This section will cover the theoretical foundations of the ECG. The basis of the ECG is based on the depolarization and repolarization of heart cells. Each time the heart beats, electrical changes occur in the heart cells and these electrical changes are recorded by electrodes attached to the body and a graph is obtained in the form of a wave. This section will also cover the electrodes of the ECG and where they are placed in the body. This theoretical basis is important for the correct and effective design of the virtual reality simulation to be developed together.

2.3.1.1. Electrocardiogram (ECG)

ECG records the potential difference of electrical signals received from the body surface by means of electrodes. The electrodes are placed on certain parts of the body such as the chest, arms and legs. These electrodes are connected to the ECG machine by means of cables and the activity is measured. The activity records are printed on a paper and interpreted by medical professionals.

The signals that control the contractions of different parts of the heart ensure that the blood continues to flow naturally. The ECG records these signals and shows the heart's rate of beat, rhythm regularity and the timing of the signals. Abnormal conditions in the ECG make it easier to diagnose heart diseases. [8]

2.3.1.2. Cardiac Conduction System

In order for the heart to function properly, electrical signals must be transmitted correctly. This transmission helps manage heartbeats and allows the heart to contract and relax. The heart's conduction system consists of the Sinoatrial (SA) node, Atrioventricular node (AV node), Atrioventricular Bundle (Bundle of His), Left and right bundle branch and Purkinje fibers.

The Sinoatrial Node (SA node) is located in the right atrium near the entrance to the superior vena and cava. It is called the heart's natural pacemaker and is a component that initiates all heartbeats and determines the heart rate. Electrical impulses spread to the atriums and cause them to contract. The Atrioventricular node (AV node) is located on the other side of the right atrium, close to the AV valve. Electrical signals pass to the ventricles using the AV node. A delay is required to allow the ventricles to completely empty and the ventricles to fill before the heart contracts. The AV node provides this electrical delay. The AV node transmits electrical signals from the SA node to the atrioventricular bundle (AV bundle of His). The bundle of His divides into right and left. The signals are then transmitted to the Purkinje fibers and spread to the ventricular myocardium.

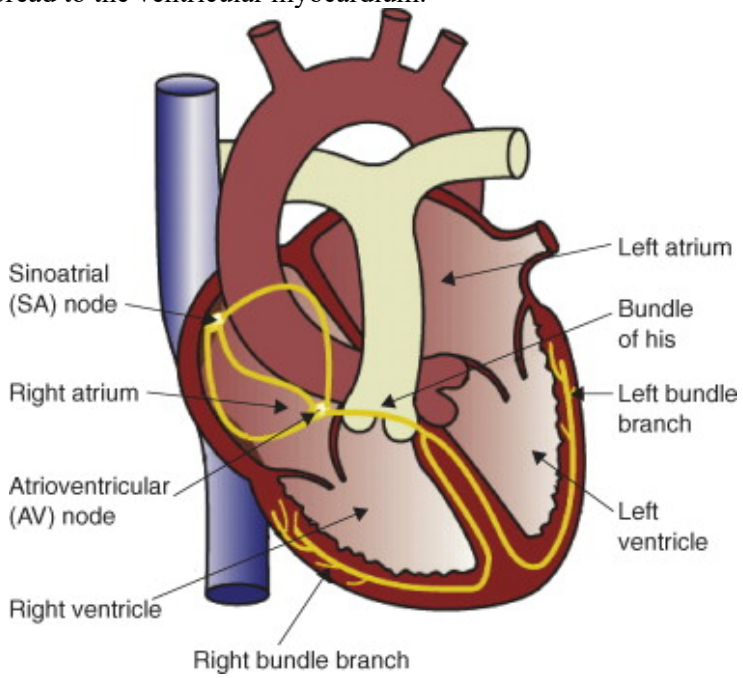

Figure 1 : Heart Conduction System [9]

2.3.1.3. Electrodes and color codes

Due to the areas of use of ECG, correct recording and interpretation are very important for heart health analysis. Each stage of the procedure should be carried out in accordance with professional standards and approved by reliable institutions. Compliance with these standards plays a critical role in ensuring the expected high accuracy rates [10].

While round-tipped electrodes are generally used for precordial leads, clamps are used for limb leads. Electrodes and clamps can have different shapes and sizes depending on manufacturers and standards.

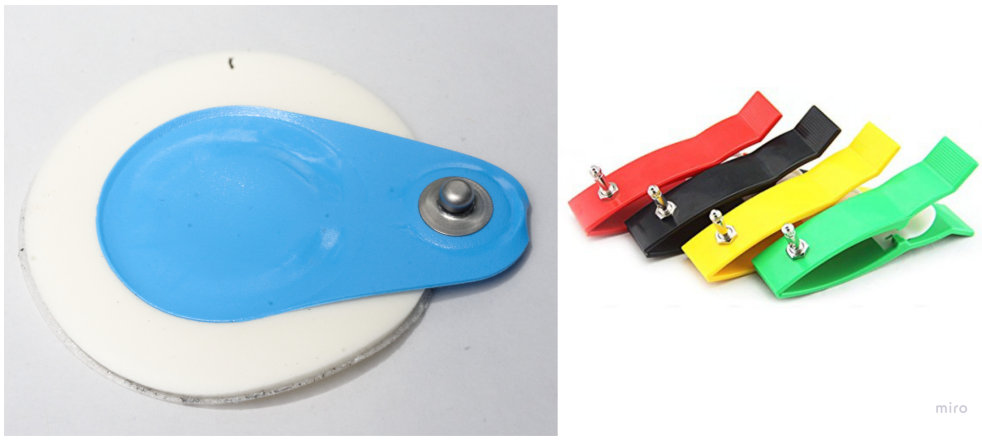

Figure 2: The lead on the left side is used for chest [11] and the clamps on the right side are used for limb leads [12].

ECG procedures have historically been published by the world's leading standardization organizations according to their own standards. Each organization, especially AHA/AAMI and IEC, has applied their own country's standards and there are differences between the colors. Although the standards have been revised in the process, this inconsistency still continues. Different models are encountered due to the production of the manufacturers according to the country's standard.

Electrodes are colored differently according to the American Heart Association and IEC - International Electrotechnical Commission. We can see these differences in the table below [13].

Electrode (AAMI/IEC)	**Color Code** (AAMI)	**Color Code** (IEC)
RA /R	White	Red
LA /L	Black	Yellow
LL /N	Red	Green
RL/F	Green	Black
V1/C1	Brown/Red	White/Red
V2/C2	Brown/Yellow	White/Yellow
V3/C3	Brown/Green	White/Green
V4/C4	Brown/Blue	White/Brown
V5/C5	Brown/Orange	White/Black
V6/C6	Brown/Purple	White/Purple

Table 1: Nomenclature and coloring of electrodes according to AHA/AAMI and IEC

2.3.1.4. ECG electrode placement

The electrodes are placed on certain parts of the body and record the heart's activity. Although there are different electrode placement methods such as 3 electrode, 5 electrode and 10 electrode placements in ECG, the 10 electrode system is accepted as the standard when placing it on the patient's body. Six precordial and 4 limb electrode placements are made. And these 10 electors create a total of 12 leads.

Precordial electrodes create 6 leads from 6 electrodes. The six precordial leads are V1, V2, V3, V4 V5 and V6 respectively. These are;

1. V1: 4th intercostal space (ICS), RIGHT margin of the sternum
2. V2: 4th ICS along the LEFT margin of the sternum
3. V3: 5th ICS, midclavicular line
4. V4: Midway between V2 and V4
5. V5: 5th ICS, anterior axillary line (same level as V4)
6. V6: 5th ICS, mid-axillary line (same level as V4), they are placed in the body. [14]

Limb clamps create 6 leads from 4 clamps. The six limb leads are Right Arm (RA), Left Arm (LA), Right Leg(RL) and Left Leg(LL) respectively. These are;

1. RA: RA clamp is usually placed on the right wrist.
2. LA: LA clamp is usually placed on the left wrist.
3. LL: LL clamp is placed on the left lower leg.
4. RL: RL clamp is placed on the right lower leg. RL lead is a neutral current where the electric current is measured as 0. It is not displayed in ECG reports and is used to minimize artifacts.

The six leads formed are, in order,

1. Lead I: Electrical potential difference between LA and RA is measured
2. Lead II: Electrical potential difference between LA and RA is measured
3. Lead III: Electrical potential difference between LA and RA is measured
4. Augmented Vector Right (aVR): Average potential difference of LA and LL is measured with respect to RA electrode.
5. Augmented Vector Left (aVL): Average potential difference of RA and LL is measured with respect to LA electrode.
6. Augmented vector foot (aVF): Average potential difference of RA and LA is measured with respect to LL electrode.

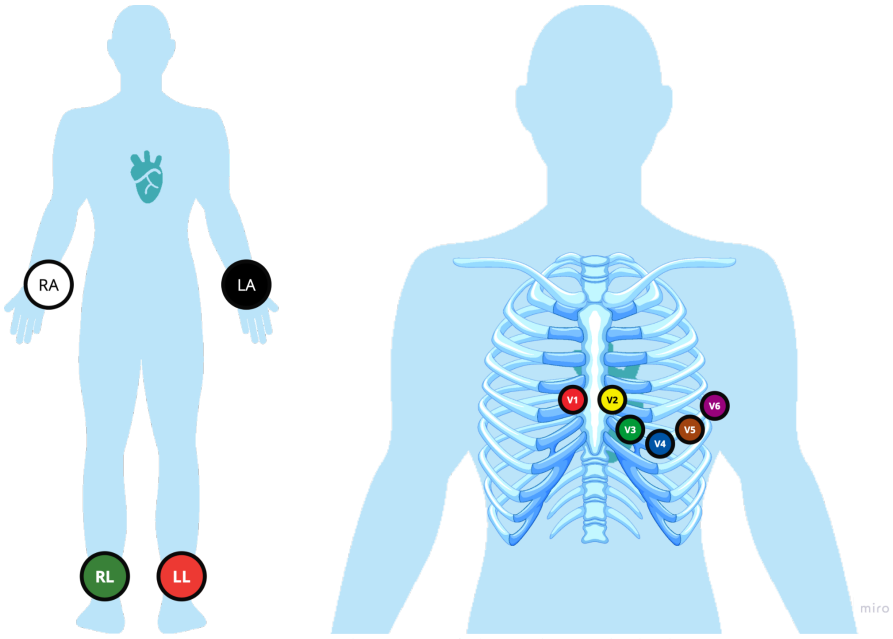

Figure 3: Placement of electrodes on the body

2.3.1.5. ECG waveforms

After all electrodes and clamps are placed and the ECG is started, the graphics start to be reflected on the monitor. Each wave of this graphic has a meaning when interpreted. The wave formations that should be in the normal ECG waveform are related to the conduction system in the heart's conduction system.

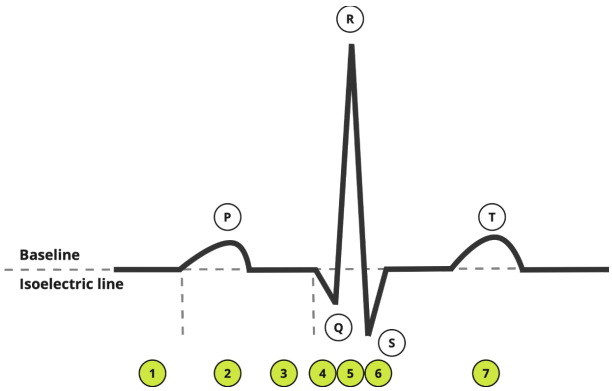

Figure 4: Normal ECG waveform

The waves in Figure 4 are formed due to the conduction in the conduction system of the heart, which we can also examine in Figure 1. Steps 1, 2 and 3 in Figure 4 represent the depolarization of the atrium, Steps 4, 5 and 6 represent the depolarization of the ventricles, and Step 7 represents the depolarization of the ventricles.

We see the P wave on the ECG as a result of the depolarization of the atrium. The conduction from the sinoatrial node moves towards the atrioventricular. Our vector forms downward and to the left. The magnitude of the vectors formed depends on the magnitude of the tissues. Since the vector makes a

positive movement, it will form a wave above the isoelectric line in the ECG, that is, in the positive direction. This is called the P wave.

The conduction speed in the atrioventricular node is very low. Therefore, it is seen as a straight line on the isoelectric line. The conduction then separates from the sinoatrial node to the right and left ventricles. The interventricular septum area needs to be depolarized. For this reason, it creates a vector in the upward and right direction. It creates a deflection in the negative direction and therefore creates a wave below the isoelectric line. This wave is called a Q wave.

For depolarization of the major part of the ventricles, the conduction is divided into two. The resultant of the vectors formed for the two-fold conduction is taken. Since the muscular structure of the left ventricle is larger than the right ventricle, the vector formed here will be larger. For this reason, the vector formed here will create a large and positive deflection. For this reason, it allows us to see a large wave above the isoelectric line. This is called the R wave.

The depolarization of the basal parts of the ventricles goes in the negative direction to depolarize all the ventricles, which creates a negative deflection, and due to the size of the muscular structure of the ventricle, we see a wave larger than the Q wave in the downward direction. This is called the S wave.

After the ventricles are depolarized, the cells repolarize and return to their initial electrical state. During this repolarization, the direction of the electrical activity becomes negative in the opposite direction of depolarization. However, since the vector itself is also upward and negative, we see a wave in the ECG above the isoelectric line, that is, in the positive direction. This wave is called the T wave. The T wave occurs larger than the P wave during ventricular repolarization because the muscle mass of the ventricles is larger than that of the atria and more electrical activity is produced.

2.3.1.6. Normal and Abnormal Findings

We can understand whether the electrical activity of the heart is normal by looking at normal and abnormal ECG findings.

For example, in a normal ECG, we expect the P wave to be in a small positive direction. The P wave vector is directed down and to the left in the frontal plane and this gives a positive P wave in lead I. However, in the case of P-Pulmonale, pulmonary vascular resistance increases, and this causes the right atrium to expand. For this reason, an increased P wave amplitude is observed in leads II and V1 [15].



Figure 5: Normal and abnormal P waves in case of P pulmonale

As another example, in the case of Right Bundle Branch Block, an inverted T wave can be observed in the V1 lead and a widened, slurred S wave in the V6 lead.



Figure 6 : ECG with Right Bundle Branch Block [16]

With many examples like these, disease diagnosis can be made by examining ECG waves and interpreting the patterns in all leads. ECG is also used to evaluate the effects of drugs on the heart and to monitor the daily rhythms of patients using heart devices. This versatility of ECG makes it indispensable for cardiac evaluations and emergency medicine.

In conclusion, ECG is a tool widely used in modern medicine and provides information about the electrical activity of the heart to health professionals quickly. It is valuable and useful for both acute and chronic conditions with its wide range of disease diagnoses. It is very important for health professionals to have accurate information about the clinical applications and interpretations of ECG and to make informed decisions.

2.3.2. Theoretical Foundation of VR

VR is the process in which an organism uses artificial stimuli to produce a targeted behavior in situations where the organism is not aware of itself.

According to this definition, the organism experiences an experience designed by the creator, such as flying, walking, and socializing. The organism can be yourself, someone else, or an animal. Some of the organism's senses are partially disabled and the usual inputs are replaced by artificial stimuli. While the organism is experiencing this experience, it seems to be unaware of the external interference, which causes it to feel the world as the real world. With this feeling, the real world and the virtual world blend, and the organism begins to accept the artificial world as natural [17].

In terms of visual immersion experience, there are a few important issues for VR. One of these is the screen resolution. Higher resolution means sharper and clearer images. Low resolution pixels can reach a level that the human eye can perceive. Since the human eye is very close to the screen on VR screens, this situation needs to be taken into consideration. For an optimal immersion experience, the pixels on the screen should be dense enough to be invisible to the human eye [18].

Another important issue is image latency and display refresh frequency. Refresh frequency is very important to achieve the feeling of “presence” in the VR experience. This is the part that makes users feel whether they are actually in the virtual environment. When the Oculus Rift and Quest 1 first came out, they had a 72Hz refresh rate. Although blur and lag were reduced thanks to the OLED screens in both of these devices. With the Quest 2, it was understood that the screen refresh rate needed to be 90Hz as a standard to bring the VR experience to a satisfactory level [19].

VR also has discussions about field of view. When we want to evaluate the interaction provided by the VR device, we need to evaluate the Field of View as well as the image quality.

The area shaded in red in Figure 7 represents the user's field of view. In order to evaluate the immersion level of the VR device, we need to pay attention to the field of view.

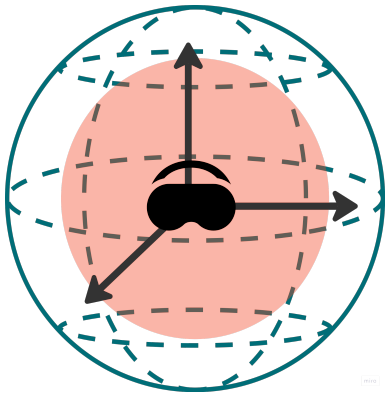

Figure 7: The area shown in red is the field of view obtained during the VR experience. (Adopted from [20])

When using a VR headset, the goal is to provide the widest FOV, but it is pointless to create a field of view larger than what the human eye can follow. Each eye has an angle of approximately 150 degrees, while the 2-eye conjunction area provides an angle of 120 degrees, and the human has a total field of view of 180 degrees.

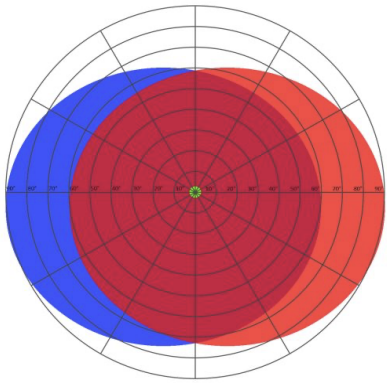

Figure 8: Normal visual field [21]

For this reason, in order to provide an impressive virtual experience, a design that mimics the real world will be needed, taking into account both FOV and resolution when developing VR.

2.3.2.1. Components of VR

In general, VR devices require a computer, head-mounted display (HMD), sensors, input devices, sound system and software components.
Computers are the machines that will process and create the three-dimensional world. Data from the input is processed, user movements are tracked, and graphics are rendered. A large amount of RAM and a good GPU are required in these devices. HMD is a screen that users can wear on their heads.

The motion sensors inside detect the user's direction and position and adjust the image accordingly. Sensors are usually integrated into the HMD. There are many sensors such as accelerometers, gyroscopes and magnetometers.
Input devices must be present for users to interact with the virtual world. These include controllers, joysticks, mouse, and hand tracking systems.

Sound systems are usually integrated into the HMD and are an important component that breaks the user's perception of the real world and allows them to feel in the virtual world. Software is the most critical part of the VR system. These are the applications that create the artificial world. There are different types of software such as games, simulations and medical ecosystems [22].



Figure 9: Meta quest 2's HMD and hand controller device [23]

2.4. Literature Review

2.4.1. Significant Studies

1. A randomized controlled study conducted with undergraduate medical students found that the use of VR had a positive effect on recognizing normal and abnormal ECG waves [24].

2. In a study on the feasibility of a VR-based platform to teach cardiac anatomy to electrophysiology trainees, we find that participants subjectively evaluated the learning method positively but observed mixed results in exam performance.[25]

3.On the effect of electrocardiography training with web-based and self-directed interventions
A study with 1206 participants observed that this training significantly increased the ECG interpretation skills of healthcare professionals [26].

2.4.2. Related Works

2.4.2.1. Industry and Current Technology

1. Acadicus ECG Simulator

This simulator is the ECG simulator section of the Pro Space simulation application developed by Acadicus. In this section, users can place ECG leads and view the 3D heart model in a pre-created scene. [27]. The simulation has VR and non-VR modes. There is an animation of the heart model. It can be interacted with online due to the Pro Space feature. However, as a minus, it can be shown that the content density is low except for the ECG lead placement and the functioning of the heart.

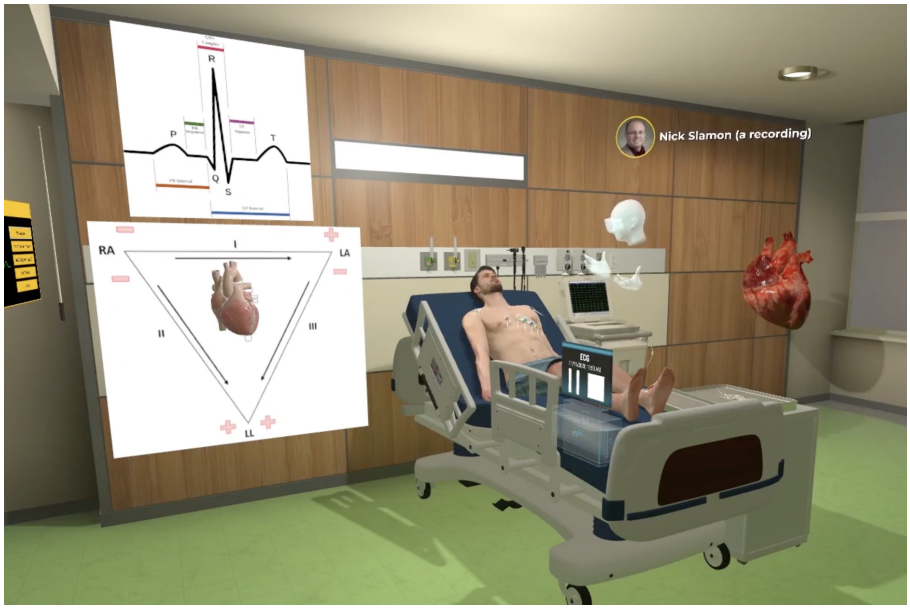

Figure 10 : Screenshot from Acadicus ECG Simulator

The MediSim application developed with this simulation is similar in design and flow, but while Medisim focuses more on explanations about the inner workings of the heart, Acadicus ECG Simulator creates an environment that will facilitate the instructor's explanation.

2. MedVR-EKG

This simulator is a simulation for ECG training developed by MedVR company. Users can place ECG electrodes on the patient's body and view ECG reports. A real hospital environment is presented as a design. And they can adjust the difficulty in assessment mode [28].



Figure 11 : Screenshot from MedVR-EKG Simulator

There are both training and evaluation modes. There are both Mixed reality and Virtual Reality modes. There is a photorealistic virtual environment. The downside is that the simulation only lasts 20 minutes.

Although the MediSim application developed with MedVR-ECG simulation is basically similar, there are differences in terms of the flow of the game. While the MediSim application provides theoretical training and then offers the chance to try it in 3D environments, the relevant application aims to evaluate more with assessment. MedVR-ECG also has patient customization and environment change features. In addition, MedVR-ECG has paid more attention to presenting Mixed Reality and Virtual Reality environments as a prominent feature.

3. Leonardo VR

LeoardoVR, developed by Medvision, was developed for Simulation-based development of clinical reasoning. There are many different areas of training in simulation such as survey, monitoring, examination, drugs, analyses, treatment, screening [29].



Figure 12: Screenshot from Leonardo VR Simulator

Patient Library is extensive. Patient bed setup is available and Heart ECG imaging is available but lead placement is not available.

The MediSim application and the Leonardo VR application are quite different in terms of operation. Leonardo VR is more interested in the general controls of the patient than in ECG learning. Although these two applications are not similar in terms of narrative, they were examined because they are Medical Simulation.

2.5. Challenges

The difficulties encountered during the use of VR technology in ECG training limit the benefits of this technology. First of all, the cost of VR equipment prevents a worldwide development. In addition, the cost of software development stages also makes these developments difficult.

Another difficulty is that the VR experience creates negative physical effects in some users. In a study [30] synthesized from research in databases such as Science Direct, Google Scholar, and PubMed, VR users were reported to have cybersickness and perceptuomotor after-effects in particular. In addition, headache and eye fatigue are among these physical effects.

In order to overcome these difficulties, in the same study, it was suggested that VR applications should undergo a prototype and content validation jointly with professional physicians and healthcare personnel in order to avoid high development costs and then produce a final product with fewer changes. For the physical effects that occur after VR use, it was suggested that research and preventive plans should be made for the behavioral changes of individuals after use and that these tools should be used as a supportive educational application rather than replacing traditional methods.

3. Methodology

3.1. Design Methodology

Before the project development phase, the game design, 3D models in the levels and 2D sprites and their contents were determined and their wireframe designs were made. Some designs were used as free assets on the internet, while some were prepared as 3D models using Blender and 2D sprites using Adobe Illustrator and made ready to be transferred to Unity.

3.1.1. Planning and wireframe phase

3.1.1.1. Content Design

The contents are collected from websites with the help of professional physician consultants. There are specific topics for each level and research has been done on these topics.

1. Level 1: Contains relevant theoretical information and graphics on understanding the heart conduction system
2. Level 2: Contains relevant information on ECG electrodes and their placement.
3. Level 3: Contains relevant information on ECG normal and abnormal waveforms

This simulation is designed for medical students and medical professionals. Before the game development begins, a user persona is created to better understand the user and to better prepare the design and content.

According to this persona, Lussie is a medical student and wants to play a simulation that provides an interactive learning experience to develop basic knowledge about ECG. Our user, who has an intermediate level of knowledge, wanted to play this simulation for fun and to provide support material for his education. Our student also wants to develop his skills in placing ECG cables and recognizing abnormal ECG waves.

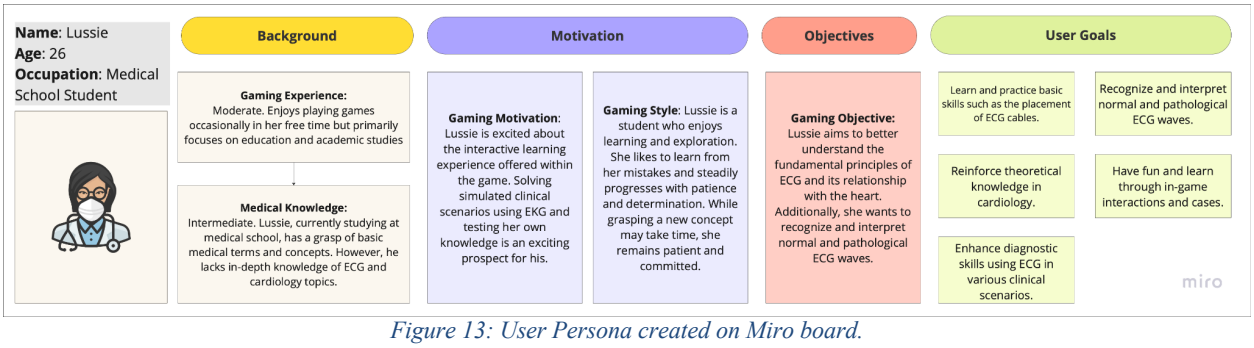

Figure 13: User Persona created on Miro board.

3.1.1.2. Level Design

All levels take place in a clinic. In order to create this environment, it is planned to design sections suitable for level design. It is planned to create a patient bed and ECG monitor for the Case section, a table and chair for the Thoracic section, and 3D heart and human models for the Interaction sections. The same loop will be used when creating levels and users will first receive theoretical training, then

complete an interactive task with three-dimensional models, and finally answer questions asked from the quiz section. There will be 3 different levels in the simulation and each level is planned to have tasks within itself. After the training levels, the user will be able to access the case section where they can place all leads, request case specific heart sounds for different abnormal conditions and view information.

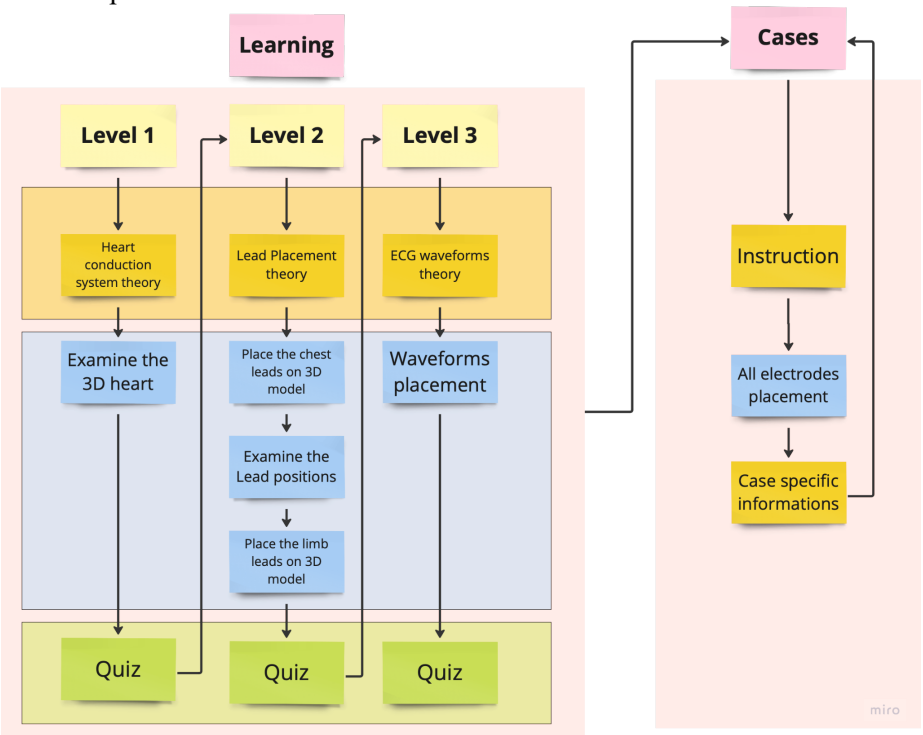

Figure 14: Design of level flow created on Miro board.

3.1.1.2.1. Theory Sections

For the theoretical section, it is planned that all levels will have the same design. In this section, it is planned to provide visual learning about the subject with images or videos, and theoretical information with the description section. It is also planned that the user will reinforce the information with a voice over created with artificial intelligence.

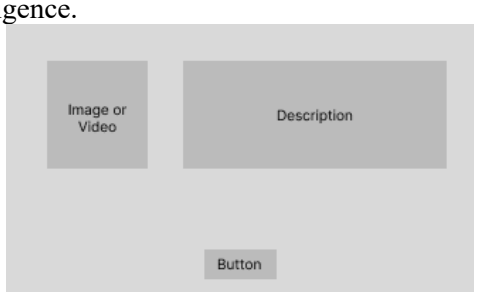

Figure 15: Wireframe design of Theory section using Figma

3.1.1.2.2. Interaction Sections

In this section, it is planned to create interaction sections with different 3D models at specific levels.

Level 1

It is designed to provide interactive learning with 3D interactive objects for the sections of the heart's conduction system taught in the theoretical section. The aim here is for users to understand where and where the conduction is in the internal structure of the heart and the directions of the vectors.

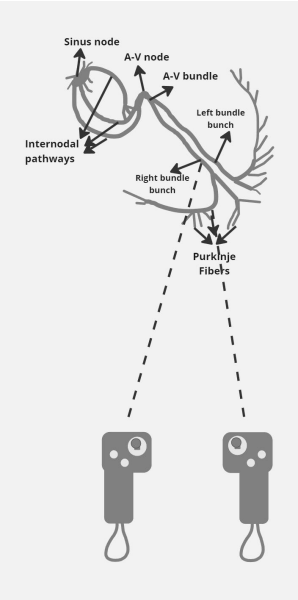

Figure 16 : Reference design for Heart Conduction system

Level 2

1. For the placement of chest electrodes taught in the theoretical section, a human chest model and electrode models were used to place the electrodes. The aim here is to train where the electrodes will be placed in the precordial section.

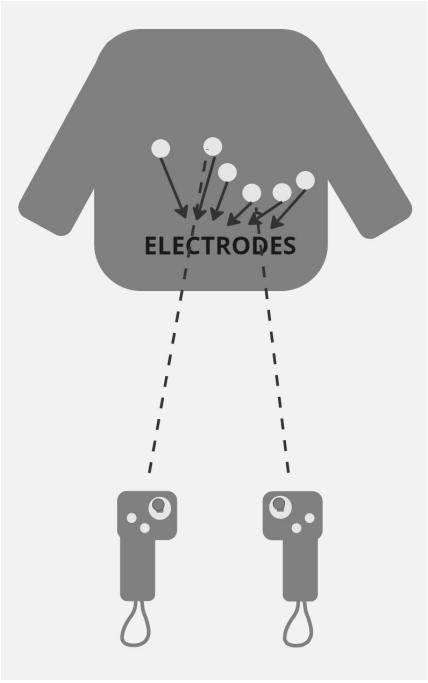

Figure 17: Reference design for chest electrodes placement game

2. Limb electrodes placement taught in the theoretical section is planned to be done with a human full body model and electrodes models. The aim here is to train where the electrodes will be placed in the limb section.

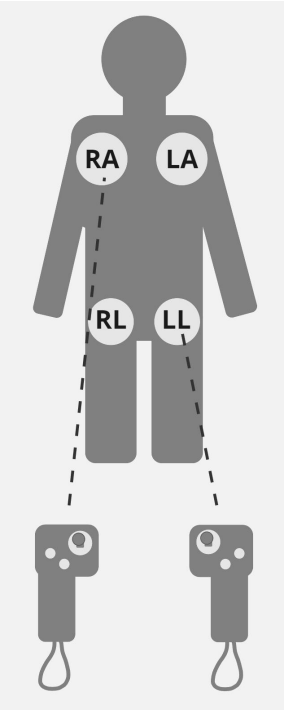

Figure 18: Reference design for limb electrodes placement

3. It is planned to examine the positions of the leads taught in the theoretical section to reinforce where they stand on the 3D model of the heart. The heart model and the positions of the leads are planned to be given as a 3D model.

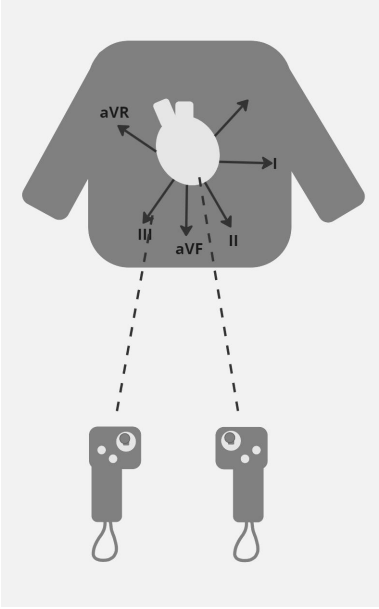

Figure 19: Reference design for lead orientation

Level 3
A small gamification is planned for the normal and abnormal situations taught in the Theory section. Here, the user will complete the missing wave piece.

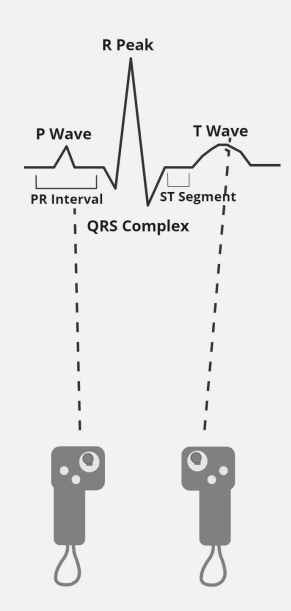

Figure 20: Reference design for ECG waveforms game

3.1.1.2.3. Quiz Sections

The Quiz Section is planned to have the same design for all levels. In order to reinforce the information learned in the Theoretical Section, a multiple-choice quiz section on these topics is planned.

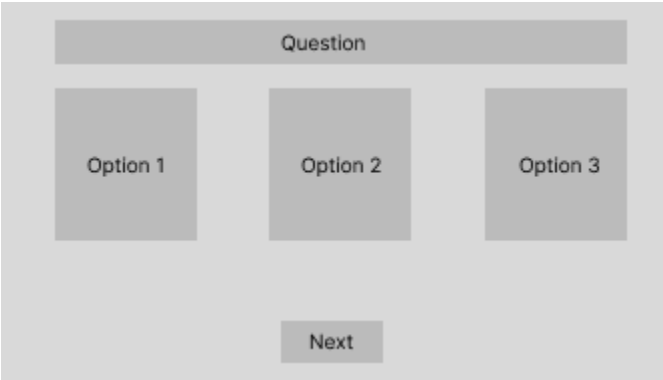

Figure 21 : Wireframe design of the Quiz section using Figma.

3.1.2. Designing and Implementation Phase

In the simulation, ready 2D images and images edited using Adobe Illustrator were prepared to be used in the Learning Section. The sources from which all 2D assets were taken are specified in the Game Assets Credit section. Similarly, ready 3D models and models edited using Blender were prepared to be used in the 3D models room design and interaction section to be used in the simulation. The sources from which

all 3D assets were taken are specified in the Game Assets Credit section. Using all these designs, 3D room and UI screens were designed in a scene on Unity.

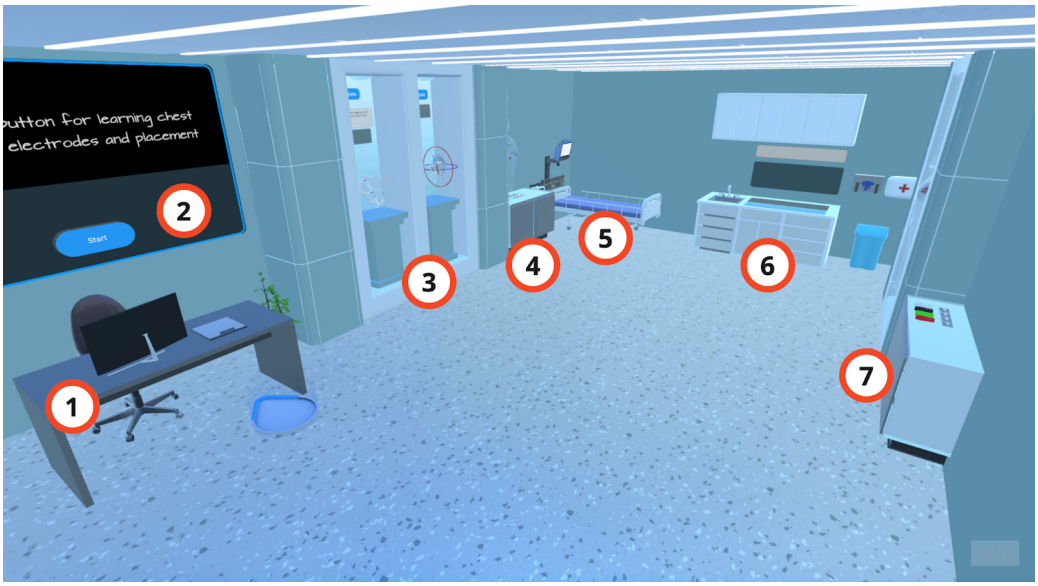

Figure 22: Room interior design of the simulation in Unity

1. Table models: A table, computer and notebook were used to make the learning part feel conceptually. This table is positioned in a way that the doctor can look at the patient's face as in real clinical environments.
2. This section is designed as a UI. The theoretical learnings in the theory section will be done here.
3. In this section, there is a heart conduction system interaction section on the left and a lead orientation interaction section on the right.
4. Here is the chest electrodes placement section.
5. Here is a nightstand prepared for the Case Section and a bed for the patient to lie on.
6. Here is the section prepared for the ECG waveforms game.
7. Here is the limb electrodes placement section.
8. The bottom part of the room is designed as a teleport area. In this section, the user can teleport to any location he wants.

Now let's explain some of them in detail.

3.1.2.1. Heart Conduction System Game



Figure 23: 3D model created for the Heart Conduction System interaction section

On this heart model, the heart sections were modeled as a separate object using Blender and implemented in Unity. Each section of the heart has a separate animation.

In this section, when students hover over the relevant area with the hand tracker, they see information about this area in the information text field. Thus, they can examine the information they learned in the theory section on the 3D model. Students who have examined all sections can complete this task by clicking the next button.

3.1.2.2. Lead orientation Game

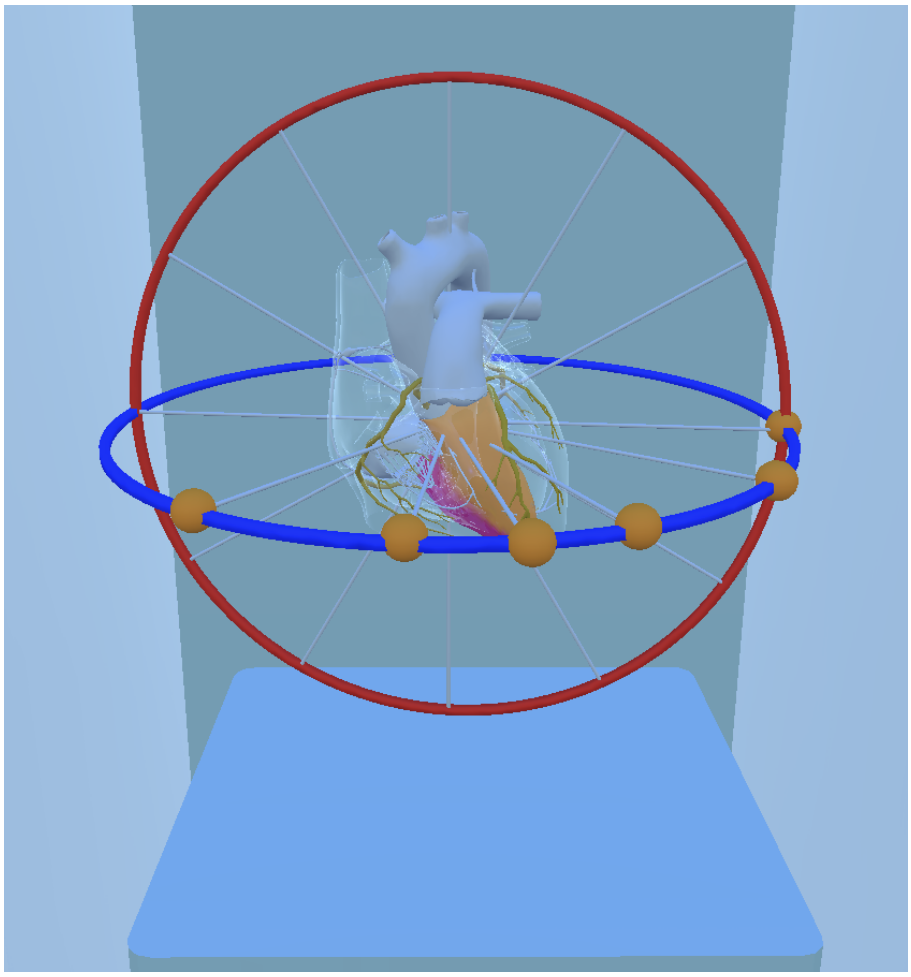

Figure 24: The 6 views of the heart in a 6 chest lead model in the game

In this part of the game, users can rotate this model with the hand tracker controller and see where the chest leads are located in the 3D model of the heart. My goal in this part was to provide a better understanding of which part of the heart the lead placement taught in the theoretical part is directed to. Students who have examined all the leads can finish this part with the complete button and complete the task.

3.1.2.3. Limb clamps placement game

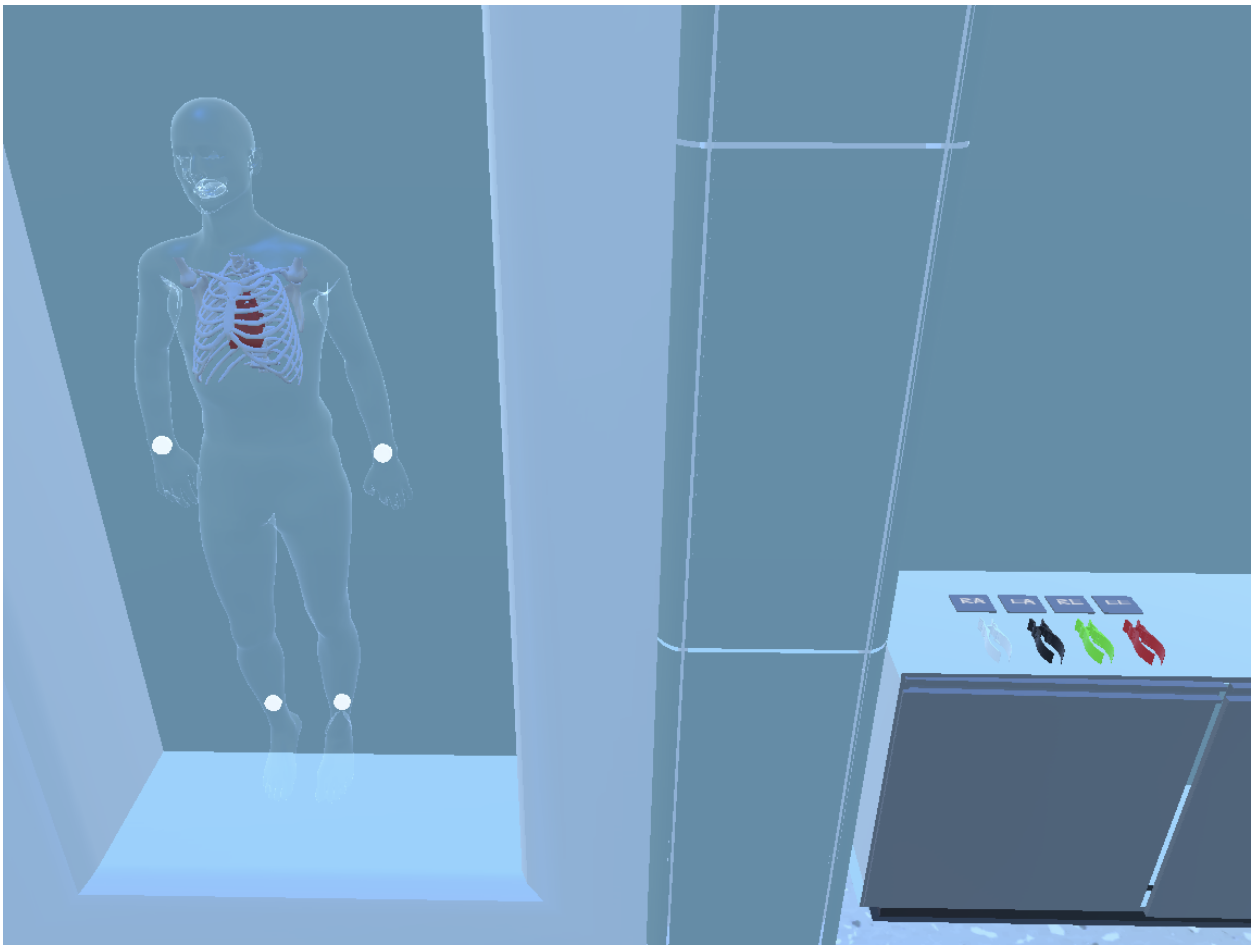

Figure 25 : Limb clamps placement game image.

In this section, students can grab the clamps seen on the right side in Figure 25 with the hand tracker control and place them on the human model on the right side. If all the clamps are placed correctly, the game ends and the task is completed. The human model and sternum models were found separately and combined. I modeled the clamps models using blender and determined and created their materials according to the AAMI standard.

3.1.2.4. Precordial electrodes placement game

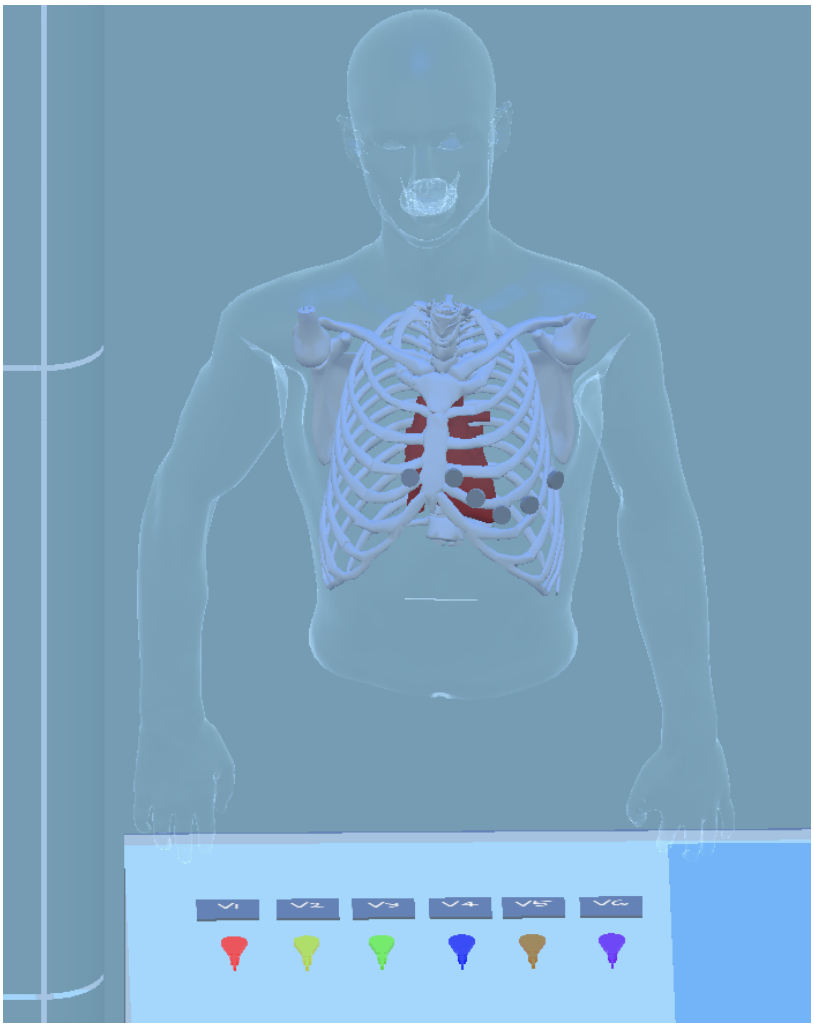

Figure 26 : The image of the Precordial electrodes placement game

In this section, students can place the chest electrodes seen at the bottom of Figure 26 on the human model at the top by grabbing them. The student who places all the electrodes correctly completes this task successfully. My aim in this section was to provide a better understanding of where the chest electrodes taught in the theoretical section should be placed on the real human model and to provide students with a realistic placement experience. The human model and sternum model were obtained separately from the internet, and I combined these two models in Blender and transferred them to Unity. I designed the electrode models myself in Blender. I materialized their colors according to AAMI standards, such as limb clamps.

3.1.2.5. ECG waveforms game

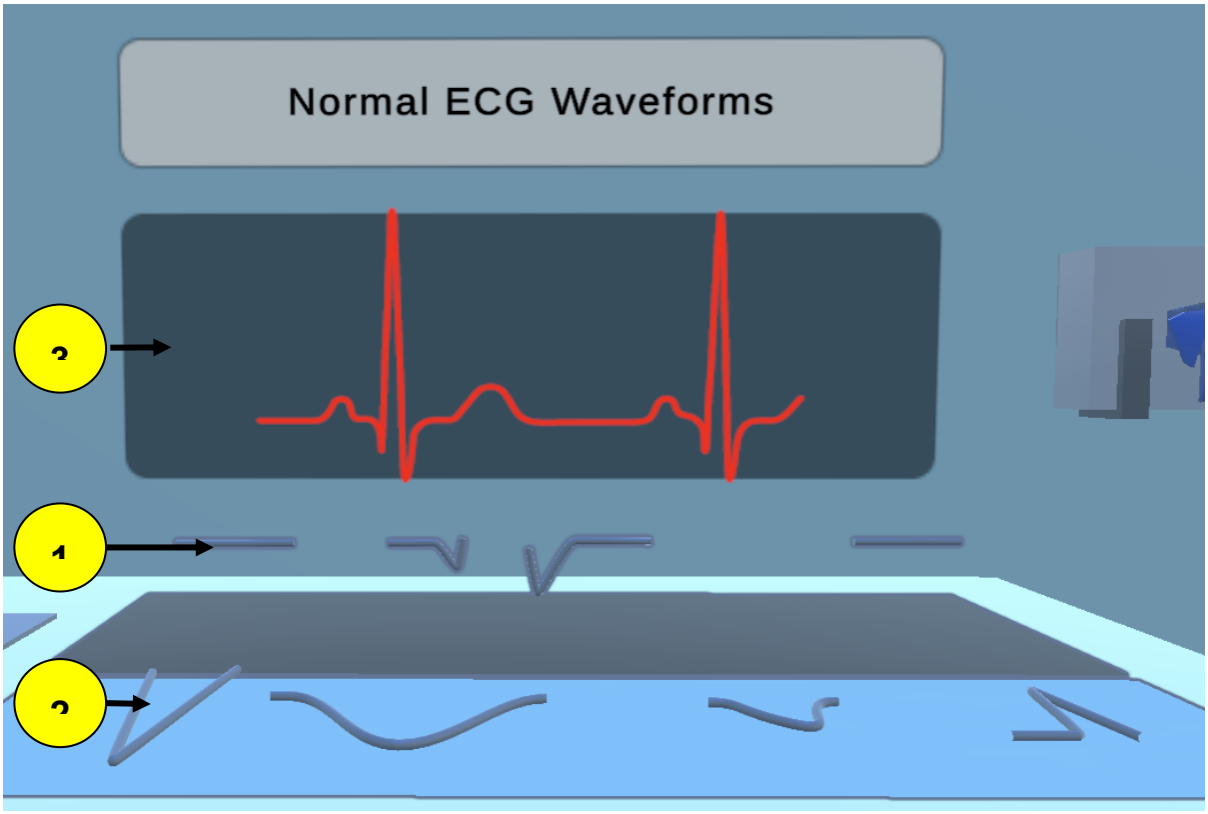

Figure 27 : The image of the ECG waveforms game

In this section, students try to complete the missing ECG waveform shown in Figure 27 in Section 1 by snapping the different pieces seen in Section 2. In Section 3, there is a reference image to help students. In addition, they hear the sound of the heart in that situation in each different waveform. They encounter four different situations until the end of the section. The student who places all situations completes this task. My aim in this section was to ensure that students see the differences in normal and abnormal waveforms of ECG that they learned in the theoretical section of Level 3 and to reinforce the sound differences by listening to the sounds. I modeled the ECG waveform models on Blender and also designed the reference sprites on Adobe Illustrator.

3.1.2.6. Case level game

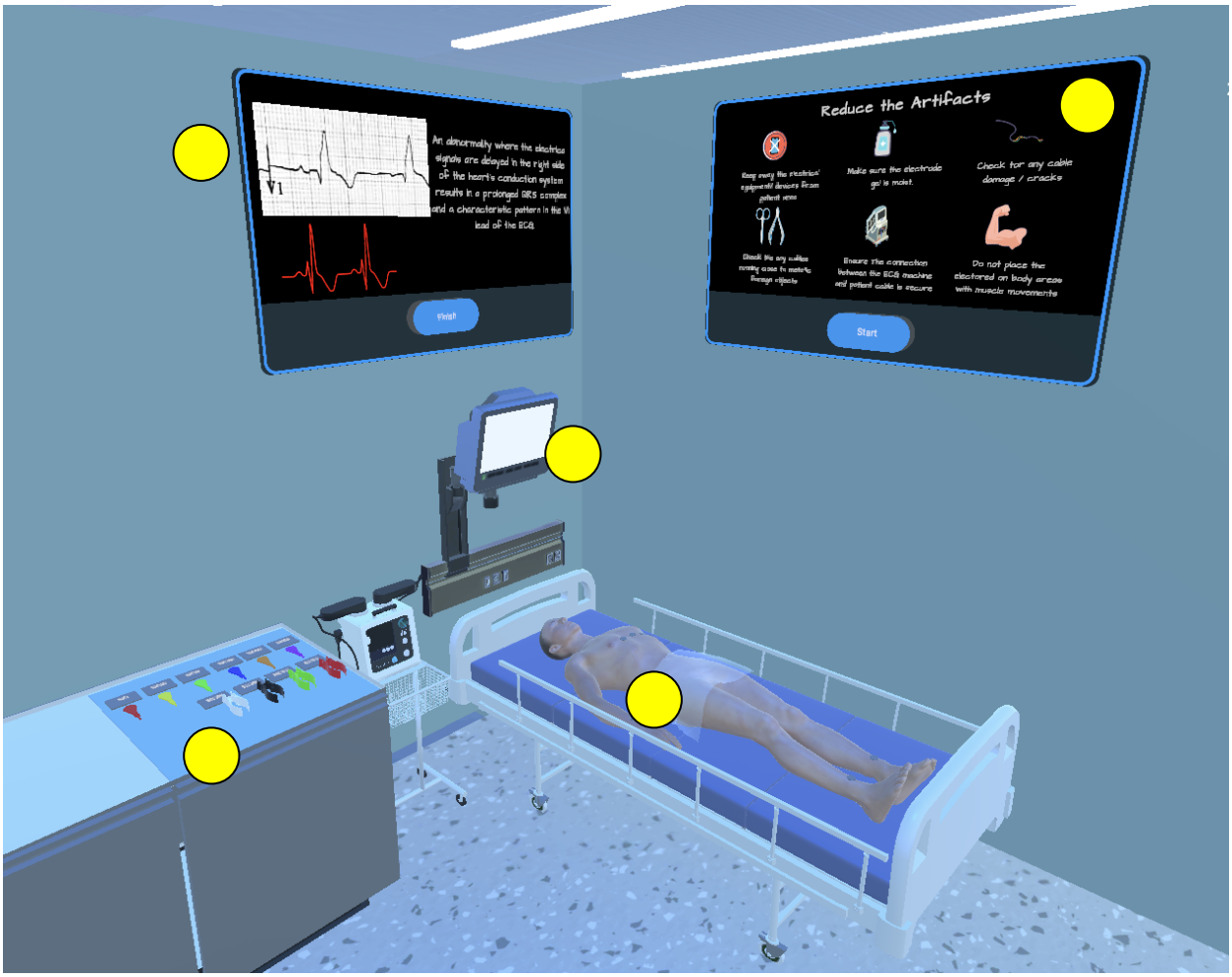

Figure 28 : The in-game view of the Case level section

This section is opened to students who have completed all training tasks. 3 different cases can be experienced and learned in this section. At the beginning of the section, reminders and procedure-appropriate slides related to ECG lead placement are provided on screen number 1 and repetition is provided. Then, electrodes and clamps are spawned in section 2. Students who place these leads on patient model number 3 receive a real ECG report related to this case on monitor number 4 according to the case they have selected. In addition, in section number 5, they see the theoretical information and graphics related to the case they have selected and also listen to the heart sound in this case. My aim in this section was to present visual, auditory and theoretical information together and to provide students with an experience that is much more difficult to obtain in the real world in a simple way. The instruction screen number 1 and the lead placement section number 2 of the section proceed in the same way for each case. This section ends when students press the finish button on screen number 5. Students can experience this section again if they wish.

Coding logic and script snippets of all games will be explained in the "Scripts" section.

3.1.3. UI designs

There is a Menu user interface in the simulation. Using this interface, the user can view their tasks, view their profile, access the case section, and exit the game. The user can open and close this menu by pressing the Y button on the left-hand tracker while in the game. There are also UI designs in the theoretical section and the quiz section.

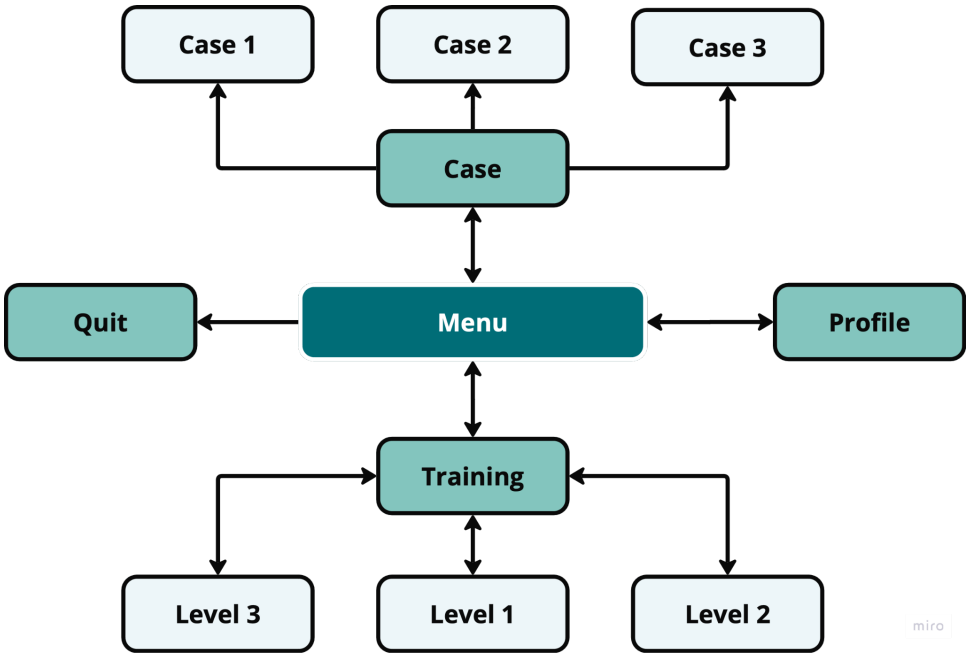

Figure 29 : User interface transaction tree

When the game is started, the main menu welcomes the user,

A. **Profile button:** The user accesses the profile panel by clicking on the Profile button. Here, the user's profile photo, experience points, badge status and degrees are displayed.
B. **Training tasks button:** The user accesses the level task choosing panel by clicking on the Training tasks button from the main menu and can view the tasks of the relevant level by clicking on the button of the desired level. As the user completes their tasks, a cross is placed on their tasks here.
C. **Case button:** If the user clicks on the case button from the main menu, the case panel opens, but if they have to complete all their tasks in the training section, they will hear a voice over stating that they must first complete all their tasks in the case section. They can access this panel when they have completed all their tasks. According to the abnormal status cases they have selected in the case panel, the objects related to the relevant level are activated within the level.
D. **Quit button:** The user can exit the game by clicking on the quit button in the main menu.

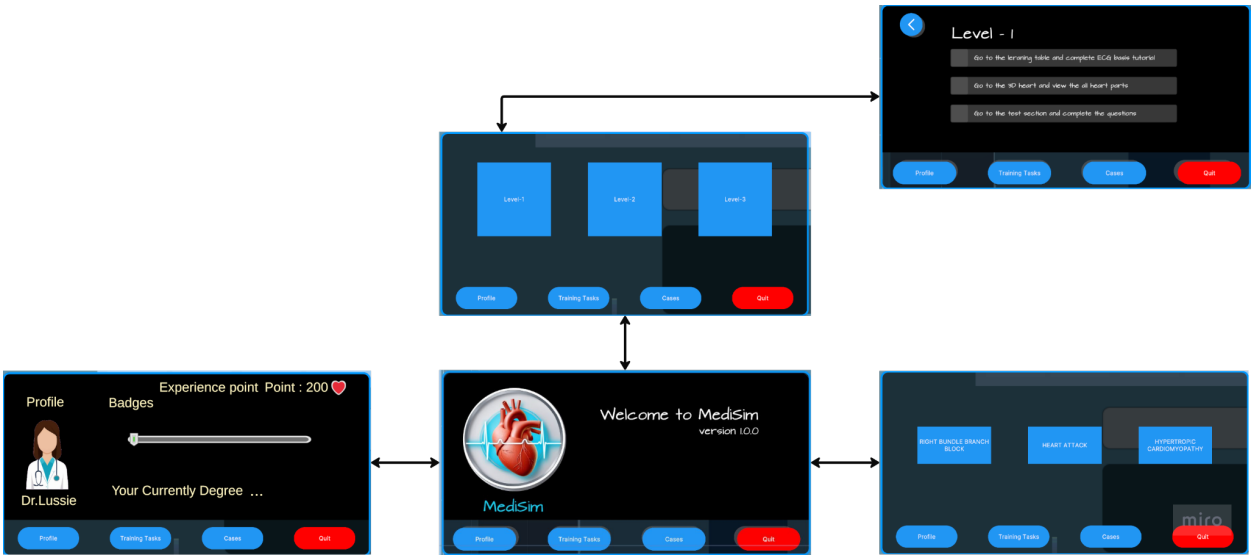

Figure 30: Menu UI transactions in the game

In the theoretical section, users can get theoretical information in the learning section of the training section. Here, in most of the levels, users can read the information as text, listen to it as a voice over, and see its representative image and animation or watch its video. This section is designed in a kind of presentation logic. Students can learn theoretical information and examine the images by switching between the relevant slides. In addition, there are interactive animations in some levels, such as the example in Figure 37. Students who read all the slides in the level can end this section with the finish button and complete their task.

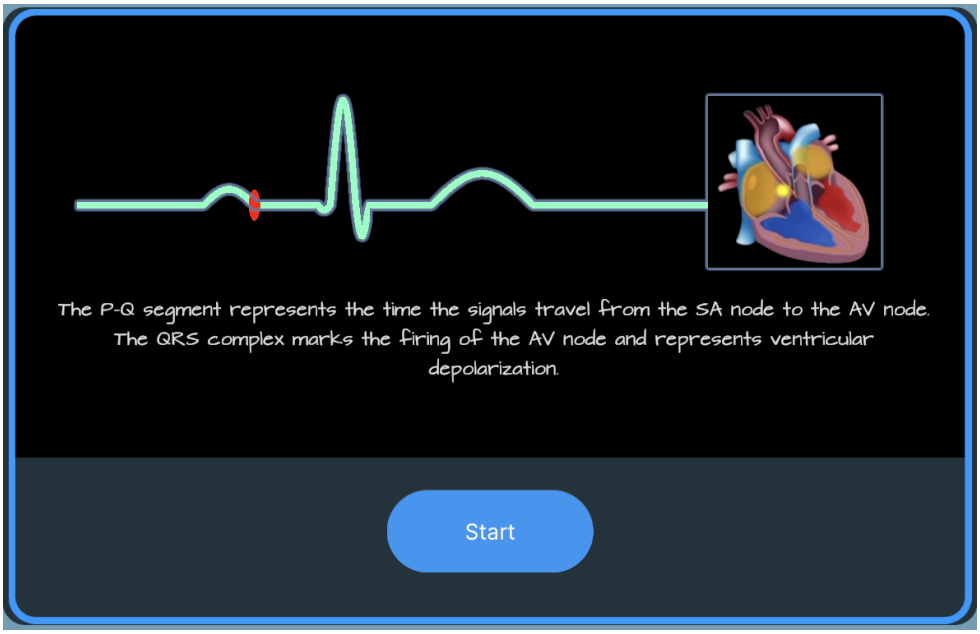

Figure 31: Menu UI transactions in the game

In the quiz section, students come after learning theoretical information and completing 3D models and interactive mini games. The purpose of this section is for students to test all the information they learned in the level. Here, there is a user interface created on Unity and three buttons located next to the answers.

You can click on the buttons with the left or right UI press button on the hand tracker controller. The next button appears in front of the students who mark the correct answer to the question to move on to the next question, if the student answers the question incorrectly, they receive both an audio and visual warning and are asked to answer the question again. Students who answer all the questions in the section complete the task and move on to the next level. This is the last task in the level.

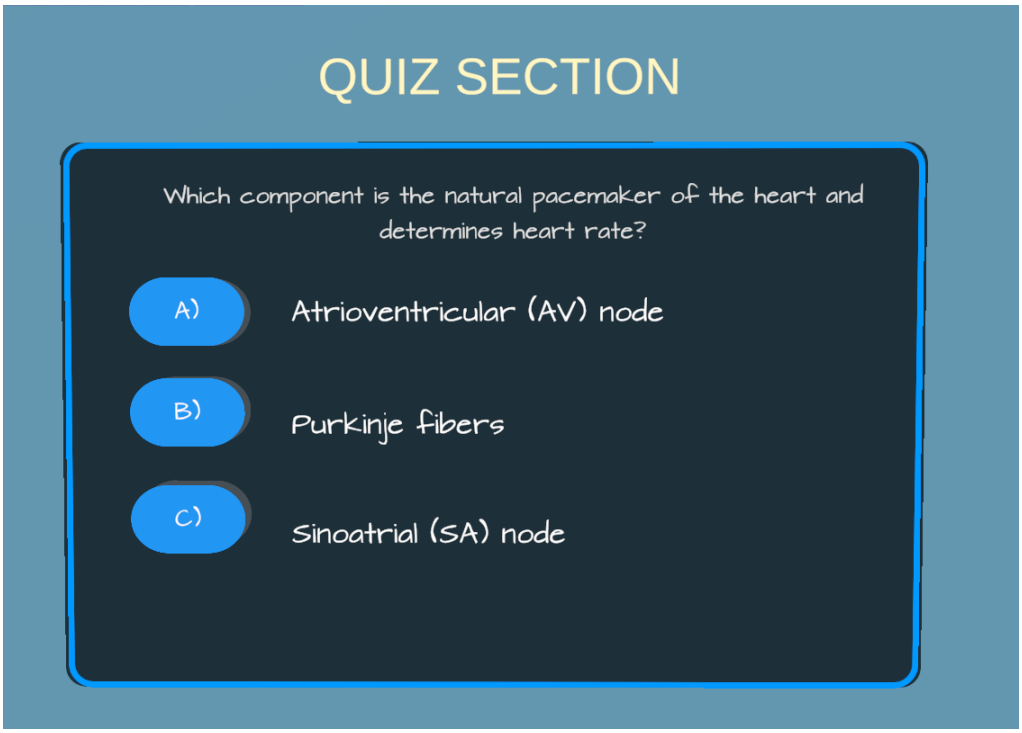

Figure 32 : Quiz section in the game

3.1.4. Point System Design

I designed a point system that allows users to earn points as they complete tasks in the simulation. These points give the user a badge and a rank at certain levels. Each task is completed, 20 XP is earned, and if the questions are answered incorrectly, the user's score is reduced by 10 XP.

Experience Point (XP)	Degree	Badge Image
20	Novice Clinician	
60	Competent Clinician	
120	Proficient Clinician	
200	Expert Clinician	

Table 2: Badges and degrees obtained with experience points

3.2. Development Methodology

Unity Game Engine was used as the game engine while developing MediSim simulation. It was coded using Visual Studio Code in C# programming language. In this section, Unity environment will be explained, and the works done in the development part of the project will be explained.

3.2.1. Unity Components

3.2.1.1. Scene

In Unity, scenes are basically places that are seen on the game screen and where the user can interact and where you work on the content you will prepare. It can include all or part of a game [31].

Since there are no simulation scene transitions in the project, there is only one scene. The scene called Game was also worked on.

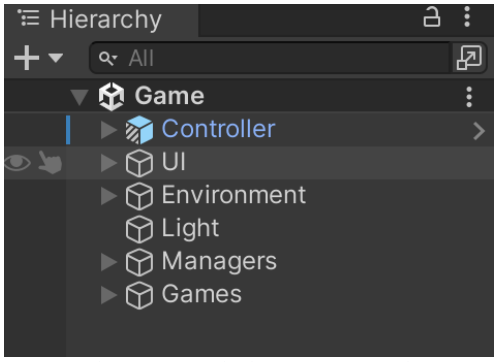

Figure 33: The scene worked on in Unity and its contents.

3.2.1.2.2. Camera

Cameras are devices that show the scene in the created world to the players. In the project, the camera is added to the XR Origin object that will follow the player who will use the VR device. In addition, the track pose drive component from the Unity XR toolkit package has been added to the camera component. This component is used to manage and update the location and route of VR devices. It reflects the movements in the physical world to the digital world. It was used as a perspective feature for camera projection.

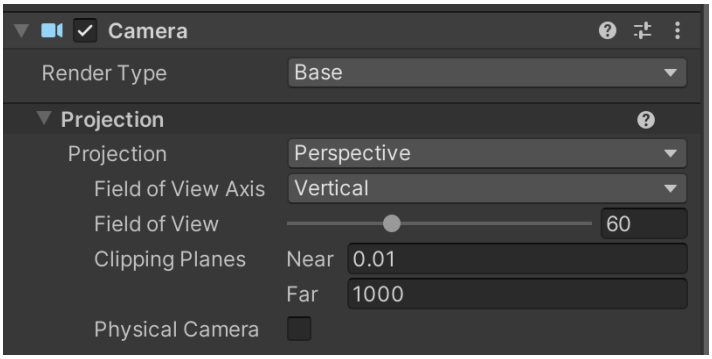

Figure 34: Camera component used in the XR Origin game object in the project

3.2.1.2.2. Lighting

Lighting in Unity is a factor that affects the appearance of objects in the scene. Lighting significantly affects the atmosphere and visual quality in a 3D environment. Unity offers us 4 different lighting options: directional, point, spot, and area light. A single directional light was used to illuminate the stage in the play.

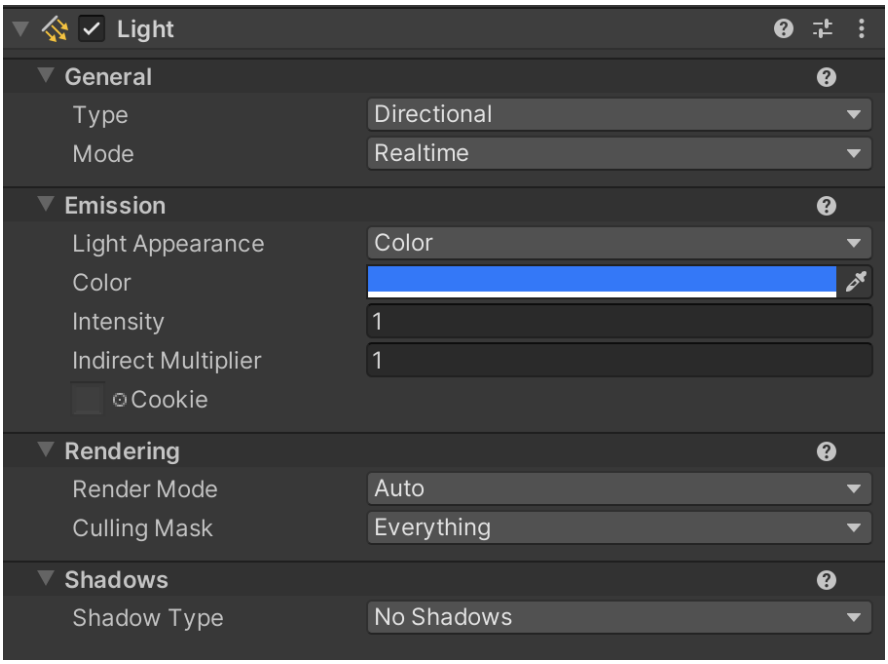

Figure 35: Directional light values used in the game scene

3.2.1.2.3. Models

Some models in the game were created from Unity 3D objects, while some were modeled by me in Blender, and some were found on internet sources and edited by me via Blender. Since designing realistic models, such as heart and human models, is a very time-consuming and complex process, such a path was followed for some models within the scope of this project. In order to import models into Unity, they must be in FBX format. Models were exported in FBX format in Blender and added to Unity.

3.2.1.2.4. Textures and Materials

In Unity, textures and materials are components that contribute to the visual quality and realism of 3D models. In general, no detailed textures were used in the game except for the chair, and all materials were selected in accordance with a certain color pattern. A transparent material was assigned to the patient model in order to show the sternum and cup model inside. In general, materials in blue, turquoise, and white tones were used in the game.

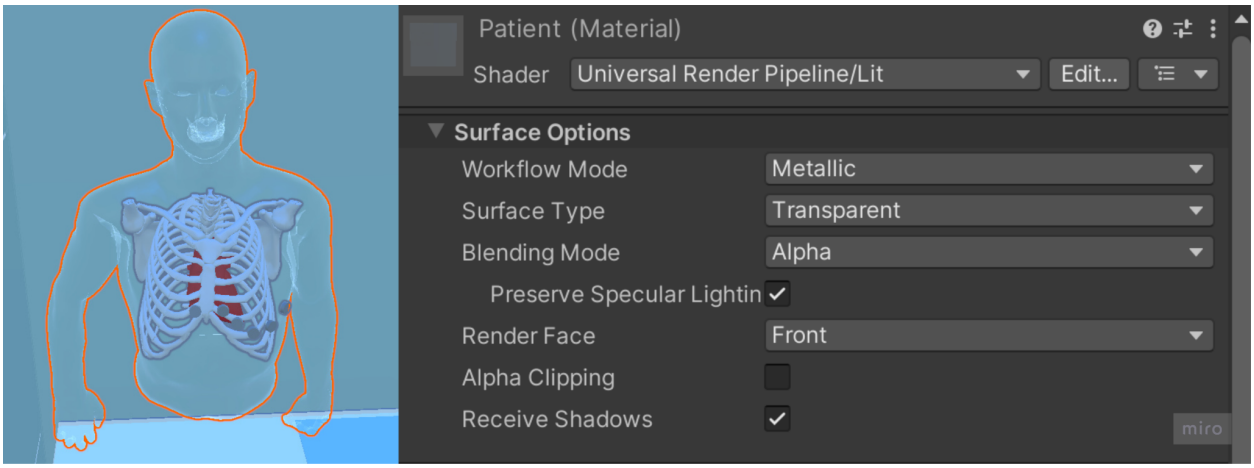

Figure 36 : Transparent material assigned to the human model surrounded by orange lines on the right

3.2.1.3. Animations

Animations work like state machines and with parameters such as trigger and bool, we ensure that the animation passes to the desired state in the script.

Some small animations were used in the project. For example, when the buttons of the electrodes in Figure 36 were pressed in part 2 of the learning, the points on the body were animated to grow and shrink. This animation was made to draw attention to the relevant point. Each electrode has its own animator and animation. When the relevant button is pressed, the "lead_scale" trigger selects the flow and the animation from the empty state to the lead_scale state. The same scale animation logic was also used in the level 1 game in Figure 23, where we showed the structures that play a role in the electrical activity of the heart in the 3D model.

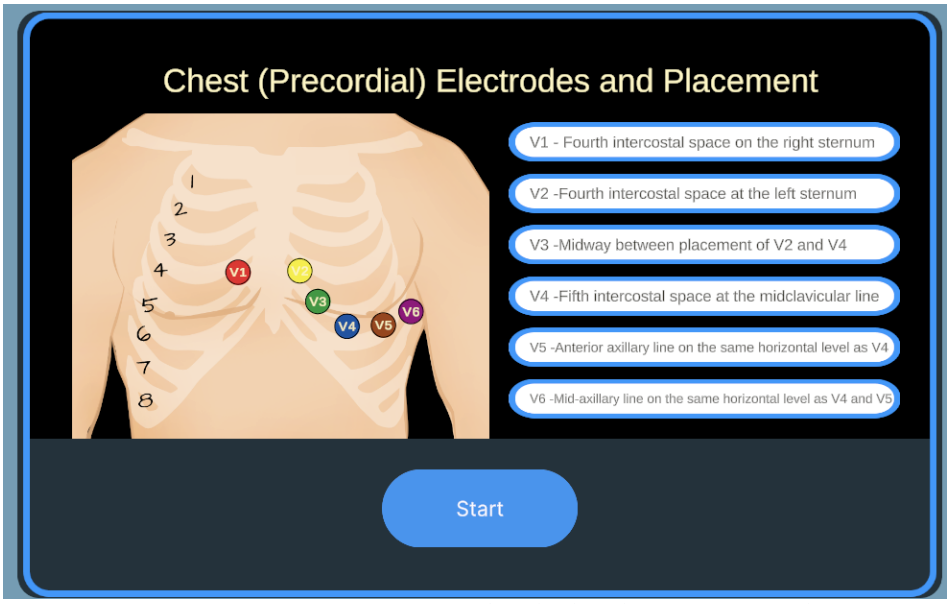

Figure 37: Learning part 2 screen where lead_scale animation is made

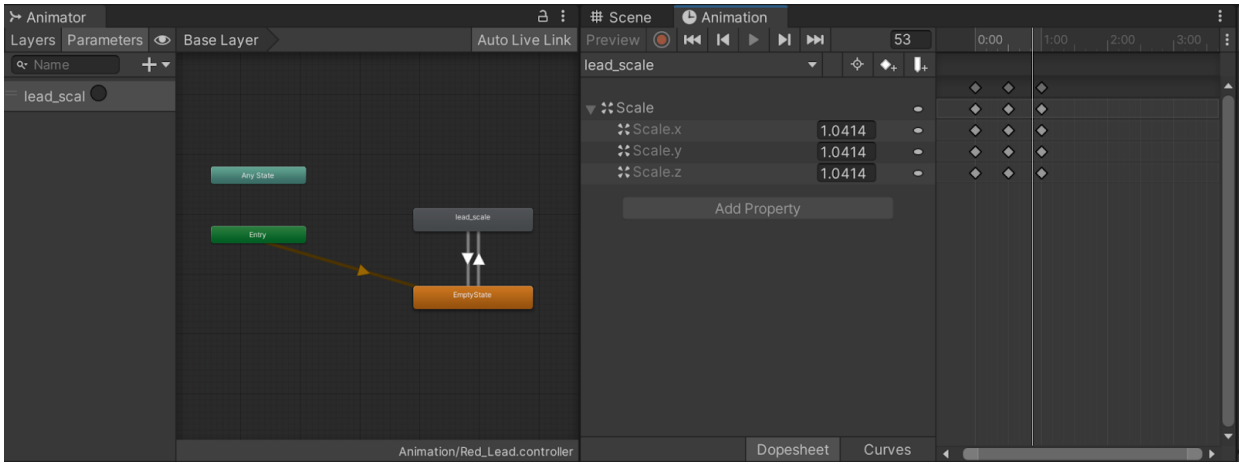

Figure 38: Lead_scale animator and animation

3.2.1.4. Controllers and Input System

Users can interact with the game with the hand trackers in their hands when the game starts. The hand trackers are managed by the buttons seen in Figure 39 and Figure 40. Here I created an XR origin game object using the methods in Unity's XR Toolkit. The descriptions of some buttons are as follows.

- UI press : Users can interact with UI objects
- Grab : Users can interact with grabs and 3D models
- Turn : Users can teleport to any location they want within the teleport area by holding down the turn thumbstick. They can rotate the camera around themselves by rotating the thumbstick
- Move : Users can move forward, backward, right and left using the move thumbstick.
- Button two (Y button) : Users can open and close the menu by pressing the Y button of the left hand tracker.

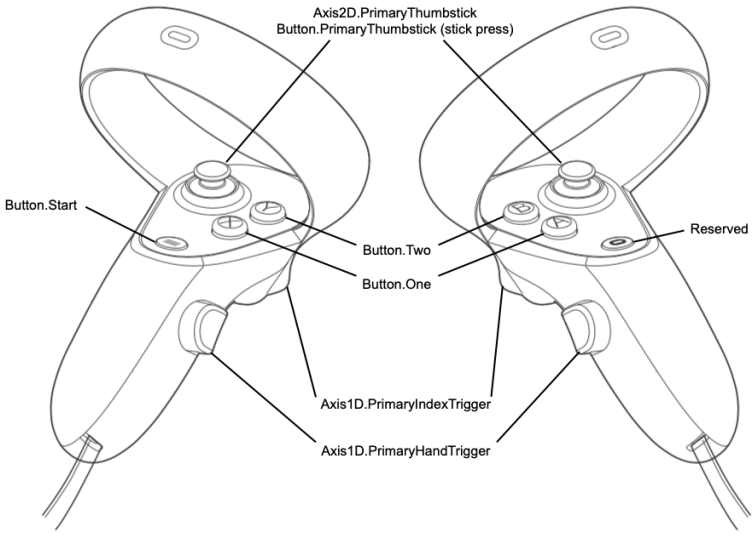

Figure 39 : Individual accessing Virtual Controller [32]



Figure 40: In-game view of hand tracker controllers

3.2.1.5. Scripts

My code structure consists of managers that manage the overall game and game scripts that control mini games. I also used Unity's XR Interaction Toolkit version 2.5.2, which was developed for VR, as an additional package for interaction with VR objects, management of hand trackers and controls of VR-related objects such as Anchor Area. Using this toolkit, I created hand tracker controllers and interactive objects and integrated them with my own scripts.

3.2.1.5.1. Main Managers

Manager scripts are generally scripts that create the main game loop and manage the game by communicating with each other. For example, when a task is completed, communication begins between many managers. TaskManager tells LevelManager to change the layout and SoundManager to play the completed sound. When Level Manager receives this information, it tells UI Manager to change the relevant UI and AnchorManager to remove the current anchor from the stage and change the position of the new anchor. In this way, all actions that occur in the game are managed by managers.

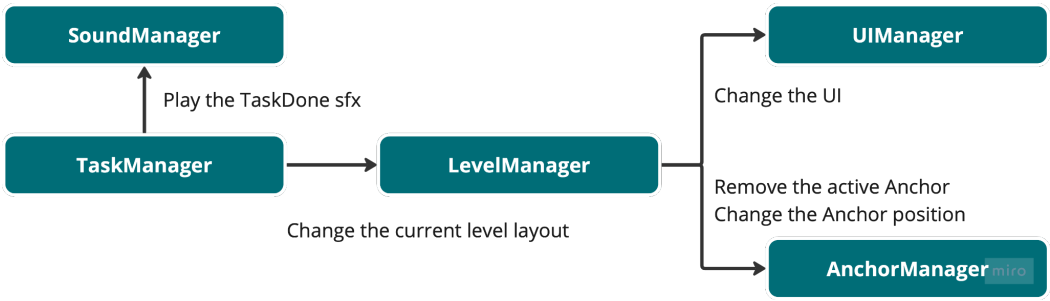

Figure 41 : An example of communication between managers when a task is completed

The parts of the game related to the User Interface are managed by UIManager. For example, by calling the OpenPanel( ) and ClosePanel( ) methods from other scripts, we can specify the panels we want to open and close wherever we want. Also, as in the OnCaseButtonClick( ) method, when a UI element is clicked, we make it do what we want it to do. We assign the name of the Level object, which is a scriptable object, to the text components of the tasks in the Training tasks section in the menu in Figure 30. Thus, we can easily change the texts of the tasks whenever we want. We can also use the panelList list to make transitions between screens in the Menu.



Figure 42 : Some important methods in UIManager

Tasks in the game are managed by TaskManager in the level. It checks the completion status of tasks in the game and notifies other managers. In the snippet we see in Figure 41, when a task or all tasks are completed in the CompleteCurrentTaskCoroutine( ) method, I changed the game's level and task status via the currentTaskIndex and currentLevelIndex values of LevelManager. Accordingly, the next task or level is passed and the necessary values in the game are changed, and the relevant arrangements are made.



Figure 43: CompleteCurrenTaskCoroutine method in TaskManager

LevelManager manages the levels in the game and most of the game in general. This script keeps track of the levels and tasks, controls them and provides transitions between levels. For example, NextTask() and NextLevel() methods decide what will happen if the player completes his task and performs the action. The voice over used in transitions between levels and the objects belonging to the relevant level are initialized with the PlayCurrentVoiceOverAndInitialize( ) method. It communicates with the

AnchorManager and sets the anchor point location of the relevant level or task. In general, it provides control of the objects related to the level in the scene and manages the game dynamically.

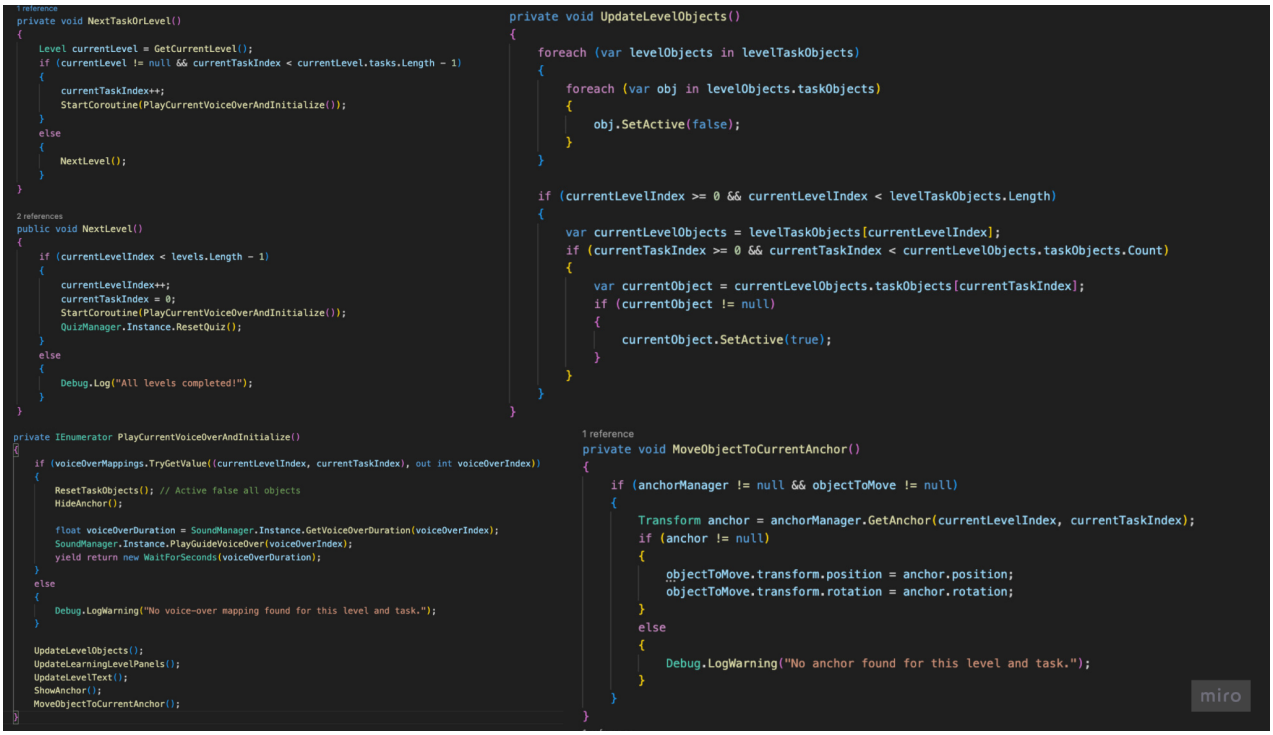

Figure 44 : Some important methods in LevelManager

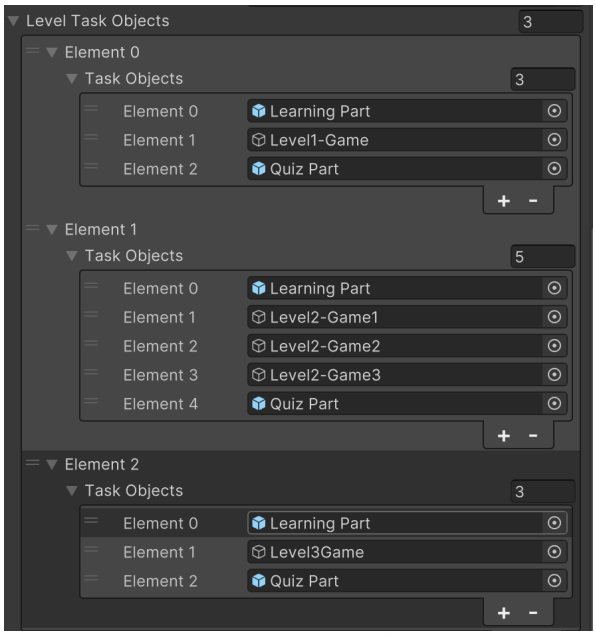

Figure 45: Level task object assignments in the Level Manager contain these objects according to the game scene task and level index.

The script that manages our score in the profile interface in the game, calculates and visually updates the points earned, and provides the player with badges when certain thresholds are reached is PointSystem. The total Point value is used to keep the total points earned by the player, and the UpdateUI( ) method updates the badgeSprites and clinicianRanks according to the point thresholds in the badgePoints list to

update the UIs in the profile menu according to these points. While the AddPoint( ) method is used to earn points, there is a DecreasePoints( ) method that reduces our score when wrong answers are given. Thanks to this class, we can reduce and increase points wherever we want in the game through other managers.

```
public void AddPoints(int points)
{
    totalPoints += points;
    UpdateUI();
}
1 reference
public void DecreasePoints(int points)
{
    totalPoints -= points;
    if (totalPoints < 0) totalPoints = 0; // point cant negative
    UpdateUI();
}

3 references
void UpdateUI()
{
    pointsText.text = "Points: " + totalPoints;

    progressBar.value = totalPoints / maxPoint;

    for (int i = 0; i < badges.Length; i++)
    {
        if (totalPoints >= badgePoints[i])
        {
            badges[i].sprite = badgeSprites[i];
            badges[i].gameObject.SetActive(true);

            rankText.text = clinicianRanks[i];

        }
    }
```

Figure 46: Level task object assignments in the Level Manager contain these objects according to the game scene task and level index.

Also among the manager scripts are the Quiz Manager, which I coded to manage the Quiz Section and output questions according to levels, the SoundManager, which I coded to control all sounds in the game and manage voice overs, and the StepManager scripts that come in the XR Toolkit package, which I use in transitions between screens in the Learning Section.

3.2.1.5.2. Game Managers and Scripts

Apart from the managers, I coded some scripts to control the mini games, especially in the 3D interactive section of the game. These scripts determine the conditions required to complete certain tasks in the game, provide this information to the managers and help organize the flow of the game.

I coded the HeartAnalysis.cs script for the Heart Conduciton System Game in Figure 23. First, the first interactable game object becomes active. With this script, when the player hovers the hand tracker

controller over the active interactable game object, the activeElementNameText and hoverTextArea fields are updated. The information of the relevant area is sent here. When this interactable object is selected with the UI press button, the next interactable object on the heart becomes active. When all the objects are selected in this way, the heart model gets a next button. The player ends this task by pressing this button.

```
2 references
private void OnSelectEntered(SelectEnterEventArgs args)
{
    var interactable = args.interactableObject as XRSimpleInteractable;
    if (interactable != null)
    {
        if (!clickedInteractables.Contains(interactable))
        {
            clickedInteractables.Add(interactable);
            //interactable.enabled = false;

            currentActiveIndex++;
            if (currentActiveIndex < interactables.Count)
            {
                interactables[currentActiveIndex].enabled = true;
                interactables[currentActiveIndex].gameObject.SetActive(true);
                UpdateActiveElementName(interactables[currentActiveIndex].gameObject.name);
            }
            CheckAllClicked();
        }
    }
}

1 reference
private void CheckAllClicked()
{
    if (clickedInteractables.Count == interactables.Count)
    {
        ShowCompleteButton();
        SoundManager.Instance.PlayDone();
    }
}
```

Figure 47: Snippet of scripts that check the list of interactable objects in the HeartAnalysis script and finish the task

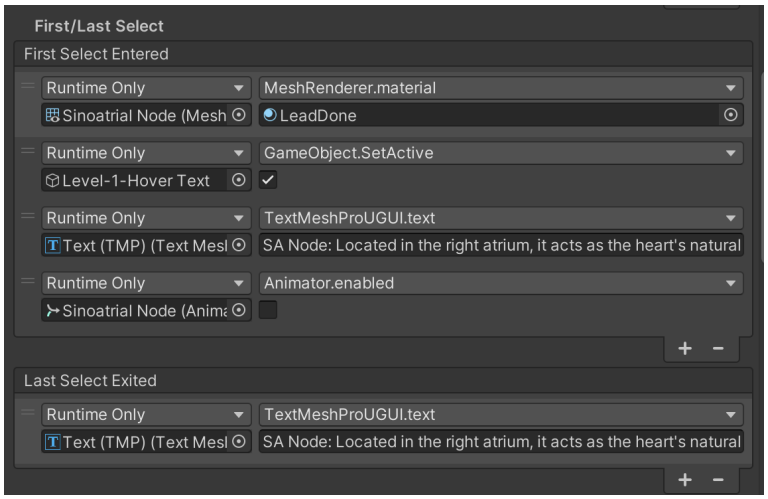

Figure 48: Actions that will occur when interactive objects are selected

I coded the SnapToPosition script for the Precordial electrodes placement game in Figure 26. This script allows us to snap the electrode object to the target patch object and controls this flow. I created a list of snapPairs from the SnapPair class, which has initialPosition and targetPosition for six different electrodes. Using this list, I ensured that the electrodes, which are XRGrabInteractable objects, would stick to that position if they had a certain proximity to their targetPosition. If the distance between the target position and the position where the interactable object was dropped was greater than the snapProximity value, I ensured that the object would return to its initial position with the ReturnToInitial( ) method. If the object was placed, I created a line with the LineRenderer in the DrawLineToHeart() method to show the vector created by the electrode towards the heart. I completed the task with the CheckAllSnapped method if all the interactable objects snap to their targets.

```
private void SnapToTarget(SnapPair pair)
{
    if (pair.targetPosition != null)
    {
        Vector3 targetPosition = pair.targetPosition.position;
        targetPosition.y -= 0.03f;
        pair.lead.transform.position = targetPosition;

        Vector3 targetEulerAngles = pair.targetPosition.rotation.eulerAngles;
        Vector3 leadEulerAngles = new Vector3(targetEulerAngles.z - 90, 270 + targetEulerAngles.y, -90);
        Quaternion leadRotation = Quaternion.Euler(leadEulerAngles);
        pair.lead.transform.rotation = leadRotation;

        Rigidbody rb = pair.lead.GetComponent<Rigidbody>();
        if (rb != null)
        {
            rb.isKinematic = true;
            rb.velocity = Vector3.zero;
            rb.angularVelocity = Vector3.zero;
        }

        Renderer leadRenderer = pair.lead.GetComponent<Renderer>();
        if (leadRenderer != null && pair.newMaterial != null)
        {
            leadRenderer.material = pair.newMaterial;
        }

        pair.lead.enabled = false;

        AudioSource audioSource = pair.lead.GetComponent<AudioSource>();
        if (audioSource != null)
        {
            audioSource.Play();
        }

        SoundManager.Instance.PlayDone();
        pair.isSnapped = true;
    }

    DrawLineToHeart(pair.lead.transform.position, heartObject.position);
    CheckAllSnapped();
}
```

```
private void ReturnToInitial(SnapPair pair)
{
    if (pair.initialPosition != null)
    {
        pair.lead.transform.position = pair.initialPosition.position;
        pair.lead.transform.rotation = pair.initialPosition.rotation;

        Vector3 leadEulerAngles = new Vector3(90, 180, 0);
        Quaternion leadRotation = Quaternion.Euler(leadEulerAngles);
        pair.lead.transform.rotation = leadRotation;

        Rigidbody rb = pair.lead.GetComponent<Rigidbody>();
        if (rb != null)
        {
            rb.isKinematic = false;
            rb.velocity = Vector3.zero;
            rb.angularVelocity = Vector3.zero;
        }

        pair.isSnapped = false;
    }
}

1 reference
private void DrawLineToHeart(Vector3 startPosition, Vector3 endPosition)
{
    LineRenderer newLineRenderer = Instantiate(lineRendererPrefab);
    newLineRenderer.positionCount = 2;
    newLineRenderer.SetPosition(0, startPosition);
    newLineRenderer.SetPosition(1, endPosition);
    activeLines.Add(newLineRenderer);
}

private void CheckAllSnapped()
{
    foreach (var pair in snapPairs)
    {
        if (!pair.isSnapped)
        {
            return;
        }
    }
    ClearActiveLines();
    TaskManager.Instance.CompleteCurrentTask();
}
```

Figure 49 : Some important methods in the SnapToPair script

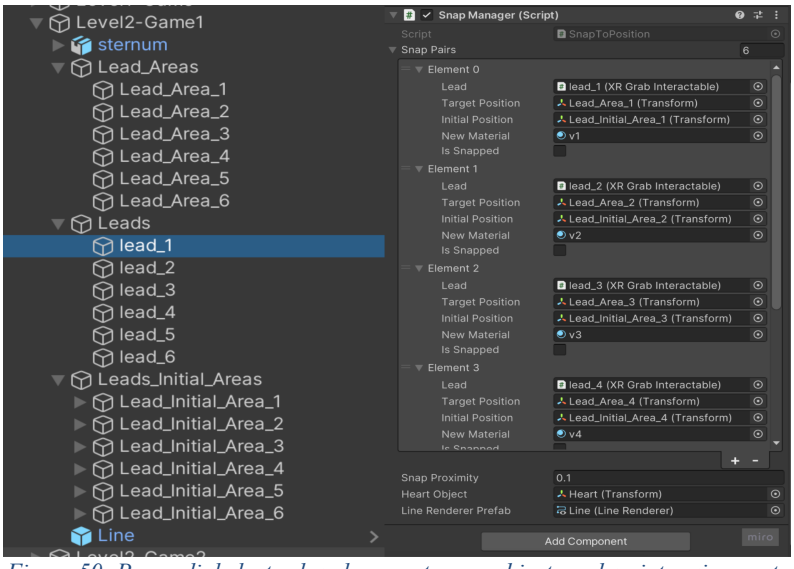

Figure 50: Precordial electrodes placement game objects and script assignments

I coded ECGSnapPosition and ECGSnapManager scripts to manage the ECG waveforms game in Figure 27. ECGSnapPosition script works with the same logic as HeartAnalysis script and also checks whether the objects are snapped or not and each sub is assigned to its own object for the game. When each game is finished, ECGsnapPosition script notifies ECGSnapManager script that the game is over via CompleteSubGame( ) method. ECGSnapManager also increases currentIndex value, updates the title information screen and reference image in the scene and activates other abnormal status game. When all games are finished, ECGSnapManager notifies TaskManager about the status and task is completed.

```
2 references
private void UpdateUIForSnapPosition(ECGSnapPosition snapPosition)
{
    foreach (var image in eCGSnapPositionsImages)
    {
        image.gameObject.SetActive(false);
    }

    foreach (var title in eCGSnapPositionsTitle)
    {
        title.gameObject.SetActive(false);
    }
    if (snapPosition != null)
    {
        int index = eCGSnapPositions.IndexOf(snapPosition);
        if (index >= 0 && index < eCGSnapPositionsImages.Count && index < eCGSnapPositionsTitle.Count)
        {
            eCGSnapPositionsImages[index].gameObject.SetActive(true);
            eCGSnapPositionsTitle[index].gameObject.SetActive(true);
        }
    }
}

1 reference
public void CompleteSubGame()
{
    if (currentIndex < eCGSnapPositions.Count)
    {
        StartCoroutine(DelayedSetNextSnapPosition(3f));
    }
    else
    {
        OnAllTasksCompleted();
    }
}
```

Figure 51 : Methods for updating UI and finishing sub-game in ECGSnapManager script

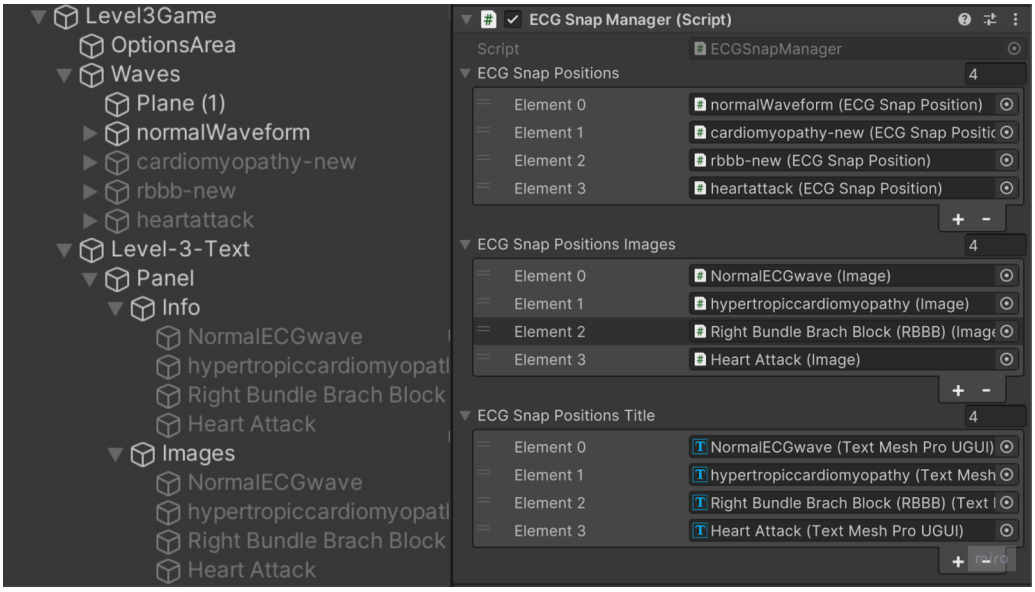

Figure 52: Game objects in the ECG waveforms game and ECGManager assignments on Unity

In some parts of the game, I made some coding for UI screens apart from the three-dimensional objects on the stage. For example, in the training section, I coded a script called ECGGenerator, which I created an animation in which I emphasized the ECG lines in level 3 and where the components of the waves in the ECG line were valid. I drew the ECG wave using the DrawECG method and LineRenderer. I gave a duration time as a time period for each component of the ECG waveform. Each wave function calculates the amplitude and duration of the wave using sinus functions and drew the wave in this way. I also created the animation through methods that determine the positions of the followerObject between two waves.

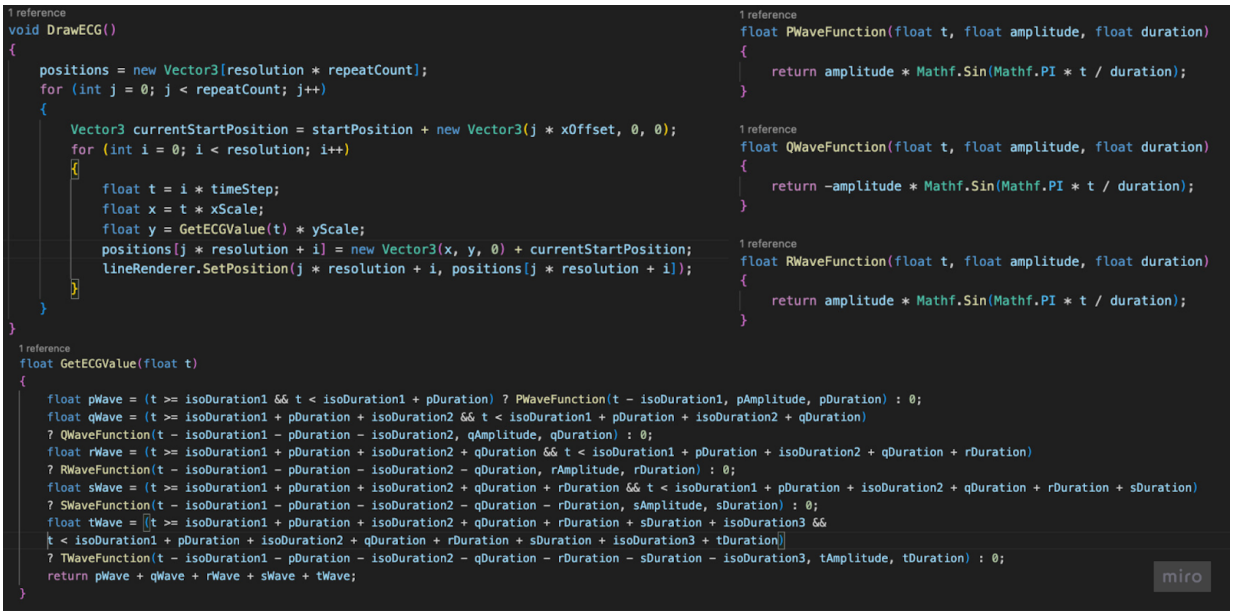

Figure 53: Methods that create ECG waveform using LineRenderer in ECG Generator script

The SimulationGameManager that I coded for the Case Level section in Figure 28 manages the Case Level flow. When the game starts, the instruction panel opens. When the player completes this section, the instruction panel is closed and the objects for the lead section are activated. Then, when all the XRGrabInteractable objects using the snapToPosition script are placed, the casePanel containing specific information about the case opens, the abnormal heart sound is heard and the ECG monitor displays the real ECG image of the relevant case on the screen. The player ends the game by pressing the finish button on the CasePanel.

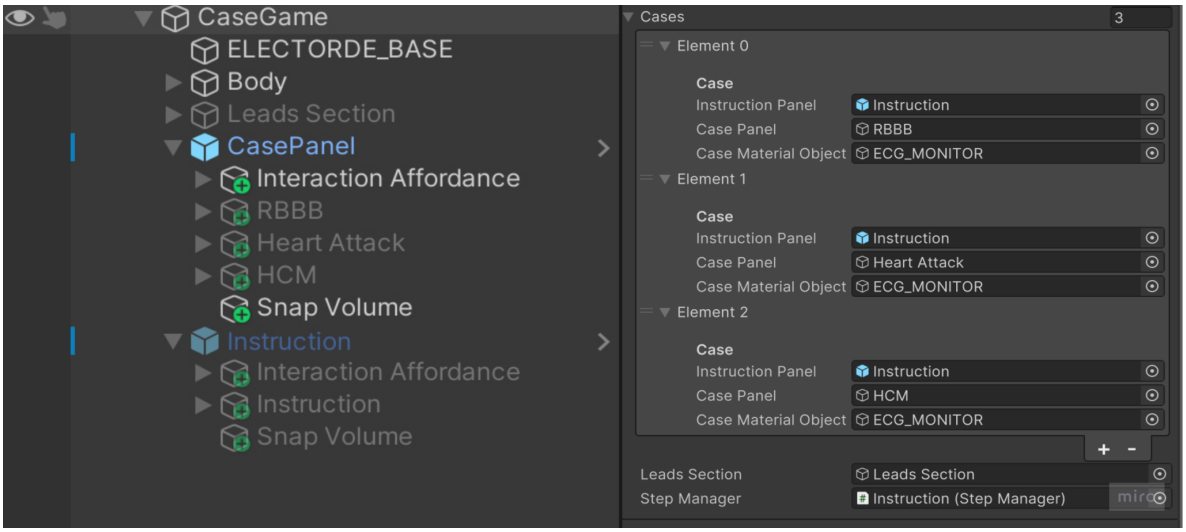

Figure 54: Three different case panels at the case level and their assignments on the inspector.

3.2.2. Technologies and Tools

A. Adobe Illustrator
Adobe's Illustrator application used for vector drawings was used to edit and create 2D designs. Vector drawings can be sized and easily edited as desired due to their structure, and this tool was used because of Illustrator's ability to export at any resolution you want.

B. Audacity
This tool is used to edit various sound files. It was used to edit wav and mp3 sound files within the scope of the project.

C. C#
The C# language is used in scripts in Unity Engine. With its object-oriented structure, it offers a very easy coding experience while coding with its suitability for game coding. C# language and its features were used in this project.

D. Meta Quest 2
It is a VR headset developed by Meta company. Within the scope of the project, the headset connected to Unity Engine with a USB-C cable was built and tested in APK format. It was used as the device on which the simulation APK was tested and run within the scope of the project.

E. Gitlab
Gitlab is a web-based version control system used to facilitate the management of software projects while developing them. The repository provided by the university was used within the scope of the project and the project was managed in this way.

F. Miro Board
Miro Board is a whiteboard application that is used for brainstorming and visualizing ideas. It was used in many areas within the project, from project management to determining and grouping tasks, from experience design to developing ideas about simulation. It was also used to explain project stages and plans more easily in the weekly supervisor meeting.

G. Play.ht
Play.ht allows users to convert written texts into real sounds using text to speech technology. Within the scope of the project, medical theoretical explanations and in-game guidance voice overs were produced with artificial intelligence using this website.

H. Figma

Figma is a web-based user interface and user experience design tool. It was used to create wireframes in the design phase of the project.

I. Leonard AI

Leonardo AI is a platform that produces visual content for game developers and digital artists using artificial intelligence technology. It was used to generate the App Icon image in the project.

J. Blender

Blender is an open source and free 3D modeling and animation editing software. It was used to edit 3D modeling and models available on the internet in the project. The models were exported in FBX format and transferred to Unity.

K. Jira

Jira is a tracking tool used for planning and managing software development projects while being developed by Atlassian. Customized Jira boards provided by the university were used in the project. A Kanban was created for the project section and this Kanban was created as Issues, Epic, Story and Task on the board.

L. Visual Studio Code

Visual Studio Code is a fast, free and platform-independent source code editor developed by Microsoft. It was used to create and test scripts within the scope of the project.

M. Unity3D Engine

Unity engine is a game engine that is especially popular in the gaming world. This engine was chosen for the development of the project due to the ability to develop on many platforms simultaneously and its suitability for VR development environments. The game can be built in Unity3D Engine and used as an APK while testing and deploying. Before the project development started, the Unity project was set up and uploaded to Gitlab. The Unity version was used as the 2022.3.18 version, which does not cause problems in VR projects.

4.Testing and Result Analysis

4.1. Survey

Testing Result

The survey was conducted to measure the impact of the VR-based ECG interpretation game on various test groups. There were 5 participants in total and the participants varied in age range, gender, profession and experience. The results show that the simulation was generally evaluated positively and was found successful.

Demographics and Background

Most of the participants were between the ages of 25-34. In terms of gender, there were four male and one female participants. The participants were from different professions such as computer engineers and doctors. This allowed the application to be evaluated both in terms of software and medicine. Only one participant had such experience with medical VR training applications, while the other participants did not have any experience.

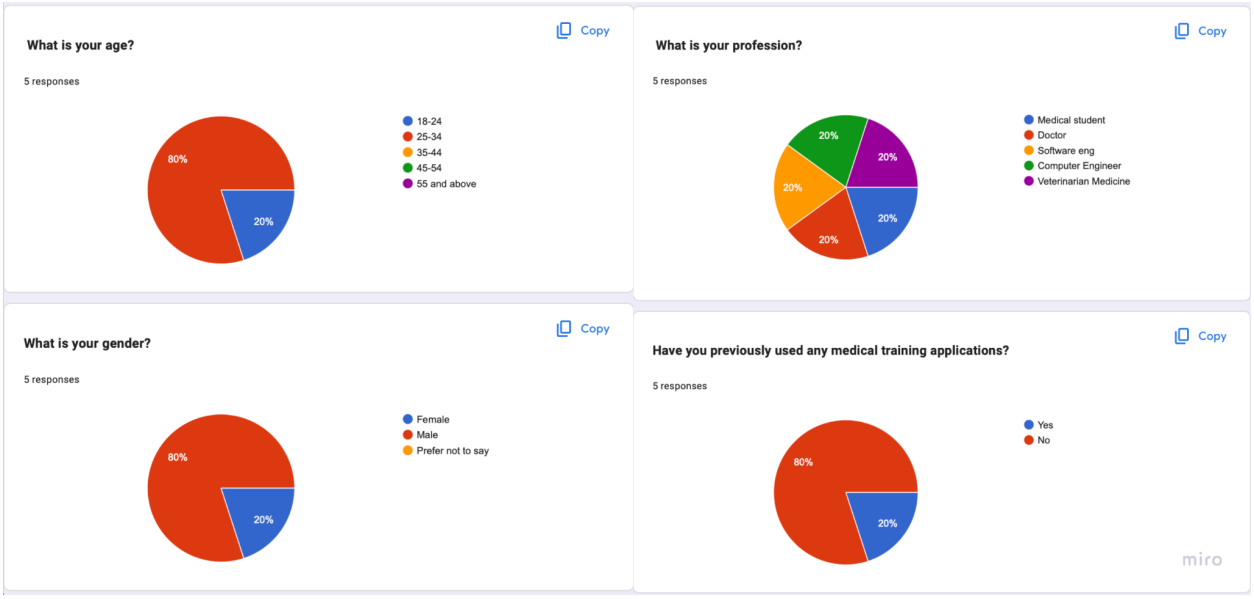

Figure 55 : A section of the results of the survey created in Google form and presented to users

Educational Effectiveness

Participants evaluated the educational content of the simulation as “Very effective” and “Effective”. All participants stated that the simulation contributed to the ability to interpret ECG reports.

User Experience

The game was evaluated more negatively in terms of user experience compared to other questions. According to the participants' feedback, the simulation was evaluated as moderately effective and

effective in terms of user experience. Only one participant evaluated it as Very effective. This was an indication that the simulation needed to be improved in terms of experience.

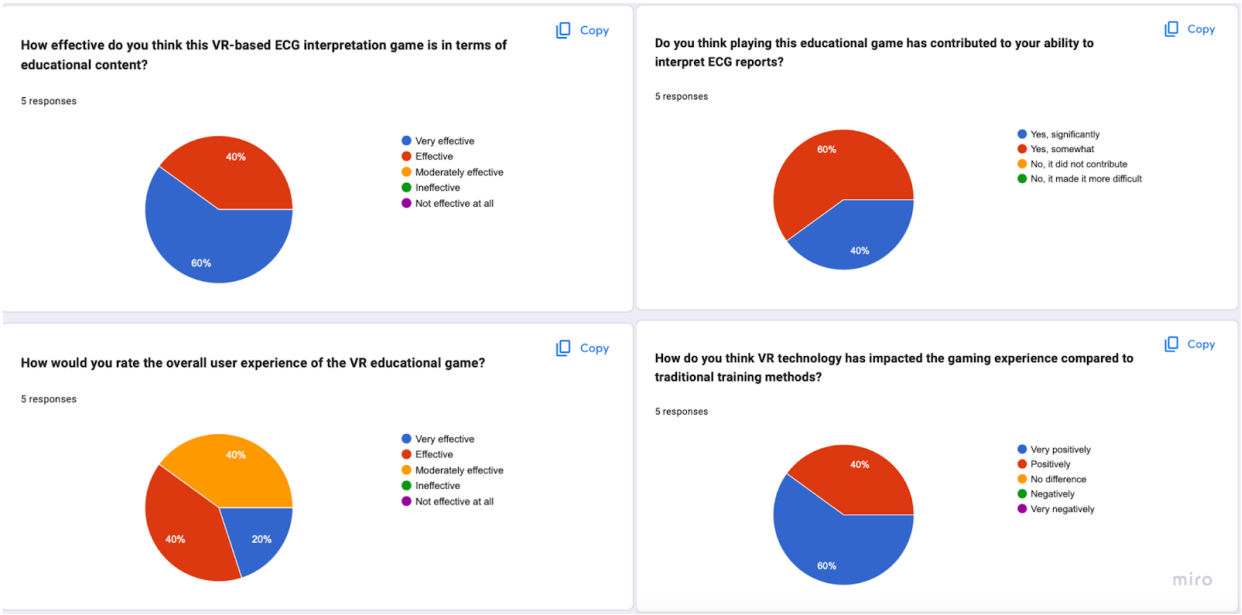

Figure 56 : A section of the results of the survey created in Google form and presented to users

Educational Content

While the effectiveness of the educational content was evaluated more positively, the scope of the educational content was evaluated more negatively. Here we can draw the following conclusion; While learning with VR simulation made the participants think that it was interesting and could contribute to their development in the subject, they thought that the simulation was limited in terms of content scope.

In addition, 60% of the participants evaluated the user interface and introductory design as “moderately user friendly”, while 40% evaluated it as “user friendly”. No one responded as “very user friendly”.

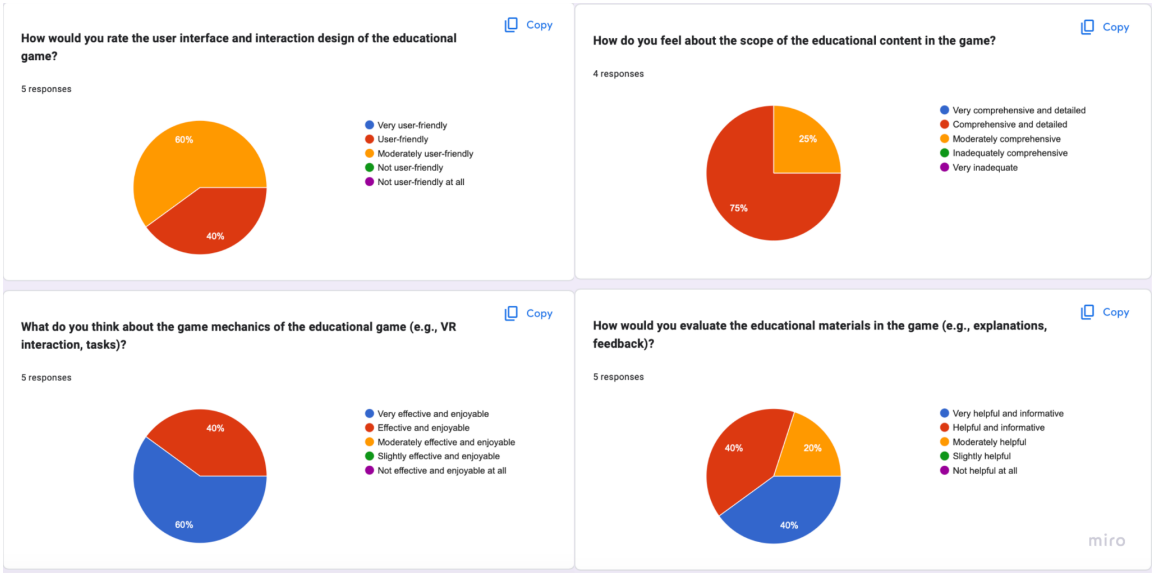

Figure 57 : A section of the results of the survey created in Google form and presented to users

All of the participants stated that they would recommend the simulation. Four of them stated that they would definitely recommend this simulation.

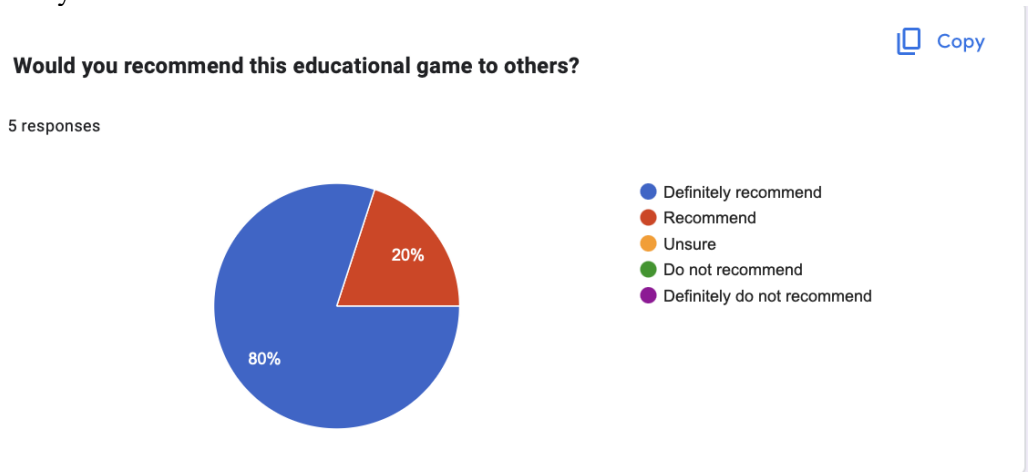

Figure 58 : A section of the results of the survey created in Google form and presented to users

As a result, the simulation aimed at improving VR-based ECG interpretation received positive feedback, especially in terms of the effect of VR technology on educational effectiveness. Although it received lower evaluations in terms of user experience and educational content, the developed application is recommended to others by all participants.

5. Project Management

While managing the project, I first determined the work breakdown structure, Gantt chart and milestones and deliverables. First of all, I determined six milestones as project planning, research and analysis, design process and prototyping, engine implementation and coding, testing and bug fixing and finalization. After the project started, I used the Gitlab repository created by the university for the project version control system. I used the Jira board created by the university as the Kanban board. When the cumulative chart of Jira in Figure 62 is compared with the Gantt chart, we see almost similar planning and project processes. During the project, there was a longer design process and prototyping process than expected, and for this reason, I started the Unity implementation and therefore coding a little late.

Task No	Task Name	Start Date	End Date	Duration (Days)	2024
	Medisim Project	3/22/2024	8/27/2024	158	March April May June July August
	1. Project Planning	3/22/2024	3/29/2024	7	
1	1.1. Creation of epic stories and tasks on Jira board	3/22/2024	3/27/2024	5	
2	1.2. Setting up the project in Unity	3/27/2024	3/29/2024	2	
	2. Research and Analysis	3/29/2024	4/22/2024	24	
3	2.1. Research and Determination of Technologies and Requirements	3/29/2024	4/5/2024	7	
4	2.2. Detailed Examination of Existing Studies	4/5/2024	4/12/2024	7	
5	2.3. Research on Contents and Use Cases	4/12/2024	4/22/2024	10	
	3. Design Process and prototyping	4/22/2024	6/8/2024	47	
6	3.1. Designing sketches in Figma environment in Unity	4/22/2024	4/27/2024	5	
7	3.2. Preparation of designs for the general concept of the simulation	4/27/2024	5/4/2024	7	
8	3.3. Making the necessary 3D and User interface designs	5/4/2024	5/18/2024	14	
9	3.4. Preparation of ECG graphs and heart sounds	5/18/2024	5/25/2024	7	
10	3.5. Development of basic simulation mechanics	5/25/2024	6/1/2024	7	
11	3.6. Testing of the prototype and improvements	6/1/2024	6/8/2024	7	
	4. Unity implementation and coding of contents	5/15/2024	7/15/2024	61	
12	4.1. Defining and coding user interactions	5/15/2024	5/29/2024	14	
13	4.2. Creation and coding of the training section	5/29/2024	6/12/2024	14	
14	4.5. Creating and coding the quiz section	6/12/2024	6/26/2024	14	
15	4.4. Creating 3D interacable games	6/26/2024	7/10/2024	14	
16	4.3. Creating and coding case section	7/10/2024	7/15/2024	5	
	5. Testing and Bugfixing	7/15/2024	8/10/2024	26	
17	5.1. Creation of system tests and unit tests	7/15/2024	7/20/2024	5	
18	5.2. Testing with VR headset	7/20/2024	7/27/2024	7	
19	5.3. Debugging and Correcting Errors	7/27/2024	7/31/2024	4	
20	5.4. Testing and Verification	7/31/2024	8/10/2024	10	
	6. Finalization and Distribution:	8/10/2024	8/22/2024	12	
21	6.1. Conducting final tests of the entire simulation and making final corrections	8/10/2024	8/14/2024	4	
22	6.2. Creating the final version of the project	8/14/2024	8/18/2024	4	
24	6.4. Completion of the documentation of the project	8/14/2024	8/27/2024	13	

Figure 59: A Gantt Chart

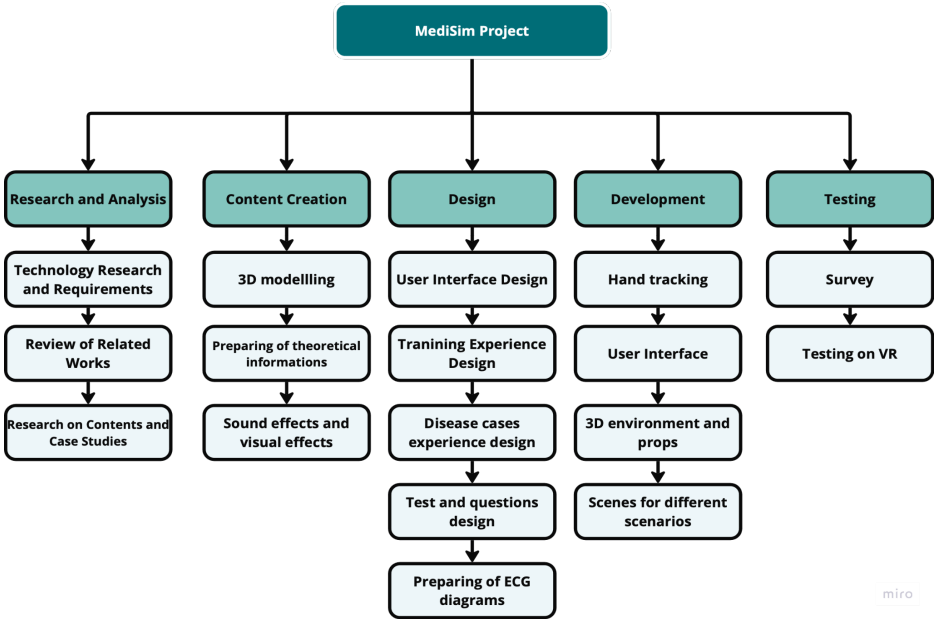

Figure 60 : A Work Breakdown Structure

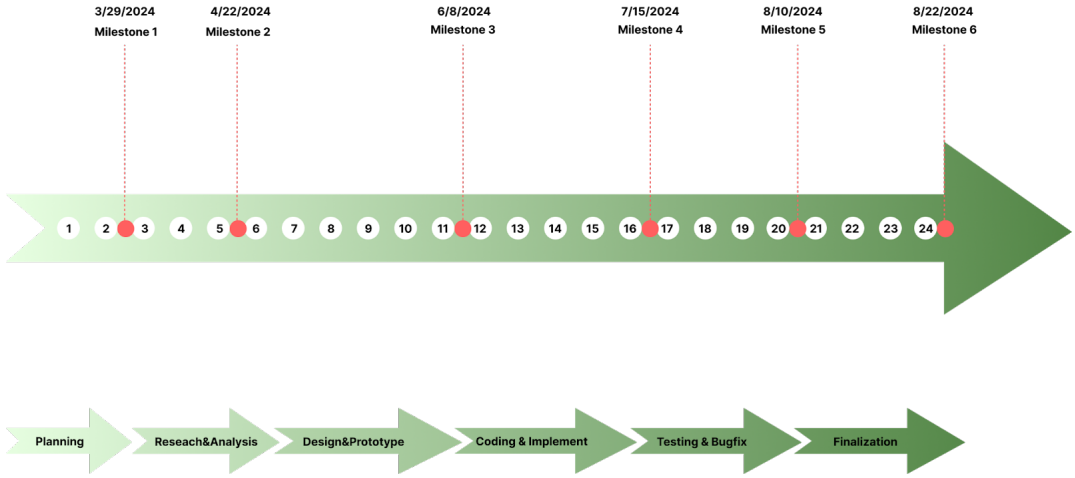

Figure 61 : A Milestones and deliverables

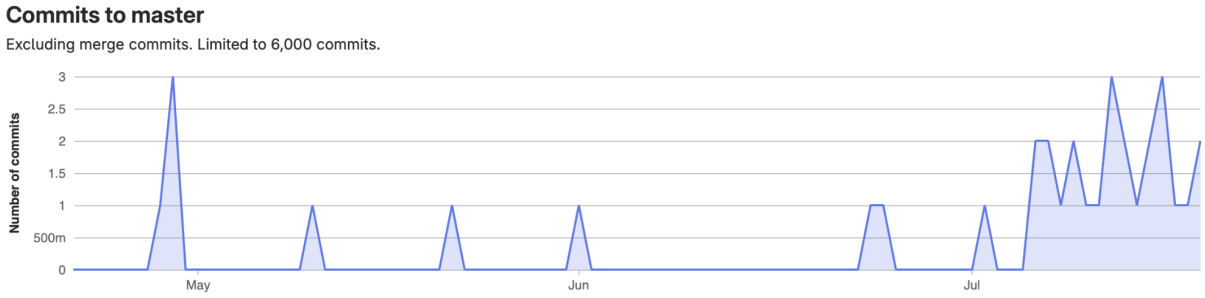

Figure 62 : Screenshot of commits made throughout the project taken from gitlab

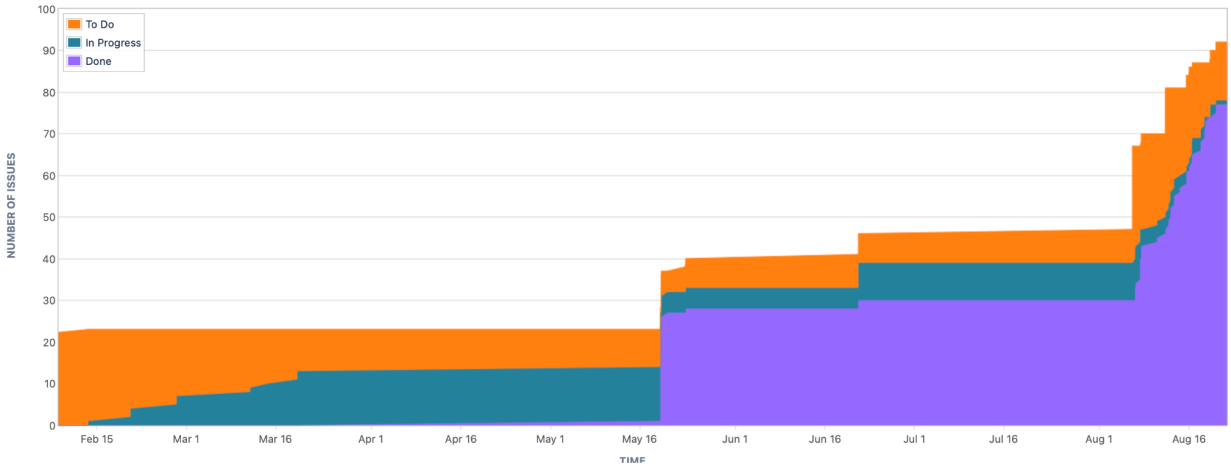

Figure 63 : Cumulative diagram which is a Kanban board showing to-do in progress and done stages.

6. Conclusions and Future Work

The aim of the project was to create a VR-based simulation that combines visual, theorical and auditory elements in a virtual environment to improve ECG interpretation skills for medical professionals and medical students. This ECG training simulation project initially began when I asked my friend, who is a professional physician, about the difficulties experienced in medical education. Then, in my research, I saw that ECG is a subject of emergency medicine and plays a critical role in preliminary diagnoses.

As I mentioned in the literature review section during the research process, I observed in many studies that especially medical students increase their relevant skills in learning through medical simulations. After the research was completed, I prepared a simulation with Unity engine. This simulation included theoretical information about the inner workings of the heart, the formation of ECG waves and the ECG lead placement sections that greatly affect the correctness of ECG reports, interactive mini games with three-dimensional interactive nail polishes that will allow them to reinforce these topics, quizzes about the topics and abnormal situation case scenes that they can experience after all the training is over.

After the development process was completed, we see that in the tests conducted on five people, all of the people who participated in the experiment found this training positive in terms of educational effectiveness, but the content was low in terms of effectiveness. In addition, according to the survey conducted at the end of the experiment, 40 percent of the people who participated in the experiment marked the user experience as moderately effective, 60 percent of the people who participated in the experiment marked the user interface and interaction design as moderately user friendly, but in addition, all of them said they would recommend this application.

As we understand from this survey, the simulation had a positive effect on people, and we understand that it was seen as a useful tool that could add something to them in terms of ECG training. In this context, although the number of people who participated in the experiment was small, we can conclude that it could be a supportive training set for ECG training.

In future studies, it would be useful to conduct similar tests on a wider user base and expand the content of the simulation with the feedback of the users, and to design the user experience by adding a tutorial that will make the game more similar to real environments and easier to play, considering the users.

References

[1] D. A. Cook, S.-Y. Oh, and M. V. Pusic, “Accuracy of Physicians’ Electrocardiogram Interpretations: A Systematic Review and Meta-analysis,” JAMA Internal Medicine, vol. 180, no. 11, pp. 1461–1471, Nov. 2020, doi: https://doi.org/10.1001/jamainternmed.2020.3989
[2] E. Elsheikh, Nurah Alkhteeb, A. Alamer, M. O. Alarfaj, Ghaida AlQarni, and Jumana Alsultan, “Medical Students’ Competency and Confidence in Interpreting Electrocardiograms at King Faisal University, Al-Ahsa,” Curēus, Oct. 2023, doi: https://doi.org/10.7759/cureus.46393
[3] “Virtual Reality (VR) for Medical Education in 2024,” www.scnsoft.com. https://www.scnsoft.com/healthcare/virtual-reality/medical-education
[4] Y. Sattar and L. Chhabra, “Electrocardiogram,” National Library of Medicine, Jun. 05, 2023. https://www.ncbi.nlm.nih.gov/books/NBK549803/
[5]Virtual Reality Society, “What is Virtual Reality? - Virtual Reality Society,” Virtual Reality Society, 2017. https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html
[6] Pfizer, “Flashback: The First ECG | Pfizer,” Pfizer.com, 2022. https://www.pfizer.com/news/articles/flashback_the_first_ecg
[7] Virtual Reality Society, “History Of Virtual Reality,” Virtual Reality Society, 2019. https://www.vrs.org.uk/virtual-reality/history.html
[8] John Hopkins medicine, “Electrocardiogram,” John Hopkins Medicine, 2019. https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electrocardiogram
[9] G. Amit and K. J. Quan, “Cardiac Pacemakers – Past, Present, and Future,” Elsevier eBooks, pp. 807–816, Jan. 2009, doi: https://doi.org/10.1016/b978-0-12-374248-3.00067-7.
[10] P. Kligfield et al., “Recommendations for the Standardization and Interpretation of the Electrocardiogram,” Circulation, vol. 115, no. 10, pp. 1306–1324, Mar. 2007, doi: https://doi.org/10.1161/circulationaha.106.180200.
[11] Wikipedia Contributors, “Electrocardiography,” Wikipedia, Apr. 06, 2019. https://en.wikipedia.org/wiki/Electrocardiography
[12] “ECG Clamp Electrode,” Eibco.co.in, 2021. https://eibco.co.in/product/ecg-clamp-electrode/ (accessed Aug. 25, 2024).
[13] “ECG Colour Codes Explained,” Biometriccables, Feb. 07, 2020. https://www.biometriccables.in/blogs/blog/ecg-colour-codes-explained
[14] M. Cadogan, “ECG Lead positioning • LITFL Medical Blog • ECG Library Basics,” Life in the Fast Lane • LITFL • Medical Blog, Jan. 31, 2019. https://litfl.com/ecg-lead-positioning/
[15] Dr Araz Rawshani, MD, PhD, “ECG interpretation: Characteristics of the normal ECG (P-wave, QRS complex, ST segment, T-wave),” Cardiovascular Education, May 30, 2017. https://ecgwaves.com/topic/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/#:~:text=detect%20this%20vector.- (accessed Aug. 25, 2024).
[16] E. Burns and R. Buttner, “Right Bundle Branch Block (RBBB) • LITFL • ECG Library Diagnosis,” Life in the Fast Lane • LITFL • Medical Blog, Aug. 01, 2018. https://litfl.com/right-bundle-branch-block-rbbb-ecg-library/
[17] S. M. LaValle, Virtual Reality. Cambridge University Press, 2023.
[18] “Resolution - Virtual Reality and Augmented Reality Wiki - VR AR & XR Wiki,” xinreality.com. https://xinreality.com/wiki/Resolution
[19] S. Butler, “How Important Are Refresh Rates in VR?,” How-To Geek, Dec. 23, 2021. https://www.howtogeek.com/758894/how-important-are-refresh-rates-in-vr (accessed Aug. 25, 2024).
[20] “File:Immersive Index In Theory larger.jpg - Wikimedia Commons,” Wikimedia.org, May 16, 2022. https://commons.wikimedia.org/wiki/File:Immersive_Index_In_Theory_larger.jpg (accessed Aug. 25, 2024).
[21] “How Normal Vision Works,” Chadwick Optical, Inc.https://chadwickoptical.com/how-normal-vision-works/#:~:text=As%20you%20can%20see%20from

[22] "What are the basic components of virtual reality?," Educative. https://www.educative.io/answers/what-are-the-basic-components-of-virtual-reality
[23] Meta, "Meta Quest 2: Immersive All-In-One VR Headset | Meta Store," www.meta.com, 2023. https://www.meta.com/gb/quest/products/quest-2/
[24] N. Iqbal, R. Kandasamy, J. O, Balasundaram R, and Karthika Jyothish, "Virtual Reality Simulation for the Acquisition and Retention of Electrocardiogram Interpretation Skills: A Randomized Controlled Trial Among Undergraduate Medical Students," Cureus, Jun. 2024, doi: https://doi.org/10.7759/cureus.62170.
[25] S. P. Seslar et al., "A PILOT STUDY COMPARING A CONVENTIONAL SLIDESHOW APPROACH WITH VIRTUAL REALITY FOR TEACHING CARDIAC ANATOMY TO ELECTROPHYSIOLOGY TRAINEES," Cardiovascular Digital Health Journal, vol. 3, no. 4, p. S13, Aug. 2022, doi: https://doi.org/10.1016/j.cvdhj.2022.07.032.
[26] A. H. Kashou et al., "EDUCATE: An international, randomized controlled trial for teaching electrocardiography," Current problems in cardiology, vol. 49, no. 3, pp. 102409–102409, Mar. 2024, doi: https://doi.org/10.1016/j.cpcardiol.2024.102409.
[27] J. Brouchoud, "ECG Simulator," Acadicus Virtual Sim Lab, Oct. 15, 2020. https://acadicus.com/ecg/ (accessed Aug. 25, 2024).
[28] "EKG | MedVR Education," MedVR Education | XR Healthcare Training Solutions, Dec. 14, 2023. https://medvr.education/solution/ekg-vr/ (accessed Aug. 25, 2024).
[29] "MedVision," Medvisiongroup.com, 2024. https://www.medvisiongroup.com/leonardoVR.html (accessed Aug. 25, 2024).
[30] T. Baniasadi, S. M. Ayyoubzadeh, and N. Mohammadzadeh, "Challenges and Practical Considerations in Applying Virtual Reality in Medical Education and Treatment," Oman Medical Journal, vol. 35, no. 3, pp. e125–e125, May 2020, doi: https://doi.org/10.5001/omj.2020.43.
[31] Unity Technologies, "Unity - Manual: Scenes," Unity3d.com, 2018. https://docs.unity3d.com/Manual/CreatingScenes.html
[32] "OVRInput," developer.oculus.com. https://developer.oculus.com/documentation/unity/unity-ovrinput/

Appendix

Appendix A

Game Assets Credits

Creator	**Asset**	**Link**	**Edited**
Play.ht	All voice over	Play.ht	No
Rich (Let's Find Out)	White noise SFX	https://www.youtube.com/watch?v=Og40mpl8VNc&t=12834s	Trim
Freepik.com	2D hand gel	https://www.freepik.com/free-vector/flat-style-hand-sanitizer_7295596.htm#fromView=search&page=1&position=23&uuid=92962c6b-82bd-4174-9a71-ee89e286bafa	Edited
Freepik.com	2D Bacteria	https://www.freepik.com/free-vector/hand-drawn-virus-collection_7363905.htm	Edited
Freepik.com	2D Arm	https://www.freepik.com/free-vector/brawny-caucasian-arm_3949298.htm	Edited
Freepik.com	2D Scissor	https://www.freepik.com/free-vector/manicure-tools-concept_9472281.htm	Edited
Freepik.com	2D Broken cable	https://www.freepik.com/free-vector/broken-electrical-cable-set-vector-illustrations-damaged-cords-power-connection-cartoon-torn-wires-with-plug-vga-usb-circuit-isolated-white-break-	Edited

		disconnect-hardware-concept_22343460.htm	
Freepik.com	2D patient bed position	https://www.freepik.com/free-vector/isometric-modern-hospital-room-ward-with-male-patient-bed_2238475.htm#fromView=search&page=1&position=2&uuid=38c94006-1bfa-4613-8c19-3ef87127e868	Edited
sinapashami	3D ECG machine	https://sketchfab.com/3d-models/ecg-f3b842e5391c4455bdaf5bc9c8daca23	No
University Medical Center Groningen	3D heart Model	https://sketchfab.com/3d-models/anatomie-van-het-hart-afleidingen-en-geleiding-ada0def6172a4b26bbce45ca0d04f418	Edited
University Medical Center Groningen	3D cardiac conduction system Model	https://sketchfab.com/3d-models/cardiac-conduction-system-20a5e36391474f2b99e1a4c94c707b47	Edited
edouard77	3D Patient Model	https://sketchfab.com/3d-models/patient-00b483f284a542899b94e99831f1ad1c	Edited
gamekorp	3D Chair Model	https://www.cgtrader.com/free-3d-models/furniture/other/office-set-two	Edited

formidabelt	3D clock Model	https://www.cgtrader.com/free-3d-models/interior/interior-office/animated-functional-analog-clock-with-drivers	No
pistonstone	3D ECG monitor Model	https://www.cgtrader.com/free-3d-models/science/medical/medical-monitor-06b3d9fb-75ba-4725-b298-48655df78200	No
Rad-Coders	3D PC model	https://assetstore.unity.com/packages/3d/environments/minimalist-archviz-bedroom-131093	Edited
forthehorde	3D Sternum Model	https://www.cgtrader.com/free-3d-print-models/art/sculptures/free-ribcage-model	No
Uğur Yakışık	3D bed stuff	https://sketchfab.com/3d-models/hospital-room-39d5f1555a26402ab1a54ffbaa8afec1	No
Veterock	3D Bed model	https://sketchfab.com/3d-models/kidman-room-907ac12c61744803b22a49efd74ec40a	Edited
graphicsrf	2D frame images	https://www.vecteezy.com/vector-art/8683989-human-internal-organ-with-heart	Edited
Thinklabs	Rbbb SFX	https://www.youtube.com/watch?v=qKuHGWdFDTo	Trim
Medzcool	Heart Attack SFX	https://www.youtube.com/watch?v=JZYGQTjDvLk	Trim

Thinklabs	Hypertrophic Cardiomyopathy SFX	https://www.youtube.com/watch?v=C79cPXPHu4s	Trim
Thinklabs	Normal ECG SFX	https://www.youtube.com/watch?v=xER8Bp4L2kM	NO