UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA ELÉTRICA

PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

GUILHERME FERNANDES DE SOUZA MIGUEL

PROPOSAL OF A GAME STREAMING-BASED FRAMEWORK FOR A TELEREHABILITATION SYSTEM

UBERLÂNDIA

Agosto de 2021

GUILHERME FERNANDES DE SOUZA MIGUEL

PROPOSAL OF A GAME STREAMING-BASED FRAMEWORK FOR A TELEREHABILITATION SYSTEM

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos necessários para a obtenção do título de doutor em ciências.

Orientador: Prof. Dr. Eduardo Lázaro Martins Naves

UBERLÂNDIA

Agosto de 2021

UNIVERSIDADE FEDERAL DE UBERLÂNDIA – UFU

Reitor: Valder Steffen Junior

Pró-Reitor de Graduação: Kárem Cristina de Sousa Ribeiro

Pró-Reitor de Pós-graduação: Carlos Henrique de Carvalho

Diretor da Faculdade de Engenharia Elétrica: Sérgio Ferreira de Paula Silva

Coordenador do Programa de Pós-Graduação Engenharia Elétrica: Luiz Carlos Gomes de Freitas

Dados Internacionais de Catalogação na Publicação (CIP) Sistema de Bibliotecas da UFU, MG, Brasil.

M636p 2021

Miguel, Guilherme Fernandes de Souza, 1984-

Proposal of a game streaming-based framework for a telerehabilitation system [recurso eletrônico] / Guilherme Fernandes de Souza Miguel. - 2021.

Orientador: Eduardo Lázaro Martins Naves.
Tese (Doutorado) - Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2021.5561 Inclui bibliografia.
Inclui ilustrações.

1. Engenharia elétrica. I. Naves, Eduardo Lázaro Martins, 1970-, (Orient.). II. Universidade Federal de Uberlândia. Programa de PósGraduação em Engenharia Elétrica. III. Título.

CDU: 621.3

Glória Aparecida
Bibliotecária - CRB-6/2047

GUILHERME FERNANDES DE SOUZA MIGUEL

PROPOSAL OF A GAME STREAMING-BASED FRAMEWORK FOR A TELEREHABILITATION SYSTEM

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia como parte dos requisitos necessários para a obtenção do título de Doutor em Ciências.

Comissão examinadora:

Prof. Dr. Eduardo Lázaro Martins Naves
(Orientador – UFU)

Prof. Dra. Angela Abreu Rosa de Sá
(Examinador – UFU)

Prof. Dr. Pedro Felipe do Prado
(Examinador – UFES)

Prof. Dr. Samuel Baraldi Mafra
(Examinador – INATEL)

Uberlândia, 04 de Agosto de 2021

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Engenharia Elétrica
Av. João Naves de Ávila, 2121, Bloco 3N - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4707 - www.posgrad.feelt.ufu.br - copel@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Engenharia Elétrica				
Defesa de:	Tese de Doutorado, 289, PPGEELT				
Data:	Quatro de agosto de dois mil e vinte e um	Hora de início:	14:00	Hora de encerramento:	16:30
Matrícula do Discente:	11813EEL002				
Nome do Discente:	Guilherme Fernandes de Souza Miguel				
Título do Trabalho:	Proposal of a game streaming-based framework for a telerehabilitation system				
Área de concentração:	Processamento da informação				
Linha de pesquisa:	Engenharia biomédica				
Projeto de Pesquisa de vinculação:	Título: RehabNet: Rede de Telereabilitação por meio de Realidade Virtual e Aumentada . Agência Financiadora: Conselho Nacional de Desenvolvimento Científico e Tecnológico. Início: 01/03/21 . Término 28/02/24. No. do Projeto na agência: 307754/2020-0. Professor Coordenador: Eduardo Lazaro Martins Naves				

Reuniu-se por meio de videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Engenharia Elétrica, assim composta: Professores Doutores: Angela Abreu Rosa de Sá - PNPD/UFU; Pedro Felipe do Prado - UFES; Samuel Baraldi Mafra - INATEL; Eduardo Lázaro Martins Naves - FEELT/UFU, orientador do candidato.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Eduardo Lázaro Martins Naves, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Pedro Felipe do Prado**, **Usuário Externo**, em 04/08/2021, às 16:20, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Angela Abreu Rosa de Sá**, **Usuário Externo**, em 04/08/2021, às 16:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Samuel Baraldi Mafra**, **Usuário Externo**, em 04/08/2021, às 16:22, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Eduardo Lazaro Martins Naves**, **Professor(a) do Magistério Superior**, em 04/08/2021, às 16:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2884611** e o código CRC **C4AA3F30**.

Referência: Processo nº 23117.043861/2021-11 SEI nº 2884611

Abstract

Nowadays, electronic games are used in areas other than entertainment, referenced in the literature as Serious Games when used in educational activities, professional training, and exergames when used in training to improve physical fitness or medical rehabilitation therapies. Through exergames, rehabilitation therapies can be developed in clinics or hospitals and at the patient's home, improving its engagement in therapy since, because they are computer games, they are understood as fun and not as a repetitive exercise. However, despite the advantages of using exergames, they introduce costs to the patient, such as the device where the game is played and the device used to control it. Another disadvantage is that the therapist does not follow up on the session unless the patient travels to the clinic or Hospital or the therapist travels to the patient's home. This work proposes the use of various new technologies to solve the different issues found in the literature, which are: build an input device via video object tracking and use it to control the exergame, which is executed on a remote game server and, by using a game streaming application, its rendered frames streamed to the patient over the Internet. Additionally, a real-time audio and video communication channel between the patient and the therapist is implemented as part of this Framework, allowing them to interact during the telerehabilitation session without needing to be in the same physical space.

Keywords: Exergames. Game Streaming. Serious Games. Remote Rehabilitation. Real-time Streaming.

List of Figures

List of Tables

List of abbreviations and acronyms

PC	Power Computer
GPU	Graphics Processing Unit
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
LIDAR	Light Detection And Ranging
IoT	Internet of Things
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
RFC	Request for Comment
RTP	Real Time Protocol
API	Application Programming Interface
HTML	HyperText Markup Language
HTML5	HyperText Markup Language version 5
WHATWG	Web Hypertext Application Technology Working Group
JS	Java Script
W3C	World Wide Web Consortium
IETF	Internet Engineering Task Force
SDP	Session Description Protocol
QoE	Quality of Experience
WLAN	Wireless Local Area Network
OpenCV	Open Source Computer Visual Library
HSV	Hue, Saturation, Value

RGB	Red, Green, Blue
SDK	Software Development Kit

Table of Contents

1 INTRODUCTION

A form of interaction in the human species is called a game, been defined by the Merriam-Webster dictionary [1] as: "A physical or mental competition conducted according to rules with the participants in direct opposition to each other." Computers are also used as a platform for playing games, be this kind of game denominated by the literature as a computer, video, or electronic game [2]. In a computer game, the player can interact with other players been this other player(s) part of the game logic or other humans locally (on the same computer or console) or remotely (over the Internet) [3–5].

Wiemeyer[6] used the term "Serious Games" as a definition of using games in a wide range of areas, including: education, military, health care, and disability rehabilitation; But the term Exergames is most commonly used as a definition of games usage for physical training and rehabilitation therapies. In rehabilitation therapy, exergames are used to increase the patient's motivation and engagement since games have: a story, tasks to be completed with multiple levels of difficulty, objectives, and type of movement; All this exergames parameters can be adjusted for each patient to met the treatment specific goals and allows the rehabilitation sessions to be executed on various places, i.e., the patient's home, clinic or Hospital [7]. In these types of games, the patient replicates movements that are presented in the games, which is based on its rehabilitation exercises requirements, and receive feedback information that is also available to the therapist about the movement that has been performed, its success rate, and areas which still needs to be improved [8]. Exergame also assists the therapist, facilitating the follow-up on the patient's evolution [9–11].

When used for remote therapy, Exergames can be considered Telerehabilitation or telehealth technology. Telehealth technologies have a fundamental role in medical filed since it enables the diagnostics, monitoring and rehabilitation process to be executed remotely between a medical professional and its patient [7,12]. According to Mantovani *et al.*; Burke *et al.*; Burke *et al.*[13–15]the use of exergames has proven to be effective when used in Telerehabilitation since it enables this process to start as early as possible and increase the patient's motivation and engagement in the therapy. By the use of Telerehabilitation, it is not necessary for the patient to be in a hospital or clinic environment, which implies cost reduction for both: patient and Hospital or clinic; and decreases the chance of infection by the patient of other diseases. This requirement is mandatory in a case of a pandemic, i.e., COVID-19, due to the collapse of most of the health systems worldwide, which makes those who do not have the option of using a telerehabilitation to interrupt its treatment while a vaccine or an effective medical treatment is discovered [16].

Until now, the advantages of using exergames as a telehealth technology was presented, but there are some issues with this technology which needs to be addressed, such as: To be able to execute an exergame at home, the patient will need to buy a gaming Personal Computer (PC) or a gaming console; There are numerous possibilities of input devices that can be used, each one having its price range; and the impossibility for the therapist to participate on the therapy session, without requiring the patient to go to a hospital/clinic or the therapist to go to the patient residence, which also implies in time and cost reduction for the patient and therapist.

Having introduced the concepts of telehealth and exergames, presenting its main advantages and disadvantages, that will be described in more detail in the Theory fundamentals session. The next session will make an overview of these telehealth technologies and the motivation of this work. Later on, the general objective and specific objectives will be presented, followed by a capture in which a Literature review is presented, followed by a Theory fundamentals chapter constituted by a breath explanation of the technologies used in this work. Then, the methodology is presented where all the steps to make the solution proposed in this thesis are shown, followed by the Results presentation and discussion of these results. Finally, the conclusion of this work is presented with suggestions for further developments.

1.1 Overview and Motivation

There are already several Telehealth technologies proposed in the literature that use exergames to improve the patient's motivation and engagement in therapy sessions. To be used as a telehealth technology, an exergame needs to have an input device to control the game and can vary depending on the patient needs, and it needs to receive specific instructions about the tasks that need to be executed by the patient on the rehabilitation session and send feedback information to a server that will process and store it, making it available for further analysis by the patient or the therapist. Various exergames frameworks were found in the literature, been classified as Client-Server, Web-Based, Layers Based, Cloud Base, and Multi-user.

Despite the Framework used for the exergame implementation, they rely on the concept that there are two computers that interact with each other, called Client and Server. The Client computer (installed in the patient's home) is responsible for: installing (if required), processing, running the game, collecting patient data, and send it to the Server; while the Server computer that is responsible for storing the patient's rehabilitation program data and information, having as the main focus, the management and reporting regarding patient performance.

However, although there are already several systems for Telerehabilitation, there are challenges in the use of these rehabilitation games at home regarding the costs of the hardware that is used to run the game and the input device that is used to control the game; and also the form in which the therapist interacts with the patient.

1.1.1 Motivation

The home-based Telerehabilitation, in addition to facilitating for the therapist, as he/she does not have to move to where the patient is, also facilitates and includes the patient. Many patients would not be able to attend the rehabilitation clinic as often as necessary to provide treatment. Also, in periods of the pandemic, when people are isolated at home, home-based Telerehabilitation has greater prominence. The areas of application of rehabilitation using remote therapy are the most diverse (stroke, cognitive, upper-limb, chronic neck pain), and most systems have the possibility to expand the type of therapy. Thus, a primary challenge for home-based Telerehabilitation is to reduce the expenses for the patient.

In this sense, we would be contributing to the popularization of this type of therapy as we would be developing systems that are accessible for all patients. Furthermore, the key solution to address this challenge is to take advantage of all the evolution of remote therapy frameworks and use scientific knowledge to reduce the need for high processing power in the patient's computer. Moreover, take advantage of the computer's native input devices (Webcam, Mouse, keyboard, etc.) to collect the data needed for the patient's rehabilitation. Thus, we would have a chance to reduce the patient's expenses with the purchase of new technological resources. In this context, with the improvement of telerehabilitation systems, which are mainly composed of well-designed exergames to improve patient engagement, it is necessary that the patient has a computer suitable for installing and running the exergame. Furthermore, it is not always possible for the patient to buy or get government assistance to buy a good computer and the necessary devices to run the exergame.

An issue that needs to be considered is that, with the increasing evolution of game engines that offers the game developer the possibility to use, consequently, needs an excellent graphical interface to run it, it is necessary that the computer that will run this software has an excellent configuration. Otherwise, it will not be possible to install the system, or the exergame will run slowly because the computer will not have the necessary processing resources. Moreover, we have to remember that the exergame will be installed and run on the patient's computer. So, if the patient does not have enough money to buy a good computer, he/she will not be able to perform a home-based rehabilitation.

For exergames that require patient motion detection, a possible solution is to use the computer's Webcam as input to capture the patient's image and implement image processing algorithms to track the motion. According to the studies of [17, 18], it has been shown that it is possible to implement body movement tracking techniques. Also, the work of [19, 20] suggests that it would be possible to use the computer's Webcam to track the gaze. However, in this context, processing the algorithms to use the Webcam as an input device for the exergame would require a high cost to the patient's computer. However, it would be possible to use, for example, the Cloud-based or Layer based frameworks to process the webcam image on the server computer. Thus, the patient's machine would not need to be high-performance.

Moreover, on the issue of the high processing power required to run the exergame, a possible solution could be a Layer framework based on streaming technique [21]. The exergame would be installed on the server computer, and the high processing would be in charge of it, and the patient's computer would only receive the necessary images of the game [22]. This solution could make it easier for many patients to have a telerehabilitation system at home, as it would not be necessary to spend much money since the computer would not need to have such high processing power.

1.2 Objectives

One of the possible solutions to these issues is to consume as little as possible the processing of the patient's computer - so the machine would not need to have such an advanced configuration. The use of the Webcam for tracking the patient's movements (body and eyes) and the execution of the exergame via streaming may be a solution worth investigating. Also, the combination of the remote therapy frameworks presented here in this review may be a new way to assist in solving these issues. These improvements may help to increase the number of patients who have access to home-based Telerehabilitation.

1.2.1 General objective

This works to propose a novel approach to home-based Telerehabilitation, using game streaming technology with real-time communication protocols with the objective to provide both patient and therapist a platform in which they can interact as if they were in the same physical space.

1.2.2 Specific objectives

To make possible the telerehabilitation process to be fully remote executed, a number of technologies should be implemented on this framework:

1. A real-time video stream from the patient to the therapist so he can be visually evaluated and motivated in real-time by the therapist.
2. A real-time video stream from the therapist to the patient, enabling the patient to get and image feedback from the therapist, validating the execution of the task.
3. A real-time data and video stream from the patient's device to the computer that hosts the game (that can be in the clinic or hospital environment or in a cloud), enabling the patient to control the game and his video to be recorded for further analysis if necessary.
4. A low-cost input device that is suitable for various therapies using the image recognition techniques from the image acquired from the patient's device.
5. A very low latency real-time video stream from the Server that hosts the game to the patient's device, with the images generated by the game engine as if the game was running on his device.
6. Real-time control of the machine that hosts the game from the therapist, allowing the therapist to set multiple aspects of the therapy into the game.

1.2.3 Thesys Contributions

All the innovative approaches implemented on this Framework allows:

- The therapy is to be conducted in a fully remote manner resulting in cost reduction. Reduced costs for both patient and therapist by not requiring them to move to the same physical location.
- The therapist to attend multiple rehabilitation sessions at the same time.
- Using various devices to be used on the remote therapy, not requiring any software installation on it.
- Using various input devices, including the Object Tracking that was also implemented on this Framework.
- A real-time audio and video communication channel between the patient and the therapist for feedback.

1.2.4 Thesys structure

Been this thesis theme introduced, as well as the general and specific objectives presented, followed by the contributions of this work, on the next chapter, a literature review will be made, followed by a chapter in which the theory and fundamentals are presented. Then, the methods that were used to implement this Framework will be explained, followed by the test results of the implementation and the discussion of these tests. Finally, the conclusions obtained by the implementation and analysis of this work are presented in the last chapter.

2 Literature Review

In this literature review, the criteria for Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) were adopted [23], been this protocol, which outlines methods for the proposed systematic review, designed in accordance with PRISMA guidelines. The entire study selection process was carried out by four reviewers and summarized in a PRISMA flow diagram shown in Figure 1.

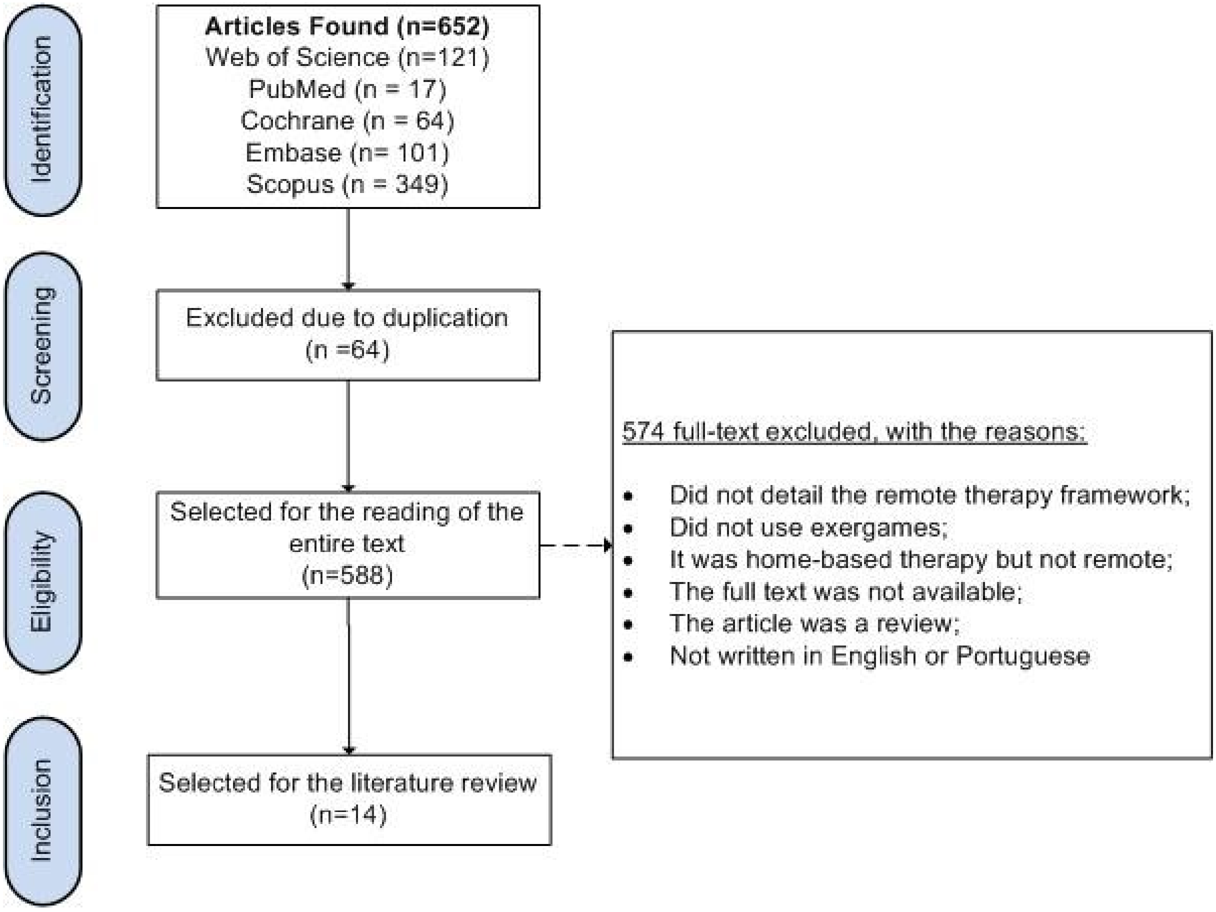


Figure 1 – PRISMA method used for the selection of articles for this review
Source: Authors(2021).

By using this criterion, eligible articles accounted for studies on remote therapy using exergames were selected, excluding studies that only used remote therapies but did not detail the Framework. This review included English and Portuguese language, been placed up to April 2021 on five databases: Web of Science, Pubmed, Cochrane, Embase and

Scopus, using the following queries: (("game*" OR "exergame*") AND ("rehabilitation") AND ("remote" OR "streaming" OR "telerehabilitation" OR "telemedicine")). Eligible studies were selected after reading the full text to compose this review following the PRISMA flow diagram selection process.

The search in the databases resulted in a total of 652 articles. Duplicated were disregarded, and 588 articles remained. After reading the pre-selected studies, 574 were excluded after reading the full text due to the following reasons: 1) Did not detail the remote therapy framework; 2) Did not use exergames; 3) It was home-based therapy but not remote; 4) The full text was not available; 5) The article was a review; 6) Not written in English or Portuguese. Therefore, the sample for this review consisted of 14 studies, as shown in Figure 1.

The 14 studies selected for this review were published between 2000 and 2021. According to Table 1, we have identified five different remote therapy frameworks in the selected studies: Client-Server, Web-Based, Layers, Cloud-Based, and Multi-users. This Framework will be discussed in more detail in session 3.

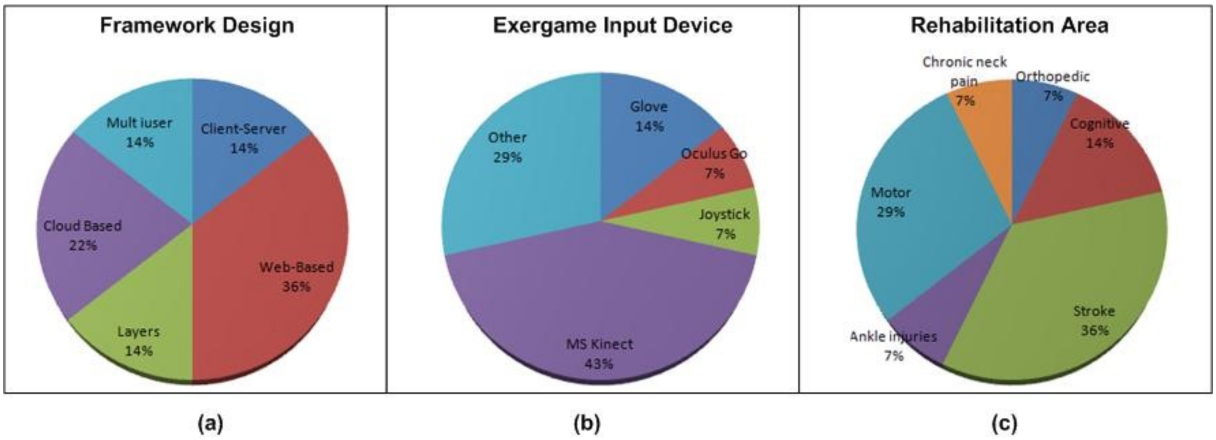


Figure 2 – Main features of the studies selected in this review: a) Types of remote therapy framework design; b) Models of exergame input devices; c) Rehabilitation area of remote therapy.
Source: Authors(2021).

Figure 1 presents an overview of the classification of the 14 studies in this review. Among the framework designs of the selected studies, 36% of them are classified as Web-based (Figure 2.a). In addition, all the studies rely on an external input device to control the exergame, highlighting that 43% of them use MS Kinect (Figure 2.b). Also, the rehabilitation area of the remote therapies of these studies are diverse, but most (36%) are applicable in the area of stroke rehabilitation (Figure 2.c).

The 14 studies selected are summarized on the table 1 below, by the framework design, rehabilitation area, the input device used and the highlights of this study:

Table 1 – Main characteristics of the articles selected for this review.

Framework Design	**Study**	**Rehabilitation Area**	**Input Device**	**Highlights**
Client-Sever	[24]	Orthopedic	Glove	In addition to recording the patient's progress the server side changes the patient's difficulty level.
Client-Server	[25]	Cognitive	Oculus Go	Uses an app with a immersive virtual reality application.
Web-Based	[26]	Stroke	Joystick	New versions are automatically installed each time a user accesses the therapy web page.
Web-Based	[27]	Ankle injuries	Wobble board	The system is simple and easy to be used by people of all ages.
Web-Based	[28]	Motor Rehabilitation	MS Kinect	Real time evaluation of the patient session.
Web-Based	[22]	Stroke	Inertial sensors	It allows setting the patient expected motor skill at any moment.
Web-based	[29]	Stroke	Glove	Mobile applications and IoT layers.
Layers	[21]	Stroke	MS Kinect	Layers: approach, technology and implementation.
Layers	[30]	Motor Rehabilitation	MS Kinect	Layers: input, game and clinician.
Cloud Based	[31]	Motor Rehabilitation	Robotic Arm	Cloud robot assisted rhabilitation system with multimodal interaction.
Cloud Based	[20]	Chronic neck pain	MS Kinect	Back-end (analytics and game zones) and front-end (patient and therapist dashboards)
Cloud Based	[19]	Cognitive	MS Kinect	It uses an Inference Subsystem (artificial intelligence)
Multi user	[32]	Stroke	MS Kinect	A central server interacting with peripheral client computers.
Multi user	[8]	Motor rehabilitation	Orthosis wrist	Multiple patients are integrating robots and games.

All the articles presented on this table, as a fundamental principle of using exergames on telehealth technologies, despite its framework design (as shown in the first column), relies on the communication between two computers: a computer that will be installed in the patient's home (Client computer) and another computer, called Server, that will be located in: a hospital or rehabilitation center(clinic) network data center, or in a server farm (cloud infrastructure); depending on the Framework used. In the second column, there is the reference for each article. In the third column, the rehabilitation area that the presented, been classified as Orthopedic, Stroke, Ankle Injuries, Chronic neck pain, Cognitive or Motor rehabilitation. The fourth column shows the input device that is used to control the exergame, been identified as Glove, Oculus Go, Joystick, Wobble board, MS Kinect, Inertial sensors, Robotic Arm, and Orthosis wrist. In the fifth column, a highlight of the article is presented.

From this table, it can be identified that there are multiple Frameworks that can be used by exergames developers. The Client-Server Framework, used on the exergame developed by Popescu *et al.*[24] and Varela-Aldás *et al.*[25], it is a strategy that relies on the communication between two entities, the client computer and the Server. On the client's computer, the exergame that the patient will use to perform the rehabilitation exercises will be installed. The client computer will send the patient's rehabilitation session data to the Server computer, which will be accessible by a therapist to assess the patient's evolution, and receives instructions from the Server to change the game difficulty or tasks to be executed. The communication between the Client and Server computer can be limited to a few packages, which is essential in cases of limited bandwidth, and since the game is installed on the client computer, all kinds of input devices can be used to control the game. A disadvantage of this strategy is that the Client computer hardware needs to support the game since it will be physically installed on that computer.

The Web-Based framework, used on the exergame developed by Reinkensmeyer *et al.*[26], Karime *et al.*[27], Pérez-Medina *et al.*[28], Zedda *et al.*[22] and Postolache *et al.*[29], it is also Client-Server based, but different from the traditional Client-Server Framework, the patient access a Web page that is hosted in a Web Server that is located on the hospital or clinic data center, the game is transferred to his computer and is executed on the Web browser. As in the traditional Client-Server Framework, the session data will be sent from the Web Browser that executes the exergame to the Web Server, which will be accessible by a therapist to assess the patient's evolution. The Web Server is used to control the game difficulty or tasks to be executed. The main advantage of using this strategy is that the exergame runs in different computer architectures and operating systems, but it imposes some limitation to the game developer since the browser environment it is not suitable for complex graphics games, and it supports a limited input devices types.

A Cloud Based Framework, used on the exergame developed by Li e Song[31], Afyouni, Murad e Einea[20], and Oliver *et al.*[19], it is a Web-Based Framework that, instead of having its Server physically located on the Hospital or clinic data center, uses a cloud environment to host its Web Server. Due to advances in virtualization technology, large corporations and Internet Service Providers has offered the possibility of renting an on-demand server with different configurations located in its server farms, so in this case, the clinic or Hospital does not need to have its own server infrastructure with all the costs involved, been this service it is known as Cloud-Based infrastructure. The same advantages and limitations of the Client-Server or Web-Based Framework are imposed to the exergames implemented with the Cloud-Based Framework, but with a reduced cost for the clinic or Hospital.

Nowadays, most of the games developed for consoles, PCs, or mobile devices can offer the player the possibility to interact with other players over the Internet, been it known as a Multi-User Framework, used on the exergame developed by Triandafilou et al.[32] and Andrade et al.[8]. This is also a possibility when using exergames, where patients can interact with each other and with the therapist in a virtual environment, making the rehabilitation more efficient in terms of engagement and motivation. A Multi-User game (and also in the exergames) imposes some difficulties for the implementation of the game itself, requires a server infrastructure to orchestrate the different player's positions and interactions, and requires a low latency connection between players and the orchestration server. Also, the same advantages and limitations of the Client-Server, Web-Based or Cloud-Based Framework are also imposed to the exergames implemented with the Multi-User Framework.

After analyzing all the studies in table 1, it can also be noticed that there is a natural evolution of these exergames, and, in some cases, a combination of framework designs can be adopted. As we have already detailed above, each of these frameworks has its advantages and disadvantages. However, the best approach should be patient-focused and not only a cost reduction of implementation difficulty. According to Table 1, all the studies presented use some kind of device to be the input control of the exergame, and, in some cases, even more, than one device is required. Indeed, all exergames need an input device to control them, but it is essential to highlight that, depending on the input device required, it may not be affordable for purchase by all patients.

Analyzing the articles presented on the Table 1 and Figure 2 by input device, most of this articles: Pérez-Medina *et al.*; Vourvopoulos *et al.*; Paraskevopoulos *et al.*; Afyouni, Murad e Einea; Oliver *et al.*; Triandafilou *et al.*; Andrade *et al.*[8, 19–21, 28, 30, 32]; uses in their solution the MS Kinect since it has a built-in camera and a Light Detection And Ranging (LIDAR), that is used for depth perception since the extraction of this information from a image is computational difficult and imprecise [33].

The Glove device, used in the work of Popescu *et al.*[24] and Postolache *et al.*[29]; that aims at detecting the movements of a hand, its fingers and joints and translate it into actions on the game [34], using the variations of the electrical resistance on a conductor to measure these fingers' displacements. The Orthosis wrist, used on the work of Andrade *et al.*[8], uses the same principle as the Glove, but it is used to captures the wrist movement and can be used in conjunction with a Glove device.

The other input methods found on the literature are: Oculus Go (used on Varela-Aldás *et al.*[25]), Joystick (Reinkensmeyer *et al.*[26]), Wobble board (Karime *et al.*[27]), Robotic Arm (Li e Song[31]). The Oculus Go is a device used to simulate a virtual 3D environment that is composed of: a screen that displays a stereo image, an inertial sensor, and a pair of joysticks, each one containing a button and an inertial sensor. A joystick is an input device consisting of: a stick that pivots on a base and reports its angle or direction to the device it is controlling and buttons (on/off and proportional); been used to control a game or as a replacement of a keyboard and Mouse for people with physical disabilities [35]. The wobble board is a musical instrument invented and popularized by the Australian musician and artist Rolf Harris and is played by holding the board lengthwise with the hands holding the edges and flicking the board outward, thereby making a characteristic "whoop-whoop" noise [36]. A Robot Arm is a device that has actuators and sensors that feedback its displacement and is used to grab and move objects with precision, but it can also be used as an input device, using its sensors to capture the displacement of its arms executed by a human.

After analyzing the methods used by this literature review, its differences in terms of Framework and input device used, it is possible to verify that in most of these works the exergame, that will be used to perform the rehabilitation exercises will run, and most of the times are also installed on its computer. The client computer will send the patient's rehabilitation session data to the Server computer, which will be accessible by a therapist to assess the patient's evolution [24], , been this communication via the Internet. This client computer will record the performance data and send it to the server computer in order to provide the therapist with the necessary information to evaluate the patient's progress [29]. In addition, the therapist can enter new settings for the patient's upcoming rehabilitation sessions and send them to the exergame that is installed on the client's computer. This implies that the quality of network services is essential for the performance of the game and this remote therapy systems [37]. Also, there are limitations regarding the cost and ease to use of input devices, and there is no article in the literature that aims to present a solution that enables the therapist to interact with the patient in real-time.

In the next session, some theory fundamentals about this work will be presented, followed by the methodology used to make this possible, the results obtained on the preliminary tests and its discussions, followed by conclusions of this work.

3 Theory and Fundamentals

This chapter will present the theory that was used to make possible the implementation of this work. Before presenting the specific protocols and technology used, it is essential to differentiate the different Frameworks used to develop exergames, that are: Client-Server, Web-Based, Layers Based, Cloud-Based, and Multiuser. Client-Server Framework

3.1 Client-Server Framework

This model of telerehabilitation system is the classic Client-Server: the exergame is installed on the computer (Client) of the patient's home, and software to store the patient session data is installed on another machine (Server). These two machines communicate via a network protocol. The primary application of this model has improved, and the type of communication between client and server computers is evolving and allowing new interactions with the user. In this framework design, the work of [37] was one of the pioneers in introducing the concept of a novel multipurpose haptic control interface. A force feedback glove for orthopedic rehabilitation is connected to the patient's computer. This Glove is an exoskeletal structure that provides forces on the patient's fingertips and contains position sensors to control an object within the exergame. Also, while wearing the rehabilitation gloves, the patient controls the system using voice commands. Data collected during the rehabilitation exercises are stored remotely at the server machine using the Internet. Also, remote consultation is allowed using a videoconferencing system. In addition, the study of [27] improved the client-server Framework for a mobile application with an immersive virtual reality application. The exergame is controlled by the Oculus Go [38] which contains the inertial measurement unit to track 3 degrees of freedom of angular movement. The significant differential of these two works is the possibility of inserting more than one input device for the user to control the exergame, using the basic Client-Server Framework.

3.2 Web Based Framework

The Web based framework has as main characteristic the use of Web services to enable telerehabilitation. The study of [39] initiated this model by inserting a game therapy: users enter the system using the web, carry out a personalized program of therapy activities and receive information on their rehabilitation progress. The system uses a joystick to control the exergames. Also, the remote therapy of [31] makes use of internet connectivity to provide a link between the patient and the therapist. The exergame provides an ankle muscle strengthening exercise using the rotation information and pressure information from the wireless wobble board as input. Furthermore, the work of [40] brought the concepts of real-time evaluation, flexible and modular software architecture to the Web-Based Framework.

Moreover, in the study of [13] the exergame is implemented in Unity 3D and is supported by a web application accessible from anywhere by medical staff and patients, allowing constant monitoring of rehabilitation progress. Furthermore, the work of [41] presents an inovation on Web Based model: The use of Internet of Things (IoT). IoT Systems focuses on data analysis of data extracted during the training session, where patients are using wearable sensing devices to interact with the exergame [26, 42, 43]. IoT-based systems are a new model of communication between humans and things: we can have connectivity to everything, anywhere, and at any time [44, 45]. The goal of IoT systems is to establish synergy between different systems and make them communicate automatically to provide important services to users [26].

3.3 Layers Based Framework

Another framework design approach for exergames in Telerehabilitation is the concept of Layers. In this model, the telerehabilitation system is completely separated into layers: a layer for management of hardware (input devices), a layer to perform the exergame, and a layer for therapy management. The work of [46] presents an architecture consisting of three main layers: hardware for device support, control panel for translation and data emulation, and Web Content for access to the rehabilitation tools. All layers are interconnected in a client-server architecture. Also, the research of [47] implemented the novelty about holistic nature: it includes the technology-agnostic approach, as it addresses the input controllers to a separate layer. This concept allows the total customization of the rehabilitation therapy to a higher level of fidelity and granularity since the therapist is able to define all the parameters of each repetition of the exercise that the patient will perform with the exergame.

3.4 Cloud Based Framework

Following a new trend in technology, some cloud-based rehabilitation systems have already been proposed. Basically, these systems use cloud services to store the therapy database and the intelligent systems that control the exergames. The work of [48] proposed an adaptive and personalized exercise. The system contains 3 main components: frontend (runs the exergame), processes (data processing and storage) and backend (cloud storage: artificial intelligence modules and data analysis). In addition, data are continuously stored on the cloud to allow real-time sessions analytics. Moreover, the study of [49] presented a new architecture of a cloud robotic system for upper-limb rehabilitation with multimodal interaction. This system consists of a cloud central server, several therapists as the server and patients as clients that may be distributed with patients in anywhere (home, community, etc). All patients and therapists are connected to the central server through cloud. Each client computer includes a robotic arm attached and a input device to control it. Furthermore, the work of [48] consists of two subsystems: cognitive and inference. The cognitive subsystem is located on both the therapist's and patient's computer. The data collected in each rehabilitation session is sent to inference subsystem which is stored in the cloud.

3.5 Multiuser Framework

The multi-user approach, allows several patients and therapists to be connected simultaneously in the same rehabilitation session. In the work of [19], stroke patients can interact with therapists in the same virtual space. The system has a central server that interacts with the patient's computers, one for each user. The server receives information from the patient computers and controls the updating of the exergame scene so that the appropriate view of the scene is shown on each patient computer. In addition, the study of [30] proposes a framework for multi-patient rehabilitation integrating robots and exergames. The architecture supports multiple rehabilitation robots that send and receive messages simultaneously. The exergame receives position and speed data from the robot and the robot receives the data representing the movement that must be performed to assist in the patient's rehabilitation.

Been this Frameworks presented, what are the protocols used for communication of the exergames over the internet? The answer for this question, Transport Layer Protocols, are introduced bellow followed by real-time communications, there are some other basic aspects to this work implementation.

3.6 Transport Layer Protocols

The most widely used Transport Layer Protocols are Transmission Control Protocol (TCP), and the User Datagram Protocol (UDP) specified in RFC7688 [50] and RFC793 [51], respectively. Transport layer protocols are responsible for creating virtual connections between processes hosted on different end systems, making it possible to exchange messages between these processes.

TCP is a connection-oriented transport protocol designed to be part of a protocol layer architecture that provides reliable and orderly communication between pairs of processes that run on hosts of different data communication networks interconnected. It can operate across a broad spectrum of communication systems, from hosts directly connected to circuit switching or packet switching networks. It also provides congestion control, allowing TCP connections to travel through a congested network link share the link equally [52].

UDP is a simple transport protocol geared to transactions that it only provides delivery of data between processes that run at different final destinations, not guaranteeing delivery, order, or duplication of messages. The lack of congestion control algorithms in UDP causes injustice in the load balancing of the various network flows. While TCP reduces the rate of sending packets to minimize network congestion, the lack of knowledge of the network status by UDP causes it to continue sending packets [53]. UDP is generally used in real-time applications in conjunction with the Real-Time Transport Protocol (RTP) specified by the RFC1889 [54].

3.7 Real-Time communication

When communicating via the Internet frequently, the message exchange between the two entities must occur in two phases: The two parties willing to communicate must send messages to each other through some mechanism, specifying what functionality they are both able to support. With these messages, they can start the communication, given that there are often many choices that can be made for communicative functionality, or, failing to find anything in common, give up on communication.

Real-Time communication is a type of communication where the generation and display of content is intended to occur closely together in time (on the order of no more than hundreds of milliseconds). Real-time media can be used to support interactive communication. A communication is classified as interactive when in communication between multiple parties, where the expectation is that an action from one party can cause a reaction by another party, and the reaction can be observed by the first party, where the total time required for the action/reaction/observation is on the order of no more than hundreds of milliseconds [55].

The UDP is used in real-time communication, in conjunction with the RTP since it implements services such as identification payload, sequence number, timestamping, and package delivery monitoring system, but does not have any flow or congestion control; that is important for both endpoints in a real-time communication [54]. Real-Time communication is used in applications like audio and video conferences, remote desktop control, game streaming, multi-player gaming, video streaming, chat apps, etc. In order to create a real-time application, the developer makes use of an Application Programming Interface (API), which is a specification of a set of calls and events, usually tied to a programming language or an abstract formal specification such as WebIDL, with its defined semantics. The APIs used in this work will be presented in the next session.

3.7.1 Real-Time communication API

Modern Web Browsers supports the HyperText Markup Language version 5, which is standardized and tested by the Web Hypertext Application Technology Working Group (WHATWG), and together with JavaScript (JS), which is an interpreted and object-based language, makes possible the creation of dynamic web pages by using multiple libraries and APIs to control its behavior. JS is used for most of the dynamic Web Pages that are available today due to its flexibility and the number of libraries and frameworks available for this programming language. For the implementation of this work, game streaming, audio, and video conference APIs were used to develop a JS application that matches the requirements specified on the specific objects (session 1.2.2)

3.7.2 audio and video conference

A variety of technology companies aiming to provide real-time video audio and data communication between modern web browsers and applications for different operational systems, such as Android, iOS, Windows, Etc.; developed its applications based on JS Application Programming Interface (API) specification, standardized by the World Wide Web Consortium (W3C) Web Real-Time Communications, Web Applications Security, and devices and Sensors Working Groups.

For browser-based applications, the model for real-time support does not assume that the browser will contain all the functions needed for an application, such as a telephone or a video conference. Instead, the browser will have the functions needed for a web application, working in conjunction with its backend servers, to implement the other required functions. This means that two vital interfaces need specification: the protocols that browsers use to talk to each other without any intervening servers; and the APIs that are offered for a JS application to take advantage of the browser's functionality [[56].

The Internet Engineering Task Force (IETF) WebRTC working group was responsible for the proposition of the standard set of communication protocols used by WebRTC applications. As defined in this working group webpage: "The RTCWEB working group was originally chartered to standardize mechanisms that provide direct interactive rich communication using audio, video, collaboration, games, Etc.; between two peers web-browsers, without requiring non-standard extensions or proprietary plug-ins, along the most direct possible path between the participants.".

Via Request For Comment (RFC) documents, the goal of the webRTC protocol specification is to specify a set of protocols that, if all are allowing applications from different companies to communicate with each other using audio, video, and data sent along the most direct path between peers. Cooperation into the protocol and an API specification makes clear for developers of different applications that want to use WebRTC, which API calls to invoke in order to implement a specific application, option, or feature [56, 57].

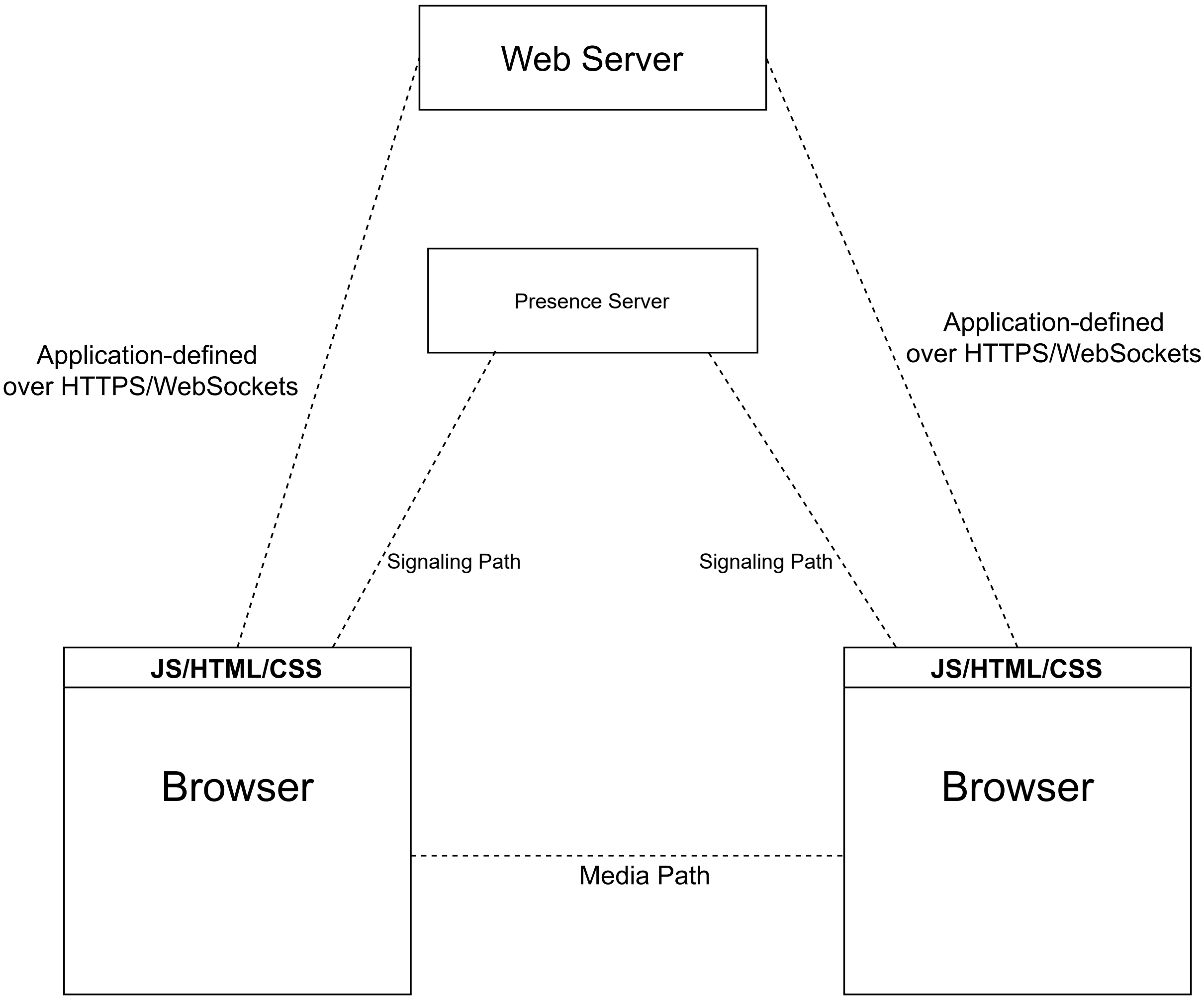


Figure 3 – Working principle diagram of WebRTC Source
Source: Authors(2021).

In Figure 3, it is noticeable that there are three main components: A Web Browser that interprets JS/HTML/CSS scripts, a Web Server that serves the Web browsers with the scripts and interacts with them with WebSockets, and the Presence Server. The Presence Server is the entity responsible for all the RTP media negotiation between all the entities, which is done using the Session Description Protocol(SDP). Once a communication channel is established between all the participants, the audio and image of them are presented to all users.

3.8 Game Streaming

Innovations of mobile devices and network interfaces enabled the development of real-time multimedia and communication services. Users can communicate and share data in real-time to a set of devices, including mobile phones, computers, tablets, smart TVs, cars, Etc. Having a dedicated gaming device, such as a console (fixed or mobile) or a Gaming Personal Computer (Tower or laptop), is expensive and not suitable for most user's daily use cases. Nowadays, tech companies with high computing power data-centers are using it is computing resources to offer users the possibility of running games that requires a high-performance Graphics Processing Unit (GPU) in their servers and have its video output streamed via a high-speed internet connection to the user's device, enabling a user to play high demanding GPU games on a low-cost device, paying only for the time that the game server was used.

In game streaming, a peer-to-peer connection is establish, one peer been the computer that runs the game (game server) and encodes, using a high-efficiency video encoder, the rendered game images and send it in a real-time low-latency video stream to the other peer (client), which decodes the received video and plays this video stream to the user. The client device sends control commands to the game server that receives it and emulates the device which sent the control commands to the game.

Game Streaming APIs are optimized audio and video real-time conference API that implements proprietary optimizations on the video and audio encoding/decoding in a one-way path (from the game server to the client) and emulates input devices like Mouse, Keyboard, and joysticks. It is important to clarify that a game, despite the high fidelity images rendered with modern game engines, is still highly compressible due to its characteristics like well-defined shadows, restricted brightness, Etc., which makes it easier for developers to optimize its game streaming APIs for such cases. A Web conference API, like WebRTC, needs to cover other cases, like the live video for real people and real scenarios, with sudden light changes e non-uniform shadows, Etc., which makes its encoders and decoders less optimal and, consequently, slower than the game streaming ones despite its similar characteristics.

The main issue when dealing with game streaming technologies, as for all real-time communication applications over the Internet, is the network latency and jitter. A high delay between the game server and the patient device decreases its Quality of Experience (QoE) due to a noticeable delay between the patient action and the related movement or action performed by the game avatar. Due to these characteristics, the delay between the game server and the patient device should be as low as possible. For this purpose, landline internet access is recommended, even when using a Wireless Local Arena Network (WLAN) to access the Internet over a mobile data internet connection (3G, 4G).

3.9 Object Tracking

Tracking objects based on color is one of the quickest and easiest methods for tracking an object from one image frame to the next. The speed of this technique makes it very attractive for real-time applications. This can be achieved by using an already existing set of image processing libraries like the Open Source Computer Visual Library (OpenCV).

OpenCV is a library with more than 2500 algorithms optimized to be used in computer vision and machine learning software. One of it many algorithms can be used for object identification and tracking, and this library can be used to develop applications in various programming languages, i.e., C++, Python, Java MATLAB, and JAVA Script; to be used by all major operating systems, i.e., Android, Microsoft Windows, and Linux; having also support for MMX and SSE instructions, for CPU processing optimization, as CUDA and OpenCL interfaces for GPU processing optimization.

Been the fundamentals presented in order to guide the authors to choose the best strategy when implementing the proposed Framework and taking into consideration the general and specific objectives of this work. In the next chapter, the methodology used to develop this Framework is presented.

4 RESEARCH DESIGN AND METHODS

In order to achieve the objectives proposed on the section 1.2, the framework architecture was designed, as presented in Figure 4, as composed by: Patient and Therapist Devices and the applications that each one executes, application server, Game Server and Internet. The lines between the boxes represents physical (solid) or logical (dashed) connections between the different entities. The patient and therapist devices run the: Game Streaming Client (which has a logical connection with both devices) and Web conference Client (that has two logical connections: one between both devices and another one with the Application Server); To control the exergame, the patient device also runs the Object detect application, that sends the movement instructions to the Game Server to control the character on the exergame.

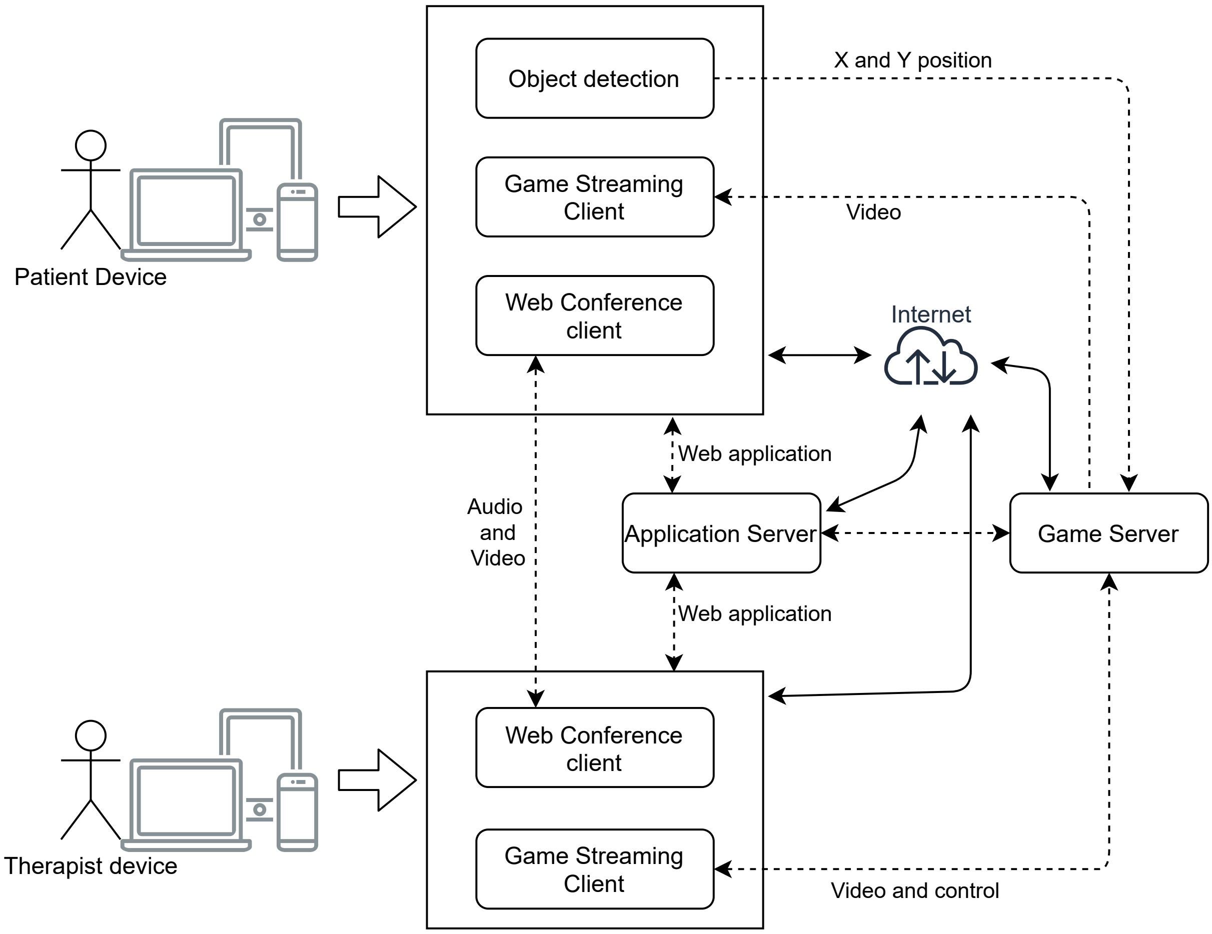


Figure 4 – Framework Architecture
Source: Authors(2021).

The implementation of this Framework was divided into the following steps:

1st: Object Tracking - Implementation of an Object tracking web application to be used as an input device on the exergame.

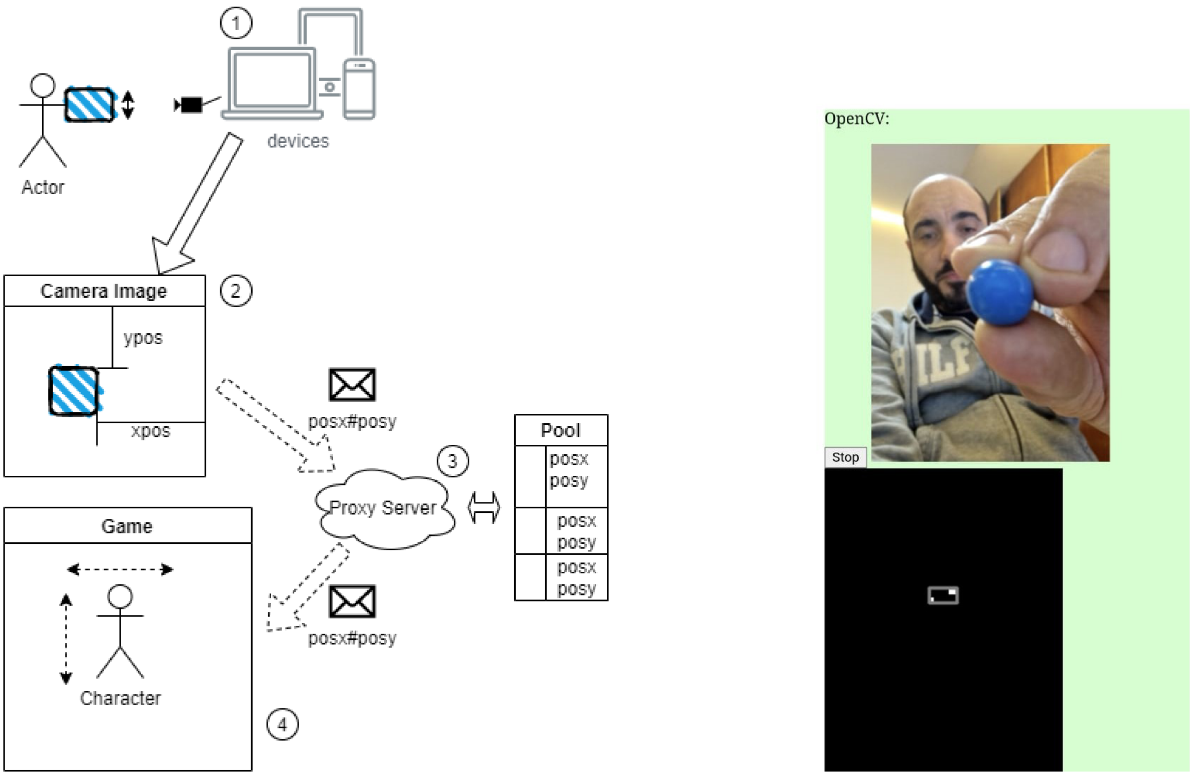
2nd: Game Streaming Client - Implement the game streaming client web application.

3rd: Web Conference Client - Implement the Web conference client for real-time audio and video communication between patient and therapist.

The implementation of these three steps will be presented in the subsequent sessions, followed by its validation tests and results from the discussion.

4.1 Object tracking

To implement the first step, the OpenCV Library was used in a JavaScript Web Application. This application captures the image acquired by the patient's device camera and, using a set of Algorithms, detects a specific color on this image and extract its position relative to the image border, as shown in Figure 5:



(a) (b)

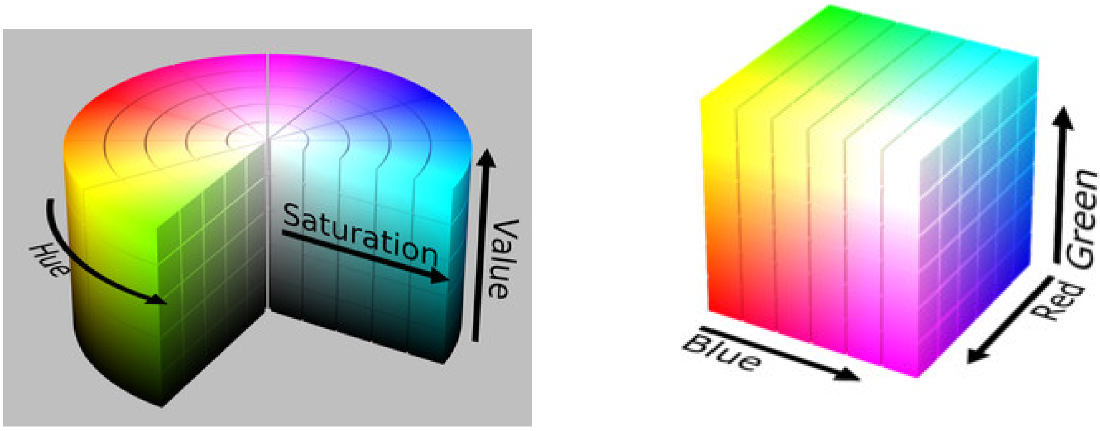
Figure 5 – Object tracking: a) Object detection schematics; b) Camera image and object detection mask;
Source: Authors(2021).

In Figure 5.a) the following steps are identified:

1. The image that was captured by the patient device
2. Calculating the position of the object relative to the image frame borders
3. This information is sent to a server which stores it into a queue
4. The queue items are sent to the game by the Proxy server

4.1.1 Detecting the object and Calculating its position (steps 1 and 2)

The webcam image is captured and stored in a matrix. This matrix has a row and column for every pixel of this image, represented in the: hue, saturation, value(HSV) colorspace, shown in Figure 6. a); The HSV model is similar to the: Red, Green, Blue (RGB) colorspace, shown in Figure 6.b); that is the most commonly used on image representation.



(a) (b)

Figure 6 – Colorspaces: a) HSV; b) RGB;
Source: OpenCV[58].

Each channel in the HSV colorspace represents:

- The Hue channel: Models the color type.
- The Saturation channel: Varies from unsaturated (shades of gray) to fully saturated (no white component).
- The Value channel: Describes the brightness or the intensity of the color.

The range of HSV in which the object color is located is used as arguments for the inRange() function of the OpenCV library, which searches it on the image, and marks the located points on an output matrix. The output of the inRange() function noise is removed using the morphologyEx(), and then the position is extracted using the boundingRect() function that created a bounding box with all the located points. The horizontal and vertical distance between the center of the bounding box and the image frame borders are used as the location of the Object.

4.1.2 Sending the location of the object to the game server via proxy

Before discussing why the Proxy server is needed, it is necessary to present the method by which the game acquired this input data. In Figure 7, it can be noticed the presence of two boxes that identifies two events (with its labels marked in red): Event

Begin Play and Event tick. The Begin Play event is activated when the game is initiated, and the event tick is generated on each frame call.

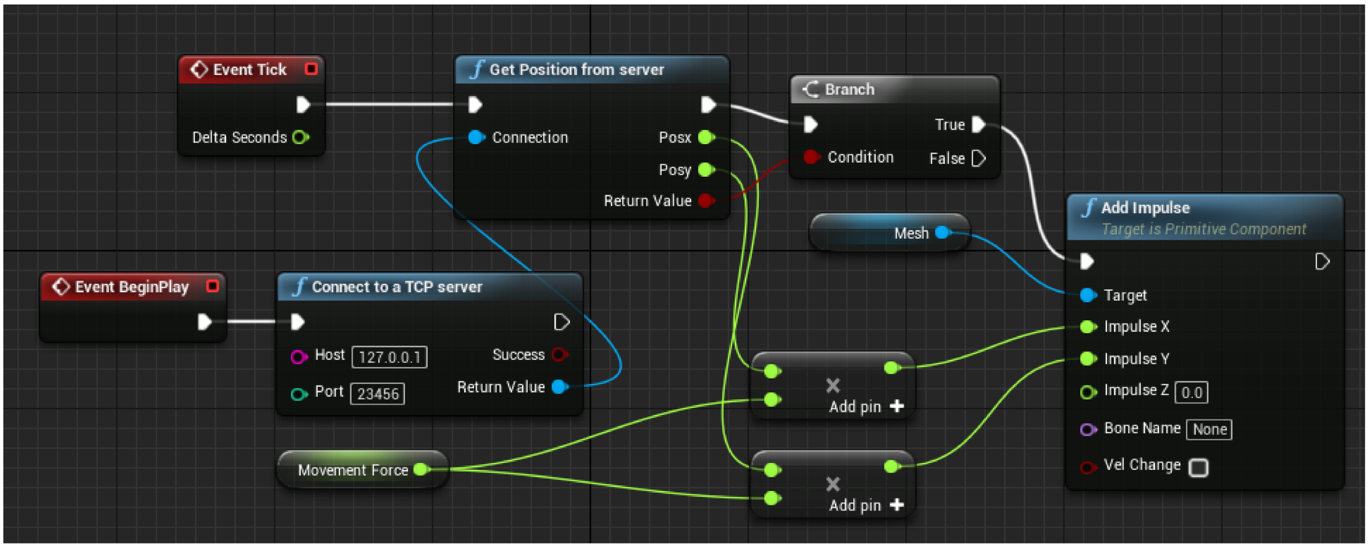


Figure 7 – The blueprint socket implementation
Source: Authors(2021).

When a BeginPlay event is generated, it calls the Connect to a TCP server function. The connection to a TCP server function establishes a TCP communication channel with the Proxy server (which in this image is located on the IP address 127.0.0.1) that listens for connections on port 23456. When established, this connection is provided as an input of the Get Position from the server function. The proxy server is used since it has a valid internet IP address, different from the game server or client that can be located in a Local Area Network (with an invalid IP Address) with a router that implements Network Address Translation, which means that it cannot be accessed directly from the Internet.

The proxy server is implemented using two sockets, one listening on the port12345 (input data), and another one was listening on the 23456 port (output data). When a connection from the patient device is established via input data port, every data point received is stored in a queue that is shared with the socket that listens for game connections on the output data port. When there is data on the queue and the game is connected to the output data socket, this data point is removed from the queue and sent to the game. A timer is used to verify, at every second, if there are points on the queue and there is no connection from the game. If this is the case, the queue is cleared, avoiding the game to receive old commands from the patient device.

On the game implementation, the Get Position from the server function checks the socket buffer for messages, and if no message is received, it returns false. If a message is received, the return value is set to true, and the Post and Posy are set to the values acquired from the message. These points are then used to move the character via the Add Impulse function by multiplying both of them by the movement force.

Figure 8 shows the tests performed to validate the use of the input device to control the game.

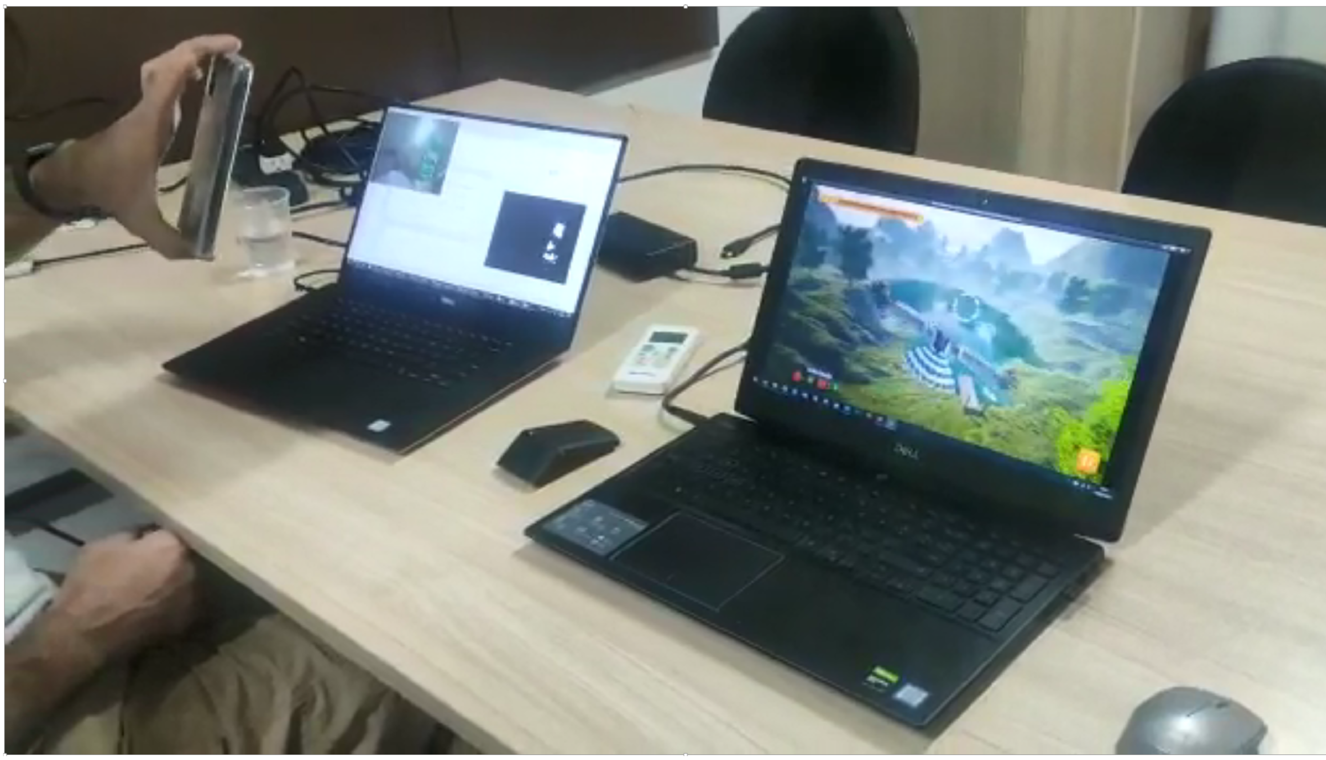
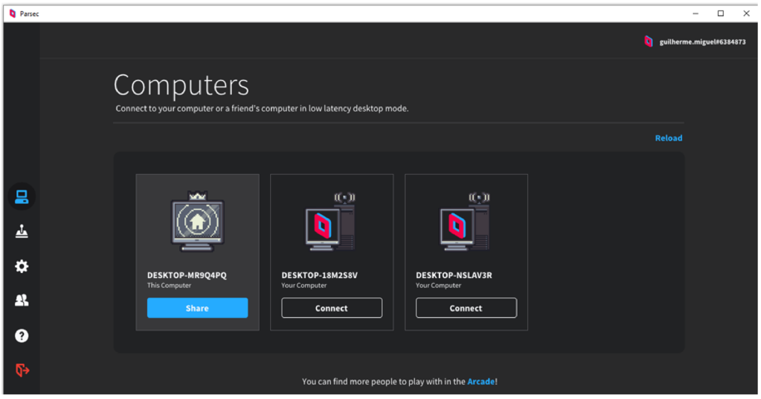


Figure 8 – Test of the Object Detection as an input device
Source: Authors(2021).

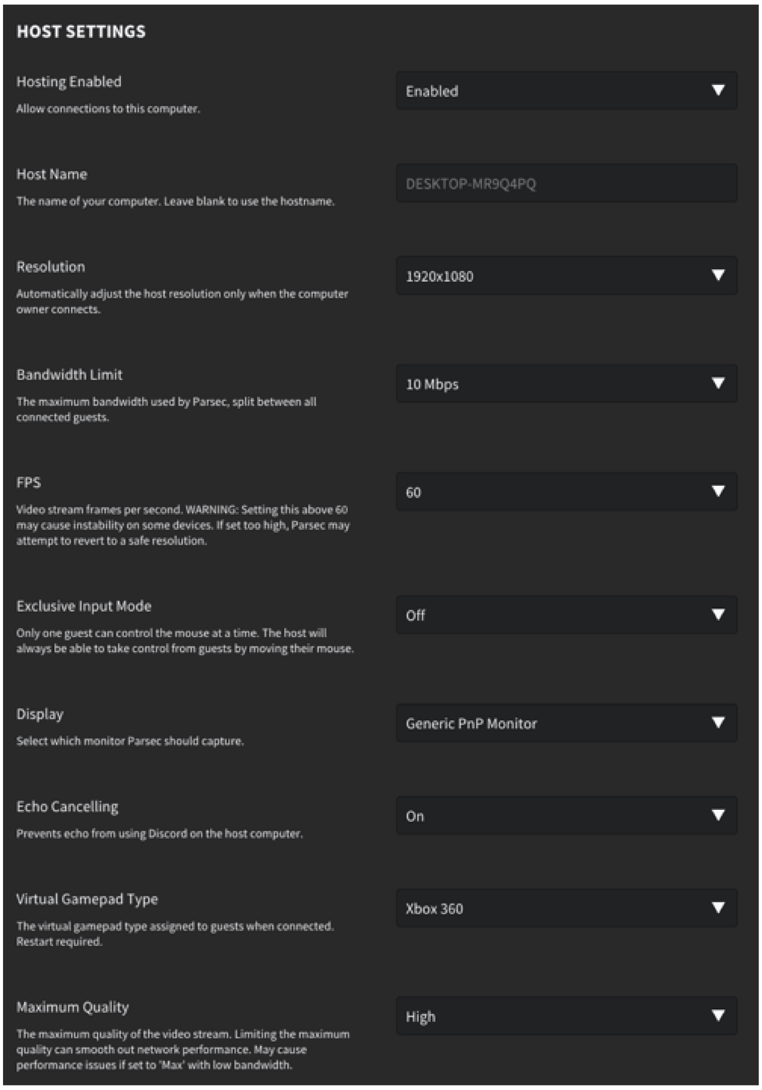
After the game control was implemented, the next step is to implement the Game Streaming Client.

4.2 Game Streaming Client

As presented on the session 3.8a Game streaming solution is a highly specialized type of ultra-low latency peer-to-peer audio, video, and input control data streaming client-server application that allows users to play games on any device by utilizing the CPU and GPU resources from a remote machine over the Internet, as the Server is responsible for: receiving the input data from the client and use it in an emulated input device (keyboard, Mouse or XBOX One and DualShock4), run the game and encode the audio and video output in a video stream codec to send it via Internet; and the client is responsible for: send the input device data to the Server, decode the video stream from the game and display it to the user.



(a)



(b)

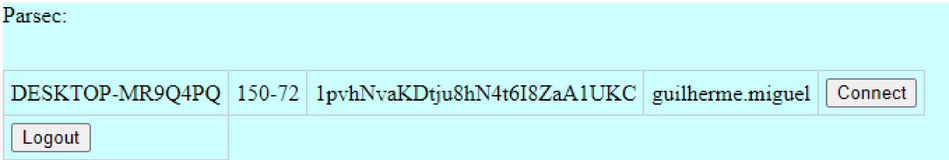
Figure 9 – PARSEC Server Application: a) the Main Screen; b) Host Configuration Options
Source: Authors(2021).

The server application's main screen, shown in Figure 9.a), shows its server ID (with a blue button in the bottom labeled SHARE) and all the other servers that are registered on same account. Figure 9.b), shows the settings that can be changed for that server, which are:

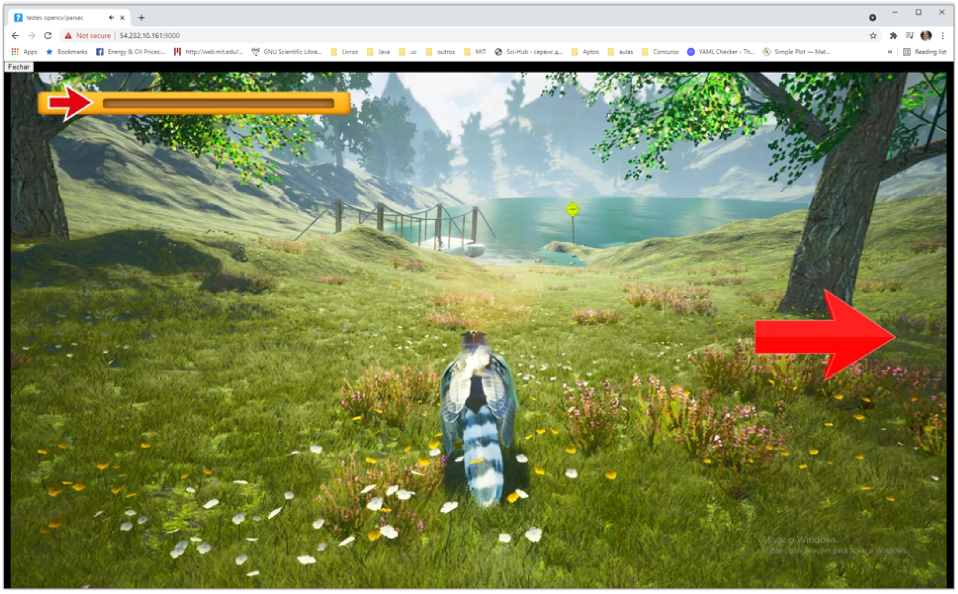
- Hosting Enable: If that server is available for sharing.
- Host Name: The windows host name.
- Resolution: The resolution of the shared screen.
- Bandwidth Limit: The total bandwidth limit (which will be divided by all the connected user on that server).
- FPS: The screen refresh rate.
- Exclusive Input Mode: If only one user will be able to control the server mouse at the time.
- Display: The display in which the screen will be captures.
- Echo Canceling: This option is not used by this framework.
- Virtual Gamepad Type: The game pad type that will be emulated by the game server.
- Maximum Quality: The maximum quality that the image will be encoded.

The PARSEC game streaming binaries for windows was installed on the machine that was used as the game server. The PARSEC Software Development Kit (SDK) JS library was used for the client Web application development since it provides the libraries, implemented with low-level computer instructions, that handles the: peer-to-peer connectivity, networking, hardware acceleration for video/audio processing, traditional input methods (keyboard and Mouse) and game console controllers (XBOX One and DualShock4) for the client-side.

(a)



(b)



(c)

Figure 10 – PARSEC containers: a) login screen; b) Available Servers; c) Game Image streamed from the game server
Source: Authors(2021).

To access the game streaming client Web application, first, the user need to log in with his PARSEC account, as shown in Figure 10.a). After the correct login and password information is provided, it is presented to the user a table with all the servers that are available to run the game, as shown in figure 10.b). Once a server is chosen, and the connection is established with it, the image of the game is presented to the user, and its control data also begins to be sent via the proxy server, which can be seen in Figure 10.c).

When the therapy session is finished, the used click on the *Fechar* button, as shown on the up left corner of figure 10.c), closing the connection with the game server and return to the last web page, shown in Figure 10.b).

However, to become a fully remote therapy session, there must be a real-time audio and video communication channel between the patient and the therapist that is implemented via a Web conference Client.

4.3 Web Conference Client

The real-time communication channel between the patient and its therapist was implemented using the WebRTC architecture, as shown in Figure 3, composed of a Web application Server and a Presence Server. The Web application server is responsible for: providing the HTML and JavaScript files and also adding, remove and broadcast this information to all participants in a conference room.

The PEERS was used as the Presence Server, responsible for managing the establishment of the peer-to-peer: audio, video, and data channels between the participants of the conference room; be its binaries installed on the same machine as the Web application server.

To initiate a Web Conference, the user is connected to the application server, informing in the browser URL the room-id which it wants to enter, as shown in Figure 11.a) and in Figure 11.b) where is shown a screen capture of the conference room.

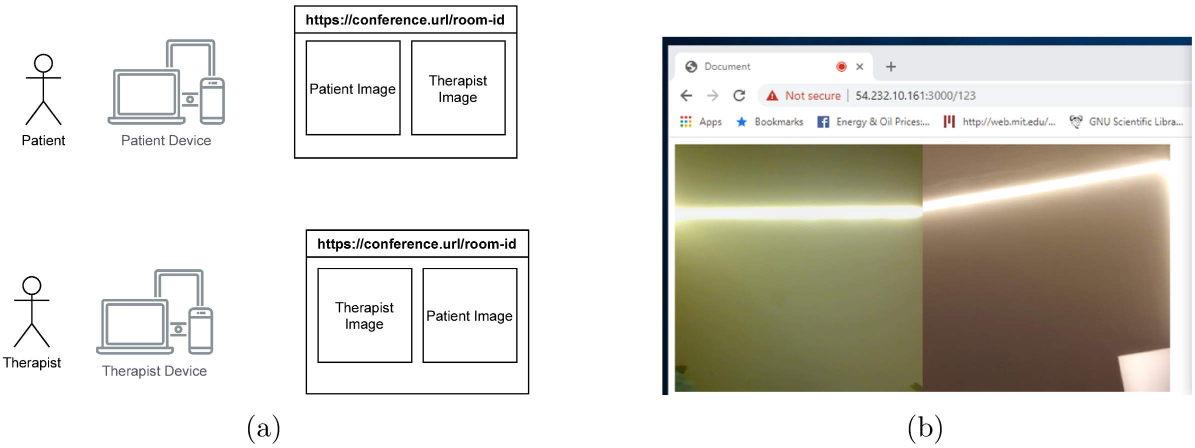
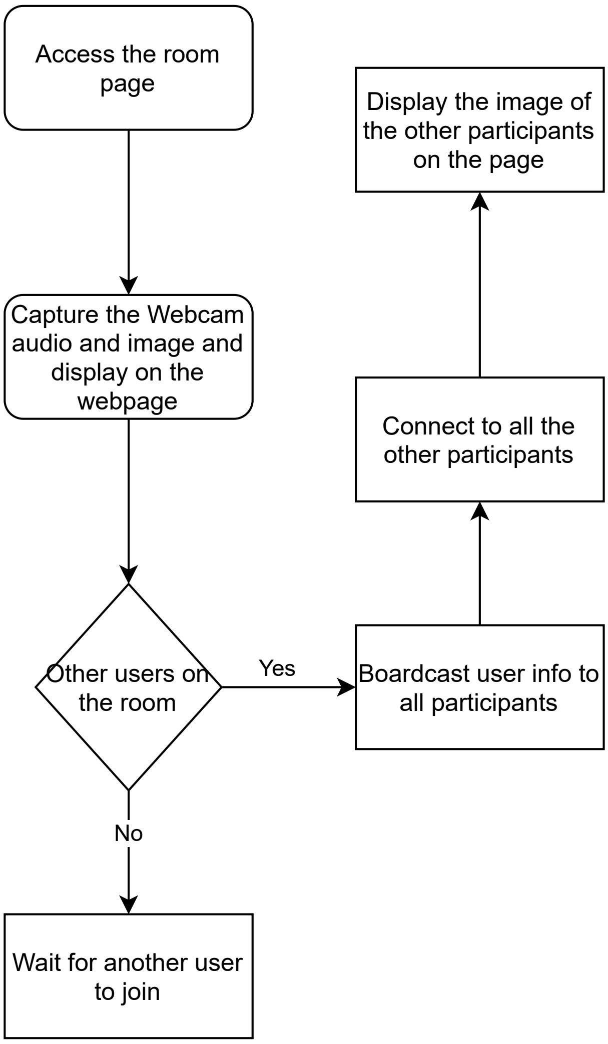


Figure 11 – Web Conference: a) Schematics; b) Web Page Example
Source: Authors(2021).

When accessing a conference room, the Web application server sends to the user the Web Page HTML and JS files and the WebCam image of the device are displayed on the browser. Then, users, web browser requests the list of all the participants in the room. Once the list is received, and there are other users in the room, the presence server is contacted to assist the establishment of a peer-to-peer connection for audio and video streaming between users. This process is shown in Figure 12.



(a)

Figure 12 – Web Conference diagram
Source: Authors(2021).

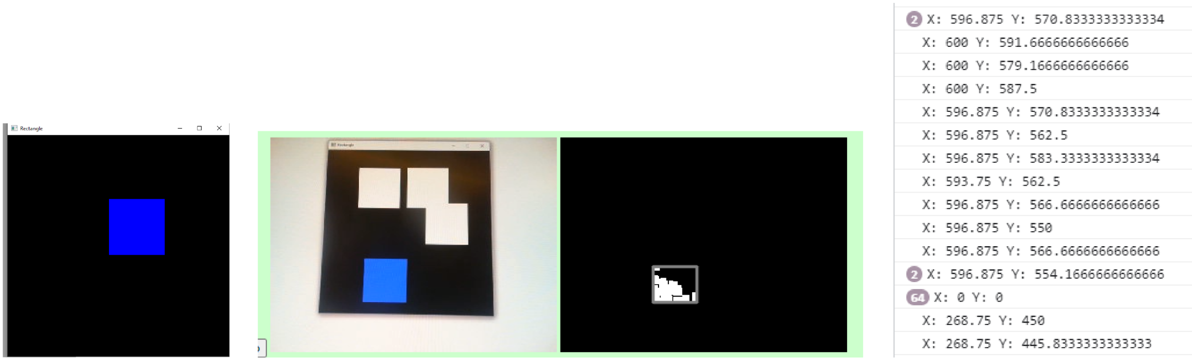
The Web Applications and the PEERJS server were installed on a Ubuntu Linux distribution at an Amazon AWS EC2 Instance with one vCPU, 1GB of RAM, HD with 30 GB, 25Gbps low-latency internet connection.

5 PRELIMINARY RESULTS

After the implementation and integration of all the applications: Object Tracking, Game Streaming Client and Web Conference Client; tests were conducted to verify the performance of the hole framework and its viability as a telerehabilitation system. This tests results and performance analysis are presented on the next sessions.

5.1 Object Tracking Test

using a Python script that draws a blue square on the screen in random positions every four seconds, as shown in Figure 13.a), been replaced by a white square after 2 seconds as in Figure 13.b). The object detection algorithm was tested using a webcam that captures the screen image, analyses it and prints the detected object position on the log console of the web browser, as shown in Figure 13.c).



(a) (b) (c)

Figure 13 – Object Detection: a) Object generator window; b) Object detection window; c) Position data acquired
Source: Authors(2021).

Figure 13.a) shows the visual output of the Python script, with a blue square and Figure 13.b) presents the image of the webcam used as a input for the object recognition algorithm and the area where the object was detected. The position data are shown in Figure 13.c). In this test, the average run time of each object detection algorithm interaction on JS was 8.958 milliseconds, been this function executed 30 times/second, which was also the frame rate of the WebCam.

The object detection algorithm then sends its data to the game server, been the time in which a input data take to be transmitted from the patient device to the game server and vice versa called network latency. The Network latency between the game patient device and the game server is provided by the PARSEC application, been presented on the next section.

5.2 Game Streaming Test

In game streaming service, the images from the game must be encoded on a video container, transmitted over the internet and, on the patient's device, be decoded and presented on the device screen, been the time consumed by all this steps expected to be as low as possible. The encode, decode and network latency times are provided by the PARSEC application and the sum of this times is referred as total time latency time.

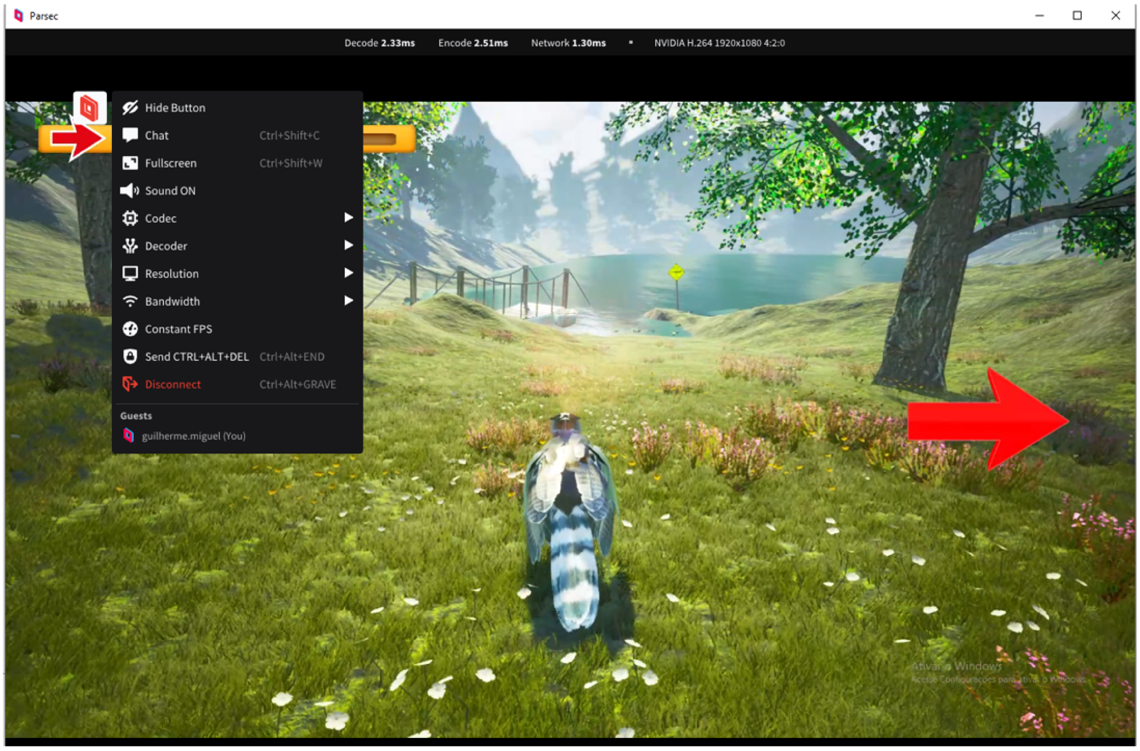


Figure 14 – PARSEC Decode, Encode, Network latency and Video codec used
Source: Authors(2021).

The network latency time between the patient device and the game server it is considered to be the same in both directions, which means that the network latency between the input data sent from the patient device to the game server is the same as the audio and video stream network delay from the game server to the patient device. The network latency is provided by the PARSEC application, as shown on the top of Figure 14.

To simulate a real scenario, different internet connection access methods where used in order to obtain the Network and total delay for the communication between the game server and the patient device, and between the patient and therapist device. For this first test, the following access type were used: HSDPA connection shared via USB cable, HSDPA connection shared via Wi-Fi Hotspot, LTE connection shared via USB cable, LTE connection shared via Wi-Fi Hotspot and with the laptop connected via a broadband Wi-Fi router. The game server was directly connected to a GPON broadband router via Ethernet Cable, been the internet service provider the VIVO operator. The mobile internet connection provider used for connecting to the internet via HSDPA or LTE was also VIVO. The Claro operator was the internet service provider via its broadband router Wi-Fi connection. Despite been the GPON and the mobile internet access been provided by the same company, its access networks for landline and mobile transport network are completely disjoint. Due to this network design, it can be assumed that the delay characteristics will not change if another mobile or broadband network operator is used by the patient.

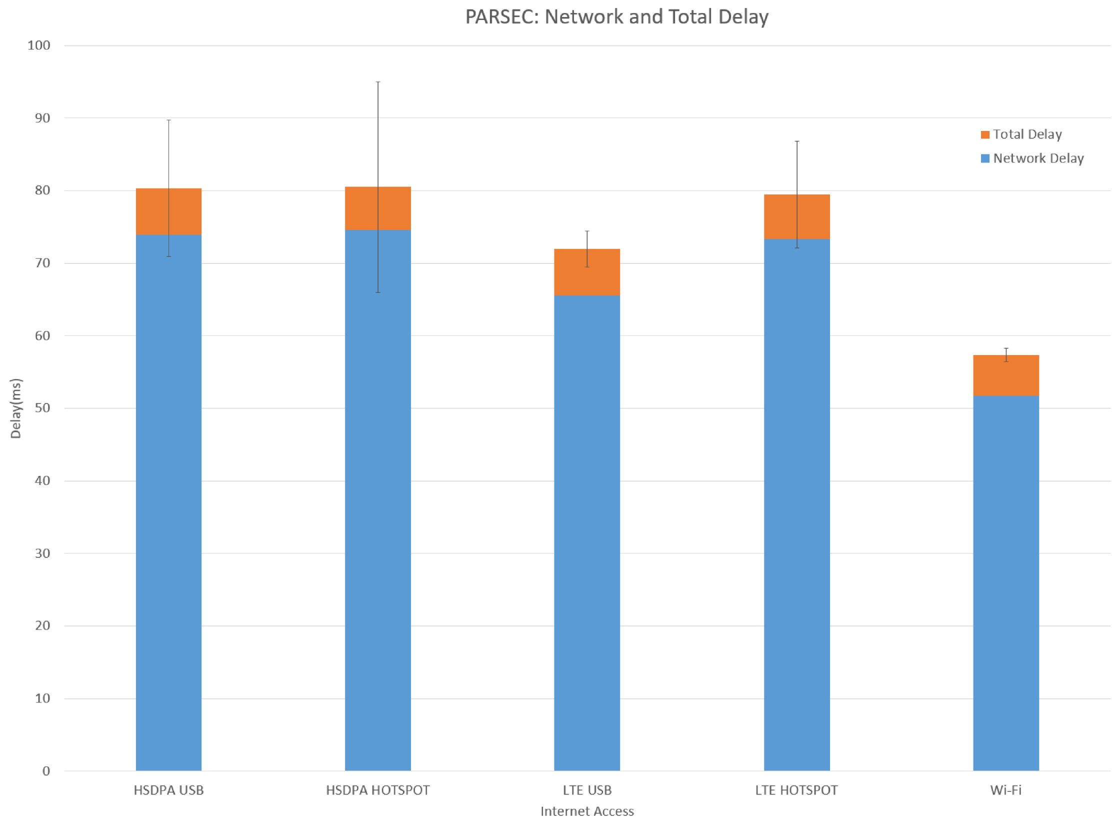
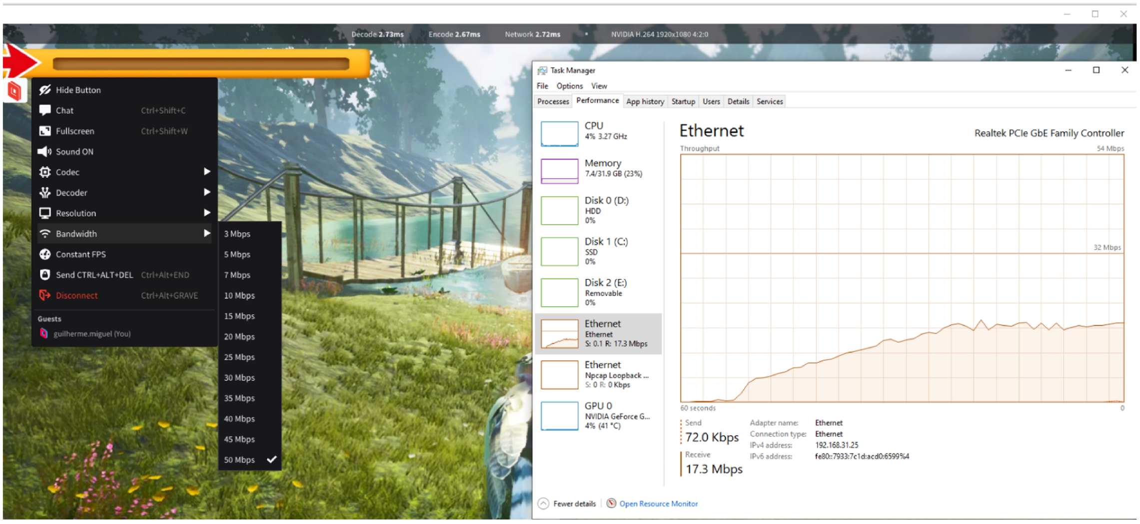


Figure 15 – PARSEC Network and Total Delay by internet access type
Source: Authors(2021).

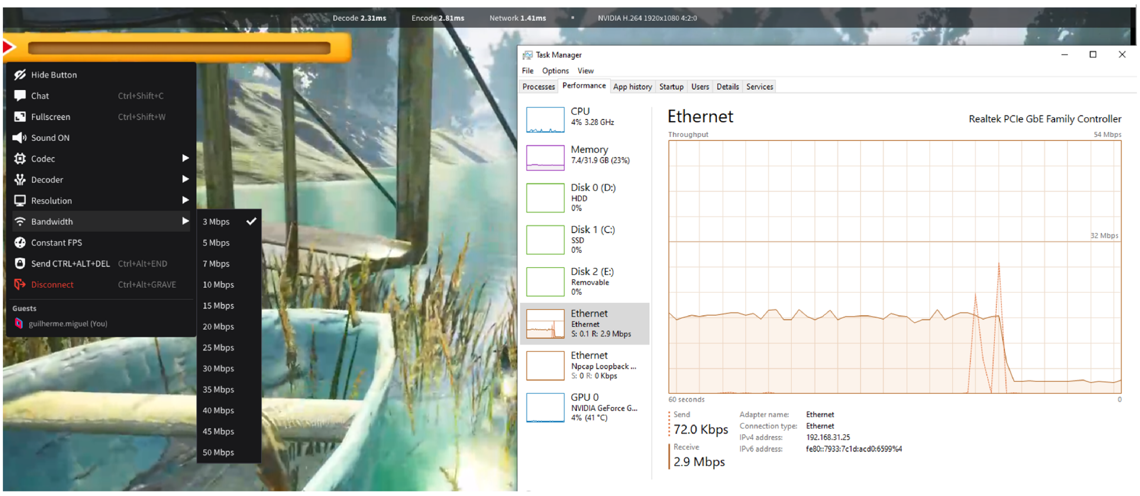
In Figure 15, a bar chart is presented having its bars representing: the Network Delay (blue) and the Total Delay (orange) both in milliseconds. The Error bars are also shown for each bar representing the standard deviation of the samples collected. When connected via HSDPA by a USB cable, the average network delay was 73.93ms and the total delay was 80.344ms on average with a standard deviation of 9.44ms. When the Wi-Fi Hotspot was used to share the HSDPA connection, the average network delay was 74.564ms and the total delay was 80.519ms with a standard deviation of 5.955ms. By sharing via USB cable a LTE internet connection, the network latency was in average 65.657ms and the total delay was 71.966ms in average, with a standard deviation of 6.399. By sharing the same LTE connection using a Wi-Fi Hotspot, the average network delay was 73.36889ms and the average total delay was 79.4611, with a standart deviation of 7.3177ms. When the laptop was using a broadband Wi-Fi connection to the internet, the network delay was 51.6575ms and the total delay was 57.35ms with a standard deviation of 0.893ms.

By analyzing the chart from Figure 15, it is clear that a Wi-Fi broadband connection should be preferred, followed by a LTE or HSDPA connection shared via USB cable. If there is no broadband Wi-Fi connection available or its not possible to use a USB share the mobile internet connection, the LTE access should be prefered over the HSDPA via Wi-Fi Hotspot, since despite its close average netowrk and total delay values, the standard deviation of the HSDPA connection should makes the video stream more unstable than when using the LTE connection.

The Network Bandwidth also needs to be taken in consideration for this analysis since the PARSEC server increases or decreases the image quality on the video stream, based on the network delay that is observed.



(a)



(b)

Figure 16 – PARSEC Bandwidth vs Image Quality: a) Unrestricted Bandwidth (Up to 50 Mbps); b) Restricted Bandwidth (Up to 3 Mbps);
Source: Authors(2021).

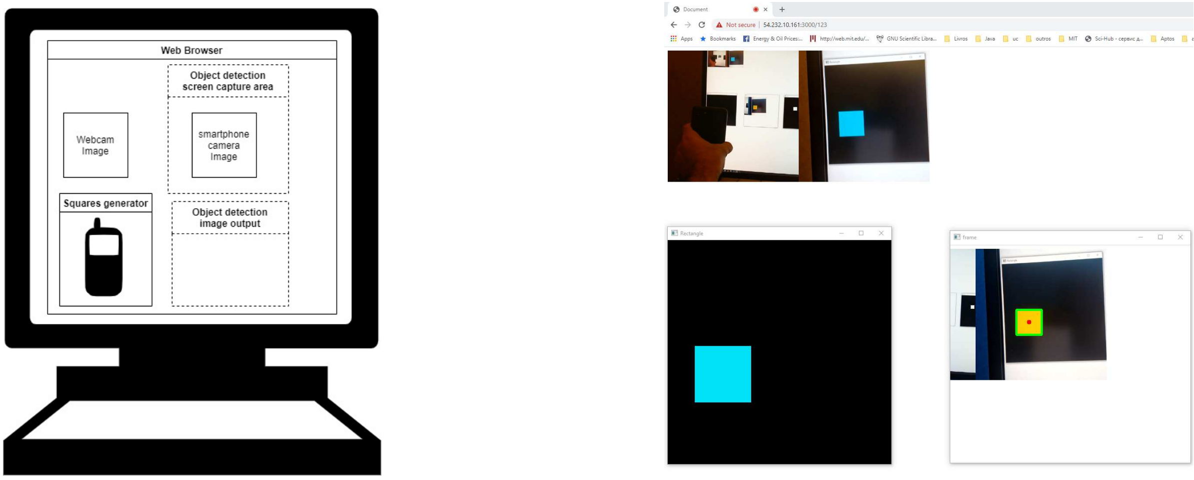
It can be observed that in Figure 16.a), it can be noticed that the initial bandwidth consumption is very low and, due to the low latency of the connection Wi-Fi connection that was used, the game server increased the image quality, which also increases the bandwidth requirement. When the total bandwidth was restricted to 3 Mbps maximum, as shown in Figure 16.b), the game image also have its image quality decreased. This test shows that the bandwidth limitation of a HSDPA connection will decrease the image quality, since the typical bandwidth of this technology is 7 Mbps. Both LTE and Wi-Fi provides a bandwidth of more than 50 Mbps, that is the maximum bandwidth used by

PARSEC Server, as shown in Figure 9.b).

All the tested access technologies can be used for game streaming, been the broadband Wi-Fi the preferred one, since, as shown by Jain *et al.*[59], the average human response time for a visual stimulus is in order of 180 to 200 milliseconds and 140–160 milliseconds for sound. By keeping the average total round trip delay (from the object detection to its corresponded image received) less than 180ms, the user won't have a low quality experience with the exergame, despite the lower the delay, the higher will be the quality of its experience with the exergame. Considering that the presented average total delay represents only the time required to go from the game server to the patient deivce or the time requiered for a package to go from the patient device to the game server, the use of a HSDPA or LTE connection shared via Wi-fi hotspot or a HSDPA connection shared via USB cable, but should be avoided, since any increase on the network delay will cause degradation on the game streaming performance, been noticed as lag in character movement, frame loss or the connection to the server been abruptly closed.

5.3 Web Conference Test

The audio and video communication channel between the patient and the therapist are also subjected to the same encode, decode and network latency times as the game streaming, been in this case a bi-directional communication, as the image from the patient and the therapist captured via Webcam to be encoded and transmitted to each other. Different from the PARSEC, the WebRTC API used does not provide any information about the encode, decode or network latency, which required an external application to measure this times.



(a) (b)

Figure 17 – WebRTC delay estimation: a) Object generator window; b) Object detection window;
Source: Authors(2021).

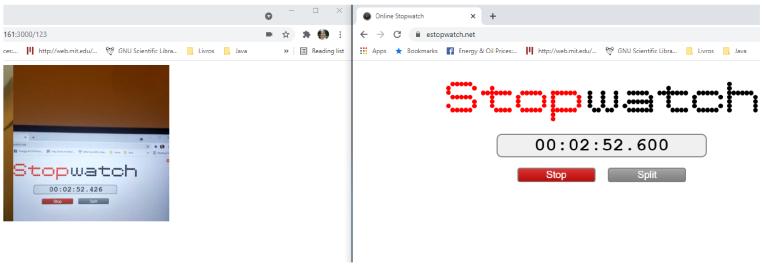
To estimate the total latency time of the communication channel between the patient and the therapist, the same python script that generates squares and the object detection algorithm was used, both registering the time in which each square is drawn (blue or white) or begins or end to be detected. To feed this data into the object recognition script, first, a conference room was created between a laptop and a smartphone. Then the smartphone camera was used to capture the squares on the generator window. This image was then transmitted from the smartphone to the web browser on the laptop. De detection algorithm captures the laptop screen portion in which the image from the mobile phone was displayed. Then, registering when a square begins and ends to be detected, the total latency can be calculated as the difference in time in which the blue square was drawn and started to be detected, or the white square was drawn, and consequently, it stops to be detected. A schematic of this system is shown in 17.a) and Figure 17.b) is a picture of the screen when the tests were executed. Also, Figure 18.a) shows the registered time in which a square was drawn in blue (Start) or white (Stop), and in Figure 18.b) when the square begins and ends its detection.

Start:	13.449517
Stop:	15.464541
Start:	17.474671
Stop:	19.482883
Start:	21.483934
Stop:	23.487546
Start:	25.498976
Stop:	27.505045
Start:	29.506027
Stop:	31.520312
Start:	33.535361
Stop:	35.550913
Start:	37.556795
Stop:	39.561597
Start:	41.563404
Stop:	43.577845

(a)

Start:	13.584
Stop:	15.577
Start:	17.572
Stop:	19.632
Start:	21.622
Stop:	23.614
Start:	25.615
Stop:	27.607
Start:	29.638
Stop:	31.668
Start:	33.700
Stop:	35.699
Start:	37.689
Stop:	39.683
Start:	41.685
Stop:	43.671

(b)



(c)

Figure 18 – WebRTC latency estimation: a) Object generator events time; b) Object detection events time; c) Latency estimation via stopwatch
Source: Authors(2021).

A second latency estimation procedure, as shown in Figure 18.c), was also conducted by opening a precise stopwatch application on the laptop and establish a communication channel with the smartphone been both presented side by side on the laptop screen. Then, with instant images of the desktop, the difference on the time shown in each image is the total delay time.

Since on the WebRTC or communication channel between patient and the therapist application test, the mobile phone represents one of the participants, it was not necessary to connect the laptop via a shared mobile internet connection. The tests was executed with the laptop connected to a broadband internet connection and the smartphone was connected to the different access networks.

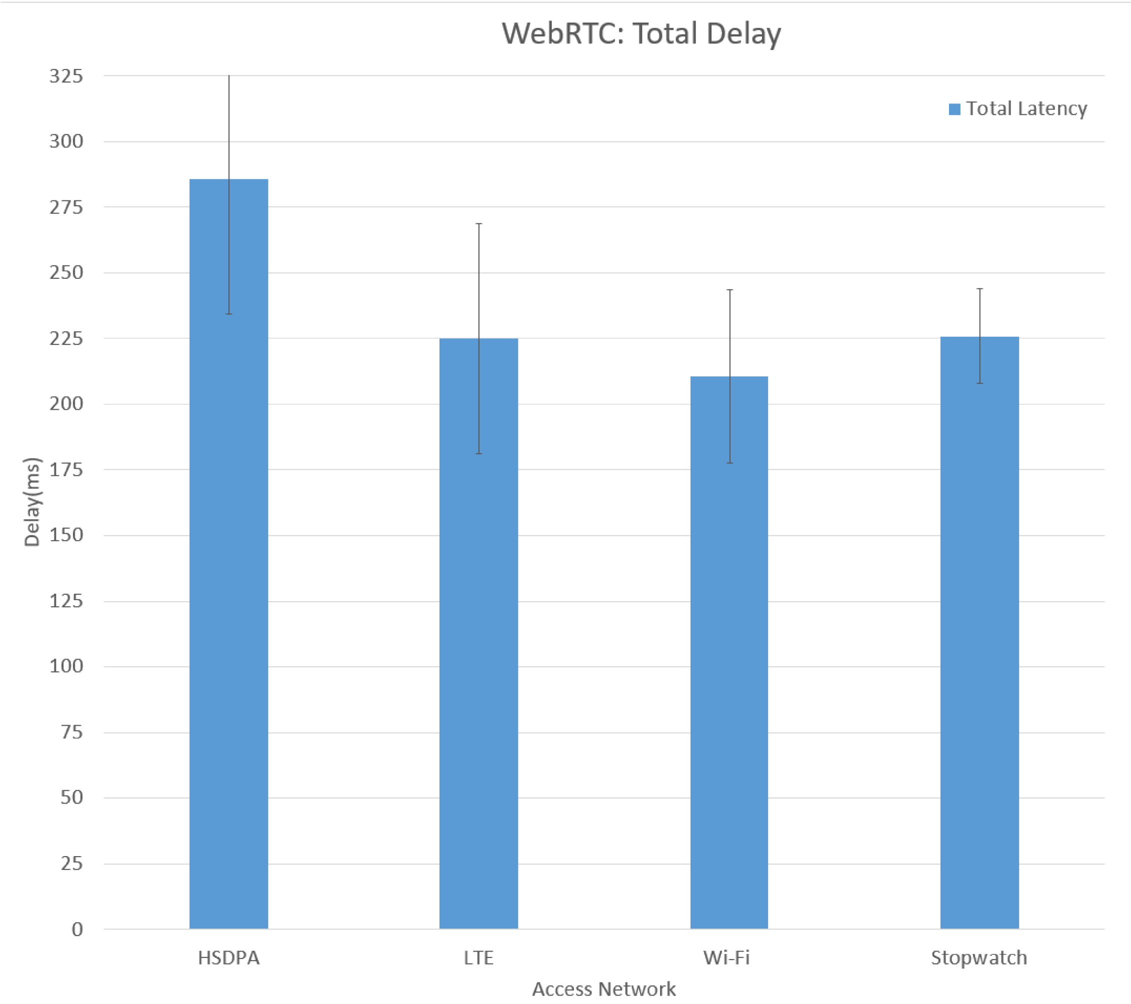


Figure 19 – WebRTC Total Delay by internet access type
Source: Authors(2021).

In Figure 19, the Total Delay time is presented for each internet access technology, been each bar representing the Total Delay between an image been captured by the smartphone and presented on the laptop. The first bar was the Total Delay when the HSDPA mobile internet connection was used, and the average total delay was 285.8331 with a standard deviation of 51.643ms. The average total delay time when the smartphone was connected to the Internet via LTE was 225.1187ms with a standard deviation of 43.7698ms. The following two bars (Wi-FI and Stopwatch) presents the average time when the same broadband WiFi connection was used, but the total delay time estimation method was different. The delay estimation shown on the bar identified as WiFi uses the same method as the HSDPA and LTE bars, been its total delay 210ms with a standard deviation of 33.03ms. The bar identified as Stopwatch uses the method described in the Figure 18.c), and the total delay time 225.95ms with a standard deviation of 18.17 seconds.

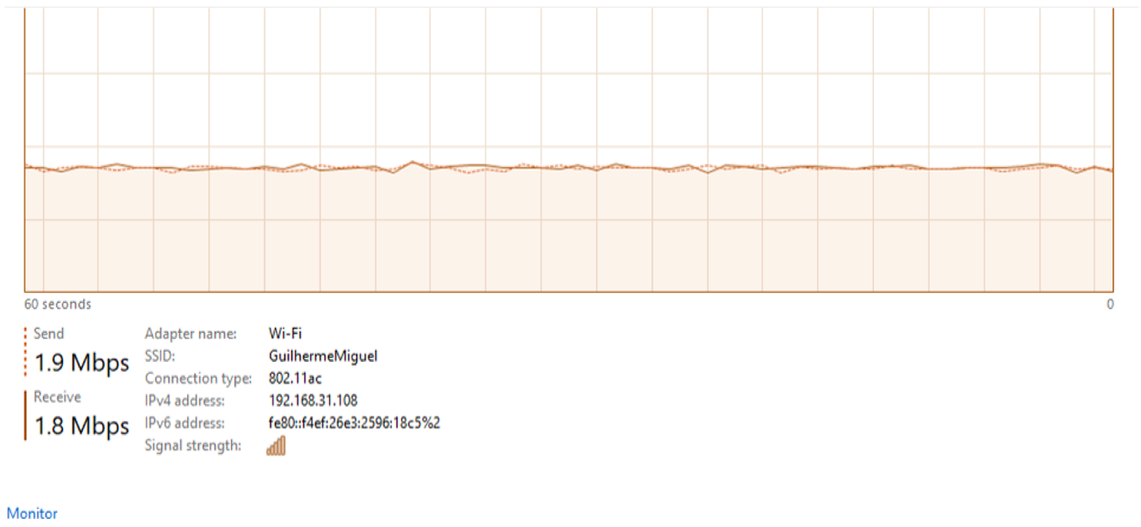


Figure 20 – WebRTC Bandwidth;
Source: Authors(2021).

The bandwidth requirement for the audio and video communication channel between the patient and the therapist is shown in Figure 20, and it uses between 1.8M and 1.9 Mbps for each audio and video stream. The communication channel between the patient and the therapist will compete with the game streaming for bandwidth, which can impose a limitation on the game streaming image quality and can also imply a higher delay if an HSDPA connection is used.

The difference observed between the total delay times of the PARSEC and WebRTC applications can be explained by the different characteristics of each application. A game streaming application has a directional video feed from the game server to the patient device, and thus, the application and video encoder can be optimized by using CPU and GPU features in order to achieve an ultra-low latency delay. On an audio and video conference application over the web, the JS APIs and resources implemented on the web browser are the ones in charge of encoding the audio and video from the user's Webcam as well as decoding the audio and video from all the participants. Since all the encode and decode processes are executed by the web browser, they will require a higher time to be executed. Also, there is required the use of buffers on this application, which also implies a higher delay for the communication. As a communication platform, it also assumes that the audio and video need to have the best quality possible, so its algorithms are tunned for quality over performance.

6 DISCUSSION

In Table 2 it can be identified that, in order to be classified as a exergame, the game must used on Cognitive or Motor impairment rehabilitation. The game also can dynamically increase/decrease its difficulty or challenge, based on adjustable thresholds defined by the therapist regarding the treatment goals, and support for reporting to the user his achievements and saving them for further analyses. The exergame can also be developed to have two or tree dimensions, but it should be considered that despite a 3D games are more immersive and interesting to play, it requires a Graphics Processing Unit for its graphics rendering. To be executed in a multi-player environment, an exergame needs to make use of a internet connection exchange data with a server which is responsible to synchronize the interactions between players.

Table 2 – Serious Games Classification

Rehabilitation Criteria	**Application Area**	Cognitive/Motor rehabilitation
	Adaptability	The game dynamically adapts its difficulty or challenge, according to the player performance and abilities
	Performance Feedback	Capability of the system to transmit to the patient the results of the interaction
	Progress Monitoring	Capability of the system to allow saving the results of patients interaction with the system
	Portability	The possibility to use the system in multiple locations: home, hospital or clinic
game Criteria	**Interaction Techlology**	The input devices used by the patient to interact with the system
	game Interface	The game virtual environment dimensionality: Two or Tree dimensions (2D or 3D)
	Competitive/ Collaborative	The game is based on competitive or collaborative tasks, in a single or multiplayer environment
	game Genre	The type of action required to complete a task on the game.

Once the exergame classified, it is important to analyze the articles referenced on the Table 1 session 2, in terms of Therapy Evaluation, Framework used,input device (if it needs to be custom build, its use complexity and estimated costs), the device that must be acquired by the patient and the clinic/hospital Infrastructure requirements and its implied costs for both cases.

Table 3 – Main characteristics of the existing solutions.

			Input Device			Patient device
Articles	Evaluation	Framework	Name	Custom	Complexity	Hardware/Software
[24]	Offline	Client-Server	Glove	x	3	Both
[25]	Offline	Client-Server	Oculos Go		1	Hardware
[26]	Offline	Web-Based	Joystick		1	
[27]	Offline	Web-Based	Wobble Board	x	2	Software
[28]	Offline	Web-Based	MS Kinect		4	Both
[22]	Offline	Web-Based	Inertial sensors	x	2	Software
[29]	Offline	Web-Based	Glove	x	3	Software
[21, 30]	Offline	Layers	MS Kinect		4	Both
[31]	Offline	Cloud Based	Robotic Arm	x	1	Software
[19, 20]	Offline	Cloud Based	MS Kinect		4	Both
[32]	Offline	Multi User	MS Kinect		4	Both
[8]	Offline	Multi User	Orthosis Wrist	x	3	Both
This proposal	Online/Offline	Cloud Based/Streaming	Object Tracking		1	

In Table 3, the columns:

- Article: References the works that is been analyzed on the respective line;
- Evalution: The possibility for the therapist to interact with the patient via an audio and video real-time communication channel;
- Framework: The framework that the exergame is based;
- Patient device Name: The commercial name of the product used as a input device;
- Patient device Complexity: The level of difficult to assembly, maintain, calibrate and use of the device, between 1 (easy) to 5 (difficult);
- Patient device Hardware/Software: If the input device requires special hardware interface and/or software installed on the patient device in order to work;

This information are summarized on Table 3 and the following considerations can be made:

1. None of the reviewed works provides a method to evaluate the remote therapy in real time by the therapist.

2. Only for the Web-Based Framework that uses a Joystick as a input device doesn't require any special hardware or software to be installed on the patient device in order to properly run the exergame.

3. The input devices that needs to be custom build has a high cost, despite its components low price, due to a its labor intensive fabrication process and constant adjustments.

4. Oculos Go requires that the patient device have a powerful CPU and GPU in order to process its input and display data.

5. MS Kinect requires that the patient device have a powerful CPU and GPU in order to process its input images and calculate the displacement data from its multiple cameras images and lidar information.

6. The input devices: Glove, Wobble Board, Ms Kinect, Inertial Sensors, Robotic Arm and Orthosis Wirst requires a specific software to be installed on the patient device to convert the input data into movement commands that will be executed by the avatar on the exergame.

7. Client-Server and Multi User exergames requires the game to be installed on the patient device, which also demands a powerful CPU and GPU in order to display the game graphics.

8. All of the proposed frameworks requires that patient to have at least a modern Computer in order to execute the exergame, which can be a limitation for the remote therapy adoption.

9. For a Clinic/Hospital, the use of Could Based Framework is the most cost effective due to its scalability, performance and price.

By the considerations above, it is possible to conclude that most of the solutions presented has limitations regarding the input device price or complexity of use, the required specialized Hardware (powerful CPU and GPU), and Software to process the signals from the input device and the exergame, that is needed to be installed on the patient Personal Computer, which also implies in higher costs for the patient. Any smartphone, laptop, or tablet nowadays has a built webcam that can be used to detect the movement of a specific object that, attached to the patient body, can be used to move the avatar in the exergame by translating the object position in relation to the image borders into avatars forward, backward or lateral movement.

Using a game streaming application allows a user to play a game that's been executed on a virtual machine, usually hosted on a cloud infrastructure streaming, and its rendered images are sent over the Internet to the user device (PC, Phone, or Tablet). When using a cloud infrastructure use, the patient will pay only for the time that the game streaming server is active (the duration of his rehabilitation session), eliminating the initial cost with the acquisition of a powerful personal computer, its maintenance, and hardware upgrades over time. The PARSEC SDK was used in this Framework to implement the game streaming capability due to its low-latency characteristics and multiple input methods support. Also, since this Framework implements the Web version of the PARSEC SDK, any: PC, smartphone, or tablet; despite its: operational system, CPU, and GPU capabilities, that has a modern Web Browser installed can be used by the patient.

By using a game streaming application, the therapist can also make use of it to access the game server on its personal device to adjust the game configurations considering the therapy goals and validate in real-time the performance of the patient on the exergame, without needing to rely only on the game performance data. The feedback between patient and therapist is also an innovation that was introduced by this framework proposal, by using a WebRTC Web Application to establish an audio/video communication channel between them.

All the innovative approaches implemented on this framework allows the therapy to be conducted in a fully remote manner results:

- Reduced costs for both patient and clinic by means of input device and computer hardware/software acquisition and maintenance.
- Allows the patient to use a portable device to execute its exergame routines
- Allow the therapist to attend multiple rehabilitation sessions at the same time.
- The therapist can give real-time feedback to the patient regarding its current condition.
- Reduce the number of times in which the patient needs to go to the clinic or the therapist to go to the patient's house in order to verify therapy evolution and consult with its medical professionals.

- Less exposure of the patient and the therapist to malicious diseases common in hospital environments.

The preliminary test conducted shown that the use of a broadband WiFi connection offers the best image quality, low delay, and jitter, which are essential factors for the overall user experience. The use of mobile access technologies such as LTE and HSDPA should be avoided, been the last one the worst-case scenario observed on the tests. However, even when using a mobile HSDPA access network, it is possible to play the game and have the audio and video communication channel between the patient and the therapist at the same time.

7 CONCLUSION

The home-based Telerehabilitation, in addition to facilitating for the therapist, as he/she does not have to move to where the patient is, also facilitates and includes the patient. Many patients would not be able to attend the rehabilitation clinic as often as necessary to provide treatment. Also, in periods of a pandemic, when people are isolated at home, home-based Telerehabilitation has greater prominence.

In this context, with the improvement of telerehabilitation systems, which are mainly composed of well-designed exergames to improve patient engagement, it is necessary that the patient has a computer suitable for installing and running the exergame. Moreover, it is not always possible for the patient to buy or get government assistance to buy a good computer and the necessary devices to run the exergame.

To tackle this issue, this work presented a framework of a low cost fully remote Telerehabilitation that uses the image obtained by the camera from the patient device (smartphone, PC, tablet) to track an object that can be held by the patient or attached to its body, and use its position relative to the image frame as the input control of the exergame. The game is executed on a game server which feeds the game with the input commands from the patient device, renders the game images, encodes it in a video codec, and transmits this information to the patient device. Also, the therapist and patient can interact in real-time via an audio and video communication channel that is also part of this proposed Framework.

All the different parts (object recognition, game streaming, and real-time communication) of this Framework were implemented and tested in order to validate its behavior in different scenarios. Since it requires an internet connection to work, the internet access via mobile networks (HSDPA and LTE) and via broadband WiFi router were tested. The mobile networks were shared by a smartphone with a laptop, via USB and Wi-FI Hotspot. After the conclusion of the tests, the results confirmed that all the interfaces on the tests were considered viable to be used, been the HSDPA mobile internet access the one with the higher delay, followed by the LTE and then the broadband WiFi, in line with the literature and the author's expectations.

Despite the satisfactory tests results and high importance for the community, more testing of this platform is required in order to validate it with patients and receiving their inputs to improve the overall system and user interface. Also, the support for other exergames and input devices must be included on the platform in further developments.

References

1 MERRIAM-WEBSTER. **Game**. Disponível em: <https://www.merriam-webster.com/dictionary/game>. Cited at page 17.

2 MERRIAM-WEBSTER. **Video Game**. Disponível em: <https://www.merriam-webster.com/dictionary/videogame>. Cited at page 17.

3 MUÑOZ, J. *et al.* Multimodal system for rehabilitation aids using videogames. In: IEEE. **2014 IEEE Central America and Panama Convention (CONCAPAN XXXIV)**. 2014. p. 1–7. Disponível em: <http://dx.doi.org/10.1109/CONCAPAN.2014.7000395>. Cited at page 17.

4 VOURVOPOULOS, A. *et al.* Rehabcity: design and validation of a cognitive assessment and rehabilitation tool through gamified simulations of activities of daily living. In: **Proceedings of the 11th conference on advances in computer entertainment technology**. [s.n.], 2014. p. 1–8. Disponível em: <http://dx.doi.org/10.1145/2663806.2663852>. Cited at page 17.

5 MUÑOZ, J. *et al.* Space connection-a multiplayer collaborative biofeedback game to promote empathy in teenagers: A feasibility study. In: SCITEPRESS. **International Conference on Physiological Computing Systems**. 2016. v. 2, p. 88–97. Disponível em: <http://dx.doi.org/10.5220/0005948400880097>. Cited at page 17.

6 WIEMEYER, J. Serious games in neurorehabilitation: A systematic review of recent evidence. In: **Proceedings of the 2014 ACM International Workshop on Serious Games**. [s.n.], 2014. p. 33–38. Disponível em: <http://dx.doi.org/10.1145/2656719.2656730>. Cited at page 17.

7 FARIA, A. L. *et al.* The benefits of emotional stimuli in a virtual reality cognitive and motor rehabilitation task: Assessing the impact of positive, negative and neutral stimuli with stroke patients. In: IEEE. **2015 International Conference on Virtual Rehabilitation (ICVR)**. 2015. p. 65–71. Disponível em: <http://dx.doi.org/10.1109/ICVR.2015.7358584>. Cited at page 17.

8 ANDRADE, K. D. O. *et al.* Rehabilitation robotics and serious games: An initial architecture for simultaneous players. In: IEEE. **2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)**. 2013. p. 1–6. Disponível em: <http://dx.doi.org/10.1109/BRC.2013.6487455>. Cited 4 times at pages: 17, 25, 27, and 64.

9 BAMIDIS, P. D. *et al.* Reviewing home based assistive technologies. In: **Handbook of Research on Innovations in the Diagnosis and Treatment of Dementia**. IGI Global, 2015. p. 317–334. Disponível em: <http://dx.doi.org/10.4018/978-1-4666-8234-4.ch017>. Cited at page 17.

10 BHATT, C.; DEY, N.; ASHOUR, A. S. Internet of things and big data technologies for next generation healthcare. Springer, 2017. Disponível em: <http://dx.doi.org/10.1007/978-3-319-49736-5>. Cited at page 17.

11 CHEN, Y. *et al.* Home-based technologies for stroke rehabilitation: A systematic review. **International journal of medical informatics**, Elsevier, v. 123, p. 11–22, 2019. Disponível em: <http://dx.doi.org/10.1016/j.ijmedinf.2018.12.001>. Cited at page 17.

12 AMORIM, P. *et al.* Serious games for stroke telerehabilitation of upper limb-a review for future research. **International Journal of Telerehabilitation**, v. 12, n. 2, p. 65–76, 2020. Disponível em: <http://dx.doi.org/10.5195/ijt.2020.6326>. Cited at page 17.

13 MANTOVANI, E. *et al.* Telemedicine and virtual reality for cognitive rehabilitation: a roadmap for the covid-19 pandemic. **Frontiers in neurology**, Frontiers Media SA, v. 11, 2020. Disponível em: <https://doi.org/10.3389/fneur.2020.00926>. Cited 2 times at pages: 17 and 30.

14 BURKE, J. W. *et al.* Optimising engagement for stroke rehabilitation using serious games. **The Visual Computer**, Springer, v. 25, n. 12, p. 1085–1099, 2009. Disponível em: <https://doi.org/10.1007/s00371-009-0387-4>. Cited at page 17.

15 BURKE, J. W. *et al.* Augmented reality games for upper-limb stroke rehabilitation. In: IEEE. **2010 second international conference on games and virtual worlds for serious applications**. 2010. p. 75–78. Disponível em: <https://doi.org/10.1109/VS-GAMES.2010.21>. Cited at page 17.

16 AGRAWAL, S.; VIEIRA, D. A survey on internet of things. **Abakós**, v. 1, n. 2, p. 78–95, 2013. Disponível em: <https://doi.org/10.5752/P.2316-9451.2013v1n2p781>. Cited at page 18.

17 TAN, L.; WANG, N. Future internet: The internet of things. In: IEEE. **2010 3rd international conference on advanced computer theory and engineering (ICACTE)**. 2010. v. 5, p. V5–376. Disponível em: <https://doi.org/10.1109/ICACTE.2010.5579543>. Cited at page 20.

18 SILVA, T. D. da *et al.* Serious game platform as a possibility for home-based telerehabilitation for individuals with cerebral palsy during covid-19 quarantine–a cross-sectional pilot study. **Frontiers in Psychology**, Frontiers Media SA, v. 12, 2021. Disponível em: <https://doi.org/10.3389/fpsyg.2021.622678>. Cited at page 20.

19 OLIVER, M. *et al.* Ambient intelligence environment for home cognitive telerehabilitation. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 18, n. 11, p. 3671, 2018. Disponível em: <https://doi.org/10.3390/s18113671>. Cited 6 times at pages: 20, 25, 26, 27, 31, and 64.

20 AFYOUNI, I.; MURAD, A.; EINEA, A. Adaptive rehabilitation bots in serious games. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 20, n. 24, p. 7037, 2020. Disponível em: <https://doi.org/10.3390/s20247037>. Cited 5 times at pages: 20, 25, 26, 27, and 64.

21 VOURVOPOULOS, A. *et al.* Rehabnet: A distributed architecture for motor and cognitive neuro-rehabilitation. In: IEEE. **2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)**. 2013. p. 454–459. Disponível em: <https://doi.org/10.1109/HealthCom.2013.6720719>. Cited 4 times at pages: 20, 25, 27, and 64.

22 ZEDDA, A. *et al.* Domomea: A home-based telerehabilitation system for stroke patients. In: IEEE. **2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)**. 2020. p. 5773–5776. Disponível em: <https://doi.org/10.1109/embc44109.2020.9175742>. Cited 4 times at pages: 20, 25, 26, and 64.

23 GHADIYARAM, D.; PAN, J.; BOVIK, A. C. Learning a continuous-time streaming video qoe model. **IEEE Transactions on Image Processing**, IEEE, v. 27, n. 5, p. 2257–2271, 2018. Disponível em: <https://doi.org/10.1109/TIP.2018.2790347>. Cited at page 23.

24 POPESCU, V. G. *et al.* A virtual-reality-based telerehabilitation system with force feedback. **IEEE transactions on Information Technology in Biomedicine**, IEEE, v. 4, n. 1, p. 45–51, 2000. Disponível em: <https://doi.org/10.1109/4233.826858>. Cited 4 times at pages: 25, 27, 28, and 64.

25 VARELA-ALDÁS, J. *et al.* Head-mounted display-based application for cognitive training. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 20, n. 22, p. 6552, 2020. Disponível em: <https://doi.org/10.3390/s20226552>. Cited 3 times at pages: 25, 27, and 64.

26 REINKENSMEYER, D. J. *et al.* Web-based telerehabilitation for the upper extremity after stroke. **IEEE transactions on neural systems and rehabilitation engineering**, IEEE, v. 10, n. 2, p. 102–108, 2002. Disponível em: <https://doi.org/10.1109/tnsre.2002.1031978>. Cited 5 times at pages: 25, 26, 27, 30, and 64.

27 KARIME, A. *et al.* Tele-wobble: A telerehabilitation wobble board for lower extremity therapy. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 61, n. 7, p. 1816–1824, 2012. Disponível em: <https://doi.org/10.1109/TIM.2012.2192338>. Cited 5 times at pages: 25, 26, 27, 29, and 64.

28 PÉREZ-MEDINA, J.-L. *et al.* ephort: towards a reference architecture for tele-rehabilitation systems. **IEEE Access**, IEEE, v. 7, p. 97159–97176, 2019. Disponível em: <https://doi.org/10.1109/ACCESS.2019.2927461>. Cited 4 times at pages: 25, 26, 27, and 64.

29 POSTOLACHE, O. *et al.* Remote monitoring of physical rehabilitation of stroke patients using iot and virtual reality. **IEEE Journal on Selected Areas in Communications**, IEEE, v. 39, n. 2, p. 562–573, 2020. Disponível em: <https://doi.org/10.1109/JSAC.2020.3020600>. Cited 5 times at pages: 25, 26, 27, 28, and 64.

30 PARASKEVOPOULOS, I. *et al.* Virtual reality-based holistic framework: a tool for participatory development of customised playful therapy sessions for motor rehabilitation. In: IEEE. **2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)**. 2016. p. 1–8. Disponível em: <https://doi.org/10.1109/VS-GAMES.2016.7590378>. Cited 4 times at pages: 25, 27, 31, and 64.

31 LI, H.-J.; SONG, A.-G. Architectural design of a cloud robotic system for upper-limb rehabilitation with multimodal interaction. **Journal of Computer Science and Technology**, Springer, v. 32, n. 2, p. 258–268, 2017. Disponível em: <https://doi.org/10.1007/s11390-017-1720-4>. Cited 5 times at pages: 25, 26, 27, 30, and 64.

32 TRIANDAFILOU, K. M. *et al.* Development of a 3d, networked multi-user virtual reality environment for home therapy after stroke. **Journal of neuroengineering and rehabilitation**, Springer, v. 15, n. 1, p. 1–13, 2018. Disponível em: <https://doi.org/10.1186/s12984-018-0429-0>. Cited 3 times at pages: 25, 27, and 64.

33 ZHANG, F.; CLARKE, D.; KNOLL, A. Vehicle detection based on lidar and camera fusion. In: **17th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. [s.n.], 2014. p. 1620–1625. Disponível em: <http://dx.doi.org/10.1109/ITSC.2014.6957925>. Cited at page 27.

34 GENTNER, R.; CLASSEN, J. Development and evaluation of a low-cost sensor glove for assessment of human finger movements in neurophysiological settings. **Journal of Neuroscience Methods**, v. 178, n. 1, p. 138–147, 2009. ISSN 0165-0270. Disponível em: <https://doi.org/10.1016/j.jneumeth.2008.11.005>. Cited at page 27.

35 LISZEWSKI, A. **If You Don't Find This Video About Robot Wheelchairs For Babies Heartwarming, You Probably Don't Have a Soul**. Disponível em: <https://gizmodo.com/if-you-dont-find-this-video-about-robot-wheelchairs-for-5905966>. Cited at page 28.

36 HALLI, L. **Rolf Harris and his Wobbleboard**. Disponível em: <https://web.archive.org/web/20150905231346/http://www.nma.gov.au/___data/assets/pdf_file/0012/5430/FriendsMar09-Rolf-Harris.pdf>. Cited at page 28.

37 HOSSEINIRAVANDI, M. *et al.* Home-based telerehabilitation software systems for remote supervising: a systematic review. **Int J Technol Assess Health Care**, v. 36, n. 2, p. 113–125, 2020. Disponível em: <https://doi.org/10.1017/S0266462320000021>. Cited 2 times at pages: 28 and 29.

38 HASSAN, Q. F. **Internet of things A to Z: technologies and applications**. [S.l.]: John Wiley & Sons, 2018. Cited at page 29.

39 KRISHNAN, C.; WASHABAUGH, E. P.; SEETHARAMAN, Y. A low cost real-time motion tracking approach using webcam technology. **Journal of biomechanics**, Elsevier, v. 48, n. 3, p. 544–548, 2015. Disponível em: <https://doi.org/10.1016/j.jbiomech.2014.11.048>. Cited at page 30.

40 LI, X. *et al.* A survey on cloud-based video streaming services. Elsevier, 2021. Disponível em: <https://doi.org/10.1016/bs.adcom.2021.01.003>. Cited at page 30.

41 MARTINS, J. **Webcam motion detection and tracking-interfaces for immersive embodied experiences**. Disponível em: <https://welcome.isr.tecnico.ulisboa.pt/publications/webcam-motion-detection-and-tracking-interfaces-for-immersive-embodied-experiences/>. Cited at page 30.

42 REGO, P. A.; MOREIRA, P. M.; REIS, L. P. A serious games framework for health rehabilitation. **International Journal of Healthcare Information Systems and Informatics (IJHISI)**, IGI Global, v. 9, n. 3, p. 1–21, 2014. Cited at page 30.

43 SARDI, L.; IDRI, A.; FERNÁNDEZ-ALEMÁN, J. L. A systematic review of gamification in e-health. **Journal of biomedical informatics**, Elsevier, v. 71, p. 31–48, 2017. Disponível em: <https://doi.org/10.1016/j.jbi.2017.05.011>. Cited at page 30.

44 SEMMELMANN, K.; WEIGELT, S. Online webcam-based eye tracking in cognitive science: A first look. **Behavior Research Methods**, Springer, v. 50, n. 2, p. 451–465, 2018. Disponível em: <https://doi.org/10.3758/s13428-017-0913-7>. Cited at page 30.

45 SHAHRESTANI, S. **INTERNET OF THINGS AND SMART ENVIRONMENTS : ASSISTIVE TECHNOLOGIES FOR DISABILITY, DEMENTIA, AND AGING**. [S.l.]: Cham : Springer International Publishing : Imprint: Springer, 2017. Cited at page 30.

46 MATAMALA-GOMEZ, M. *et al.* The role of engagement in teleneurorehabilitation: a systematic review. **Frontiers in Neurology**, Frontiers, v. 11, p. 354, 2020. Disponível em: <https://dx.doi.org/10.3389/fneur.2020.00354>. Cited at page 30.

47 (N.D.), M. **Azure Kinect DK**. Disponível em: <https://www.oculus.com/>. Cited at page 30.

48 OCULOS. **Oculos Go**. Disponível em: <https://www.oculus.com/>. Cited at page 31.

49 MOHER, D. *et al.* Preferred reporting items for systematic reviews and meta-analyses: The prisma statement. **PLOS Medicine**, Public Library of Science, v. 6, n. 7, p. 1–6, 07 2009. Disponível em: <https://doi.org/10.1371/journal.pmed.1000097>. Cited at page 31.

50 POSTEL, J. **RFC 768: User Datagram Protocol**. 1980. Disponível em: <http://dx.doi.org/10.17487/RFC0768>. Cited at page 32.

51 POSTEL, J. **RFC 793: Transmission Control Protocol**. 1981. Disponível em: <http://dx.doi.org/10.17487/RFC0793>. Cited at page 32.

52 GOFF, D. P. T. Unified transport layer support for data striping and host mobility. **IEEE J. Select. Areas Commun**, v. 22, p. 737–746, 2003. Disponível em: <https://doi.org/10.1109/JSAC.2004.826001>. Cited at page 32.

53 FLOYD, K. F. S. Promoting the use of end-to-end congestion control in the internet. **IEEE/ACM Trans. Netw.**, IEEE Press, Piscataway, NJ, USA, v. 7, n. 4, p. 458–472, ago. 1999. ISSN 1063-6692. Disponível em: <http://dx.doi.org/10.1109/90.793002>. Cited at page 32.

54 Audio-Video Transport Working Group *et al.* **RFC 1889: RTP: A Transport Protocol for Real-Time Applications**. 1996. Status: PROPOSED STANDARD. Disponível em: <http://dx.doi.org/10.17487/RFC1889>. Cited 2 times at pages: 32 and 33.

55 SINGH, V. *et al.* **Considerations for Selecting RTP Control Protocol (RTCP) Extended Report (XR) Metrics for the WebRTC Statistics API**. RFC Editor, 2018. RFC 8451. (Request for Comments, 8451). Disponível em: <http://dx.doi.org/10.17487/RFC8451>. Cited at page 33.

56 ALVESTRAND, H. T. **Overview: Real-Time Protocols for Browser-Based Applications**. RFC Editor, 2021. RFC 8825. (Request for Comments, 8825). Disponível em: <http://dx.doi.org/10.17487/RFC8825>. Cited at page 34.

57 W3C. **W3C**. Disponível em: <https://www.w3.org/TR/webrtc/>. Cited at page 34.

58 OPENCV. **Thresholding Operations using inRange**. Disponível em: <https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html>. Cited at page 42.

59 JAIN, A. *et al.* A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students. **International Journal of Applied and Basic Medical Research**, Wolters Kluwer–Medknow Publications, v. 5, n. 2, p. 124, 2015. Disponível em: <https://doi.org/10.4103/2229-516x.157168>. Cited at page 57.