INSTITUTO POLITÉCNICO
DE BRAGANÇA
Escola Superior
de Tecnologia e Gestão

Integration of Heterogeneous Components for Dynamic Adaptation of Rehabilitation Simulations

Isaac Van-Deste Marcelino - 41754

Thesis presented to the School of Technology and Management in the scope of the Master in Informatics.

Supervisors:
Prof. Rui Pedro Sanches de Castro Lopes

Bragança
October, 2025

INSTITUTO POLITÉCNICO
DE BRAGANÇA
Escola Superior
de Tecnologia e Gestão

Integration of Heterogeneous Components for Dynamic Adaptation of Rehabilitation Simulations

Isaac Van-Deste Marcelino - 41754

Thesis presented to the School of Technology and Management in the scope of the Master in Informatics.

Supervisors:
Prof. Rui Pedro Sanches de Castro Lopes

Bragança
October, 2025

Dedication

To myself. In recognition of all the effort, perseverance, and countless hours devoted to the research, development, and writing of this work. This thesis stands as a testament to the dedication, passion, and resilience that I have invested throughout this journey, starting from my Bachelors degree. The person and professional I am today are, in part, the result of embracing this project with unwavering commitment and care, overcoming challenges, and pushing myself beyond limits to bring this work to be concretized. This dedication reflects, not only the technical and intellectual growth I have achieved, but also the personal journey of persistence, self-discovery, and determination that this thesis represents.

Acknowledgment

Firstly, I would like to express my deepest gratitude to my supervisor, Rui Pedro Lopes, for his tireless mentorship, guidance, and unwavering support throughout these years. His dedication, patience, and constant encouragement have been invaluable, not only in guiding my research but also in shaping me into the professional I am today. I am truly thankful for everything he has done for me, both academically and personally.

I would also like to extend a very special thanks to Júlio Castro Lopes for his tireless assistance and dedication over the past four years. Beyond his invaluable support in the development of this work, I am deeply grateful for the experience we shared, from collaborative research and discussions to turning our efforts into published articles and attending at conferences together. His guidance and partnership have made this journey productive and memorable.

I would like to express my gratitude to everyone involved in this project, as it is the result of a combination of efforts. Special thanks to João Vieira, Ana Sofia Rodrigues, Gonçalo Pinto, and Jecé Xavier for their contributions, which made this work possible.

I am also grateful to all my friends for their support throughout my Bachelors and Masters courses, for sharing countless hours of study, and for growing together with me during this journey.

Last but not least, I would like to express my deepest gratitude to my parents and my family for their unconditional love, support, and encouragement throughout my life. Their belief in me, guidance, and constant presence have been my foundation and a source of strength in every step of this journey. I am truly grateful for everything they have done to help me reach this point.

This work is funded by the European Regional Development Fund (ERDF) through the Regional Operational Program North 2020, within the scope of Project GreenHealth - Digital strategies in biological assets to improve well-being and promote green health, Norte-01-0145-FEDER-000042. This work has been supported by FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UIDB/05757/2020.

Abstract

This dissertation investigates the integration of digital health technologies, artificial intelligence, and immersive environments to design a rehabilitation system for individuals who suffers from schizophrenia disease. Schizophrenia is a multifaceted complex mental disorder, that often leads to loss of physical and cognitive skills, reducing social interaction, with a high relapse rates, that turns rehabilitation a long-term challenge. Virtual Reality has been emerging as a promising tool for rehabilitation, sice it provides a safe, controlled, and motivating therapeutic scenarios. Upon this, the present work introduces a VR-based serious game capable of dynamically adapting its difficulty in response to physiological and behavioural signals. To achieve this solution, several different data from the patient, such as heart rate, body motion rate, and facial expression recognition were continuously monitored and processed to infer the patient's stress levels in real time. The stress estimate is then integrated into a Dynamic Difficulty Adjustment mechanism, formulated in a Reinforcement Learning algorithm. A Deep Q-Network agent was trained within a simulated environment to learn optimal policies that balance challenge and engagement, ensuring that rehabilitation sessions remain neither frustrating nor trivial. Then, a distributed, event-driven architecture was designed to support the integration of heterogeneous modules, including the Virtual Reality application, a web dashboard, the dynamic difficulty system and a back office server, by using publish-subscribe communication, with Apache ActiveMQ Artemis, and containerized deployment, using Docker, to ensure scalability, modularity, and real-time interoperability. Experimental validation showed that the agent successfully learned to maintain stable levels of difficulty and stress, confirming the feasibility of adaptive rehabilitation driven by physiological signals. Overall, this dissertation demonstrates how the convergence of VR, distributed systems, and artificial intelligence can be harnessed to create adaptive rehabilitation tools that are not only technically robust but also clinically meaningful. The framework provides a foundation for future clinical studies, offering potential applications beyond schizophrenia, including in neurological rehabilitation, stress management, and personalized digital health therapies.

Keywords: Distributed Systems, Event-Driven Architecture, Microservice Architecture, Docker Deployment, Artificial Intelligence, Reinforcement Learning, Virtual Reality

Resumo

Esta dissertação investiga a integração de tecnologias de saúde digital, inteligência artificial e ambientes imersivos para projetar um sistema de reabilitação para indivíduos que sofrem de doença esquizofrênica. A esquizofrenia é um transtorno mental complexo multifacetado, que muitas vezes leva à perda de habilidades físicas e cognitivas, reduzindo a interação social, com altas taxas de recaída, tornando a reabilitação um desafio de longo prazo. A Realidade Virtual tem vindo a emergir como uma ferramenta promissora para a reabilitação, proporcionando cenários terapêuticos seguros, controlados e motivadores. Com isto, o presente trabalho apresenta um jogo sério, baseado em Realidade Virtual capaz de adaptar dinamicamente sua dificuldade, em resposta a sinais fisiológicos e comportamentais. Para alcançar essa solução, vários dados diferentes do paciente, como frequência cardíaca, taxa de movimento corporal e reconhecimento da expressão facial, foram continuamente monitorizados e processados para inferir os níveis de stress do paciente em tempo real. Esta estimativa de stress é então integrada num mecanismo de ajuste dinâmico de dificuldade, formulado com base num algoritmo de aprendizado por reforço. Um agente de uma rede Deep Q-Network foi treinado num ambiente simulado para aprender políticas ideais que equilibram o desafio e o envolvimento, garantindo que as sessões de reabilitação não permaneçam frustrantes ou triviais. Em seguida, uma arquitetura distribuída e orientada a eventos foi projetada para suportar a integração de módulos heterogéneos, incluindo a aplicação em Realidade Virtual, um dashboard web, o sistema de dificuldade dinâmica e um servidor de back office, usando comunicação publish-subscribe, usando o Apache ActiveMQ Artemis, e implantação em conatainers, usando o Docker, para garantir escalabilidade, modularidade e interoperabilidade em tempo real. A validação experimental mostrou que o agente aprendeu com sucesso a manter níveis estáveis de dificuldade e stress, confirmando a viabilidade da reabilitação adaptativa impulsionada por sinais fisiológicos. Em síntese, esta dissertação demonstra como a convergência da Realidade Virtual, sistemas distribuídos e Inteligência Artificial pode ser aproveitada para criar ferramentas de reabilitação adaptativas que não são apenas tecnicamente robustas, mas também clinicamente significativas. A estrutura fornece uma base para estudos clínicos futuros, oferecendo aplicações potenciais além da esquizofrenia, incluindo reabilitação neurológica, gerenciamento de stress e terapias de saúde digital personalizadas.

Palavras-chave: Sistemas Distribuídos, Arquitetura baseada em eventos, Arquiteturas por Microsserviços, Implementação em Docker, Inteligência Artificial, Aprendizagem por Reforço, Realidade Virtual

Contents

List of Tables

List of Figures

Acronyms

AI Artificial Intelligence.

AMQP Advance Message Queue Protocol.

ASD Autism Spectrum Disorder.

BMR Body Motion Rate.

CART Classification and Regression Tree.

CNN Convolutional Neural Networks.

DDA Dynamic Difficulty Adjustment.

DQN Deep Q-Network.

DRL Deep Reinforcement Learning.

ECG Electrocardiogram.

EGB Ensemble Gradient Boosting.

FER Facial Expression Recognition.

GAN Generative Adversarial Networks.

gRPC Google Remote Procedure Call.

GSR Galvanic Skin Response.

HAR Human Action Recognition.

HR Heart Rate.

HRV Heart Rate Variation.

IPC Interprocess Communication.

IQ Intelligent Quotient.

JMS Java Message Service.

JSON Javascript Object Notation.

LSTM Long Short Term Memory.

MDP Markov Decision Process.

MJD Moving Joint Descriptor.

ML Machine Learning.

MQTT Message Queuing Telemetry Transport.

MSE Mean Squared Error.

NB Naive Bayes.

NoSQL Not only SQL.

PNS Parasympathetic Nervous System.

PPO Proximal Policy Optimization.

REST Representational State Transfer.

RL Reinforcement Learning.

RMI Remote Method Invocation.

RPC Remote Procedure Call.

SNS Sympathetic Nervous System.

SOAP Simple Object Access Protocol.

STOMP Simple Text Oriented Messaging Protocol.

SUS System Usability Scale.

SVM Support Vector Machine.

TCP Transmission Control Protocol.

UDP User Datagram Protocol.

UI User Interface.

UX User Experience.

VR Virtual Reality.

XGB Extreme Gradient Boosting.

XML Extendible Markup Language.

Chapter 1

Introduction

This dissertation results from part of the work done in the scope of the GreenHealth project, in order to find digital technologies for innovative techniques of rehabilitation [1]. The main focus is the development of techniques of rehabilitation and maintain the mental health of people diagnosed with schizophrenia.

There are evidences that confirm that digital technologies can have positive benefits on rehabilitation by promoting physical activity using Virtual Reality (VR), wearables and mobile applications, since these type of devices can offer real-time data and personalized insights [2]. In the scope of this work, VR is used with a serious game as the main tool in the rehabilitation scenario. Games usually are designed for entertainment. Nevertheless, serious games are designed “for a primary purpose other than the pure entertainment” [3]. They can be used in vast fields such as rehabilitation, education, well-being, training, health care and many others [4]. VR has been proved that it is a safe tool for rehabilitation patients with schizophrenia, and it is more effective than the traditional sessions [5].

Schizophrenia is a mental illness that affects about 20 million people over the world, according to the World Health Organization [6]. It is characterized by a loss of contact with reality by changing the way of thinking, perception and behaviour in the patient’s environment.

The symptoms can be structured in different groups, being them positive, negative and cognitive as described in the figure 1.1. Positive symptoms included delirious, hallucinations, perturbations in the way of thinking and an unorganized behaviour. The negatives include alogia, isolation and social housing, abulia and apathy, affective planing and psychomotor slowing. The cognitive are related to lack of attention, memory loss, reduced Intelligent Quotient (IQ), language difficult and degradation of executive functions [7].

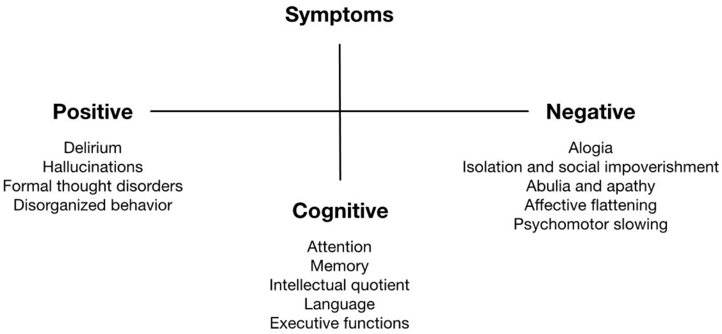


Figure 1.1: Schizophrenia symptoms as a multifaceted disease

Antipsychotic medication is the basis of treatment for the sick person. This form of treatment acts mainly on the positive symptoms of the disease, with a frank reduction of these symptoms and a reduction in the risk of relapse. Additionally, most available medicines only can be used on the positive symptoms and there are no medicine approved for the negative and cognitive symptoms. Thus, the main focus of this project will be applied on these type of symptoms.

1.1 Overall Architecture

The rehabilitation session consists of a VR serious game in which the patient is monitored throughout the session. It is intended that the game needs to meet several specific conditions, to be considered as safe and suitable for the patient. Biometric and physiological data from the patient, such as Heart Rate (HR), facial expression and body posture need to be analyzed, processed and adapted to the patient, based on Artificial Intelligence (AI) techniques. Also, data collection needs to be performed using non-intrusive mechanisms. For that, HR signals will be collected by a smartwatch, since is was considered safe by the medical team. A camera will also be used to assess patient's body posture and facial expressions. Finally, the data must be stored on a back-end server and needs to be shown to the caregivers with a front-end application. Figure 1.2 represents the overall architecture of rehabilitation process.

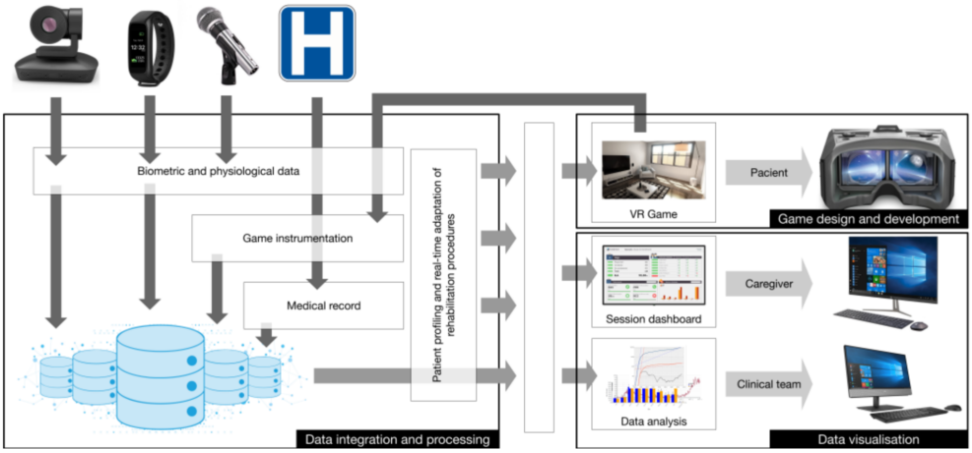


Figure 1.2: Overall architecture for rehabilitation [1]

1.2 Objectives

The work detailed in this thesis has the main objective to develop an architecture capable to integrate several distinct devices, being them, the VR application, the back-end server, the front-end and the data analyzed by AI. The game must be in accordance with the patient, so it cannot be easy nor difficult. In addition, traditional difficulties on games relies on an easy, medium and hard level, which cannot adapts to the patient [8]. For that, Dynamic Difficulty Adjustment (DDA) will be applied to the game, consisting of a Reinforcement Learning (RL) system, who analyze the data of the patient (VR, Heart Rate Variation (HRV), facial expressions and body posture) to infer the difficult of the game for each specific person (Figure 1.3).

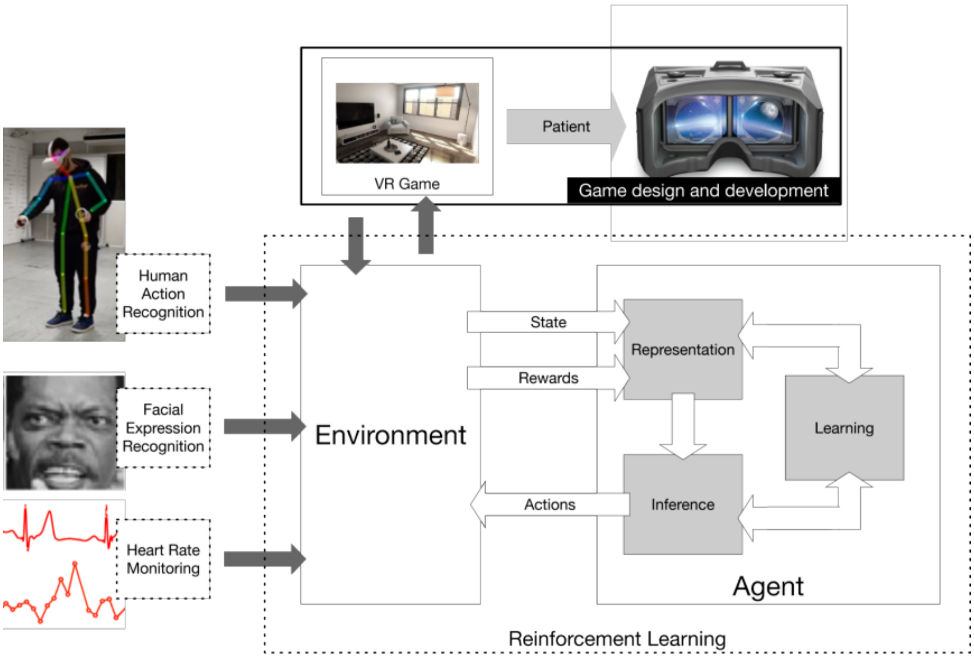


Figure 1.3: Integration System

1.3 Structure

This dissertation is structured into 6 chapters, as follows: Chapter 1 introduces this dissertation, focusing on the main purpose of this work; Chapter 2 focused on the overview of relevant studies in the field of VR, Rehabilitation and Serious Games, methodologies for integrating heterogeneous systems, and finally, relevant studies the Dynamic Difficulty Adaption method used; Chapter 3 details every component and shows the proposed architecture to integrate these heterogeneous components; Chapter 4 introduces the Dynamic Difficulty Adaption algorithm used in the rehabilitation scene; Chapter 5 explains the architecture for integrating all the components; Chapter 6 presents the conclusion of the thesis.

Chapter 2

Related Work

This chapter presents relevant approaches on several topics important for the development of this dissertation. Starting by VR and Rehabilitation, the objective is to find other works that used this type of technology in rehabilitation scenarios. Then, it will be discussed how heterogeneous systems are complex to integrate and what technologies exists that can help to establish communication between these systems. Finally, it will be explained the DDA and detailed some related work on this field.

2.1 Virtual Reality Applications in Rehabilitation

Bosch-Barceló et al. [9] evaluate the possibility of using a treadmill rehabilitation in a Gamified VR Environment for people with Parkinson's Disease. The simulation has three different walking scenarios (countryside, city, and park) with a difficulty adjustment based on five parameters, being them speed, visibility, path width, obstacles, and distractors. Visibility difficulty is handled by the appearance of fog or nightlight; obstacles include traffic cones, cardboard boxes and bricks, and the participant needs to avoid the obstacles executing a skip movement in the treadmill. For a better immersive scenario the path is filled with threes, buildings or fences, according to the scenario, to create a dense environment to the patient; the distractors are used based on the environment: in countryside, the patient sees animals moving on the scenario, on the city there are cars moving around and construction noised are heard and in the park, footballs can cross the path and birds can flight in front of the participant. To test the study, eight participants with Parkinson's Disease with ages between 45 and 80 years between Stage II and Stage III in the Hoehn&Yahr scale [10] having the ability to walk for ten or more minutes unassisted and having a minimum score of 24 in the Mini-Mental State Examination. To evaluate the overall performance the authors used usability methods such as System Usability Scale (SUS), Assistive Technology Usability Questionnaire for people with Neurological diseases (NATU Quest), Simulator Sickness Questionnaire (SSQ) and Independent Television Commission Sense of Presence Inventory (ITC-SOPI). With those metrics, this rehabilitation scenario proved to be effective in persons with Parkison's Disease, where the participants state that this rehabilitation tool improves their independence. SUS score has achieved a value of 74.82 ± 12.62 out of 100, and NATU Quest a score of 4.49 ± 0.62 out of 5. As future work the authors found it necessary to improve this type of scenario to more complex environments adapted to a specific person to guarantee a better training progression.

Martins et al. [11] introduced a serious game in VR for persons with Autism Spectrum Disorder (ASD). The game was introduced on a base game developed by the authors, named TravelTrain that consists of teaching ASD individuals to take buses to the city where the player needs to validate the ticket, sit down, click on the button to stop the bus, and other related activities. Three mini-games were developed to be integrated with the TravelTrain game. Those three minigames are:

1. Facial Expression Recognition (FER): the player learns how to identify facial expressions. This game consists of 6 components representing 6 different emotions (Happy, Anger, Disgust, Sad, Scare and Surprise) and one avatar is centered on the scenario. Then, two difficulties are inserted, the easier where the game presents the facial expression and the name of the that expression, and the hardest, that only shows the name of the expression. This difficulty levels help to accommodate different difficulties to learn how the patient identifies the emotions, which is crucial

for social interaction;

2. Facial expression manipulation: the player needs to manipulate the expression on an avatar to match the emotion on the proposed scenario. For that, the player needs to make adjustments on facial parts, such as mouth, lips, upper teeth, lower teeth, forehead, eyebrows, tongue, and eyes, with a horizontal bar where the player can interact to make those modifications. The difficulty levels are the same of the first game and the objective on this second game is to deepen the understanding of players and their proficiency in recognizing and adjusting the subtle complexities of facial expressions, which is crucial for effective emotional communication;

3. Facial expression production game: is similar to the second game, but the player needs to match the required emotion using his own face. The patient face was tracked using a face tracker device that connects to the upper lip and chin of users. Like the second game, this one has the same two difficulties. The objective of this game is to reinforce the association between emotional states and facial expressions.

To validate this hypothesis, the authors made a pilot test that include 10 participants, where 5 of them were people with ASD and the other 5 serve as control group. Several metrics were used. Quantitative metric recorded the time played on each session and the achieved score. Semi-Qualitative feedback was gathered by using a post-test questionnaire. Although the control group obtained better results than the group of the patients, there was a positive acceptance about the game usability and it is a promising tool to help people with ASD to improve their emotional capabilities.

Robitaille et al. [12] tested an immersive VR simulation with sensors to detect changes in the emotional state of persons, such as an HR sensor and a Galvanic Skin Response (GSR) connected to the index and middle fingers of the users non-dominant hand. They used an HTC Vive VR system with additional Vive Trackers strapped to the individual's waist. Twelve users, six men, and six women were recruited for the experimental session, which included four tasks: three calm scenes and one stressful scene, played randomly for each participant. During the day, the calm scene consisted of a virtual forest with the sound of birds, and the participants were asked to hit 50 butterflies. The stressed scene featured the same environment at night, but the participant was instructed to hit bees while surrounded by monsters and the sounds of wolves, screaming people, and bees. Those bees can be in one of two possible states, each represented by a different colour on the bug. When it is yellow, it is harmless; when it is red, it is angry. They calculated the speed of the hands during the session and segmented it into time intervals, which they refer to, as fast motions. Fast motions were found to be more prevalent when a participant was in the stress scene. Finally, they trained a decision tree with the maximum speed of the motion and the HR at the same moment, achieving an accuracy of 80% approximately. They also trained the decision tree with only the maximum speed, getting an accuracy of 70% approximately.

Raffe and Garcia [13] investigated the use of commercial skeletal tracking and VR technologies to improve gameplay interfaces in fall prevention exercise games for older adults. The study used the StepKinnection game as a reference and investigated the use of Microsoft Kinect for skeletal tracking and HTC Vive for head tracking and VR visualization. The authors carried out a self-reflective study to assess various avatar positioning modes for accurate stepping motions, physical comfort, and user engagement. While the study reveals promising opportunities for developing engaging step training games, it also identifies limitations, particularly in terms of accessibility, safety concerns, and technical challenges associated with the combined technologies. The paper emphasized the need for additional research into alternative interaction modalities to improve accessibility and engagement in game-based fall prevention training programs for the elderly.

Ishaque et al. [14] designed a system to analyze and evaluate stress under VR simulations. The authors acquired several vital signs, such as: Electrocardiogram (ECG), GSR and respiration signals. ECG and respiration signals were used to measure the impact of the simulations on the Sympathetic Nervous System (SNS) and the Parasympathetic Nervous System (PNS). An increase in the SNS leads to an increase in the HR and an increase in the PNS activity leads to relaxation, decreasing the HR. GSR was used to measure the changes in electrodermal activity, such as skin temperature and sweat activity.

In this study, fifteen subjects participated in the simulations, being 7 men and 8 women. For data acquisition, the procedure was divided into 4 phases, each one with duration between 5 and 7 minutes. The first one for control, where the subjects were asked to read a book or do nothing. The second and third stages, the subjects were playing a VR game of a rollercoaster to induce stress. In the final stage, the subjects played a fishing game in VR. ECG signals, GSR and respiration signals were collected during the procedure and used for stress evaluation. Several Machine Learning (ML) models were used, some of them for binary classification of stress and others to classify 5 different classes of stress from the rollercoaster VR game. For the binary classification, the Ensemble Gradient Boosting (EGB), Support Vector Machine (SVM), Naive Bayes (NB) and a personalized Classification and Regression Tree (CART) were used. For 5 class classification, a dataset of the VR game was developed using the data from the rollercoaster VR game. Extreme Gradient Boosting (XGB), Decision Trees were used for this classification. In both scenarios, the CART model showed to be the best model, achieving an accuracy of 87.75% for binary classification and 72.22% for 5 class classification.

Harrison et al. [15] studied the effectiveness of VR on anxiety evaluation, in female soccer players. Thirteen healthy players participated in the study, consisting of the perception of anxiety levels and the penalty kick performance. For data acquisition, the authors used two DELSYS Trigno Avanti accelerometer sensors, one placed at the fifth lumbar vertebrae and the second placed on the anterior thigh of the kicking leg. The HR of the players was captured using a Polar H10. The players used the *Oculus Quest* VR goggles with the Liminar VR app calm scene. The procedure was divided into three blocks, each one consisting of five penalty kicks for each player. The first block served as baseline data, the second as high-stress situation and the last one, was the intervention of the VR in the stress situation. The VR scene revealed positive effects in the reduction of the anxiety levels of the players, where all the participants said that the VR helped them to relax with 11 of them (out of 13) showing reductions in the anxiety levels.

Ma et al. [16] assessed the accuracy and validity of Mystic Isle, a rehabilitation game, using the Microsoft Kinect V2 sensor, for assessing motion in stroke patients'. In this study, the authors compared the data from the Kinect V2 with the Vicon system, the industry-standard optical motion capture system, for a range of full-body movements. The participants completed trials in sitting and standing positions, with the results showing high correlation coefficients and signal-to-noise ratios for arm joints, while hip joints showed less stability. Mystic Isle showed potential for clinical assessments and home-based rehabilitation, especially with the potential for remote monitoring and data collection, despite certain limitations, such as sample homogeneity and possible difficulties with stroke patients' movement patterns.

Lidstone et al. [17] presented a pilot study for automated and scalable computerized assessment of motor imitation in children with ASD using a single 2D camera. Motor imitation is the ability to observe and mimic the actions or movements of others. For children with ASD, it is a crucial skill for the development of social skills, forming social bonds, and observant learning. The goal of their research was to compare the OpenPose skeleton (2D method) to the well-established Kinect 3D and Human Observation Coding (HOC), evaluating the feasibility of using only a 2D camera. Significant correlations were found between OpenPose 2D, Kinect 3D, and HOC, indicating concurrent and construct validity, though the Kinect 3D CAMI method demonstrated superior discriminating ability. Furthermore, motor imitation scores from all methods were significantly associated with social-communication impairments in children with ASD. According to the study, OpenPose 2D provides a scalable and accessible method for assessing motor imitation in children with ASD, potentially facilitating therapeutic interventions and the development of serious games that target social and motor skills.

2.2 Integration of Heterogeneous Systems

Heterogeneity is one of the main challenges in the development of distributed systems. A distributed system is defined as hardware or software components that are located in a computer network and coordinate their actions only by parsing messages [18]. Nowadays, applications, programming languages, computer architectures and operating systems have increased a lot the heterogeneity of the systems and how complex they are. To abstract all this complexity and to offer common interfaces, it was necessary to create a middleware layer.

Coulouris et al. [18] state that in order to define a distributed system, it is necessary to define the entities that are communicating; how they communicate between themselves; what are the roles and responsibilities of each one and how they are mapped on the physical architecture. Communication is essential in this type of systems. With communication, applications can interchange information, ensure coordination and synchronization. According to the book, there are three major types of communication paradigms:

- interprocess communication;
- remote invocation;
- indirect communication.

Interprocess Communication (IPC) refers to the simplest and basic type of communication between processes. It may be synchronous or asynchronous, if the process that sends the message waits or no for a response, respectively. IPC methods includes sockets and message-passing primitives, and support for multicast communication. Sockets is composed by a network address and a number of port, and serve as an endpoint to send and receive data. Usually, sockets are over Transmission Control Protocol (TCP), but in some specific cases they can be over User Datagram Protocol (UDP). Message passing refers to exchanging structured messages through `send` and `receive` operations.

Remote invocation is a specific form of IPC where a process can invoke procedures or methods from a remote entity. Remote Procedure Call (RPC) and Remote Method Invocation (RMI) are two well-know examples of remote invocation. RPC, turns possible for a program to execute co-routines or procedures in the same computer or in a remote computer, by calling a remote procedure as if they are in the same local address space. The goal, is turning distrusted computing very similar to conventional programming. RMI extends the RPC to be used in a distributed platform of objects in applications developed in Java.

Indirect communication, as the name itself says, is a form of communication between entities through an intermediary, ensuring no direct coupling between the sender and the receivers. The existent uncoupling used by the intermediary have two different properties: space uncoupling, where the sender does not know or need to know the identity of the receiver(s) and time uncoupling, where the sender and the receiver(s) can have independent lifetimes.

Group communication, publish-subscribe systems, message queues and distributed shared memory are examples of indirect communication. Group communication refers to a paradigm where a message is sent to a group and the message is delivered to all members of the group (multicast communication). The sender does not know the identities of the receivers and processes may joint or leave a group.

Publish-subscribe systems, also known as distributed event-based system, is a type of system where publishers, the ones who sent messages, publishes structured events to an event service, usually a broker, that is responsible to send the messages to all the subscribers that ate interested (subscribed) to that event. This type of communication is used in heterogeneous systems, offering an asynchronously communication, since there is no need of synchronization between the two entities that are part of the data exchange, implying that the both publishers and subscribers are decoupled.

Message queues provides a point-to-point service by using a concept of message queue as form of indirect communication. Thus, producers sends messages to a specific queue that are received by the consumers. As the system uses as queue as an indirect communication, the queuing policy follows the first-in-first-out algorithm.

Distributed shared memory provides an abstraction of shared memory in distributed systems. It is used to share data between computers who do not share the same physical memory. When data is not locally available, then it is fetched from the distributed shared memory.

In distributed systems, it not enough to send messages. It is mandatory to define how the message is represented, so both entities can understand and can communication with each other, otherwise, the system will not work as expected.

Abstract representation corresponds to a common format in which each process can encode and decode information. With this abstraction, it is possible to hide the implementation details, specific for each operative system or programming language, benefiting a transparent communication. Thus, it is possible to serialize data to be transmitted over internet protocols, and then retrieved on the other side to be reconstructed again in the original format. This allows, portability and preserving on the data that is transmitted, being independently of the operative system or programming language.

Over the years, many data interchange formats have been developed to address these challenges. Examples include text-based formats such as Extendible Markup Language (XML) and Javascript Object Notation (JSON), as well as more efficient binary formats such as Protocol Buffers (Protobuf), Apache Avro, and many others.

XML was one of the earliest markup languages developed to structure and interchange data [19]. It follows a structured and hieratical format, using tags to define elements. Due to its highly flexibility, the language became popular in web services, forming the base of communication of the Simple Object Access Protocol (SOAP) that uses XML to send formatted messages. Despite its flexibility, the language was very criticized due to its verbosity and increased message size. This motivated the emergence of develop alternative formats to offer a more compact and efficient representation of data.

JSON has emerged as an alternative solution to XML [19]. This format introduces a text-based data format that can be easily read by humans and for machines to parse. It follows a key-value data representation, and due to its simplicity, it became the standard data interchange in modern distributed systems. JSON is very used in RESTful APIs and web application.

Protobuf is a open-source data format serialization platform developed by Google[1]. It was designed to be smaller and faster than XML. Data schemas and services are defined in a ".proto" file that serve as a contract between the entities that will interchange data. With

---

[1] https://developers.google.com/protocol-buffers/

this, the client knows how to create a message and how to send it, and the server knows how to receive that message and return a response. The definition file can be compiled with `protoc`, generating the necessary structures that are invoked by both client and server. The compiled code can be in several languages, such as C++, Python, Java, Go, and many more programming languages.

Apache Avro is a data serialization framework developed in the Apache Hadopp ecosystem [20]. The schema is defined in JSON, although, they are converted to a binary format, being extremely efficient and compact. It is compatible with several programming languages, such as Java, Python, C, C++, PHP, Ruby, Rust, JavaScript and Perl. Avro is well-know for its usability in Apache Kafka, an event streaming platform developed by Apache, that uses publish-subscribe systems to offer communication between servers and clients.

2.3 Dynamic Difficulty Adjustment

DDA refers to a set of techniques in game design and artificial intelligence that automatically adjust the challenge of a game to match the player's current abilities, preferences, or state. Instead of relying on static "easy," "normal," or "hard" modes, DDA continuously monitors player performance and engagement indicators such as success rates, reaction times, mistakes, or even physiological signals and uses this information to adapt gameplay elements in real time. The goal is to keep players in the so-called "flow zone,", as shown in Figure 2.1, where the game is neither too easy (leading to boredom) nor too hard (leading to frustration).

This adaptive mechanism can take many forms: changing enemy behaviour, adjusting resource availability, modifying puzzle complexity, or even altering narrative pacing. By tailoring the experience dynamically, DDA seeks to maximize enjoyment, learning, and immersion while also accommodating a broader range of player skills. It has been applied not only in entertainment games but also in educational and serious games, where keeping learners motivated is essential. In essence, DDA personalizes the game experience, making

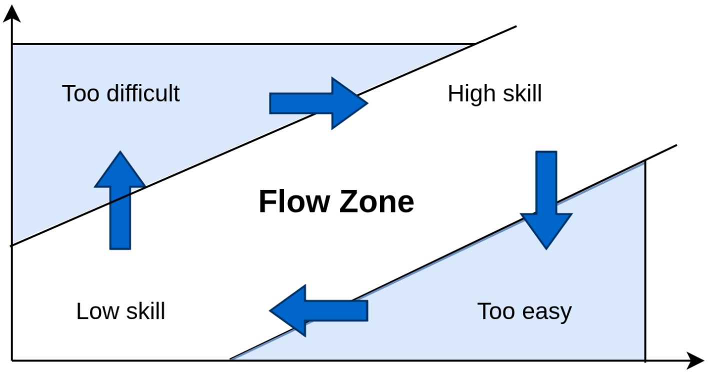


Figure 2.1: Flow zone

it more responsive, inclusive, and engaging for diverse players.

2.3.1 Reinforcement Learning

RL, a subfield of AI, has attracted a lot of interest since it can be used to teach machines how to learn from their environment and how to make decisions [21], [22]. An artificial agent that interacts with a given environment learns the best behaviour to use by attempting to maximize some concept of cumulative reward [23]. By receiving feedback in the form of rewards or penalties based on its actions, the RL algorithm enables an agent to learn optimal strategies. RL was several applications [24], ranging from games [25], [26], resource management [27], personalized recommendations [28], and robotics [29].

To define a RL environment, it is necessary to have this concpets in mind:

- **Agent**: the decision-maker. The agent interacts with the environment, receives feedback, and aims to maximize the cumulative reward by learning an optimal policy;
- **Observation space**: the information available to the agent at each step. It can be discrete or continuous, depending on the problem setup;
- **Environment**: the system in which the agent operates. The environment receives the agent's action, updates its internal state, and returns a new observation and a reward;

- **Action space**: defines all the set of possible actions that the agent can chose;
- **Policy**: the strategy that the agent follows to select actions. It maps observations to actions, indicating what action to take in a given state;
- **Reward function**: the feedback signal received from the environment. It tells the agent whether an action was beneficial or not for achieving the objective. This is a critical component, as it guides the learning process and defines how the agent is penalized or rewarded.

The RL has its principles in the Markov Decision Process (MDP), where a learning agent must be able to sense the state of its environment to some extent and must be able to take actions that affect the state. Also, the agent must have a goal or goals relating to the state of the environment. The MDP include these three aspects, sensation, action and goal. So, the observation space defines the current state s, the agents possible adjustments in game difficulty form the action space a, and the reward function R guides learning towards maintaining optimal stress levels. This framework allows the RL agent to continuously interact with the environment, receive feedback based on the players physiological and emotional signals, and improve its policy π over time. Figure 2.2 illustrates this interaction loop, highlighting the flow of states, actions, and rewards that underpin the decision-making process.

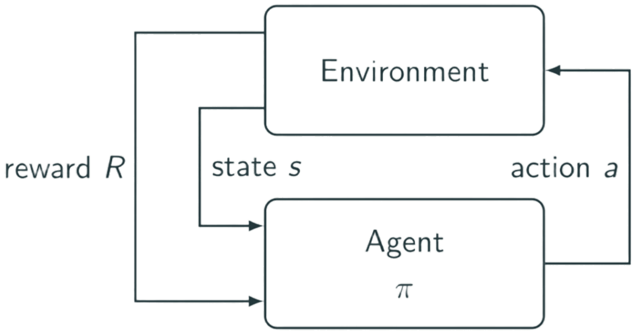


Figure 2.2: Markov Decision Process representation of Reinforcement Learning

2.3.2 Reinforcement Learning for DDA

DDA can be done solely based on game instrumentation (how the player is performing in the game, for example, if he/she is completing a task in the expected time), although it was concluded that in some applications physiological information is a critical factor to dynamically adapt the game [30]. This is mainly applicable in rehabilitation games, where the focus of the game is not on entertainment, but in rehabilitation [31].

DDA with RL algorithms can be applied in different types of applications, not only games for rehabilitation or only for diversion, but is also applied in an educational field, such as mathematical tools [32], [33]. Although its primary applications focus on its inclusion in games (serious and non-serious) [34], [35], providing user satisfaction and also enabling skill development and personalization, resulting in a more immersive and enjoyable overall experience.

Kamikokuryo et al. paper [36] focuses on the application of RL techniques, specifically the Multi-Armed Bandit algorithm, for DDA in VR rehabilitation. The RL approach is used as a decision-making framework, where the distance metric derived from the latent space representation is employed as a reward. The Multi-Armed Bandit algorithm, specifically the Boltzmann and Sibling Kalman filters, is used to learn the distance evolution in the latent space and adjust the game parameters accordingly. The findings show that the RL technique is effective in quickly adjusting game difficulty based on the patient's motor control abilities, resulting in a personalized and engaging rehabilitation experience. The "arms" in the Multi-Armed Bandit represent various actions or choices that the agent can make, for physical rehabilitation procedure in VR. The actions in this case correspond to the outcome of the movements or tasks in the VR game, which correspond to one of six pre-established movements: square, cylinder, heart, infinity, sphere and triangle. As expected, the difficulty of drawing these shapes in the air changes, and should be adapted to the player/patient skills. The latent space represents the patient's current effort in drawing the shape (hand motion), and it is used to compute a distance measure to the reference shape. This measure serves as the reward in the Multi-Armed Bandit algorithm: the lower the distance, the more accurate the movement of the patient. In this situation, this means that the patient is doing well and the difficulty should be increased to a more difficult shape (action suggested by the RL algorithm). This information is then used by the MAB algorithm to update the agent's internal knowledge and influence the action selection. In this paper, the RL environment refers to the VR rehabilitation environment in which the Multi-Armed Bandit algorithm is used to dynamically adjust the game difficulty based on the patient's motor control abilities.

Cheruki and Glavin's work [37] presented a novel application of their Skilled Experience Catalogue technique for DDA in fighting games using RL. An RL agent's policy is stored during training at regular intervals and milestones are used to match the opponent's skill level. By adapting the agent's skill level, the RL agent successfully balances two out of three fixed-strategy opponents. However, the technique is limited by the upper skill level attainable by the RL algorithm used. As part of future work, they plan to explore advanced RL algorithms, such as Deep Reinforcement Learning (DRL), to enhance the skill cap and improve game balance. The available moves or strategies that the RL agent can choose during gameplay are referred to as actions, which, is this case, correspond to the movement and actions in the game, in a total of 7. The specific actions available to the agent are determined by the agent's energy level and the game environment. The state space for the FightingICE game is a complex data structure that includes all the game information, such as energy, (x,y) position, speed, and damage, among many others. The reward is a signal that the RL agent receives in response to its actions and how they affect the game. Several reward and penalty conditions are mentioned in the paper, including rewards for damage done to the opponent and penalties for damage received, energy wasted, or missed attacks. The environment is the FightingICE platform, a Java-based fighting game framework used for the paper's experiments.

Rajabi et al. [38] proposed a method for automatically generating well-balanced levels in 2D platformer games using Generative Adversarial Networks (GAN). The levels are randomly generated and evaluated for solvability using RL. The GAN is trained to generate new levels that aim to balance requirements specified by the level designer. The approach generates balanced levels with an accuracy of 83.6%, providing a potential solution to the challenge of creating balanced games. The actions in the game include moving right, moving left, and jumping, and the game's state includes the player's current position, the positions of platforms, coins, the starting point, and the ending point, from the game screen image. It also indicates whether the player has collected coins or reached the end of the level. The reward system is intended to motivate the agent to achieve specific goals such as earning coins or reaching the end, which results in a positive score or reward for the agent. Failure to complete the game or falling off a platform results in a negative score. The environment includes the game's map, which is made up of platforms, coins, a starting point, and an ending point that provides feedback (rewards) to the agent based on its actions and current state.

Biemer and Cooper [39] presented a method for generating game levels that adapt to player performance and preferences. The authors formulate the level generation problem as a MDP and solve the MDP using Adaptive Dynamic Programming. There are four directors (the directors aim at adjusting the game to ensure the desired player experience [40]) introduced: Random, Greedy, Policy Iteration, and Adaptive Policy Iteration. When player proxies change, the results show that Adaptive Policy Iteration outperforms the other directors in terms of reward and adaptability. The director, which is the agent, makes decisions on how to assemble levels using the MDP described in the paper. Since they are dealing with two distinct case studies in this paper, the states in Case Study 1 with Mario N-Grams represent different prior-level slices. The states in Case Study 2 with Icarus Segment Generation represent level segments. Regarding the rewards, they mention the use of custom reward functions that represent designer preference and player enjoyment. The MDP's rewards are initialized by a custom reward function, and there is also a dynamic reward function that takes into account designer and player preference, as well as how many times the player has played a given state. The environment refers to the level generation process and the MDP is used as a framework to model and generate levels. The directors interact with the MDP to assemble levels based on the defined actions, states, and rewards. The paper concludes with a discussion of the challenges and potential future directions for this approach, which includes DDA and multi-objective optimization.

Reis et al. [41] proposed a DDA-based methodology for achieving automatic video game balance. The balance task is designed as a meta-game, with actions within the meta-game changing the game rules. The authors used an RL technique in which an agent acts as a game master and learns the best policy by playing a meta-game. In addition, the authors propose a multi-agent system training model for the game master agent, in which it competes against multiple agent opponents with varying behaviours and skill levels. The outcomes demonstrate the game master's decision-making as well as the preliminary development of a framework for automatic game balance. The proposed methodology demonstrates the capability of RL in achieving game balance and emphasizes the significance of game adaptation in improving gameplay engagement and enjoyment. The game master agent, which assumes the role of an opponent in the game, performs actions in the game environment to make the maze more difficult for the player. The players are allowed to move up, down, left, and right. The master agent perceives the meta-state space as a standard environment so that it can infer the skills of the player and act based on it. The rewards are used to guide the learning process and encourage the game master to meet the game's balance objectives; the specific reward function is designed by the game designer and is mentioned in the formulation of the game adaptation problem. In the meta-game environment, the game master agent interacts with the base game and other player agents to learn the best strategy for game adaptation.

Rahim et al. [42] defined a RL algorithm for dynamic difficulty adjustment in a Visual Working Memory game. Unlike most prior studies, which treat difficulty as a discrete choice between a few predefined levels, this approach defines difficulty as a continuous variable that combines the number of targets, their connectedness, and spatial distribution. By doing so, the system avoids the rigid, step-based adjustments of earlier methods and instead adapts game challenge smoothly in real time. The novelty lies in introducing a continuous stateaction space for DDA and training it with Proximal Policy Optimization (PPO), allowing the model to generalize across a vast set of possible tasks. In contrast to rule-based methods or discrete RL solutions, this continuous framework provides more nuanced personalization and better maintains player engagement. The environment is a Visual Working Memory game where players memorize a hexagonal grid with highlighted targets and then recall them by clicking on the correct cells. Performance is scored as the ratio of correct to total targets. Task difficulty is modelled continuously using three factors: the number of targets, the number of connected components, and their spatial distribution. This continuous metric enables a nearly infinite range of task variations. The RL formulation defines the state as the previous difficulty and score, the action as choosing the next difficulty value, and the reward as how close the players accuracy remains to the optimal 85% success rate. PPO was trained first with simulated agents and then fine-tuned with human players. In experiments with 52 participants, the RL-based DDA significantly improved scores, win rates, and subjective experience compared to rule-based baselines, demonstrating that continuous RL can provide more balanced and engaging gameplay.

Table 2.1, presents a summary of the action space, observation space and rewards of the papers described in this section.

Table 2.1: Environments

Author	**Action Space**	**Observation Space**	**Rewards**
Kamikokuryo et al. [36]	discrete (6)	continuous ($\mathbb{R}^2$)	continuous (low reward for low difference to the expected hand movement)
Cheruki and Glavin's [37]	discrete (7)	frame and game data (data structure with Character, Attack, HitArea information)	continuous - damage dealt or taken and energy spent
Rajabi et al. [38]	discrete (3)	frame and game data (15x8 pixels)	continuous ($\mathbb{R}^3$) (collect coins and complete the level)
Biemer and Cooper [39]	discrete (3)	frame and game data (image)	continuous (collect coins and complete the level)
Reis et al. [41]	discrete (4)	frame and game data (image)	continuous (reaching the end of the maze, navigating through free spaces and avoiding walls)
Rahim et al. [42]	continuous	frame and game data (6x6 hexagonal grid)	continuous (maximized when player accuracy stays close to 85%)

2.4 Summary

In this chapter, it was explored the three key domains of this dissertation, VR in rehabilitation, the integration of heterogeneous systems and the definition and explanation of DDA methods.

First, research on VR demonstrates the potential of immersive realities to support diverse patients groups. Studies shows efficiency on diseases such as the Parkinson's disease and ASD. Also, other important scenario was included, the fall prevention, which is very useful on more elderly people. Overall, VR proved effective in improving independence, emotional recognition, stress assessment, motor imitation, and engagement in therapeutic scenarios.

Second, the integration of heterogeneous systems was discussed as a fundamental requirement on distributed application. Middleware and communication paradigms were reviewed, presenting s first approach on how existent implementation can solve the proposed problem on this dissertation.

Finally, it was introduced the RL and DDA, and how these techniques can continuously adapts the game challenge based on player performance or physiological states. RL provides a robust framework for implementing DDA, with applications that ranges rehabilitation, education and diversion.

Chapter 3

Towards DDA: Data and Models

To move towards the DDA, it is necessary to define all the data needed to use in that system. Stress levels during rehabilitation are a extremely useful indicators, by providing the patient physiological response, but also engagement, workload and emotional state. The difficulty must be calculated in order to reach an optimal zone. If the difficulty is too low, patients may experience boredoom and disengagement. On the other side, if the game is too difficult, it can lead to frustration and even worse, to a more difficult recover. To achieve this, stress must be estimated from multiple physiological data. This chapter explores the research made to understand different types of data, and by using this data together can improve the construction of a better stress inference model.

3.1 Human Action Recognition

With Human Action Recognition (HAR), it is possible to automatically detect the human action on different situations [43]. HAR can be performed trough the appliance of skeleton extraction methods in RGB images, or through image segmentation techniques. For the purpose of the thesis, only a skeleton-based approach was studied, with the intuition of isolate the body joints and proceed the calculation of the Body Motion Rate (BMR), comparing three skeleton extraction algorithms, OpenPose, BlazePose and YOLO-Pose [44].

OpenPose is currently one of the most popular skeleton extraction algorithms. After processing each frame, the main keypoints, that make the body image skeletons, are obtained. Each skeleton is represented by 18 points in a 2D coordinate system, where each coordinate is the pixel position of a joint (keypoint) in the frame [45]. Due to OpenPose's deep network structure, it can identify more than one person in the frame by identifying anatomical location, the body points, and then joining those points using affinity fields.

YOLO-Pose is also a 2D algorithm based on the YOLO object detection framework [46]. The detected skeleton is represented by 17 keypoints, each one having a pair of 2D coordinates and a confidence value. For this work, version 7 of YOLO-Pose was used.

BlazePose is a real-time inference 3D skeleton technique designed by Google for mobile devices [47]. The model presents high-fidelity body posture tracking, by using RGB video frames, to infer 33 3D landmarks and a background segmentation mask for the entire body. The first 11 landmarks are used for the face, and the other 22 points are used for the body.

Figure 3.1 shows the skeleton representation of the three algorithms.

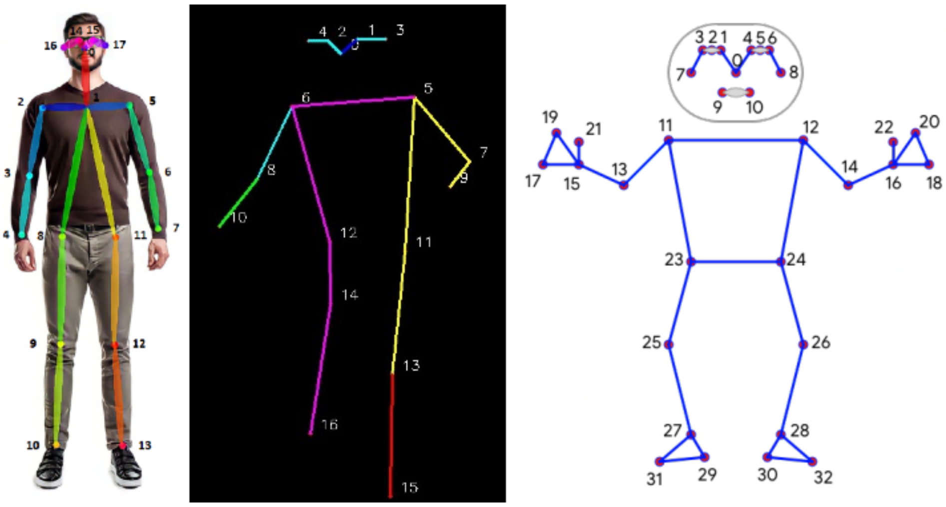


Figure 3.1: OpenPose, YOLO-Pose and BlazePose skeleton representations

As mentioned above, only skeleton extraction methods were used to infer HAR. For that, several methods were discussed to evaluate which method is better to use [44]. A dataset was chosen, and the three skeleton algorithms described before were tested using two different methods on each one. Figure 3.2 represents the evaluation methodology.

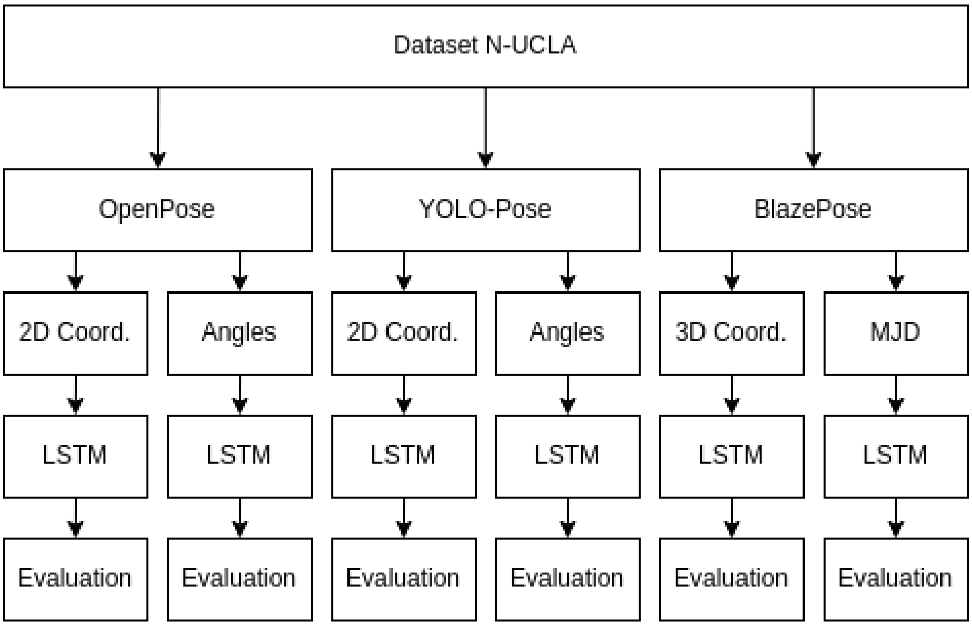


Figure 3.2: Evaluation methodology

The first technique (Coordinates) normalizes the 2D coordinates extracted by OpenPose and YOLO-Pose. The second (Angles) computes the angles of each joint's segment when determining the angles between joints. The third method makes use of the Moving Joint Descriptor (MJD) [48]. All the algorithms that were discussed employed the coordinates of every skeletal joint of its respective original skeleton. It is important to acknowledge that not every feature extraction technique discussed in this section was utilized for every skeleton algorithm. Since the coordinates are already normalized in BlazePose, keypoint normalization was only performed in the OpenPose and YOLO-Pose algorithms. Since the 3D formulation would be significantly more complicated, the calculation of the angles between the joints was restricted to the 2D techniques. BlazePose normalized coordinates are the sole sets of data to which the MJD has been applied because this descriptor depends on 3D data for classification.

3.1.1 Coordination Normalization

OpenPose algorithm outputs 18 joints and YOLO-Pose outputs 17 joints. Image coordinates are used to represent both joints. Finding the lowest and highest values of x and y in each frame can help normalize them. Every real coordinate can be normalized with these four points. A standing static individual is used to show this reference in Figure 3.3.

Figure 3.3: Representation of the rectangle referential for 2D skeleton normalization.

Equation 3.1 is used to compute the new coordinates. For each joint:

$$(new_x, new_y) = (\frac{x - min_x}{max_x - min_x}, \frac{y - min_y}{max_y - min_y}) \tag{3.1}$$

where:

min_x - Minimum x of all joints.

min_y - Minimum y of all joints.

max_x - Maximum x of all joints.

max_y - Maximum y of all joints.

3.1.2 Angles Between Joints

One possible method of deriving significant information from a human movement is through the angles that the skeleton creates. For this purpose, the primary joints in the skeleton are represented by 12 defined angles for OpenPose and 8 formed angles for YOLO-Pose. Each angle is the result of a set of three vertices. All the keypoints are presented in Figure 3.1.

For the OpenPose angle computation, the following set of joints was used: 1 - (0, 1, 2); 2 - (1, 2, 3); 3 - (2, 3, 4); 4 - (0, 1, 5); 5 - (1, 5, 6); 6 - (5, 6, 7); 7 - (0, 1, 8); 8 - (1, 8, 9); 9 - (8, 9, 10); 10 - (0, 1, 11); 11 - (1, 11, 12); 12 - (11, 12, 13).

For YOLO-Pose, the following set of joints was used: 1 - (6, 8, 10); 2 - (6, 12, 14); 3 - (12, 14, 16); 4 - (5, 6, 8); 5 - (6, 5, 7); 6 - (5, 7, 9); 7 - (5, 11, 13); 8 - (11, 13, 15).

As a next step, the three points were used to calculate the angles formed by each vertex sequence. In order to achieve this, two new corresponding vertex definitions were generated:

Transformed Vertex 1:

$$newvertex_1x = vertex_2[x] - vertex_1[x] \tag{3.2}$$

$$newvertex_1y = vertex_2[y] - vertex_1[y] \tag{3.3}$$

Transformed Vertex 2:

$$newvertex_2x = vertex_3[x] - vertex_2[x] \tag{3.4}$$

$$newvertex_2y = vertex_3[y] - vertex_2[y] \tag{3.5}$$

Each skeleton segment is represented by a vector, and NumPy's function *arccos* is used to calculate the angle between vectors (Figure 3.4).

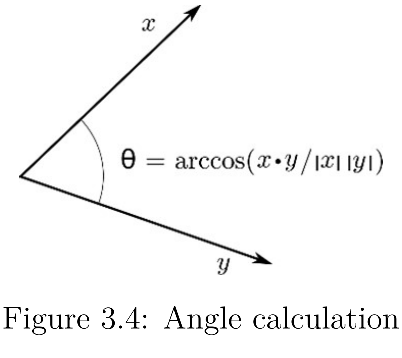


Figure 3.4: Angle calculation

3.1.3 Moving Joint Descriptor

The methodology employed by the MJD bears some changes to the work of Kamel et al. [48]. Out of the 20 joints, they have only utilized 13, and the BlazePose algorithm's original 33 joints were used in this work. Moreover, they employed a central critical point, the hip joint, that BlazePose is unable to identify. This was addressed by utilizing the middle point between the closest joints to generate a hip point estimator (Figure 3.5). The point centered on the hip is one of the most stable regions of the human body, so it makes sense to choose it as the origin of the spherical coordinates. Based on this, each subsequent joint will have two angles and the distance from the reference point.

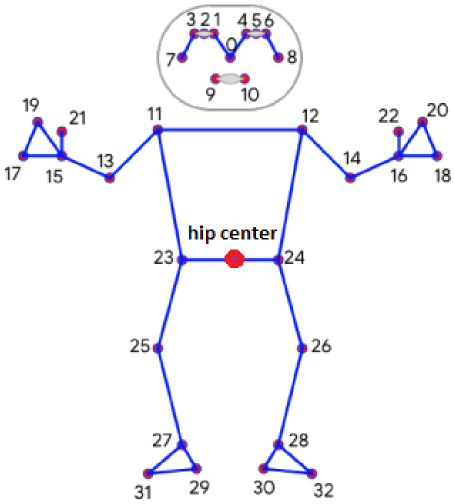


Figure 3.5: Hip Center Point - BlazePose Skeleton

The z coordinate can be traduced in a boost on the detected human skeleton, as it also brings depth into account. The real coordinates of every joint are automatically normalized by BlazePose's method, therefore the coordinates were converted back to the real coordinates. After this restoration procedure, Equation 3.6 was used to convert the data into spherical coordinates.

$$(r, \theta, \phi) = (\sqrt{x^2 + y^2 + z^2}, \arccos{(z/r)}, \arctan 2(y, x)) \tag{3.6}$$

3.1.4 Results

The N-UCLA dataset [49] is composed of ten action categories: picking up with one hand, picking up with two hands, dropping trash, walking around, sitting down, standing up, donning, doffing, throwing, and carrying. Each action was performed by ten actors. This dataset has the particularity that most videos are short, containing just the number of frames from the beginning of the action to the end.

A Long Short Term Memory (LSTM) was selected based on the literature review and its sequence-handling capabilities. The model was used to categorize a given sequence, identifying it as a single entity among the ten actions in the N-UCLA dataset. Various hidden layer sizes, namely 48, 100, and 300, were tested. Nonetheless, these values turned out to be the optimal answer.

Table 3.1 shows the F1 score of all the scenarios tested, being the Angles technique on OpenPose model the best one for classification of HAR, achieving a F1 score of 74.5%.

Table 3.1: Results on N-UCLA dataset (F1 score) [44]

	OpenPose		YOLO-Pose		BlazePose	
LSTM	2D Coord.	Angles	2D Coord.	Angles	3D Coord.	MJD
48	0.733	0.745	0.718	0.735	0.584	0.476
100	0.722	0.711	0.715	0.724	0.536	0.511
300	0.711	0.703	0.678	0.661	0.557	0.444

This table provides a base for the Body Motion Rate (BMR) calculation, which is being explained later in this chapter.

3.2 Facial Expression Recognition

Human face plays a crucial role to understand the human emotions. Facial expressions reflect the person's internal emotions that can be understandable without verbalization [50], [51]. Facial emotions include happiness, sadness, anger, fear, surprise, disgust among others. Since all individuals are different, the same emotion can trigger different emotional states on a specific person, resulting in a different reaction of that emotional on the individual's face.

For this work specially, FER tries to infer facial expressions under VR googles, which occlude the eyes. Nevertheless, the eyes have been demonstrated to carry significant information for emotion recognition, playing a pivotal role in conveying affective states. Previous studies have shown that even when other facial features are hidden or obstructed, the eyes provide critical cues for accurately identifying emotions such as happiness, sadness, anger, and surprise. This highlights the challenge of FER in VR environments, where the occlusion of key facial regions can reduce recognition accuracy, yet emphasizes the enduring importance of the eyes in understanding human affect [52].

All the work in FER field was developed by Rodrigues et al. [50], [51], [52], [53], [54]. Their research begins by identifying the strong relationship between facial expressions and emotions, and the individual differences in emotional expression. They then developed a ML system capable of detecting seven different human emotions in real time [50]. Subsequently, they demonstrated the importance of the eyes in facial classification and showed how occlusion can decrease the accuracy of ML models [52]. By using the FER+ dataset, they developed an occlusion algorithm, by drawing a rectangle in the eye area to simulate the presence of VR goggles (Figure 3.6).

Figure 3.6: FER+ dataset with occlusion algorithm [51]

Finally, their study was extended to VR simulations, where they proposed a system to classify the seven emotions into three groups (positive, neutral, and negative) under a simulated VR environment (Figure 3.7).

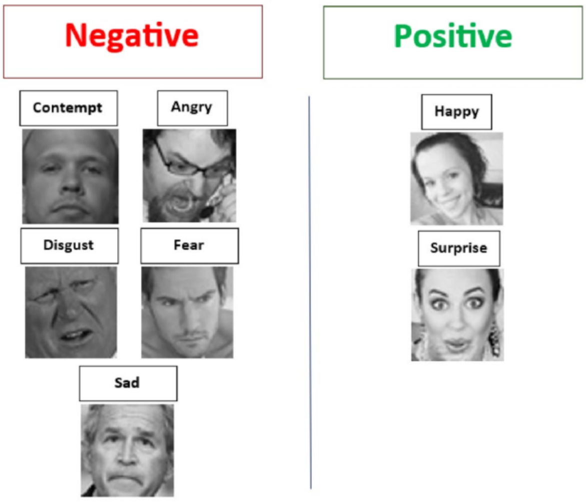


Figure 3.7: FER grouping into negative and positive groups [54]

To address the challenge of occlusion, they employed different Convolutional Neural Networks (CNN) architectures and produced ensemble models combining the three most accurate networks. The best-performing model achieved an accuracy of 85.7% without occlusion and 82.7% with occlusion [51].

3.3 Heart Rate

HR provides essential information regarding stress, and HR measurements enable the monitoring of its temporal evolution. To facilitate HR acquisition, an automatic annotator was developed to label HR sequences within a web-based application [55].

Data is collected through an an `Apple Watch SE`, using the HeartMonitor application [1]. This application produces a CSV file with the measurement of the HR, that was used to create a dataset [56]. By the end of the collection of data, the HR dataset achieved a format of 3 columns: `start_time`, `end_time` and `BPM`. The sampling varies between, approximately, 2.5 and 3.5 seconds. Two cameras (smartphone cameras) were used to record the video, later used for BMR calculation, yielding a total of 25 videos. To ensure that BMR and HR are synchronized (sharing the same timestamp), a script was developed

---

[1] (https://zachsim.one/projects/heartmonitor)

to insert a subtitle in each recorded video indicating the current time (hours, minutes and seconds), in the format `(HH:mm:ss,SSS)->Date:HH:mm:ss,ms`.

However, HR measurement alone is insufficient to infer stress. Factors such as physical activity, dehydration, and emotional states can all elevate HR, making it impossible to determine whether the individual is experiencing stress solely based on HR. For that reason, it is also necessary to consider some of these variables to exclude situations of elevated HR that are not associated to stress. In this context, the most influencing variable is the physical activity that directly influences the HR. In other words, intense physical activity leads to elevated HR just like light activity or sedentary should lead do low HR.

3.4 Body Motion Rate

BMR is an index created to measure the motion of the body of a person [56]. The higher the number, the faster and wider the movement. BMR is calculated based on the angle variation per unit of time (angular speed), due to be the better HAR technique as described before.

In other words, BMR can be calculated by summing the absolute value of the key-point's angle variation in pose estimation. Thus, the formula is obtained by adding the differences of all the angles (12 for OpenPose, 8 for YOLO-Pose) between two pictures in the sequence of movements per unit of time (Equation 3.7). In this equation, the superscript (i) corresponds to the frame i, and the $t^{(i)}$ corresponds to the time registered in the frame i.

$$BMR^{(i)} = \frac{\sum_{n=1}^{num_{angles}} \hat{a_n}^{(i)} - \hat{a_n}^{(i-1)}}{t^{(i)} - t^{(i-1)}} \tag{3.7}$$

As mentioned above, the increase in the HR can be the result of several factors, such as stress, dehydration or physical activity. It is possible to infer stress from increasing

HR as long as the other factors are excluded. For example, high HR and high BMR may mean that the person is performing a physical activity. On the other hand, high HR and low BMR may be a sign of stress. In this sense, to be able to infer stress, it is necessary to capture both the HR and the BMR. BMR and HR monitoring information will be incorporated into the stress inference model (figure 3.8) simultaneously, aiming to produce a model that, through the integration of this information, will estimate the stress level by outputting a value. In this context, synchronizing the timestamps of the smartwatch and camera is crucial to the applications capability to analyze each individual moment [57].

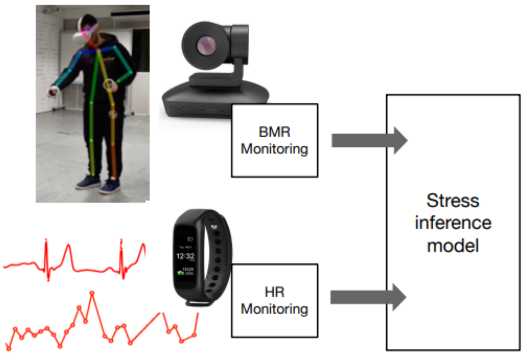


Figure 3.8: Heart Rate and Body Motion Rate as parameters for the Stress Inference Model [56], [57]

3.5 Summary

This chapter explored the data fource and models required to estimate stress for the DDA system. Three main physiological (HR) and behavioural (BMR and FER) signals were identified as essential inputs. Together, these multimodal signals provide a more accurate representation of the users state than any single metric alone.

HAR was studied in order to extract skeleton-base features to proceed for the BMR calculation. Differente feature extraction methods were explored, such as coordinate normalization, angles between joints and moving joint descriptor for three different skeleton algorithm: Open Pose, BlazePose and YOLO-Pose.

Facial expression was adapted for VR environments, playing a crucial role on understanding human emotions. Facial emotions were grouped in three goups: negative, neutral and positive, and algorithms with occlusion were studied to see the impact of the eyes in FER. The best result was achieved, by using ensemble models, l achieved an accuracy of 85.7% without occlusion and 82.7% with occlusion.

HR data was collected via smartwatch, providing a non-intrusive way of collection and monitoring cardiovascular activity. HR alone is insufficient to infer stress, since it can be elevated by physical activity or other factors. Thus, the combination of HR with BMR is crucial: high HR with low BMR suggests stress, while high HR with high BMR reflects physical effort.

Finally, the BMR was a index defined mathematically as the angular variation of skeleton joints over time, quantifying movement intensity. The more higher the index, more wider and faster the movement is. Together with HR, it forms a basis to build a stress inference model.

Chapter 4

Dynamic Difficulty Adjustment

As referred in chapter 2, DDA aims to adjust, dynamically, a difficulty of a game. The main objective is to match the best gameplay, keeping the engagement, and maintain an optimal level of challenge to each specific player. Thus, instead of having the classic difficulty levels, such as easy, medium and hard, the difficulty is computed in real-time according to the player. DDA can be achieved by using RL algorithms, where an agent interacts with a given environment and learns the best behaviour to use, by attempting to maximizing some concept of reward. When receiving feedback from the environment in the form of penalties or rewards, the algorithm enables the agent to learn the best optimal strategy.

4.1 Environment

The use of RL for DDA depends on the training of an agent that responds to an environment in the sense of lowering the difficulty level in case of increasing stress and increasing the difficulty otherwise. For that, it is necessary to create an environment that responds to the increase/decrease actions and provide a measurement of the physiological signals as the state.

In other words, the main idea of the RL algorithm is that the agent is able to infer the person's stress by using three physiological metrics (HR [55], BMR [56], [57] and FER

[50], [51]). With this, the proposed algorithm will be applied as a DDA mechanism, on a controlled scenario, for the rehabilitation of patients who suffer from schizophrenia.

As discussed in chapter 2, to setup the RL algorithm, it is necessary to define the agent, the observation space, the environment, the action space, the policy and the reward function.

4.1.1 Observation Space

The input parameters, which form the observation space, are HR, BMR and FER. HR and BMR are continuous variables, as they represent physiological measurements sampled over time. In contrast, facial expression is treated as a discrete variable, with values of -1, 0, and 1 corresponding to negative, neutral, and positive emotional states, respectively.

With this observation space, the agent aims to infer the patient's stress level. As said in chapter 3, conceptually, a high HR combined with a high BMR does not necessarily indicate elevated stress, since it may simply reflect physical activity. Similarly, low HR and low BMR may represent a state of rest or relaxation. However, a high HR combined with a low BMR can be a strong indicator of psychological stress, and this is interpreted by the agent as a sign of elevated stress. Facial expression data acts as an emotional modulator in this process, providing additional reinforcement that influences the stress inference. Based on these combined signals, a stress score is computed to quantify the player's stress level. If the agent fails to reduce the stress by appropriately lowering the game's difficulty over a certain period, the session should be paused, and the player needs to calm down with the help of the occupational therapist who is supervising the rehabilitation session.

4.1.2 Action Space

The action space (that defines the policy) corresponds to what the agent should do by interpreting the state of the environment at some point of time. This action should be done in order to adapt the difficulty to the player's current performance, to maintain an optimal level on the rehabilitation process and support, also, the player's engagement. With this, it was developed an action space with discrete values of -1, 0 and 1 corresponding to decrease, maintain and increase the difficulty of the game, respectively, during the session.

Stress is calculated based on the observation state (HR, BMR and FER), and it is expressed based in an interval score of 0 to 5.

Table 4.1 describes all the variables involved in the formulation of the observation space and action space structure of the rehabilitation game. Continuous metrics, such as HR and BMR, capture real-time physiological activity, while discrete metrics, such as facial expressions, stress scores, and difficulty levels, represent categorical or ordinal aspects of the player's state and game configuration. The action space defines the discrete set of adjustments the agent can apply to the games difficulty.

Table 4.1: Metrics types

Metric	**Variable Type**	**Range**
Heart Rate	Continuous	$[40; 200]$
Body Motion Rate	Continuous	$[0; 1000]$
Facial Expressions	Discrete	$\{-1, 0, 1\}$
Difficulty	Discrete	$\{1, 2, 3, 4, 5\}$
Action Space	Discrete	$\{-1, 0, 1\}$
Stress	Discrete	$\{0, 1, 2, 3, 4, 5\}$

4.1.3 Policy

The policy corresponds to the strategy made by the agent, when taking an action a for a given state s (Equation 4.1).

$$\pi(s) = a \tag{4.1}$$

Nevertheless, the goal of RL is to the policy that maximize the reward. Since that it is possible to have many possible policies, the main idea is to find the optimal policy π^* that selects the action a that leads to be best return expected. The optimal policy is represented in equation 4.2.

$$\pi^*(s) = \arg\max_a Q^*(s,a) \tag{4.2}$$

which means that, for each state s, the agent chooses the action a that maximizes the optimal action-value function $Q^*(s,a)$.

The optimal action-value function is defined recursively through the Bellman optimal equation [58]. It follows an intuition, that if it is possible to know all the possible actions $a^{'}$ for every state after action a $(s^{'})$, then the optimal strategy is to select the best action $a^{'}$ that gives the maximum expected value (Equation 4.3).

$$Q(s,a) = R(s) + \gamma\,\mathbb{E}\left[\max_{a'} Q(s',a')\right] \tag{4.3}$$

4.1.4 Agent

The agent is who is responsible to learn an optimal policy that maps states to actions. For this work, a Deep Q-Network (DQN) was chosen as the agent, since is one of the most widely used reinforcement learning algorithms, especially in discrete action spaces, being simple and computationally efficient. Also, DQN has proven to be reliable with discrete actions spaces.

The network architecture was kept simple to ensure stability and computational efficiency. Two intermediate dimension hidden layers with 64 units each, allow capturing non-linear relationships without incurring excessive overfitting or calculation costs. The output corresponds to the Q-values of each action, and the policy is defined as the action with the maximum Q-value. Q-value corresponds to a measure of the quality of taking a certain action s in a certain state s.

Formally, the Q-value function is defined as shown in equation 4.4 [58].

$$Q(s,a;\theta) \approx \mathrm{DQN}(s) \tag{4.4}$$

where θ are the network parameters updated by minimizing the Bellman error.

4.1.5 Reward function

The reward is the feedback or reinforcement signal received by the RL agent after performing an action in a specific state. The reward in a rehabilitation game can be based on a variety of factors, including the user's performance in completing the rehabilitation task or the achievement of therapeutic goals. The reward is used to assess how far the user has progressed in the rehabilitation process. As such, the reward function is designed to reflect the agents ability to maintain the player's stress within an acceptable range while maintains an optimal difficulty during game-play. The reward was defined as a sum of five distinct components. All these components work together to achieve the same objective, which is, prioritize the stress of the player, by moderate the difficulty of the game. More arrangements were done in order to avoid unnecessary change of actions and reward consistency of the signals. In addition, these results of the five terms were the best from 20 different training sessions.

As a result, the 5 components of the reward function were named Stress Term, Difficulty term, Action Cost, Physiological Signals and Similarity. These terms are explained bellow.

Stress Term

The stress component, denoted as R_{stress}, was designed with the objective of keeping the patient's stress level within an optimal range. Maintaining engagement during rehabilitation requires a balance: if stress is too low, the patient may lose motivation and not be fully engaged with the activity, otherwise, if stress is too high, the patient may experience discomfort or even adverse effects, which are undesirable.

Since stress is represented as a discrete variable within the range $[0, 5]$, the interval $[2, 3]$ was defined as the optimal zone. If stress falls within this interval, the reward encourages the agent to maintain the current state. Conversely, if stress is outside the optimal range, the agent is penalized unless it takes the corrective action. Thus, R_{stress} is defined as shown in equation 4.5.

$$R_{\text{stress}}(stress, action) = \begin{cases} +3.0, & \text{if } 2 \leq stress \leq 3 \ \wedge \ action = \text{maintain} \\ +2.0, & \text{if } 2 \leq stress \leq 3 \ \wedge \ action \neq \text{maintain} \\ +3.0, & \text{if } stress > 3 \ \wedge \ action = \text{decrease} \\ -8.0, & \text{if } stress > 3 \ \wedge \ action \neq \text{decrease} \\ +3.0, & \text{if } stress < 2 \ \wedge \ action = \text{increase} \\ -8.0, & \text{if } stress < 2 \ \wedge \ action \neq \text{increase} \end{cases} \tag{4.5}$$

Difficulty Term

Defined as $R_{difficulty}$, the objective is to maintain a moderate difficulty level during the rehabilitation session. If the difficulty is too low, the game-play becomes too easy and does not challenge the patient, which may reduce engagement. On the other hand, if the difficulty is too high, it can cause frustration, fatigue, or even risk of injury. Difficulty is represented in a discrete interval with range $[1, 5]$. The ideal interval was defined between $[2, 4]$, promoting engagement while ensuring safety and comfort. Equation 4.6 demonstrates the reward of $R_{difficulty}$.

$$R_{\text{difficulty}}(difficulty) = \begin{cases} +3.0, & \text{if } 2 \leq difficulty \leq 4 \\ -3.0, & \text{if } difficulty = 5 \\ -7.0, & \text{if } difficulty = 1 \\ -1.0, & \text{otherwise} \end{cases} \tag{4.6}$$

Action Cost Term

Defined as R_{action}, the objective is to reduce the impatience of the agent to change their action, when it do not needs to. The rule is give a reward of +0.5 if the action chosen was "maintain" and stress is in a controlled interval, not too low, neither too high. In both cases, reward is penalized. Equation 4.7 defines the behaviour of the R_{action}.

$$R_{action}(action, stress, difficulty) = \begin{cases} +1,0, & \text{if } action = \text{maintain } \wedge \; 2 \leq stress \leq 3 \\ -5,0, & \text{if } action = \text{increase } \wedge difficulty = 5 \\ -2.0, & \text{if } action = \text{increase } \wedge difficulty = 4 \\ -0.5, & \text{if } action = \text{decrease } \wedge difficulty = 1 \\ +0.2, & \text{if } action = \text{decrease } \wedge difficulty = 5 \\ 0.0, & \text{otherwise} \end{cases} \tag{4.7}$$

Physiological Signals Term

Defined as R_{phys}, this component integrates the normalized physiological signals (HR, BMR and FER) in order to align the reward with the patients physiological state. Both HR and BMR are normalized to the range $[0, 1]$, while FER is already discrete in the interval of $\{-1, 0, 1\}$. A linear combination of these signals, with specific weights (α, β, γ), defines their contribution to the reward. The goal is to penalize values outside the healthy or expected ranges and reward coherent physiological behavior during the rehabilitation task, as shown in equation 4.8.

$$R_{\text{phys}} = \alpha \cdot HR_{norm} + \beta \cdot BMR_{norm} + \gamma \cdot FER_{norm} \tag{4.8}$$

where:

$$HR_{norm} = \frac{HR - HR_{min}}{HR_{max} - HR_{min}}, \quad BMR_{norm} = \frac{BMR - BMR_{min}}{BMR_{max} - BMR_{min}}$$

α, β, γ were defined heuristically, based on experimental tests. Several simulations with different combinations and chose the values that resulted in stable rewards and consistent agent behaviors. This way, α was given a value of -0.024. This negative value penalizes a high HR. β was given a value of 0.0366, a positive value that rewards movement. γ was defined with a positive small value of 0.04, since it is a emotional stabilizer. Neutral or positive emotions are rewarded and negative emotions are penalized.

The most important thing was to maintain relative coherence: penalize HR when it is high, reward high BMR, and adjust with FER. Small variations of these weights do not qualitatively alter the behavior, but these concrete values were those that gave better stability to the R_{phys} term.

Similarity Term

The Similarity Term R_{sim} plays a crucial role in the agent learning. The main idea behind this, is to have the physiological signals coherent with each other. When HR and BMR are coherent, it must be rewarded. It does not matter if the values are both high or low, since who decides if it is a good or bad for the learning of the agent are the other terms of the reward. Otherwise, if HR and BMR are incoherent, it must be penalized. This Similarity Term measure how close are these signals.

All starts with the three physiological terms normalized in the interval of $[-1, 1]$. Then, it is calculated the similarity between two signals, as shown in equation 4.9. If a and b are equal, $sim(a, b)$ is 1, meaning maximum coherence. On other hand, the more difference exists between a and b, the more incoherence exists, leading $sim(a, b)$ to 0.

$$sim(a,b) = 1 - \frac{|a-b|}{2} \tag{4.9}$$

With the similarity defined, it is calculated the instant similarity S_{inst} as defined in equation 4.10.

$$S_{\text{inst}} = \frac{sim(HR, BMR) + sim(HR, FER) + sim(BMR, FER)}{3} \tag{4.10}$$

To have a more robust system, it was considered also the recent history of similarities, which is calculated by the same way, described in equation 4.11. A windows of 10 values was used in this calculation.

$$S_{\text{hist}} = \frac{sim(\overline{HR}, \overline{BMR}) + sim(\overline{HR}, \overline{FER}) + sim(\overline{BMR}, \overline{FER})}{3} \tag{4.11}$$

Finally, the R_{sim} is calculated by a linear combination of the similarities with specific weights (ω_{inst} and ω_{hist}), respectively for the instant similarity and historic similarity. Equation 4.12 shows the calculation for R_{sim}.

$$R_{\text{sim}} = \omega_{\text{inst}} \cdot S_{\text{inst}} + \omega_{\text{hist}} \cdot S_{\text{hist}} \tag{4.12}$$

To ω_{inst} was given a value of 0.6 and ω_{hist} a value of 0.4. This choice gives more importance to the instant similarity, allowing the agent to react quickly to sudden changes in the signals, while still taking into account the short-term consistency of the patient through the historical similarity. In this way, the system is both reactive and robust against noise.

With this, the Reward function R is the sum of all terms, as shown in equation 4.13.

$$R = R_{stress} + R_{difficulty} + R_{action} + R_{phys} + R_{sim} \tag{4.13}$$

4.2 Training of the Agent

An initial training of the agent was made in order to validate the environment setups. The DQN model was developed using the PyTorch framework. The training has a duration of 10000 episodes. The exploration policy was used, with epsilon (ϵ) starting a 1, decaying gradually until it reaches a minimum of 0.05. The target network is updated every 50 iterations, which helps stabilize the learning process. In addition, Adam was used as an optimization function with a learning rate of 0.001, a replay buffer of 10000 transitions and sampling batches of size 1024. A discount factor gamma (γ) is set to 0.99, making the algorithm more impatient to get a faster reward. For the loss function, it was chosen the Huber Loss, since it is the recommended for this model[1]. The Huber

---

[1] https://docs.pytorch.org/docs/stable/generated/torch.nn.HuberLoss.html

Loss takes the advantages of the Mean Squared Error (MSE) loss and the L1Loss. When the error is small, the loss behaves more likely the MSE (L2 Loss), providing a smooth quadratic penalization that facilitates optimization through gradient descent. In other hand, for larger errors, it behaves like a L1Loss, becoming more sensitive to outliers. This combination allows the Huber Loss to benefit from the smoothness of L2 near zero while maintaining the robustness of L1 for large deviations.

The replay buffer, also known as Replay Memory, has a crucial role in the training of the DQN model. It stores the transitions that the agent observes, that can be reused during training. This prevents the model to put to much correlation in the data. With this, the transitions that build up a batch are decorrelated. Replay Memory has been proved that it stabilizes and improves the DQN training procedure.

Data of the observation space (HR, BMR and FER) was artificially generated. HR and BMR were generated using a normal distribution, with a mean of 60 and variance of 5 for HR and a mean of 100 and a variance of 20 for BMR. FER was generated differently. As an emotional modulator, FER plays a crucial role in order to infer stress more precisely. Thus, FER is derived from HR and BMR through a probabilistic model. Specifically, when HR is high and BMR is low, FER is more likely to indicate negative emotions, reflecting elevated stress levels under inactivity. Conversely, moderate HR combined with low BMR suggests a balanced probability between neutral and negative emotions, indicating moderate stress. In cases where BMR is very high, FER tends to shift towards positive emotions, which may be associated with intense physical activity rather than stress. Finally, under low or normal BMR, FER is more likely to indicate a neutral state, consistent with light activity or rest.

In order to obtain better results on FER generation, it was labelled as a categorical variable conditioned on the physiological signals (Equation 4.14) :

$$\begin{aligned} \text{FER} \mid HR, BMR &\sim \text{Cat}\Big(p_{-1}(HR, BMR), p_0(HR, BMR), p_1(HR, BMR)\Big), \\ \text{FER} &\in \{-1, 0, 1\}. \end{aligned} \tag{4.14}$$

with the probability vector defined piecewise by the observed regime, as shown in equation 4.15:

$$\big(p_{-1}, p_0, p_1\big) = \begin{cases} (0.6,\ 0.3,\ 0.1), & \text{if } HR > 110 \ \wedge \ BMR < 0.25, \\ (0.4,\ 0.4,\ 0.2), & \text{if } HR > 95 \ \wedge \ BMR < 0.25, \\ (0.1,\ 0.4,\ 0.5), & \text{if } BMR > 1.5, \\ (0.2,\ 0.6,\ 0.2), & \text{otherwise.} \end{cases} \tag{4.15}$$

Sampling then proceeds as

$$\text{FER} \sim \text{Categorical}\Big(\big(p_{-1}, p_0, p_1\big)\Big)\ ,$$

where $-1, 0$, and 1 denote negative, neutral, and positive facial emotions, respectively.

The results were saved in a Data Frame with columns for reward (Figure 4.1), average stress (Figure 4.2), average difficulty (Figure 4.3), number of actions decreased, maintained and increased for each episode (Figure 4.4).

By observation the figures, it is clear that the model can learn the policy, as the reward is climbing up during the episode session. The reward starts to stabilize between 1500 and 2000 episodes, showing convergence of the agent of the agent. This stabilization is important, due to many failed attempts on training several agents had several oscillations in the reward formula, showing a non-convergence of the agent.

Stress started in a high level (4 out of 5), demonstrating some struggle in the beginning of the training session. With time, the agent learns to reduce the stress and maintain it in an optimal range, between the 2 and 4 values, as it was expected when defined the reward policy.

The difficulty also started high, maintaining an ideal difficulty between the average of 2.2 to 2.4, which matches with the conception made before, being the optimal range between 2 and 4, out of 5. The behaviour of the agent is correct, avoiding difficulties that are too low (1) or too high (5).

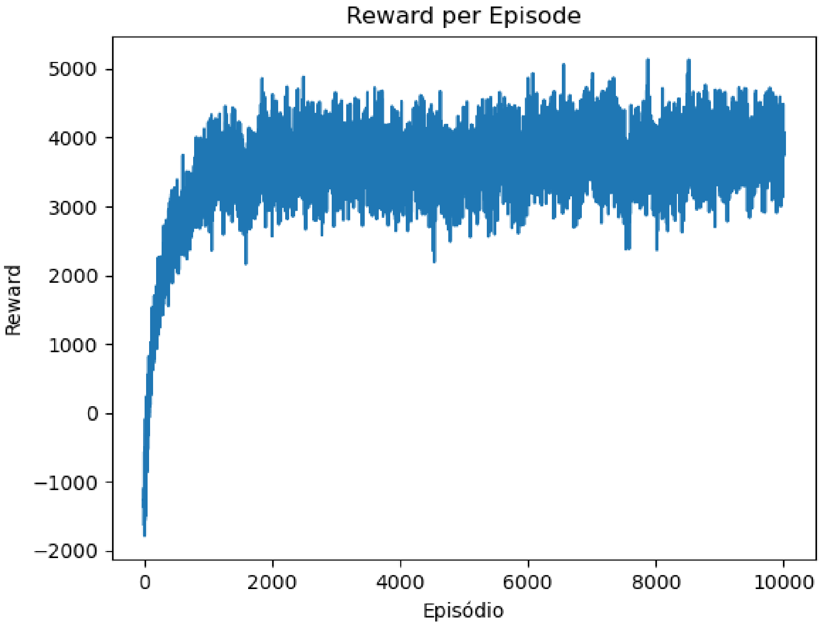


Figure 4.1: Reward during training

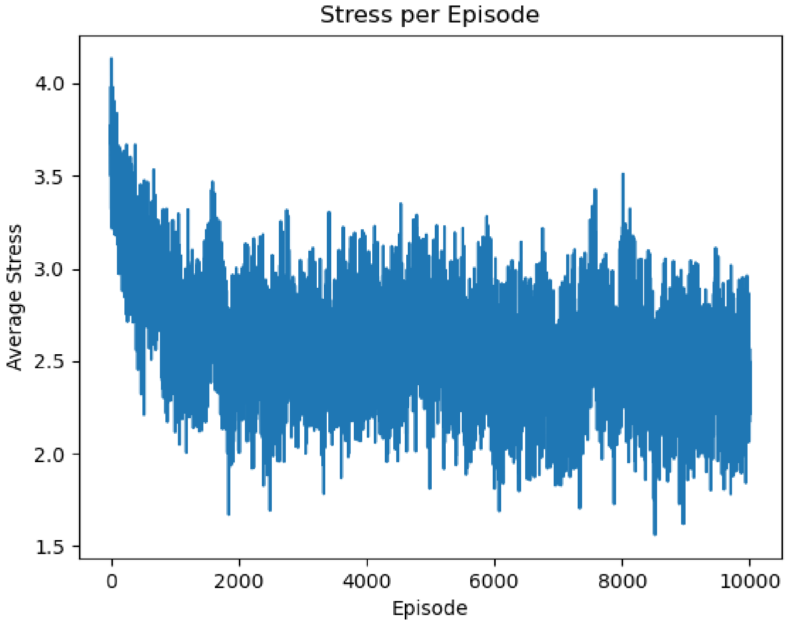


Figure 4.2: Stress during training

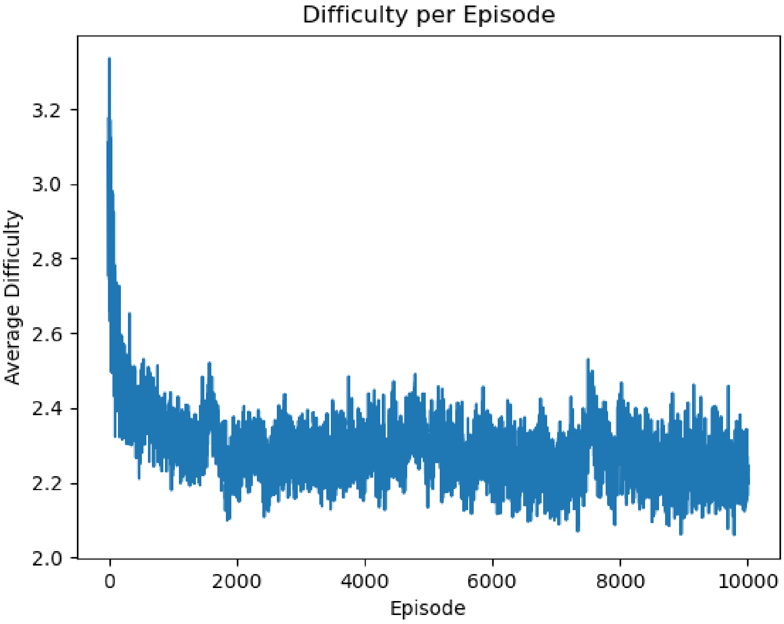


Figure 4.3: Difficulty during training

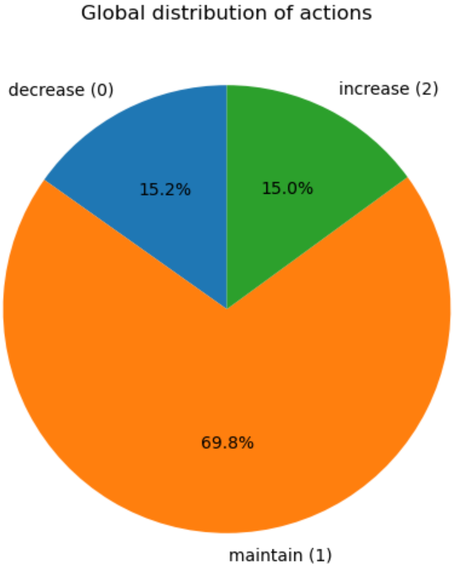


Figure 4.4: Actions during training

With the action pie chart, it is possible to see that the agent prefers to maintain the difficulty of the game, avoiding oscillations, by increasing or decreasing the difficulty without need. Having an almost equal distribution in the increase and decrease actions (almost 20%), shows that the agent only increases or decreases the difficulty of the game only when necessary.

With this, it can be concluded that the agent follows a good policy, maintaining a good interval of stress and difficulty. The metrics are not to high neither too low, ensuring that the agent learns the policy in a correct way. In addition, it should be noted that the agent learns with synthetic data, following a normal distribution. As so, it is expected that the are no outliers in the training process, giving the agent the capability to expore better decision scenarios. Nevertheless, it is still needed to validate the agent, causing some interference in the agent, in order to understand how it behaves under abnormal conditions.

4.3 Validation

Although the RL algorithm reaches a positive result, it is still necessary to validate the conception made before. The main idea in this approach is to send data to the environment, to cause interference and evaluate how the agent responds.

To develop this scenario, it is mandatory to have the environment running, and something that cause some type of interference in real-time. For that, RPC was used, due to its simply implementation. Specifically, gRPC, an open-source RPC library made by Google [59], was used. This library uses Protobuf as abstract data definition, making use of the "proto" file to specify the structure of the messages and methods. The contract created is then understandable on both client and server, so they can establish a communication.

The message defined is a simple request with two double values, being them HR and BMR values. Then, it was necessary to create a server to listen for requests from the client and also capable to make a change in the environment. As shown in Figure 4.5 the client simply sends the defined message as a change request to the server, which then implements the change through environment interference.

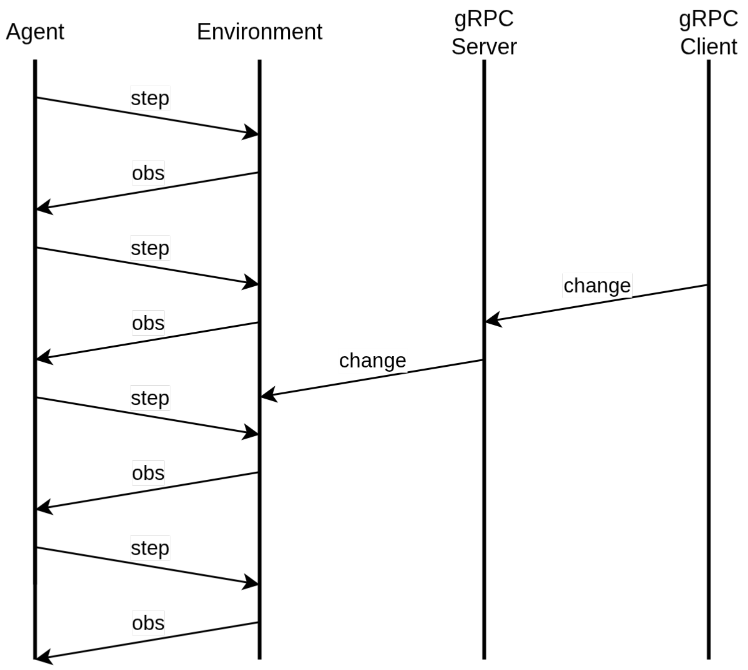


Figure 4.5: Stress during training

In an abstract form, the agent is continuously interacting with the environment. When no interference is received, physiological data is computed by using the normal distribution. In contrast, when a change request is received by the server, it injects that message into the environment, using those values in the next stepping of the agent.

To enable real-time data visualization, it was employed a Grafana dashboard connected to InfluxDB, a time-series database. For simplicity and ease of configuration, both Grafana and InfluxDB were employed in a persistence Docker container, exposing the services to the local machine through port forwarding.

The overall system architecture is built with Python co-routines, with the gRPC server listening for requests from the Google Remote Procedure Call (gRPC) client and propagate the changes to the environment. The RL agent runs continuously in another co-routine, and whenever the environment state is modified, the agent determines and executes an appropriate action. Subsequently, environment-related data, including the observation space, game difficulty, and stress level, are registered into InfluxDB, with another co-routine, and made available for real-time monitoring and analysis in Grafana. Data sent to Grafana includes the observation space (HR, BMR and FER) and additional info, corresponding to the stress and the difficulty.

The complete architecture for this validation is described in Figure 4.6. Each dashed ellipse corresponds to a co-routine in python. Both gRPC and the agent needs to interacts with the environment. The environment writes to the InfluxDB, through the InfluxDB writer co-routine, after every step of the environment.

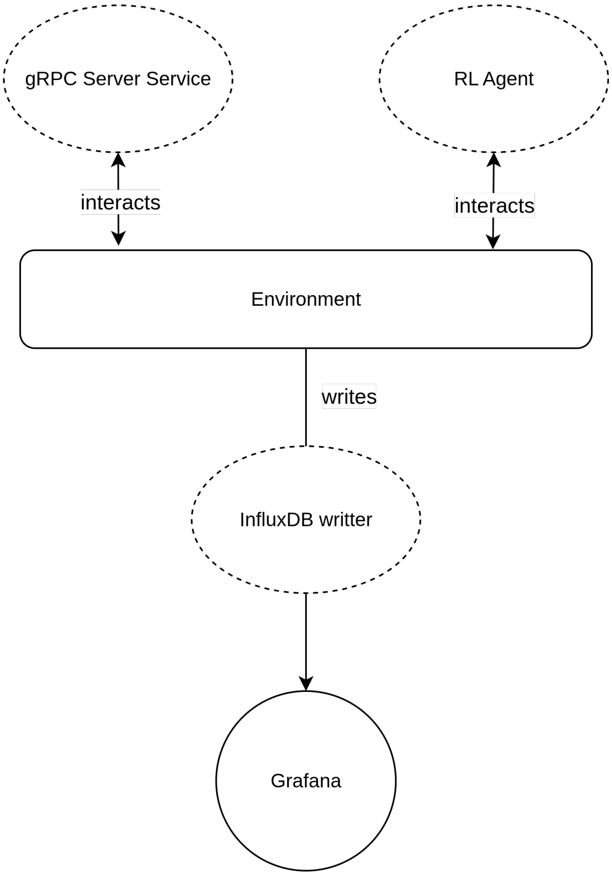


Figure 4.6: Overall System for Validation of the DDA

Finally, to validate the DDA system, the three co-routines works together to show information on the Grafana dashboard. In the meantime, interference was applied to the environment and the agent behaviour was observer in the dashboard. Figure 4.7.

Through the observation of the dashboard, it is possible to see in the left side up the physiological data of the HR and BMR, on the left side down, data that corresponds

Figure 4.7: Observation Space with additional info on the Grafana dashboard

to FER. This division was strictly necessary due to the difference of scales between the metrics. In addition, HR and BMR are not normalized in this validation for easier understanding of all the metrics. On the right side is located the stress and the difficulty during time.

Interferences are very easy to locate, as they are the peaks of the time-series graph. When a peak appears on the dashboard, usually it is more easy to see on the HR series. As shown in the Figure 4.7, interference was introduced in order to generate stress. When stress climbs up, the first action that the agent must do is to decrease the difficulty, in order to try to balance the stress of the patient, to prevent more stress. This, is what is detailed in the figure, assuring effectiveness on the decision-making of the agent trained.

4.4 Summary

This chapter described, trained and validated a DDA algorithm based on RL. Observation space relies on three physiological signals (HR, BMR andFER), that are used to infer the patient stress in real time. The action space was defined as the what the agent could do, which was decrease, maintain ans increase the difficulty of the game. A DQN was selected as the agent, that followed a carefully designed reward function, that combines several terms: stress, difficulty, action cost, physiological signals and similarity between signals. These terms, helped the agent to go to a optimal range of stress and difficulty, not oscillating to much the difficulty when it is not necessary, maintaining an optimal range of difficulty and stress, that maintains a good experience at rehabilitation. The agent was trained with synthetic data, who successfully converges to a stable policy, maintaining both stress and difficulty at desirable levels and preferring to keep the difficulty unchanged except when adjustments are necessary.

Validation was made by doing interference in the environment, by using gRPC and monitoring the agent response with a Grafana dashboard. Validation confirms the ability of the agent to response accordingly to stress peaks by lowering difficulty, demonstrating the systems robustness and real-time adaptability.

Chapter 5

Architecture for Integration

This chapter details the architecture to integrate the DDA into a web application dashboard for the medical team to use in rehabilitation sessions. It will be explored how DDA was integrated, using distinct protocols to ensure communication between different platforms, including technologies like WebSockets, message-oriented middleware, and message parsing are useful to ensure data consistency and efficiency in communication between different platforms.

5.1 Entities

The back-office relies on a distributed platform, with several components, each one with a well-defined task and behaviour. The reason why there are several components is justified by the quantity of different parts that are supposed to be integrated. These components will be divided into modules, as explained bellow:

- **Module 1 - Back-end Server**: an application developed with the SpringBoot framework, that centralizes all data. It is composed by a non-relational database in a Docker container. It is a versatile server with controllers exposed for Representational State Transfer (REST), Advance Message Queue Protocol (AMQP) and Websocket services;

- **Module 2 - Physiological Servers**: composed by four services, one for each physiological signal extracted, HR, BMR and FER. The fourth one is a specific service dedicated to the DDA, being a specialized service that receives data from the other three services and process the difficulty adjustment in real-time;
- **Module 3 - Front-end Application**: corresponding to the web application that will be used by the medical team during the rehabilitation sessions;
- **Module 4 - VR Application**: the application in VR with the serious game that will be used by the patients during each session.

5.1.1 Module 1 - Back-End

The back-end is the core of the entire system. It centralizes all data that came from other components and guarantees data consistency, integrity and availability. Built with Spring Boot, it relies on a Layered Architecture, where different layers work together as a single unit of software. In an horizontal-view perspective, the bottom layer corresponds to the Data Layer, responsible for handling the data system where the information is stored. Then, there is the Persistence Layer, who is responsible to manage data storage and retrieval. The Business Layer corresponds to all services created in the application, being a bridge between the Persistence Layer and the Application Layer. The Application Layer consists on all controllers that exposes all the aspects of the functional requirements of the entire system.

Layers can be open or close, if one layer can bypass another layer for direct access, when needed or no. For this architecture, it was followed a closed layer purpose, in order to maintain a strict interaction of layers only with the layer above or beyond, ensuring better control and encapsulating.

Data and Persistence Layers

These layers are responsible to store and retrieve all data. The database itself was built with MongoDB, a Not only SQL (NoSQL) database, inside a persistent Docker container.

Due to the non-relational nature of the database, which stores data in flexible JSON-like documents, it was the best choice to store the data structures generated by the system. The entities existent in the database are: `PatientModel` to store all data related to the patient, `BodyMotionRateModel` to store information related to BMR, `FaceModel` for information related to FER, `HeartRateModel` for information of HR, `DDAModel` for information of the DDA, `RehabilitationModel` for information of the front-end to store data of the rehabilitation session and `TaskModel` which corresponds to a daily task activity in the VR game. All data is retrieve with the help of the Spring Boot Repositories, making it available for the Service Layer.

Service Layer

The Service Layer corresponds to all the Spring services implemented, which holds the business logic. This intermediate layer ensures that the Application Layer cannot request data to the Persistence Layer, ensuring that each defined layer only does an atomic and well defined task.

Application Layer

The Application Layer is responsible to expose communication with external components. Since the application have distinct components to integrate, it expose distinct forms of communication. REST, WebSockets and publish-subscribe communication are integrated in the several controllers of the Spring Boot application. This layer follows an Event-Driven Architecture paradigm, a popular asynchronous distributed architecture pattern, where the system components communicate through events, that can be notifications of some occurrence in the system changes [60]. This architecture enables real-time responsiveness, scalability, and modularity. Components that integrates this type of communication are loosely coupled, meaning that applications can function without being tightly linked to one another. So, applications does not know who they are sending messages. They only know what they consume, dispatching the received message with the relevant components to respond accordingly. Although there are numerous advantages of using this type of architecture, there is a main disadvantage, that increases the complexity of this type of systems, being related with data consistency. It is mandatory to have coordination methods that ensures data synchronization and data consistency.

Figure 5.1 demonstrates a visual layered scenario of the architecture developed.

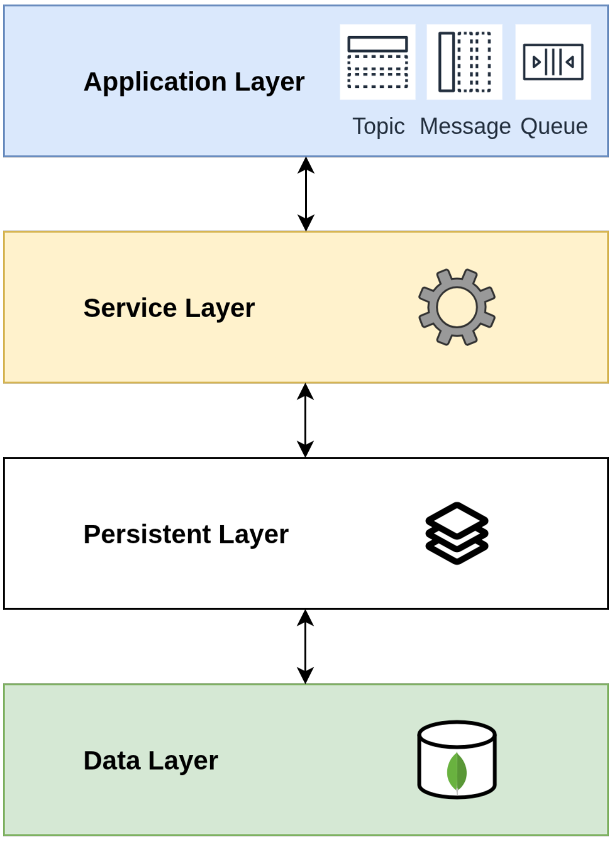


Figure 5.1: Layered Architecture for Back-End

5.1.2 Module 2 - Physiological Servers

The three physiological server, named by HR Server, BMR Server and FER Server, are three tiny services that encapsulates each one of PyTorch models, and communicate data with the Spring Boot application through the Simple Text Oriented Messaging Protocol (STOMP). This protocol was designed to send text-based messages in a message-oriented middleware environment. STOMP connects to the message broker by using a set of commands. For example, `SEND` to publish a message to the message broker, `CONNECT` to initiate the connection with the broker, `SUBSCRIBE` to subscribe to a specific topic or queue, and `UNSUBSCRIBE` to stop from receiving messages from a subscribed topic.

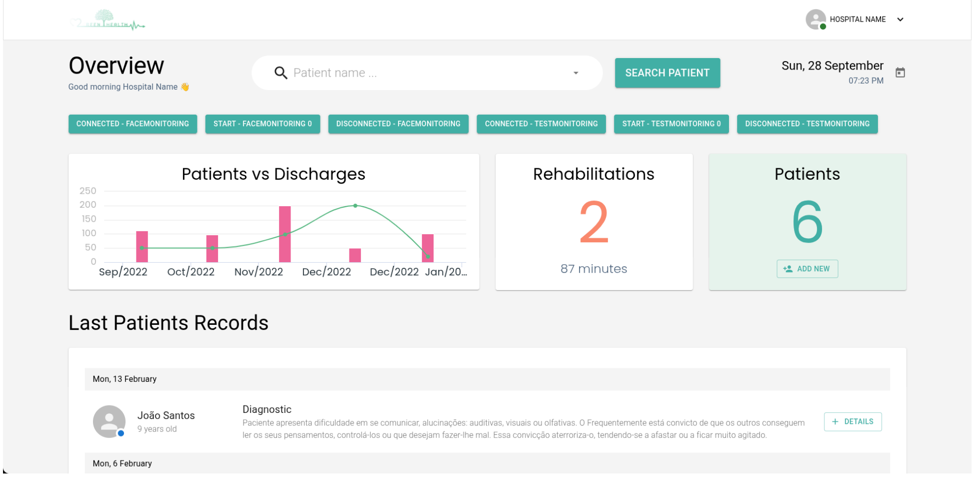
The fourth one corresponds to the DDA. It it necessary to have a special service of the RL algorithm that receives the metrics from the three servers and process the dynamic difficulty adaptation in real-time. This server also communicates through the STOMP protocol to the Spring Boot application.

5.1.3 Module 3 - Front-End Application

Developed by Neto et al. [61] the front-end application relies on a web dashboard for the healthcare professionals to use during the rehabilitation scenario. The web application was developed having several attention to the User Interface (UI) and User Experience (UX), by using a User-Centered Design principle, that focus on the addressing the user's needs, preferences and processes during the design process. To validate the case, a user satisfaction using the SUS was realized, resulting on a usability score of 81.6 points out of 100.

The web application was built using the React framework, with the TypeScript language. The system provides tools for a specific health centre to see their patients, the number of rehabilitations and diagnosis for each patient (Figure 5.2). It is possible to search for patients in a search bar (Figure 5.4) and see the detailing information of the patient selected, such as the number of rehabilitations, body movement, body temperature, facial expressions, the record data for each rehabilitation and the progress of each game activity.

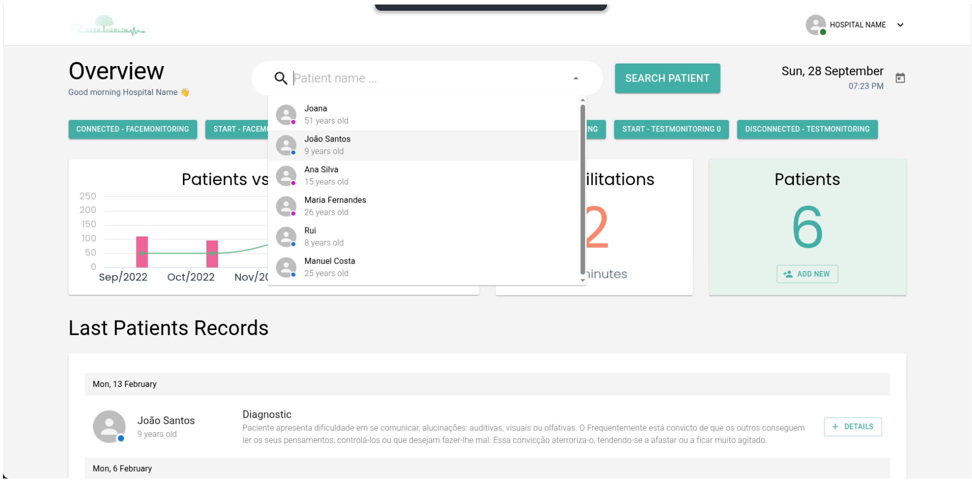
The rehabilitation dashboard is a special page that needs to be updated in real time, so it cannot have asynchronous communication such as a RESTful service. In order to make the page dynamic and ensure data retrieval over time, a websocket was built in this page. The construction of the websocket will be explained later in this chapter. In Figure 5.5 elements such as the HR, facial recognition, body movement, game activities and session time are displayed in this page.



João Santos
9 years old

Diagnostic
Paciente apresenta dificuldade em se comunicar, alucinações: auditivas, visuais ou olfativas. O Frequentemente está convicto de que os outros conseguem ler os seus pensamentos, controlá-los ou que desejam fazer-lhe mal. Essa convicção aterroriza-o, tendendo-se a afastar ou a ficar muito agitado.

Figure 5.2: Web Application Overview

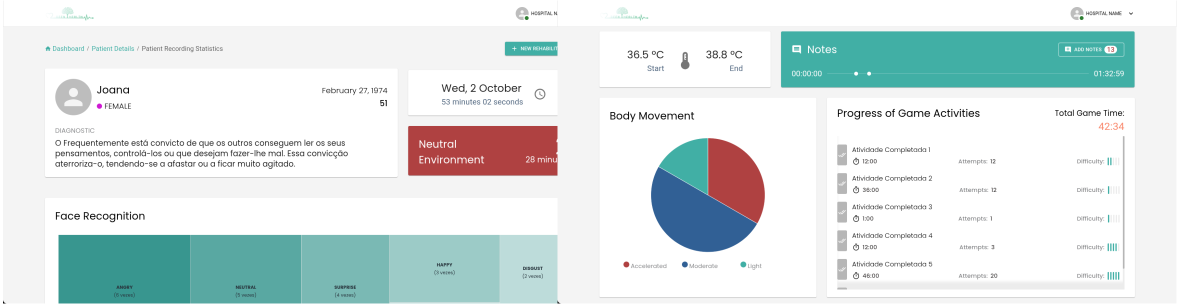


Joana
51 years old
João Santos
9 years old
Ana Silva
15 years old
Maria Fernandes
26 years old
Rui
8 years old
Manuel Costa
25 years old

João Santos
9 years old

Diagnostic
Paciente apresenta dificuldade em se comunicar, alucinações: auditivas, visuais ou olfativas. O Frequentemente está convicto de que os outros conseguem ler os seus pensamentos, controlá-los ou que desejam fazer-lhe mal. Essa convicção aterroriza-o, tendendo-se a afastar ou a ficar muito agitado.

Figure 5.3: Search for health centre’s patients



Joana
FEMALE
February 27, 1974
51

DIAGNOSTIC
O Frequentemente está convicto de que os outros conseguem ler os seus pensamentos, controlá-los ou que desejam fazer-lhe mal. Essa convicção aterroriza-o, tendendo-se a afastar ou a ficar muito agitado.

Atividade Completada 1 — 12:00 — Attempts: 12 — Difficulty:
Atividade Completada 2 — 36:00 — Attempts: 12 — Difficulty:
Atividade Completada 3 — 1:00 — Attempts: 1 — Difficulty:
Atividade Completada 4 — 12:00 — Attempts: 3 — Difficulty:
Atividade Completada 5 — 46:00 — Attempts: 20 — Difficulty:

(a) Details on patient information, diagnosis and facial details

(b) Details on body temperature, body movement, and game activity progress

Figure 5.4: Daily Action in VR during Rehabilitation

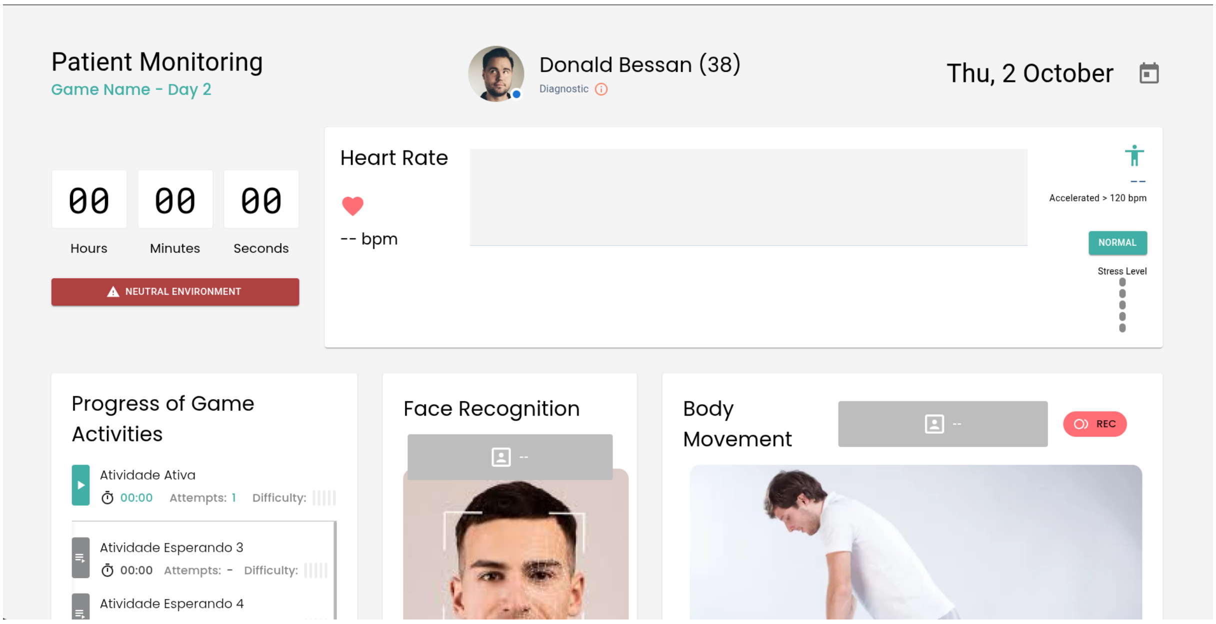


Figure 5.5: Rehabilitation Dashboard Page

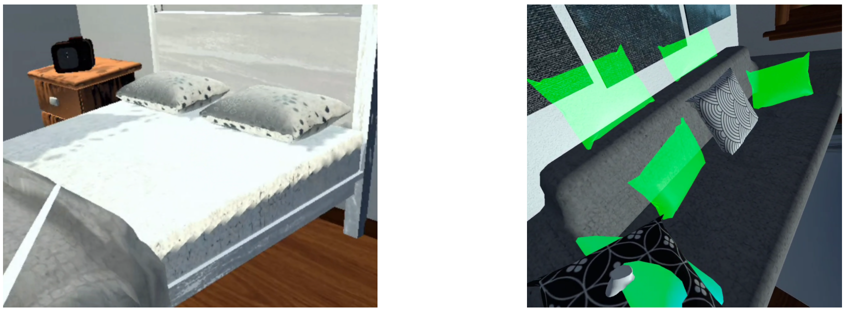
5.1.4 Module 4 - VR Application

Developed by Pinto et al. [62], the VR application was constructed following the recommendations of the Local Health Unit of Bragança. Built with the Unity engine, the game is composed of three scenarios of a house, being them the bedroom, the living room and the bathroom. For each part of the house there are a variety of activities to be done in each room. For example, if the patient is in the bedroom, it must make the bed, open the window curtains and order the pillows. If the patient is in the living room, it must put several books arranged by colours in a bookshelf and put the cushions on the sofa. If the patient is in the bathroom, it must brush their teeth and put dirty clothes in the washing machine (Figure 5.6).

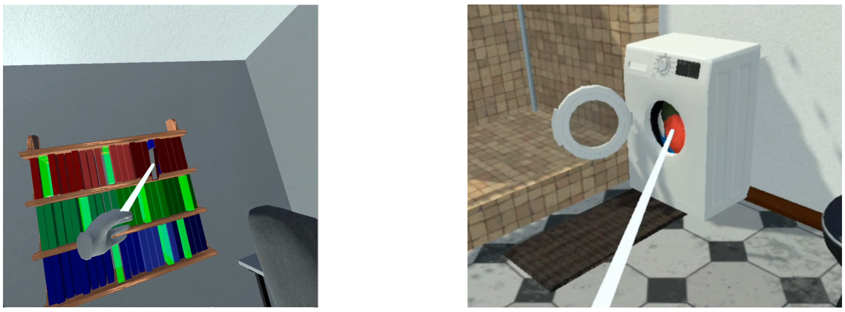
The house-based scenario was think for the persons who suffer from the negative and cognitive symptoms of the disease, as described in chapter 1, where patients demonstrate loss of physical skills and social isolation. The game is a tool, that these type of patients can use in order to gain the skills they have lost, due to the disease.

The application in each room have a mobile smartphone where patients can interact and see the tasks they have done and record time for each task. When a task is completed,

(a) Patient's Tasks (b) Make the bed



(c) Arrange bed cushions (d) Put the cushions on the sofa



(e) Arrange the books in a bookshelf (f) Washing dirty clothes

Figure 5.6: Daily Action in VR during Rehabilitation

the application communicates to the Spring application, through AMQP the necessary data that the server need to feed-forward the web dashboard.

5.2 Event-Driven Architecture

Defined all the entities that compose the entire system, it began the phase to explain in detail how every module communicates with the Spring application.

Figure 5.7 shows a visual representation of the message-queue system, that each module of the system follows.

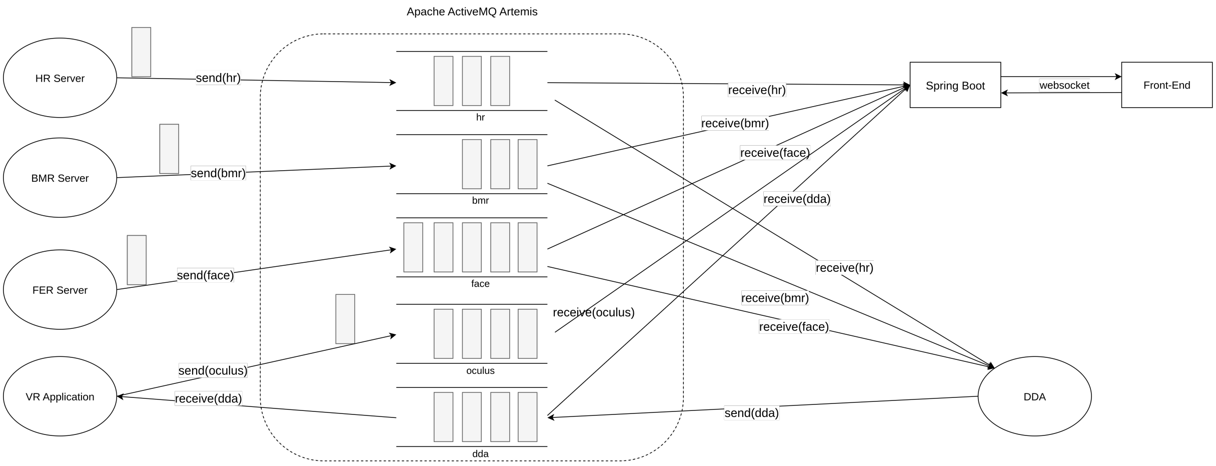


Figure 5.7: Message Queue Architecture

Since message-queue systems are part of indirect communication, it is necessary to have an intermediary between the entities that sends the message and who receives them. For that a Apache ActiveMQ Artemis was chosen. The Artemis broker is an open-source software written in Java having a high performance, low latency and delivery guarantee[1]. It supports a wide range of message protocols, including: AMQP, Message Queuing Telemetry Transport (MQTT) , STOMP, Core (the native Artemis protocol), OpenWire, and Java Message Service (JMS). This flexibility allows Artemis to integrate seamlessly with heterogeneous systems and multiple communication paradigms, making it a robust choice for integration this modules.

---

[1] https://activemq.apache.org/components/artemis/documentation/latest/book.pdf

As mentioned in the Figure 5.7, all the modules follows an Event-Driven architecture, where all modules are loosely coupled to each other, and they communicate trough events. As also said before, each module is autonomous and independent, which means that any update across some module, does not affect the entire system.

The Spring Boot application as a centralized back-end and the source of all information, needs to receive data from all the modules on the network. His functions is to guarantee the hosting of data, and communicate all the necessary to the web dashboard through the WebSocket, and also using REST service.

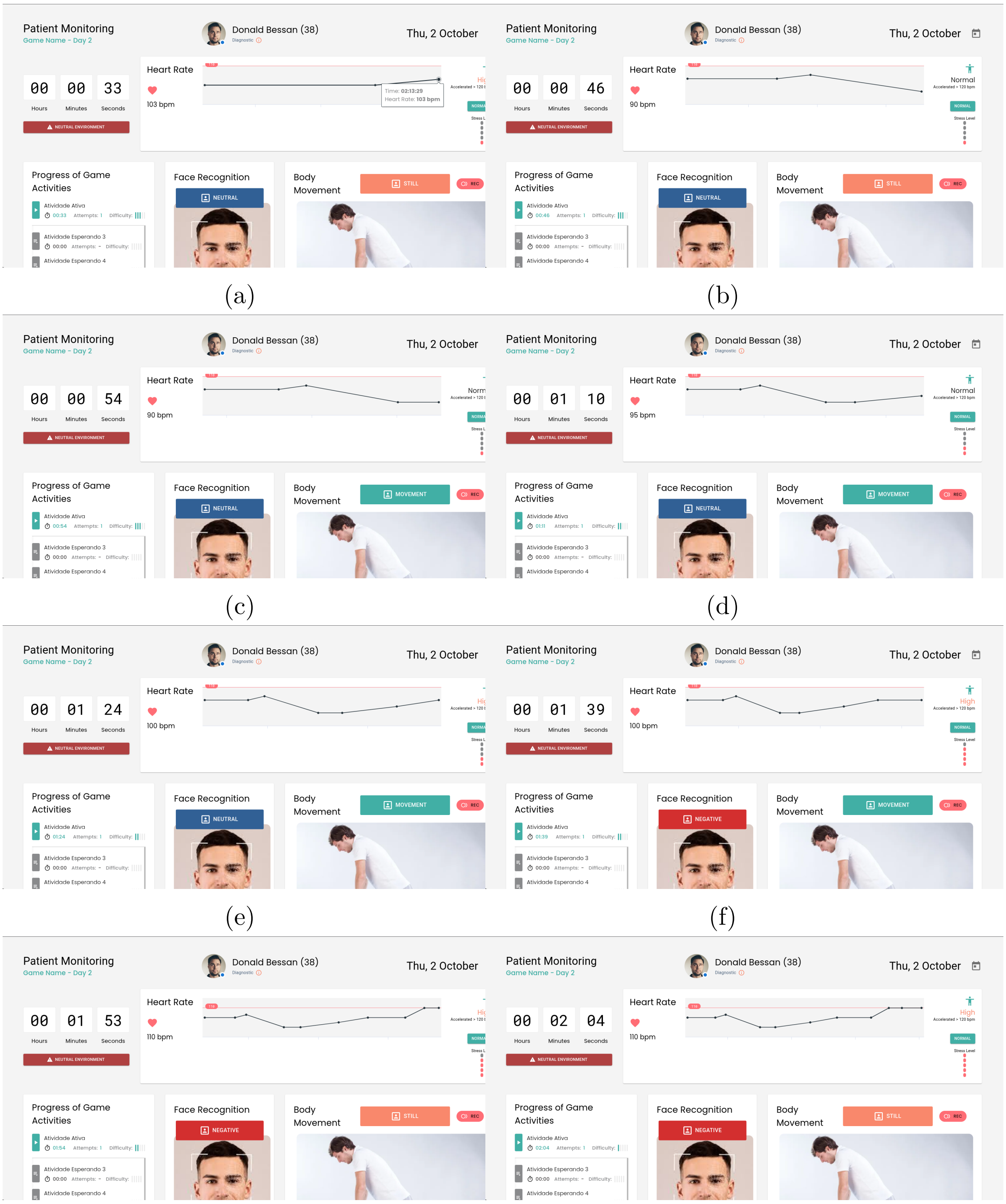
The HR, BMR and FER servers, are responsible to take real time data of the heart, movement of the body and facial expressions, respectively, in real-time, in order to feed forward the other components of the system.

The DDA system, must make use of the data published by the physiological servers, and adapt dynamically the difficulty of the game, based on the present metrics. Then, it should publish the difficulty information to the respective queue, where the VR game and the Spring Boot application must retrieve that data.

Finally, the VR application, should receive information on the DDA system, and must update the game tasks, as they are being completed, with the included current difficulty of the game.

All the messages of all topics were set to be in the JSON format, due to its simplicity and quickly data format. In audition, Spring Boot has its own library for data serialization, named Jackson, who has extremely important on converting data from JSON to Java Objects.

As a result, the system became transparent when all services are working together, by showing in the web dashboard all the information necessary to be used by the healthcare professionals during rehabilitation, as detailed in Figure 5.8.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.8: Rehabilitation Dashboard receiving data through WebSocket

5.3 Summary

This chapter presented the architecture designed to integrate the DDA system into a web rehabilitation platform. The solution was divided into several modules, being them a back-end server, physiological servers, the front-end dashboard, and the VR application. Each module has a well-defined role and communicates through standardized protocols to ensure interoperability and data consistency.

The back-end server was developed with Spring Boot, and acts as the system core, following a Layered Architecture, managing data persistence with MongoDB, and exposes all the multiple communication interfaces (REST, WebSockets ans publish-subscribe communication), following an Event-Driven architecture. The physiological servers encapsulate machine learning models for HR, BMR, and FER estimation, as well as a dedicated DDA service that processes multimodal data to adapt game difficulty in real time. The front-end provides a user centred dashboard for healthcare professionals, offering a real-time monitoring of the patients' physiological performance, under the rehabilitation session. The VR application developed in Unity, includes a home scenario, where patients must do daily task activities.

Communication between all modules is enabled by an Event-Driven architecture built on Apache ActiveMQ Artemis, a broker that supports multiple messaging protocols, such as AMQP, STOMP, MQTT, and many others. Messages are exchanged in JSON format, ensuring lightweight and transparent data serialization across heterogeneous components.

This architecture ensures scalability, modularity and responsiveness, enabling healthcare professionals to monitor rehabilitation sessions in real time and patients to benefit from dynamically adapted VR scenarios that optimize training engagement and effectiveness.

Chapter 6

Conclusions

This dissertation has presented the conception, design, and implementation of an adaptive rehabilitation framework that integrates VR, distributed architectures, and RL. The work focused specifically on schizophrenia rehabilitation, a domain in which maintaining engagement, motivation, and therapeutic continuity is particularly challenging. The system demonstrates how physiological (HR) and behavioural (BMR and FER) signals can be used to guides the rehabilitation process in real time. Combining HR with body behaviour and facial expressions, it was possible to propose a stress inference model that captures both physical and emotional states of the patient. These stress levels were then used to build a Reinforcement Learningbased Dynamic Difficulty Adjustment system, that aims to adjust the difficulty of the VR application in real time, keeping a engaged session during rehabilitation. Results from training and simulation confirmed that the RL agent was capable of learning stable policies, adjusting task difficulty to maintain an appropriate balance between challenge and comfort. Moreover, the agent showed the ability to avoid extreme variations in stress and difficulty, maintaining a good effort level, keeping an average level on both stress and difficulty.

In order to integrate such distinct types of components, distributed, event-driven architecture was built, enabling communication between the different modules, being them the back office, the physiological modules, the web dashboard and the VR application. Publish-subscribe system communication was the communication patter chosen, since it ensures flexibility, scalability, and robustness, which are critical for integrating heterogeneous components in healthcare environments. The use of containerized deployment further enhances reproducibility and portability, facilitating future extensions and clinical deployment.

More than technical contributions, this work also tries to bridging even more medicine with digital technologies, known as digital health. Adaptative rehabilitation systems have been showing their potential to improve traditional approaches, opening pathways to a more personalized and patient-centred care, as these systems can reduce dropout rates, improve therapeutic adherence, and extend rehabilitation to home-based or remote contexts.

Future work includes clinical trials to evaluate the impact of this system in real patients with schizophrenia to ensure the capability of this system to really help people around the world.

Bibliography

[1] R. P. Lopes et al., "Digital Technologies for Innovative Mental Health Rehabilitation," en, *Electronics*, vol. 10, no. 18, p. 2260, Sep. 2021, ISSN: 2079-9292. DOI: `10.3390/electronics10182260`. [Online]. Available: `https://www.mdpi.com/2079-9292/10/18/2260`.

[2] R. d. Almeida, A. Soutelo, P. Portugal, and A. Marques, "The use of digital technologies to promote physical activity in mental healthcare settings: A scoping review," *Advances in Rehabilitation*, vol. 37, no. 4, pp. 29–40, 2023, Publisher: Termedia, ISSN: 1734-4948. DOI: `10.5114/areh.2023.133146`. [Online]. Available: `https://www.advrehab.org/The-use-of-digital-technologies-to-promote-physical-activity-in-mental-healthcare-settings-A-scoping-review,125,51890,1,1.html`.

[3] B. Bonnechère and S. Van Sint Jan, "Chapter 39 - Rehabilitation," in *DHM and Posturography*, S. Scataglini and G. Paul, Eds., Academic Press, Jan. 2019, pp. 541–547, ISBN: 978-0-12-816713-7. DOI: `10.1016/B978-0-12-816713-7.00039-8`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B9780128167137000398`.

[4] F. Laamarti, M. Eid, and A. El Saddik, "An Overview of Serious Games," en, *International Journal of Computer Games Technology*, vol. 2014, no. 1, p. 358 152, 2014, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2014/358152, ISSN: 1687-7055. DOI: `10.1155/2014/358152`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1155/2014/358152`.

[5] M. Macedo, A. Marques, and C. Queirós, “Virtual reality in assessment and treatment of schizophrenia: A systematic review,” en, *Jornal Brasileiro de Psiquiatria*, vol. 64, pp. 70–81, Mar. 2015, Publisher: Instituto de Psiquiatria da Universidade Federal do Rio de Janeiro, ISSN: 0047-2085, 1982-0208. DOI: `10.1590/0047-2085000000059`. [Online]. Available: `https://www.scielo.br/j/jbpsiq/a/38pttC9zNHjmV9TZNM6WGsG/?lang=en`.

[6] A. Barbato, W. N. f. M. H. Initiative, and W. H. O. D. o. M. H. a. P. o. S. Abuse, “Schizophrenia and public health,” en, 1997, Accepted: 2012-06-16T23:37:41Z Number: WHO/MSA/NAM/97.6 Publisher: World Health Organization. [Online]. Available: `https://iris.who.int/handle/10665/63837`.

[7] A. Rampino, R. M. Falcone, A. Giannuzzi, R. Masellis, L. A. Antonucci, and S. Torretta, “Strategies for Psychiatric Rehabilitation and their Cognitive Outcomes in Schizophrenia: Review of Last Five-year Studies,” en, DOI: `10.2174/1745017902117010031`. [Online]. Available: `https://clinical-practice-and-epidemiology-in-mental-health.com/VOLUME/17/PAGE/31/`.

[8] J. C. Lopes and R. P. Lopes, “A review of dynamic difficulty adjustment methods for serious games,” en, in *Optimization, Learning Algorithms and Applications*, A. I. Pereira, A. Koir, F. P. Fernandes, M. F. Pacheco, J. P. Teixeira, and R. P. Lopes, Eds., ser. Communications in Computer and Information Science, Cham: Springer International Publishing, 2022, pp. 144–159. DOI: `10.1007/978-3-031-23236-7_11`.

[9] P. Bosch-Barceló, M. Masbernat-Almenara, O. Martínez-Navarro, C. Tersa-Miralles, A. Pakarinen, and H. Fernández-Lago, “A gamified virtual environment intervention for gait rehabilitation in Parkinsons Disease: Co-creation and feasibility study,” en, *Journal of NeuroEngineering and Rehabilitation*, vol. 21, no. 1, p. 107, Jun. 2024, ISSN: 1743-0003. DOI: `10.1186/s12984-024-01399-6`. [Online]. Available: `https://doi.org/10.1186/s12984-024-01399-6`.

[10] M. M. Hoehn and M. D. Yahr, "Parkinsonism: Onset, progression, and mortality," *Neurology*, vol. 17, no. 5, pp. 427–442, 1967. DOI: `https://doi.org/10.1212/WNL.17.5.427`. [Online]. Available: `https://www.neurology.org/doi/10.1212/WNL.17.5.427`.

[11] J. J. Martins, M. Castelo-Branco, and M. Simões, "Immersive Virtual Reality Serious Game for Facial Expression Training," in *2024 IEEE 12th International Conference on Serious Games and Applications for Health (SeGAH)*, ISSN: 2573-3060, Funchal, Portugal, Aug. 2024, pp. 1–7. DOI: `10.1109/SeGAH61285.2024.10639571`. [Online]. Available: `https://ieeexplore.ieee.org/document/10639571/?arnumber=10639571`.

[12] P. Robitaille and M. J. McGuffin, "Increased affect-arousal in VR can be detected from faster body motion with increased heart rate," in *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '19, New York, NY, USA: Association for Computing Machinery, May 2019, pp. 1–6, ISBN: 978-1-4503-6310-5. DOI: `10.1145/3306131.3317022`. Accessed: Mar. 27, 2023. [Online]. Available: `https://dl.acm.org/doi/10.1145/3306131.3317022`.

[13] W. L. Raffe and J. A. Garcia, "Combining skeletal tracking and virtual reality for game-based fall prevention training for the elderly," in *2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)*, ISSN: 2573-3060, May 2018, pp. 1–7. DOI: `10.1109/SeGAH.2018.8401371`.

[14] S. Ishaque, N. Khan, and S. Krishnan, "Physiological Signal Analysis and Stress Classification from VR Simulations Using Decision Tree Methods," en, *Bioengineering*, vol. 10, no. 7, p. 766, Jul. 2023, Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2306-5354. DOI: `10.3390/bioengineering10070766`. Accessed: Feb. 8, 2024. [Online]. Available: `https://www.mdpi.com/2306-5354/10/7/766`.

[15] K. Harrison, E. Potts, A. C. King, and R. Braun-Trocchio, "The Effectiveness of Virtual Reality on Anxiety and Performance in Female Soccer Players," en, *Sports*,

vol. 9, no. 12, p. 167, Dec. 2021, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2075-4663. DOI: `10.3390/sports9120167`. Accessed: Feb. 16, 2024. [Online]. Available: `https://www.mdpi.com/2075-4663/9/12/167`.

[16] M. Ma, R. Proffitt, and M. Skubic, “Validation of a Kinect V2 based rehabilitation game,” en, *PLOS ONE*, vol. 13, no. 8, e0202338, Aug. 2018, Publisher: Public Library of Science, ISSN: 1932-6203. DOI: `10.1371/journal.pone.0202338`.

[17] D. E. Lidstone, R. Rochowiak, C. Pacheco, B. Tunçgenç, R. Vidal, and S. H. Mostofsky, “Automated and scalable Computerized Assessment of Motor Imitation (CAMI) in children with Autism Spectrum Disorder using a single 2D camera: A pilot study,” *Research in Autism Spectrum Disorders*, vol. 87, p. 101 840, Sep. 2021, ISSN: 1750-9467. DOI: `10.1016/j.rasd.2021.101840`.

[18] G. F. Coulouris, J. Dollimore, T. Kindberg, and G. Blair, *Distributed Systems: Concepts and Design*, 5th. Boston, MA / London: Pearson / Addison-Wesley, 2011, ISBN: 9780132143011.

[19] W. Wei et al., *An extensive study on text serialization formats and methods*, 2025. arXiv: `2505.13478 [cs.SE]`. [Online]. Available: `https://arxiv.org/abs/2505.13478v1`.

[20] D. Vohra, “Apache avro,” in *Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools*. Berkeley, CA: Apress, 2016, pp. 303–323, ISBN: 978-1-4842-2199-0. DOI: `10.1007/978-1-4842-2199-0_7`. [Online]. Available: `https://doi.org/10.1007/978-1-4842-2199-0_7`.

[21] M.-J. Kim, J. S. Kim, and C. W. Ahn, “Evolving population method for real-time reinforcement learning,” en, *Expert Systems with Applications*, vol. 229, p. 120 493, Nov. 2023, ISSN: 0957-4174. DOI: `10.1016/j.eswa.2023.120493`. Accessed: Jul. 13, 2023. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0957417423009958`.

[22] M. Reymond, C. F. Hayes, D. Steckelmacher, D. M. Roijers, and A. Nowé, "Actor-critic multi-objective reinforcement learning for non-linear utility functions," en, *Autonomous Agents and Multi-Agent Systems*, vol. 37, no. 2, p. 23, Apr. 2023, ISSN: 1573-7454. DOI: `10.1007/s10458-023-09604-x`. Accessed: Jul. 13, 2023. [Online]. Available: `https://doi.org/10.1007/s10458-023-09604-x`.

[23] Y. Gui, D. Tang, H. Zhu, Y. Zhang, and Z. Zhang, "Dynamic scheduling for flexible job shop using a deep reinforcement learning approach," en, *Computers & Industrial Engineering*, vol. 180, p. 109 255, Jun. 2023, ISSN: 0360-8352. DOI: `10.1016/j.cie.2023.109255`. Accessed: Jul. 13, 2023. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0360835223002796`.

[24] Y. Mothanna and N. Hewahi, "Review on Reinforcement Learning in CartPole Game," in *2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, ISSN: 2770-7466, Nov. 2022, pp. 344–349. DOI: `10.1109/3ICT56508.2022.9990767`.

[25] J. Pfau, A. Liapis, G. N. Yannakakis, and R. Malaka, "Dungeons & Replicants II: Automated Game Balancing Across Multiple Difficulty Dimensions via Deep Player Behavior Modeling," *IEEE Transactions on Games*, vol. 15, no. 2, pp. 217–227, Jun. 2023, Conference Name: IEEE Transactions on Games, ISSN: 2475-1510. DOI: `10.1109/TG.2022.3167728`.

[26] T. Zhu, M. Ma, L. Chen, and Z. Liu, "Optimal strategy of the simultaneous dice game Pig for multiplayers: When reinforcement learning meets game theory," en, *Scientific Reports*, vol. 13, no. 1, p. 8142, May 2023, Number: 1 Publisher: Nature Publishing Group, ISSN: 2045-2322. DOI: `10.1038/s41598-023-35237-x`. Accessed: Jul. 13, 2023. [Online]. Available: `https://www.nature.com/articles/s41598-023-35237-x`.

[27] J. Delaney, S. Dowey, and C.-T. Cheng, "Reinforcement-Learning-Based Robust Resource Management for Multi-Radio Systems," en, *Sensors*, vol. 23, no. 10, p. 4821, Jan. 2023, Number: 10 Publisher: Multidisciplinary Digital Publishing Institute,

ISSN: 1424-8220. DOI: `10.3390/s23104821`. Accessed: Jul. 13, 2023. [Online]. Available: `https://www.mdpi.com/1424-8220/23/10/4821`.

[28] M. Agrebi, M. Sendi, and M. Abed, “Deep reinforcement learning for personalized recommendation of distance learning,” English, *Advances in Intelligent Systems and Computing*, vol. 931, pp. 597–606, 2019, ISBN: 9783030161835, ISSN: 2194-5357. DOI: `10.1007/978-3-030-16184-2_57`.

[29] C. Qiu, Z. Wu, J. Wang, M. Tan, and J. Yu, “Multiagent-Reinforcement-Learning-Based Stable Path Tracking Control for a Bionic Robotic Fish With Reaction Wheel,” English, *IEEE Transactions on Industrial Electronics*, vol. 70, no. 12, pp. 12 670–12 679, 2023, ISSN: 0278-0046. DOI: `10.1109/TIE.2023.3239937`.

[30] J. C. Lopes and R. P. Lopes, “A Review ofăDynamic Difficulty Adjustment Methods forăSerious Games,” en, in *Optimization, Learning Algorithms and Applications*, A. I. Pereira, A. Koir, F. P. Fernandes, M. F. Pacheco, J. P. Teixeira, and R. P. Lopes, Eds., ser. Communications in Computer and Information Science, Cham: Springer International Publishing, 2022, pp. 144–159, ISBN: 978-3-031-23236-7. DOI: `10.1007/978-3-031-23236-7_11`.

[31] P. Rodrigues, M. Fonseca, and P. Lopes, “Physiological-Based Difficulty Assessment for Virtual Reality Rehabilitation Games,” en, in *Proceedings of the 18th International Conference on the Foundations of Digital Games*, Lisbon Portugal: ACM, Apr. 2023, pp. 1–4, ISBN: 978-1-4503-9855-8. DOI: `10.1145/3582437.3587187`. Accessed: Jul. 17, 2023. [Online]. Available: `https://dl.acm.org/doi/10.1145/3582437.3587187`.

[32] A. Bonti, M. Palaparthi, X. Jiang, and T. Pham, “TuneIn: Framework Design andăImplementation forăEducation Using Dynamic Difficulty Adjustment Based onăDeep Reinforcement Learning andăMathematical Approach,” en, in *Ad Hoc Networks and Tools for IT*, W. Bao, X. Yuan, L. Gao, T. H. Luan, and D. B. J. Choi, Eds., ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Cham: Springer International Publishing,

2022, pp. 229–241, ISBN: 978-3-030-98005-4. DOI: `10.1007/978-3-030-98005-4_17`.

[33] M. Orsoni, A. Pögelt, N. Duong-Trung, M. Benassi, M. Kravcik, and M. Grüttmüller, "Recommending Mathematical Tasks Based onăReinforcement Learning and Item Response Theory," en, in *Augmented Intelligence and Intelligent Tutoring Systems*, C. Frasson, P. Mylonas, and C. Troussas, Eds., ser. Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2023, pp. 16–28, ISBN: 978-3-031-32883-1. DOI: `10.1007/978-3-031-32883-1_2`.

[34] E. Pagalyte, M. Mancini, and L. Climent, "Go with the Flow: Reinforcement Learning in Turn-based Battle Video Games," English, in *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents (IVA 2020)*, 2020, ISBN: 978-1-4503-7586-3. DOI: `10.1145/3383652.3423868`.

[35] F. Glavin and M. Madden, "Skilled Experience Catalogue: A Skill-Balancing Mechanism for Non-Player Characters using Reinforcement Learning," English, in *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games, CIG 2018*, ISSN: 2325-4270, vol. 2018-August, 2018, ISBN: 978-1-5386-4359-4. DOI: `10.1109/CIG.2018.8490405`.

[36] K. Kamikokuryo, T. Haga, G. Venture, and V. Hernandez, "Adversarial Autoencoder and Multi-Armed Bandit for Dynamic Difficulty Adjustment in Immersive Virtual Reality for Rehabilitation: Application to Hand Movement," en, *Sensors*, vol. 22, no. 12, p. 4499, Jan. 2022, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: `10.3390/s22124499`. Accessed: Jul. 13, 2023. [Online]. Available: `https://www.mdpi.com/1424-8220/22/12/4499`.

[37] A. Cherukuri and F. G. Glavin, "Balancing the Performance of a FightingICE Agent using Reinforcement Learning and Skilled Experience Catalogue," in *2022 IEEE Games, Entertainment, Media Conference (GEM)*, ISSN: 2766-6530, Nov. 2022, pp. 1–6. DOI: `10.1109/GEM56474.2022.10017566`.

[38] M. Rajabi, M. Ashtiani, B. Minaei Bidgoli, and O. Davoodi, “A dynamic balanced level generator for video games based on deep convolutional generative adversarial networks,” *Scientia Iranica*, vol. 28, no. Special issue on collective behavior of nonlinear dynamical networks, pp. 1497–1514, Jun. 2021, Publisher: Sharif University of Technology, ISSN: 1026-3098. DOI: `10.24200/sci.2020.54747.3897`. Accessed: Jul. 13, 2023. [Online]. Available: `https://scientiairanica.sharif.edu/article_22082.html`.

[39] C. F. Biemer and S. Cooper, *Level Assembly as a Markov Decision Process*, arXiv:2304.13922 [cs], Apr. 2023. Accessed: Jul. 13, 2023. [Online]. Available: `http://arxiv.org/abs/2304.13922`.

[40] D. Gravina, A. Khalifa, A. Liapis, J. Togelius, and G. N. Yannakakis, “Procedural Content Generation through Quality Diversity,” in *2019 IEEE Conference on Games (CoG)*, arXiv:1907.04053 [cs], Aug. 2019, pp. 1–8. DOI: `10.1109/CIG.2019.8848053`. Accessed: Jul. 13, 2023. [Online]. Available: `http://arxiv.org/abs/1907.04053`.

[41] S. Reis, L. P. Reis, and N. Lau, “Game Adaptation by Using Reinforcement Learning Over Meta Games,” en, *Group Decision and Negotiation*, vol. 30, no. 2, pp. 321–340, Apr. 2021, ISSN: 1572-9907. DOI: `10.1007/s10726-020-09652-8`. Accessed: Jul. 13, 2023. [Online]. Available: `https://doi.org/10.1007/s10726-020-09652-8`.

[42] M. Rahimi, H. Moradi, A.-h. Vahabie, and H. Kebriaei, *Continuous Reinforcement Learning-based Dynamic Difficulty Adjustment in a Visual Working Memory Game*, arXiv:2308.12726 [cs] version: 1, Aug. 2023. DOI: `10.48550/arXiv.2308.12726`. Accessed: Aug. 24, 2025. [Online]. Available: `http://arxiv.org/abs/2308.12726`.

[43] D.-Q. Vu, T. Thu, N. Lê, and J.-C. Wang, “Deep Learning for Human Action Recognition: A Comprehensive Review,” *APSIPA Transactions on Signal and Information Processing*, vol. 12, Jan. 2022. DOI: `10.1561/116.00000068`.

[44] J. C. Lopes, I. Van-Deste, and R. P. Lopes, “Exploring Human Action Recognition for Rehabilitation Game Application,” in *2024 IEEE 12th International Conference on Serious Games and Applications for Health (SeGAH)*, ISSN: 2573-3060, Funchal, Portugal, Aug. 2024, pp. 1–8. DOI: `10.1109/SeGAH61285.2024.10639556`. [Online]. Available: `https://ieeexplore.ieee.org/document/10639556/?arnumber=10639556`.

[45] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 172–186, Jan. 2021, Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, ISSN: 1939-3539. DOI: `10.1109/TPAMI.2019.2929257`.

[46] D. Maji, S. Nagori, M. Mathew, and D. Poddar, *YOLO-Pose: Enhancing YOLO for Multi Person Pose Estimation Using Object Keypoint Similarity Loss*, arXiv:2204.06806 [cs], Apr. 2022. Accessed: Sep. 27, 2023. [Online]. Available: `http://arxiv.org/abs/2204.06806`.

[47] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, *BlazePose: On-device Real-time Body Pose tracking*, arXiv:2006.10204 [cs], Jun. 2020.

[48] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen, and D. D. Feng, “Deep Convolutional Neural Networks for Human Action Recognition Using Depth Maps and Postures,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 9, pp. 1806–1819, Sep. 2019, Conference Name: IEEE Transactions on Systems, Man, and Cybernetics: Systems, ISSN: 2168-2232. DOI: `10.1109/TSMC.2018.2850149`.

[49] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, “Cross-view action modeling, learning and recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2014.

[50] A. S. F. Rodrigues, J. C. Lopes, R. P. Lopes, and L. F. Teixeira, “Classification ofăfacial expressions under partial occlusion forăvr games,” in *Optimization, Learning Algorithms and Applications*, A. I. Pereira, A. Koir, F. P. Fernandes, M. F. Pacheco, J. P. Teixeira, and R. P. Lopes, Eds., Cham: Springer International Publishing, 2022, pp. 804–819, ISBN: 978-3-031-23236-7.

[51] A. S. Rodrigues, J. C. Lopes, and R. P. Lopes, “Facial Expression Recognition inăVirtual Reality Simulations,” en, in *Optimization, Learning Algorithms and Applications*, A. I. Pereira et al., Eds., Cham: Springer Nature Switzerland, 2024, pp. 33–48, ISBN: 978-3-031-77426-3. DOI: `10.1007/978-3-031-77426-3_3`.

[52] A. S. Rodrigues, J. C. Lopes, and R. P. Lopes, “Eye Importance in Facial Expression Recognition,” in *2024 IEEE 12th International Conference on Serious Games and Applications for Health (SeGAH)*, ISSN: 2573-3060, Aug. 2024, pp. 1–8. DOI: `10.1109/SeGAH61285.2024.10639602`. Accessed: Sep. 5, 2025. [Online]. Available: `https://ieeexplore.ieee.org/document/10639602`.

[53] A. S. Rodrigues, J. C. Lopes, J. Vieira, and R. P. Lopes, “Partial Occlusion in Facial Expression Recognition: A Systematic Literature Review,” in *2024 IEEE 3rd International Conference on Intelligent Reality (ICIR)*, Dec. 2024, pp. 1–8. DOI: `10.1109/ICIR64558.2024.10976805`. Accessed: Sep. 5, 2025. [Online]. Available: `https://ieeexplore.ieee.org/document/10976805`.

[54] A. S. Rodrigues, J. C. Lopes, and R. P. Lopes, “From Constricted Models to State-of-the-Art for Facial Expression Classification,” en, *SN Computer Science*, vol. 6, no. 6, p. 651, Jul. 2025, ISSN: 2661-8907. DOI: `10.1007/s42979-025-04178-9`. Accessed: Sep. 5, 2025. [Online]. Available: `https://doi.org/10.1007/s42979-025-04178-9`.

[55] J. C. Lopes, J. Vieira, A. Antunes, L. Deusdado, and R. P. Lopes, “Automatic Annotation of Heart Rate Sequences,” en, in *2023 IEEE 11th International Conference on Serious Games and Applications for Health (SeGAH)*, Athens, Greece: IEEE,

Aug. 2023, pp. 1–6. DOI: `10.1109/SeGAH57547.2023.10253814`. Accessed: Oct. 28, 2024. [Online]. Available: `https://ieeexplore.ieee.org/document/10253814/`.

[56] J. C. Lopes, J. Vieira, I. Van-Deste, and R. P. Lopes, "An Architecture for Capturing and Synchronizing Heart Rate and Body Motion for Stress Inference," in *2023 IEEE 11th International Conference on Serious Games and Applications for Health (SeGAH)*, ISSN: 2573-3060, Athens, Greece, Aug. 2023, pp. 1–7. DOI: `10.1109/SeGAH57547.2023.10253815`. [Online]. Available: `https://ieeexplore.ieee.org/document/10253815/?arnumber=10253815`.

[57] J. C. Lopes, I. Van-Deste, J. Vieira, and R. P. Lopes, "Stress Inference in a Virtual Reality Game for Rehabilitation with Body Motion and Heart Rate," in *2024 IEEE 3rd International Conference on Intelligent Reality (ICIR)*, Dec. 2024, pp. 1–8. DOI: `10.1109/ICIR64558.2024.10976783`. [Online]. Available: `https://ieeexplore.ieee.org/document/10976783`.

[58] V. Mnih et al., *Playing Atari with Deep Reinforcement Learning*, arXiv:1312.5602 [cs], Dec. 2013. DOI: `10.48550/arXiv.1312.5602`. Accessed: Sep. 21, 2025. [Online]. Available: `http://arxiv.org/abs/1312.5602`.

[59] Google, *Grpc*, `https://grpc.io/`, 2025.

[60] I. Van-Deste, B. Costa, R. P. Lopes, and A. I. Pereira, "Event-driven management system for smart pill dispensers," in *Computer and Communication Engineering*, F. Neri, K.-L. Du, A.-A. San-Blas, and Z. Jiang, Eds., Cham: Springer Nature Switzerland, 2025, pp. 217–228, ISBN: 978-3-031-71079-7. DOI: `10.1007/978-3-031-71079-7_18`.

[61] J. X. Neto, J. C. Lopes, T. F. Naves, and R. P. Lopes, "User-centered dashboard design in serious vr game for cognitive rehabilitation," in *Computer and Communication Engineering. CCCE 2024*, ser. Communications in Computer and Information Science, F. Neri, K.-L. Du, A.-A. San-Blas, and Z. Jiang, Eds., vol. 2192, Springer, Cham, 2025, pp. 229–241, ISBN: 978-3-031-71078-0. DOI: `10.1007/978-3-031-71079-7_19`.

[62] G. Pinto, B. Barroso, N. Rodrigues, M. Guimarães, and E. Oliveira, “The Effects And Viability Of Video Games On The Rehabilitation Of Schizophrenic Patients: A Systematic Review,” in *2023 IEEE 11th International Conference on Serious Games and Applications for Health (SeGAH)*, ISSN: 2573-3060, Aug. 2023, pp. 1–7. DOI: `10.1109/SeGAH57547.2023.10253801`. Accessed: Sep. 28, 2025. [Online]. Available: `https://ieeexplore.ieee.org/document/10253801/`.