IPS Instituto Politécnico de Setúbal
Escola Superior de Tecnologia de Setúbal

ANDRÉ MIGUEL NAMORADO CANHOTO ANTUNES

COMPUTATIONAL FRAMEWORK TO SUPPORT SERIOUS GAMES DESIGN FOR THERAPY OF CHILDREN WITH SPECIAL NEEDS

Dissertação do Mestrado em Engenharia de Software

ORIENTADOR

Doutor Rui Miguel Neves Gonçalo Madeira

setembro 2021

ANDRÉ MIGUEL NAMORADO CANHOTO ANTUNES

COMPUTATIONAL FRAMEWORK TO SUPPORT SERIOUS GAMES DESIGN FOR CHILDREN WITH SPECIAL NEEDS

JÚRI

Presidente:
Doutor Cláudio Miguel Garcia Loureiro dos Santos Sapateiro, ESTSetúbal/IPS

Orientador:
Doutor Rui Miguel Neves Gonçalo Madeira, ESTSetúbal/IPS

Arguente:
Doutor Fausto José da Silva Valentim Mourato, ESTSetúbal/IPS

setembro 2021

Dedico esta dissertação a todos aqueles que possam vir a beneficiar dela. Espero que seja útil e ajude a tornar o mundo um lugar mais sustentável.

Acknowledgements

I wish to thank all the people who helped me conclude this study.

My family and friends, who were always supportive and motivating, and cats, who witnessed the growing of this work, looking over my shoulder, and meowing their suggestions.

I thank my dissertation's supervisor, Prof. Rui Madeira, who was always available, and helpful, directing me in the search for the most focused, rigorous, and complete work.

IPS and ESTSetúbal provided a knowledge base and tools, together with a stimulating research environment, and a team of human resources, allowing this research.

CRESCE COM AMOR therapy clinic, namely therapists Carolina Bernardo and Ilona Tóth, and all the children were invaluable. This work is for them.

Abstract

Children with special needs must perform physiotherapy exercises for rehabilitation and regaining functionality. Due to the repetitive nature of the rehabilitation and the characteristics of the population, serious games (SG) are used to motivate and promote engagement while exercising. The complexity and extreme variation of focused pathologies require flexibility and adaptation strategies from solutions. Therapists and curators can benefit from more intuitive tools allowing a more efficient usage of technology.

This research proposes the conceptualization of a framework model to support the design of serious games for children with special needs and a platform that implements this concept. The model identifies four major areas: therapeutic domain, technology, game, and analysis. The framework abstracts the technology allowing the use of multiple sensors and actuators, easing the implementation of smart environments. The model includes the definition of games based on levels and sequences of actions, which model therapeutic exercises.

The platform allows the creation of game actions towards specific therapies. The setting of the game action scope, concerning the intended therapies, allows the system to suggest particular actions based on the patient profile and current needs. The platform allows posterior view and analysis of recorded results. A comparative analysis of patient’s platform usage data and results is possible. Platform usage data and results can be shared and compared between clinics. An API allows external applications and devices to integrate with the platform.

A study with patients using a game connected to the platform and obtained results are presented and discussed in this document.

Keywords: Serious games, special needs, children, web-based platform, therapy, cerebral palsy.

Resumo

As crianças com necessidades especiais devem efectuar exercícios de fisioterapia para reabilitação e recuperar funcionalidades. Devido à natureza repetitiva da reabilitação, e às características da população-alvo, são utilizados jogos sérios para motivar e promover a adesão durante os exercícios. A complexidade e extrema variação das patologias focadas requerem flexibilidade e estratégias de adaptação para as soluções. Os terapeutas e cuidadores podem beneficiar de ferramentas mais intuitivas permitindo um uso mais eficiente da tecnologia.

Esta dissertação propõe a concepção de um modelo de enquadramento para apoio no desenho de jogos sérios para crianças com necessidades especiais e de uma plataforma computacional que implemente o conceito. O modelo identifica 4 áreas principais: domínio, tecnologia, jogo e análise. O enquadramento abstrai a tecnologia permitindo a utilização de múltiplos sensores e actuadores, facilitando a implementação de ambientes inteligentes. O modelo inclui a definição de jogos baseados em níveis e sequências de acções. As acções modelam exercícios terapêuticos.

A plataforma permite criar acções de jogo distintas para terapias específicas. A definição do âmbito do jogo, relativamente às terapias-alvo, permite ao sistema sugerir acções específicas, com base no perfil do paciente e as suas necessidades no momento. A plataforma permite a posterior visualização e análise dos resultados. É possível a análise comparativa dos resultados, e dados de utilização, entre pacientes. Resultados e dados de utilização da plataforma podem ser partilhados e comparados entre clínicas. Uma API permite que aplicações e dispositivos externos possam ser integrados com a plataforma.

Foi efectuado um estudo com pacientes utilizando um jogo ligado à plataforma e os resultados são apresentados e discutidos neste documento.

Palavras-chave: Jogos sérios, necessidades especiais, crianças, plataforma Web, terapia, paralisia cerebral.

Index

Figures

Tables

Acronyms

ADHD	*Attention-Deficit Hyperactivity Disorder*
AI	*Artificial Intelligence*
API	*Application Programming Interface*
AR	*Augmented Reality*
AV	*Augmented Virtuallity*
BAN	*Body Area Network*
BFMF	*Bimanual Fine Motor Function*
BMI	*Brain-Machine Interface*
CCI	*Child-Computer Interaction*
CFCS	*Communication Function Classification System*
CGI	*Computer-Generated Imagery*
CP	*Cerebral Palsy*
DBMS	*Database Management System*
DL	*Deep Learning*
DM	*Data Mining*
DSL	*Domain-specific Language*
DSML	*Domain-specific Modelling Language*
EMG	*Electromyography*
ESTS	*Escola Superior de Tecnologia de Setúbal*
ETL	*Extract, Transform and Load*
FPS	*Frames Per Second*

GMFCS	*Gross Motor Function Classification System*
GMFCS-E&R	*Gross Motor Function Classification System – Expanded & Revised*
GML	*Graphical Modelling Language*
GPL	*General-purpose Programming Language*
HCI	*Human-Computer Interaction*
HMI	*Human-Machine Interaction*
HTTP	*HyperText Transfer Protocol*
ICF	*International Classification of Functioning, Disability and Health*
ICT	*Information and Communications Technology*
IMU	*Inertial Measurement Unit*
IoT	*Internet of Things*
IR	*Infrared*
IS	*Information System*
JPK	*Just Physio Kidding*
JPKT	*Just Physio Kidding Therapist*
JPKW	*Just Physio Kidding Web*
LSS	*Level of Sitting Scale*
MACS	*Manual Ability Classification System*
MDD	*Model-Driven Development*
MDE	*Model-Driven Engineering*
ML	*Machine Learning*
NoSQL	*Not only SQL Databases*
NUI	*Natural User Interface*

OOP	*Object-Oriented Programming*
QUEST	*Quality of Upper Extremity Skills Test*
REST	*Representational State Transfer*
RCT	*Randomized Controlled Trial*
RGB	*RGB color model (Red Green Blue color components)*
SDK	*Software Development Kit*
SG	*Serious Game*
SLAM	*Simultaneous Localization and Mapping*
SMA	*Spinal Muscular Atrophy*
SPCM	*Seated Postural Control Measure*
SUS	*System Usability Scale*
SXGA	*Super Extended Graphics Array*
TUI	*Tangible User Interface*
UI	*User Interface*
URL	*Uniform Resource Locator*
USB	*Universal Serial Bus*
UX	*User Experience*
VGA	*Video Graphics Array*
VR	*Virtual Reality*
XML	*eXtended Markup Language*
WHO	*World Health Organisation*
WSN	*Wireless Sensor Network*

Chapter 1

Introduction

This research aims to provide tools for helping therapists and curators, and children with special needs successfully achieve their rehabilitation interventions, contributing for a better quality of living. The dissertation's focus is the conceptualization of a framework model to support the design of serious games for children with special needs and a platform that implements this concept.

1.1. Problem

Patients with special needs (SN) present conditions that limit their skills, in some way, during some time. These conditions can be chronic and temporary [1]. Patients with SN can be recovering from temporary injuries like broken arms, episodic events like stroke, or dealing with chronic conditions like cerebral palsy (CP). Patients with this kind of impairments cannot usually use the generic e-health tools available and need specific solutions. Solutions designed for patients with SN must be adaptive by nature, allowing extra degrees of personalization, to be efficiently used by most patients [2], who often require interventions in multiple simultaneous therapy areas.

Users with CP require SN [3] and most children with CP live with relevant motor and posture impairments. It is a chronic condition and physiotherapy is one of the main tools for motor function rehabilitation and cognitive stimulation. Children have distinct characteristics than adults [4]. There is a need for specific tools directed for children with CP. Tools addressing the main factors at stake (e.g., children, CP, therapy protocols, progression monitoring and analysis) are rare and expensive.

One of the reasons for patients losing focus and interest in physiotherapy exercises is their repetitive nature. Having to perform the same exercise for a long time is tedious and tiresome for anyone, which is even more relevant when dealing with children and specifically children with SN.

1.2. Motivation

Motivation for this research is built on the mentioned factors, trying to provide solutions to address them.

A frequently used technique when trying to promote user engagement and motivation is the gamification of tasks [5]. Another way to motivate users is through the implementation of particular serious games (SG), which are games where the main purpose is not entertainment [6]. The game concept is used to integrate what would otherwise be a set of long and tedious exercises. Naturally, the game itself must be appealing, dynamic and progressive to capture the target audience's attention and keep that focus. Actuation devices are used as a mean for creating more immersive environments [7], providing other means of interaction with users, and promoting interest and motivation.

Customization and personalization are also key factors when designing solutions for children with SN. Systems must adapt to multiple patients with multiple degrees of conditions. Even applications that focus on only one impairment must deal with the distinct and unique condition of each patient. Developing inclusive technology is needed for providing functionalities to users that cannot find answers on current solutions.

The technological and social advances of the XXI century allow for more efficient interventions in the context of e-health. The Internet of Things (IoT) revolution is changing the way we interact with technology. Technology is becoming embedded in the environments making these spaces smart. Smart environments and smart rooms are spaces that continuously interact with users, responding and adapting, in a non-intrusive way, providing an extra level of user experience [7]. Smart spaces used for e-health use sensors to assess and monitor patient’s status and actions and respond to users’ requests adapting according to therapeutic goals. Data analytics and machine learning (ML) techniques are used to extract information from the sampled data, generating knowledge so systems can learn from it. Therefore, data availability and quality are of utmost importance.

Moreover, the continuity and improvement of the line of research already pursued in previous work is needed since specific needs of therapists and patients have been identified during sessions and interviews at a stakeholder clinic.

1.3. Solution

The dissertation’s focus is the conceptualization of a framework model to support the design of serious games for children with special needs and a platform that implements this concept. The model identifies 4 major areas: therapeutic domain, technology, game, and data analysis. The framework abstracts the technology allowing the use of multiple sensors and actuators, easing the implementation of smart environments. The model includes the definition of games based on levels and sequences of actions. The actions model therapeutic exercises. The platform allows the creation of game actions applicable for specific therapies. The setting of the game action scope, regarding the intended therapies, allows the system to suggest specific actions, based on the patient profile and current needs. The platform allows posterior view and analysis of recorded results. A comparative analysis of patient's usage and results is possible. Platform usage data and results can be shared and compared between clinics.

Expected results are a framework model and a platform prototype providing tools for applying serious games as a complement to traditional therapy interventions. The platform should provide a REST API for interaction with external applications and devices. Validation tests with patients and therapists should be performed.

1.4. Research questions

Considering the problem and challenges presented, this research is expected to provide answers to some questions:

RQ1: Can this framework provide a more intuitive tool, abstracting technology requirements, allowing therapists to design serious games, by using a specific modeling language?

RQ2: Does the model of action and parameters flexible enough to allow representation of therapeutic exercises?

RQ3: Can this framework help therapists design serious games for children with special needs, by suggesting relevant actions, based on the patient profile?

RQ4: With this platform, can therapists benefit from being able to compare results and exercise samples between patients?

RQ5: Can developers allow their applications to integrate the platform by consuming an API?

This research provides a conceptual model and an implemented computational platform as tools for trying to answer these questions.

1.5. Main contributions

The main contributions made by this work are the following:

- A **framework model** was conceptualized and a **computational platform** was implemented. The model includes the definition of games based on levels and sequences of actions of therapeutic exercises. The platform implements the model and allows for particular games definition towards personalized therapies. The platform was deployed online, providing most of it's functionality, and therapists and patients can already use it. The API allows the integration of external applications and devices.
- A **study with children** using a game connected to the platform is presented in this dissertation.
- The **model and platform are being used** by Gonçalo Roxo in the research for his

Masters' Thesis, at NOVA Sciences and Technology School. The implemented research prototype integrates with the PLAY platform to get game structure and configuration data and share results and samples.

- **Publications**: two papers have been written and submitted to international conferences for dissemination of achieved results and contributions.
    - A full paper, "PLAY – Model-driven Platform to Support Therapeutic Serious Games Design", was submitted and accepted to the International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH2021) in Leuven, Belgium, in November 2021.
    - A work in progress (WIP) paper, "Modeling Serious Games Design towards Engaging Children with Special Needs in Therapy", was submitted and accepted to CHI PLAY 2021 (ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play), occurring on October 2021, in New York (virtual format).

1.6. Document structure

This document contains six chapters with the following structure:

- Chapter 2 (Background) makes an overview of the background, with focus on the domain, gaming and technology areas, and an analysis of relevant related work.
- Chapter 3 (Concept) presents the conceptual model, explaining the multiple contextual areas and relations, and detailing the main components.
- Chapter 4 (Platform) presents the developed platform prototype, and a brief description of platform functionalities and usage. Games in the platform are instantiations of the proposed model and are described as integration examples.
- Chapter 5 (Results) presents the results obtained during this research and validation tests with therapists.
- Chapter 6 (Conclusions) presents the conclusions of this research and introducing some research pointers for possible future work.

Chapter 2

Background

Multiple areas of knowledge and technologies are needed to build e-health solutions. Definition for a person with special needs and the concepts of functionality and disability depend on several components. The World Health Organization (WHO) provides a classification system designated International Classification of Functionality, Disability and Health (ICF). The ICF classification system is a valuable tool to understand the component relations and dependencies. Several pathologies associated with special needs are also presented, to have a clear picture of the impairments, limitations, and constraints of patients with those conditions. Cerebral palsy patients require special needs. Cerebral palsy patients can present distinct and multiple simultaneous conditions, covering almost all aspects of special needs. Relevant aspects of cerebral palsy in the context of rehabilitation are described.

When researching about solutions for children with special needs, there is a need to understand what is distinct about children as an intervention target. Specifically, in which ways a child differs from a teenager or adult, in the context of e-health rehabilitation application design. There is also a need to understand how children with special needs differ from children that do not have those needs, in the context of the use of technology. It is very important to know how these impairments affect user interaction. HCI technologies are the basis for interface design. Gamification and serious games are techniques used for promoting user engagement and motivation. This chapter makes an overview of serious games and gamification in the context of physiotherapy for children with special needs. Games and game design have relevant insights that are discussed.

Technological advances provide the tools to build more efficient e-health solutions. The pervasive and ubiquitous computing paradigms are each day more present. Smart environments are a state-of-the-art use of multiple technologies for e-health solutions in rehabilitation. Sensors are used for tracking and measuring user interaction and biometric signals. NUI sensors are used to provide more natural ways of interaction. Actuators can be used to extend application feedback to users in a more immersive way. Smart gateways are used for interfacing between sensors and actuators and provide additional services. An overview of available sensors for e-health applications focusing on NUI is made. Biometric and wearable solutions are also considered. Sensor fusion is used to provide solutions to multiple rehabilitation needs. A lookup on some relevant literature allows to perceive the applications and usages of such techniques. Overcoming the screen limitation is achieved using gateways, the internet, or some other communication technology, to control external devices, promoting more immersive environments. Other emerging technologies allowing innovative approaches in e-health applications are considered and analyzed. Data analytics, AI and ML can be used for detecting patterns, extracting information and learn. This allows applications to adapt adjusting to users' needs. Some relevant related work is discussed.

Relevant preliminary work on a related project is presented. This external platform is later used as example of instantiation of the proposed model.

2.1. Children with special needs

The combining factors of the target audience being children and requiring special needs, makes interventions in this domain very challenging. It is vital to understand children and their characteristics. Special needs is a very broad term that must be well perceived to understand all the implications and related constraints.

2.1.1. *Children*

Children as a target audience have specific characteristics that must be accounted for. Children have specific interests and life experience. The things, interests and emotions that drive the user's actions are different in a child or an adult.

Child-Computer Interaction (CCI), a more specific area of HCI, is an field of research related with studies in design and evaluation of interactive systems developed for children [4]. The areas where CCI research is more relevant are education and health. The motivation behind CCI research is to find how can we design products addressed to the needs of the children and offer them a positive experience using them. The challenges presented are not only the design of products for children but also the fact that these are children with limitations and impairments. Products for children must be innovative, interesting, and attractive enough to achieve the goal of stimulating the child to learn or perform the required task.

Children have different learning strategies and preferences that should be taken in account. The special needs mean some disability or disabilities that require distinct ways of communication and interaction. This leads to the importance of a participatory design where users are part of the product design process. Children develop their cognitive ability in different ways and speeds and can vary a lot from traditional models. The model Piaget proposed, despite being generalist, is useful in establishing some development stages. Jean Piaget, swiss psychologist, was a leading player in children cognitive development analysis and research. He showed that children present knowledge and experience gaps and understand the world in a very different way from adults. Piaget's model [8] establishes 4 stages during child cognitive development (Table 1).

Table 1 - Piaget stages of child cognitive development

Stage	Age	Relevant cognitive characteristics
Sensorimotor	0-2	cognitive process very dependent on perception through senses low interaction skills
Preoperational	2-7	short span attention capability only one thing in memory at a time difficulty of abstraction primary reading and writing skills
Concrete Operational	7-11	unable to formulate hypothesis difficulty on complex abstractions associative capabilities reading and writing capabilities
Formal Operational	11+	cognitive processes are like those of adults taste and interests differ from those of adults

Using age as a guide can be useful. We must take into account, however, that designing an application for a too young audience can be as problematic as designing for a too old audience. Children are very well capable of assessing their own skills. When faced with technologies designed for younger audiences, it can be considered offensive or not interesting. Each age presents different HCI challenges.

Cognitive process development is important but not the only issue conditioning children use of technology. Cultural differences in each age group, and their preferences and interests, are also relevant factors. Understanding what is interesting for a child means acknowledging their cognitive abilities and the cultural dependent aesthetic sensibilities. Child characteristics relevant for HCI research context are dexterity, speech, reading, experience, and interaction style.

Technology for children is usually applied in education or entertainment. In products with both purposes the term used is edutainment. Child-oriented products can include specialized software and/or hardware. The use of Tangible User Interfaces (TUI) when used in education with young children have been systematically reviewed in [9], with the analysis of 155 studies published between 2001 and 2019.

2.1.2. *Understanding special needs*

Children requiring special needs are defined as those who have a disability, disorder, or a combination of these that makes learning or other activities difficult. The current US Electronic Code of Federal Regulations (CFR) [10] defines *Child with a disability* as a child "having an intellectual disability, a hearing impairment (including deafness), a speech or language impairment, a visual impairment (including blindness), a serious emotional disturbance (referred to in this part as “emotional disturbance”), an orthopaedic impairment, autism, traumatic brain injury, another health impairment, a specific learning disability, deaf-blindness, or multiple disabilities, and who, by reason thereof, needs special education and related services.” In Europe, current regulation follows the International Classification of Functioning, Disability and Health (ICF) convention from the World Health Organisation (WHO) to define functioning and disability (Table 2).

Table 2 - WHO ICF Term definitions in the context of health

Term	Description
Functioning	Umbrella term for body functions, body structures, activities, and participation. It denotes the positive aspects of the interaction between an individual (with a health condition) and that individual’s contextual factors (environmental and personal factors).
Disability	Umbrella term for impairments, activity limitations and participation restrictions. It denotes the negative aspects of the interaction between an individual (with a health condition) and that individual’s contextual factors (environmental and personal factors).
Body functions	The physiological functions of body systems (including psychological functions).
Body structures	Anatomical parts of the body such as organs, limbs, and their components.
Impairments	Problems in body function and structure such as significant deviation or loss.
Activity	The execution of a task or action by an individual. Participation - Involvement in a life situation.
Activity limitations	Difficulties an individual may have in executing activities.
Participation restrictions	Problems an individual may experience in involvement in life situations.
Environmental factors	The physical, social, and attitudinal environment in which people live and conduct their lives. These are either barriers to or facilitators of the person’s functioning.

The ICF provides a contextual model of the component’s interactions revealing the importance of each individual component and the possibility of interaction between all of them. The multidimensionality of the disability is evidenced in the diagram, shown in Figure 1.

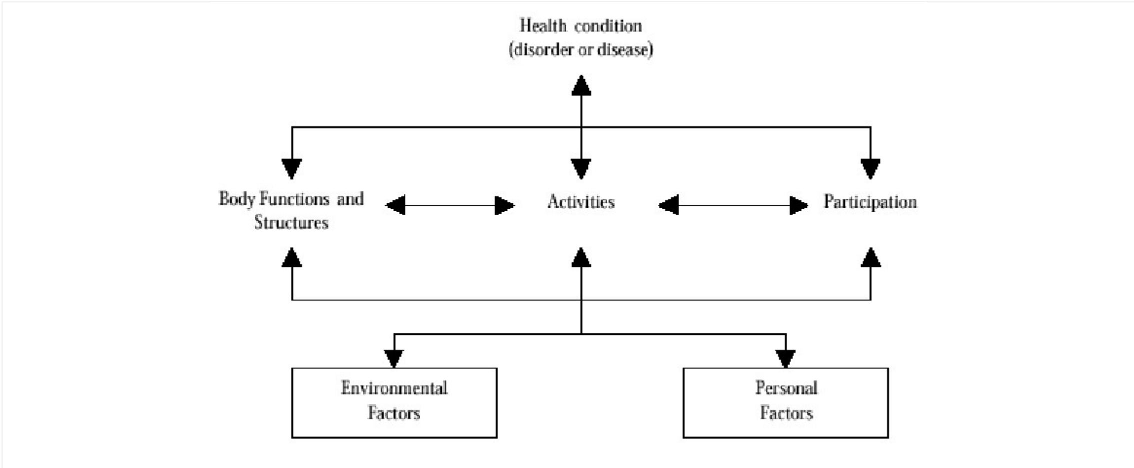

Figure 1 - Interactions between the components of ICF (WHO 2001:18)

The ICF is divided in two parts, and each part has two components. Impairments can also refer to delays or lags in the development of body functions, during childhood and adolescence. The first part of the ICF deals with functioning and disability, with two components, one focusing on the functions and structures of the body, and the other in activities and participation. The second part of the ICF allows classification of contextual factors, also with two components, the first one dealing with environmental factors and the second with personal factor. All components can be expressed in positive or negative terms, as can be observed in Table 3.

Table 3 - ICF overview

	Part 1		Part 2	
	Functioning and Disability		**Contextual Factors**	
Components	Body Functions and Structures	Activities and Participation	Environmental Factors	Personal Factors
Domains	Body functions Body structures	Life areas (Tasks, actions)	External	Internal
Constructs	Physiological changes in body functions Anatomical changes in body structures	Capacity Executing tasks in a standard environment Performance Executing tasks in the current environment	Facilitating or hindering impact of features of the physical, social, and attitudinal world	Impact of attributes of the person

<table>
<tr><td rowspan="2">Positive</td><td>Functional and structural integrity</td><td>Activities participation</td><td rowspan="2">Facilitators</td><td rowspan="2">-</td></tr>
<tr><td colspan="2">Functioning</td></tr>
<tr><td rowspan="2">Negative</td><td>Impairment</td><td>Activity limitation Participation restriction</td><td rowspan="2">Barriers and/or hindrances</td><td rowspan="2">-</td></tr>
<tr><td colspan="2">Disability</td></tr>
</table>

Targeting these children present an extra challenge than traditional children when designing applications. The ICF has a specific version for children and youth [1]. The classification system has two levels (Table 4).

Table 4 - ICF One-level classification

Context	**Description**
Body functions	Mental functions; Sensory functions and pain; Voice and speech functions; Functions of the cardiovascular, hematological, immunological, and respiratory systems; Functions of the digestive, metabolic and endocrine systems; Genitourinary and reproductive functions; Neuromusculoskeletal and movement-related functions; Functions of the skin and related structures
Body structures	Structures of the nervous system; The eye, ear, and related structures Structures involved in voice and speech; Structures of the cardiovascular, immunological, and respiratory systems; Structures related to the digestive, metabolic and endocrine systems; Structures related to the genitourinary and reproductive systems; Structures related to movement; Skin and related structures
Activities and participation	Learning and applying knowledge; General tasks and demands; Communication; Mobility; Self-care; Domestic life; Interpersonal interactions and relationships; Major life areas; Community, social and civic life
Environmental factors	Products and technology; Natural environment and human-made changes; Support and relationships; Attitudes; Services, systems, and policies

The ICF has a generic scale for classification that can be seen in Table 5. This is a scale with 5 steps, from 0 to 4, with 0 being the worst condition. The scale is applicable for measuring body functions and structures, activities and participation, and environmental factors.

Table 5 - ICF generic qualifier scale

Context	Code	Description	Scale
Body functions, Body structures, Activities and participation, Environmental factors	0	No	0-4%
	1	Mild	5-24%
	2	Moderate	25-49%
	3	Severe	50-95%
	4	Complete	96-100%
	8	Not specified	-
	9	Not applicable	-

Multiple pathologies require special needs, as can be seen in Table 6.

Table 6 - Pathologies requiring special needs

Pathology	Impairments
Cerebral palsy	Severe motor and cognitive impairments
Autism, ADHD	Behavioral disorders
Down syndrome, Asperger syndrome	Cognitive impairments
Deafness	Hearing impairments
Blindness	Visual disorders with or without total loss of vision
Dyslexia	Reading disorder
Dyspraxia	Developmental coordination disorder
Cystic fibrosis	Breathing issues

2.2. Cerebral palsy and related pathologies

Cerebral palsy (CP) refers to a group of posture and movement disorders that are attributed to nonprogressive disturbances that occurred in the developing fetal or infant brain [3]. First description of the pathology is credited to William John Little, an English orthopedic surgeon, in 1843 [11]. CP has been a field of study for, at least, the last 170 years. This condition presents significant motor disorders associated with disturbances of sensation, cognition, communication, perception, behavior, and seizure disorders. It is the most common cause of chronic disability in children occurring 2 to 2.5/1000 births [12].

Multiple factors contribute to this condition. Some relevant maternal predisposing factors are chronic diseases like diabetes, epilepsy, or mental illness. Events before and during pregnancy can increase the risk. Even during birth, a bad positioning of the fetus or vaginal infections can be relevant factors. Premature newborns weighting less than 1500g have the risk of developing the condition increased 20-fold. Genetic forms of cerebral palsy can be hereditary and multiple genetic factors have been associated with the condition. Some causal factors are environmental conditions, gene anomalies, metabolic dysfunctions, infections, traumatic delivery, and hypoxic ischemic encephalopathy, that is the most common cause of brain damage in newborns.

2.2.1. *Cerebral disorders*

Several disorders are associated with cerebral palsy, as shown in Table 7. Most patients do not work at all and present a severe cognitive delay.

Table 7 - Cerebral Palsy associated disorders

Disorder	**Description**
Motor features	Voluntary and involuntary movements Tonus anomalies Uncoordinated reflexes and strength Graphomotor capacity anomalies
Cognitive	Language Difficulty in solving problems Low verbal and non-verbal quotient Memory deficit
Behavioral	Attention deficit Poor control Social interaction deficit Aggressivity
Communication	Speech and hearing disorders

2.2.2. *Motor impairments*

The different degree of affection and areas affected determine the types of impairments the patient presents. Definition and classifications of cerebral palsy and impairments have evolved over the years and with research. There are several relevant classifications based on area of analysis. Some term definitions help understand the kind of impairments and their consequences to patient's motor functionality. Depending on the severity of the motor impairment it can be a paresis or paralysis. Paresis is a muscular weakness that prevents or makes the movement more difficult. Paralysis is the complete loss of muscle function.

CP classification based on location of lesions

- **Hemiparesis** or Monoparesis is the weakness of one entire side of the body.
- **Hemiplegia** is the complete paralysis of one entire side of the body.
- **Diplegia** is the paralysis of symmetrical parts of the body.
- **Quadriparesis** or Tetraparesis is the weakness of all members and torso of the body.
- **Quadriplegia** or Tetraplegia is the paralysis of all members and torso of the body.

CP classification based on muscular tonus

- **Dyskinetic** is the status of involuntary and recurring movement.
- **Ataxic** is the inability to activate the correct muscles for a movement.
- **Spastic** is characterized by resistance to forced joint movement that varies with speed and direction.
- **Mixed** is a combination of two or more of the previous conditions.

Positive signs are those who lead to involuntary movement or movement pattern.

- **Chorea** is derived from an ancient Greek word meaning dance and describes quick movements of the feet and hands.
- **Tremor** is an involuntary muscle contraction and relaxation. Can affect hands, arms, eyes, face, head, vocal folds, trunk, and legs.

Negative signs are those that generate insufficient muscle activity or control.

- Weakness
- Reduced selective motor control
- **Ataxia** is the inability to activate the correct muscles for a movement.

Movement resistance

- **Spasticity** is characterized by resistance to forced joint movement that varies with speed and direction.
- **Rigidity** is determined by resistance to forced joint movement, independent of speed, strength, or joint angle. There is no return of limb to previous state when forced to change position.

Involuntary movements

- **Dyskinetic** is the status of involuntary and recurring movement.
- **Athetosis** is characterized by involuntary, convoluted, and writhing movements of the fingers, hands, toes, and feet. In some cases, also the arms, legs, neck, and tongue are affected.
- **Ballismus** is characterized by repetitive, involuntary ballistic movements, often violent and with large amplitude. It is common for arms and legs to move together.
- **Dystonia** is characterized by involuntary muscle contractions leading to twisting, repetitive movements and/or abnormal postures.

Users requiring special needs have impairments limiting their actions meaning that traditional interaction methods are usually not applicable. It is common for patients to have multiple impairments. The specific user pathologies can make it difficult to hold a mouse or use the keyboard. Usage of non-intrusive sensors is required to effectively capture user interaction without affecting that same interaction.

2.2.3. *Related Pathologies*

Motor disorders have associated pathologies and disturbances that can occur in some other areas. These disorders include disturbances of sensation, perception, cognition, communication, and behavior. Epilepsy and secondary musculoskeletal problems are also related conditions. Hearing impairment is a condition that is common on cerebral palsy patients (Table 8).

Table 8 - Cerebral Palsy related pathologies

Pathology	**Description**
Seizures	Distinct seizure categories: absence, atonic, clonic, myoclonic, tonic-clonic. All categories present distinct involuntary motor features.
Dysphagia	Patients have swallowing difficulties, usually with pain. Gagging and coughing, regurgitating, drooling, weight loss, are some of the issues.
Oral health	Gingivitis, tooth decay and cavities, drooling, mouth traumas
Vision problems	Difficulty reading or writing, learning issues. Low vision.
Cognitive	Attention, behavior, and emotion disorders. Comprehension and decision-making issues. Memory and recognition issues.
Digestive	Constipation, incontinence, swallowing issues, vomiting, aspiration
Skin problems	Ulcers, infections, irritations, ringworms, boils, and pimples
Respiratory problems	Bronchitis, asthma, bronchopulmonary dysplasia, pneumonia
Behavior and Emotion disorder	Isolation, anger, depression, anxiety
Physical and motor impairments	The most common issues in patients with CP. From mild spastic movements to complete loss of mobility.
Autism	7% of cerebral palsy patients have autism. One of the most frequent forms of autism in CP patients is Asperger’s Syndrome.
Sleep disorders	Common in CP patients due to other pathologies issues.
ADHD	Impulsiveness and hyperactivity, lack of focus and attention

2.2.4. *Pathology evaluation*

Evaluation of the degree of this pathology has been using with the same method for 150 years. The method is based in relevant importance and motor affection. More recent methods of evaluation include etiology and timing, type, and severity of events. Not only the motor affection is considered but also the anatomical distribution and the associated dysfunction.

Variables considered when evaluating CP are motor involvement and muscular tonus. Muscular tonus can be spastic, dyskinetic, ataxic or mixed. Gross motor function and bi-manual fine motor activity as well as the use of hands in routine situations are also considered. There are scales for evaluation of communication, mode of walking and dystonic impairment.

The Gross Motor Function Classification System (GMFCS) uses a 5-level system to classify motor functionality, presented in Table 9. Proposed in 1997 by Palisano et al., was originally only applicable to age ranges 0-2 years and 2-12 years. The GMFCS was subject of revision in 2007 by Palisano et al. [13], to include an extra band for ages 12 to 18, increasing the range of application. The new revised version is designated GMFCS-E&R. This tool is valid for CP patients only.

Table 9 - Gross Motor Function Classification System (GMFCS)

Level	Description
I	Can walk indoors and outdoors and climb stairs without using hands for support Can perform usual activities such as running and jumping Has decreased speed, balance, and coordination
II	Can climb stairs with a railing Has difficulty with uneven surfaces, inclines or in crowds Has only minimal ability to run or jump
III	Walks with assistive mobility devices indoors and outdoors on level surfaces May be able to climb stairs using a railing May propel a manual wheelchair and need assistance for long distances or uneven surfaces
IV	Walking ability severely limited even with assistive devices Uses wheelchairs most of the time and may propel own power wheelchair Standing transfers, with or without assistance
V	Has physical impairments that restrict voluntary control of movement Ability to maintain head and neck position against gravity restricted Impaired in all areas of motor function Cannot sit or stand independently, even with adaptive equipment Cannot independently walk but may be able to use powered mobility

The Manual Ability Classification System (MACS) is a tool for classifying the use of hands, manipulating objects in daily activities, in children aged in the range of 4-18 years old. The system has 5 levels, with 1 being the level where the child can manipulate objects with ease and successfully and 5 when the child cannot manipulate object at all and has severe limitations on any activity, requiring total assistance. The Mini-MACS is a specific version of the system directed for children aged 0-4 years old [14].

Table 10 - Manual Ability Classification System (MACS)

Level	Description
I	Handle objects successfully with ease
II	Handle most objects but with less quality and/or speed
III	Handle objects with difficulty
IV	Handle selected objects needing adaptation
V	No handling of objects and severe limitations in any action

The Seated Postural Control Measure (SPCM) and the Level of Sitting Scale (LSS), proposed by Fife et al. in 1991 [15], are used to assess posture and functionality. SPCM and LSS were also a subject of further study and validation, in 2011, by Field et al. in [16]. The SPCM is a scale of 22 seated postural alignment items and 12 functional movements. Items can score in 4 levels, from 0 to 3, with 0 being severe and 3 normal.

Table 11 - Level of Sitting Scale (LSS)

Level	Description
I	unplaceable
II	supported from head downward
III	supported from shoulder or trunk downward
IV	supported at pelvis
V	maintains position, does not move
VI	shifts trunk forward, re-erects
VII	shifts trunk laterally, re-erects
VIII	shifts trunk backwards re-re-erects

The Communication Function Classification System (CFCS) follows the guidelines of GMFCS and MACS and is used for evaluating communication abilities in patients with CP. Has a 5-level scale with 1 representing the less severe and 5 the most severe level of disability.

Table 12 - Communication Function Classification System (CFCS)

Level	Description
I	A person independently and effectively alternates between being a sender and receiver of information with most people in most environments.
II	A person independently alternates between being a sender and receiver with most people in most environments, but the conversation may be slower.
III	A person usually communicates effectively with familiar communication partners, but not unfamiliar partners, in most environments.
IV	The person is not always consistent at communicating with familiar communication partners.
V	A person is seldom able to communicate effectively even with familiar people.

The Quality of Upper Extremity Skills Test (QUEST) is used for accessing patient performance of exercises. Therapists can use it to measure dissociated movements, grasps, weight bearing or protective extension. Individual scores of the tested areas as well as the mean score are registered. The areas evaluated with the QUEST tool are:

- Dissociated Movements
- Grasps
- Weight Bearing
- Protective Extension
- Hand Function
- Spasticity
- Cooperativeness

2.2.5. *Physiotherapy and Rehabilitation*

Patients with some medical conditions, motor pain or motor disabilities need physiotherapy to recover lost functionality. Physiotherapy, also known as physical therapy, is used for improving patients' physical functions. Physiotherapists apply physical interventions for rehabilitation and patient education, making use of physical observation, diagnosis, and prognosis for that effect. Physiotherapy focuses on the treatment of chronic and acute pain conditions and physical impairments of multiple origins.

Rehabilitation is the process of restoring a previous state or condition. Depending on the pathologies, the rehabilitation process is usually a multidisciplinary work. Professionals involved can include medical doctors, physiotherapists, occupational therapists, psychologists, speech therapists, prosthetists, sports coaches, and social workers.

The rehabilitation lifecycle usually begins with an evaluation of kinetic functionality, checking movements and motor functions. After the initial evaluation, therapists can develop a rehabilitation plan, identifying and scheduling the needed exercises and movements for patient recovery. Duration of therapy and time for recovery varies and depends on cause and degree of disability. Therapies can last from a few weeks to several months. The same exercise movements must be performed repeatedly to promote kinetic improvement. Repetition of the movements is important in motion recovery because it helps to develop some muscle and joint gains and implicit patient learning of the function. Patients' performance of exercises is evaluated by the therapist. The efficacy of the therapy plan is verified and re-adjusted as needed.

Time spent in clinical therapy is much less than home therapy time. Patient usually exercise in their home environments. Home training is usually not supervised. This leads to several issues. Assessment of the performance quality is one such issue. The quality of the exercise performance itself, the possibility of doing wrong exercises and the compensation of movements with some wrong body position or movement are variables to consider. Without being physically present, the therapist has difficulty in determining if the exercises were correctly performed. Motivation is another great challenge. The therapist can help motivate the patient, using voice and gestures, when exercising in the clinic, but when the patient is alone at home that does not happen. All these issues are relevant when performing therapy. Technology is being used to try to address some of these issues. Interactive systems are being developed trying to promote engagement and motivation of patients. These tools help therapists complement their conventional therapies.

Interactive systems used in this context implement augmented or virtual reality solutions, using biosensors and biofeedback tools. The haptic glove is the most used wearable device. In the 2015 study by daGama et al. [17], 31 systems were referenced, 30 being assistive systems for rehabilitation.

Exercises are usually modeled using some identified techniques:

- Avatar control or game
- Predefined task stimulating one or more movements
- Mimic a guided task, reference video or generated image

Movement recognition and assessment of movement performance are applied using several techniques:

- Tracking of hand position
- Tracking upper limb joints on reach activities
- Tracking the angle computed on the relevant plane for the movement

Patients in need of physical therapy can benefit from exercises for posture, balance, gait, and functionality assessment. Speech therapy can exercise oral production and articulation and has been combined, in some cases, with Electromyography (EMG) for muscle area activity detection or Galvanic Skin Response (GSR) for detecting psychological and emotional responses. Music therapy can be used to exercise memory and cognitive competence or monitoring active cerebral areas or biofeedback stress [18] [19]. Occupational therapy can be used measuring graphical evaluation, activities of daily living (ADL) evaluation or sensorial evaluation. Cognitive therapy is used for cognitive stimulation (Table 13).

Table 13 - Common exercise areas by therapy

Therapy	Exercise Areas
Physical	posture; balance; gait; functionality
Speech	EMG, sweat, oral production, articulation
Occupational	graphical evaluation, ADL evaluation, sensorial evaluation
Music	memory; cognitive competence; active cerebral areas, biofeedback stress
Cognitive	cognitive stimulation

Gamification techniques and serious games can be used to promote engagement and motivation for the execution of the often long and repetitive exercises. Videogames have been identified as the preferred means of leisure by adolescents with CP in a study by Shikako-Thomas et al. [20].

2.3. Game Design

What are games? Sociological and psychological aspects of games are the subject in the field of game studies. In an essay called 'Homo Ludens', from 1938, the historian Johan Huizinga defines a game as voluntary activity, apart from ordinary life, but capturing the user focus and attention. No material interest or profit is gained. Has its own rules and time and space boundaries. Promotes the creation of social groupings sharing that common interest. In an article from 1975 [21], Avedon and Sutton-Smith describe a game an "*exercise of voluntary control systems in which there is an opposition between forces, confined by a procedure and rules in order to produce a disequilibrial outcome.*". A more recent definition from Salen and Zimmerman, in 2004, describes a game as "*a system in which players engage in an artificial conflict, defined by rules, that result in a quantifiable outcome*" [22].

Play or the act of playing is a slightly different concept. An analogy using OOP terms can be made interpreting play as an instantiation of a game. Games and play are directly related.

Games, in general, follow a set of principles:

- Rule-based system
- Existence of variable and quantifiable outcomes
- Different outcomes are assigned different values
- Player makes effort to influence outcome
- Player feels emotionally attached to the outcome
- Consequences of activity are optional and negotiable

Designing and implementing a game is not a trivial task, integrating multidisciplinary knowledge and expertise. That does not mean that a game cannot be designed and implemented by a single person. It is just common knowledge that is rare to master all aspects of game design. Different team members have distinct roles. Roles can include Level Designer, User Interface Designer, Mechanics Designer, Art Director and Developer among others. The following sections present a few guidelines that should be considered during the process of designing a game.

2.3.1. *Narrative*

What is the story? Narrative is a very important feature to account for when designing a game. Making the actions in the game contextualized can promote user engagement and immersion. It is this contextualization of game elements and dynamics that sometimes help promote motivation and interest. A game should have well defined stages for a supporting story. A prologue or backstory helps make an introduction and explain the game elements and gameplay. The story should develop during the game. There is a need to define what happens during the beginning (the early stages), middle (heart of the game) and ending (final stages). An epilogue is useful to prevent leaving the story in an unfinished state. It explains what happens after the game ends and how the game elements will remain after the user finishes the tasks.

2.3.2. *Mechanics design*

Game mechanics is the logic that drives the game. What movements should a game have? Game movements must reflect the kind of therapy exercise that is being modeled. If an exercise requires the hand to reach a specific location or move in a distinct way, a target object or goal can be placed in a position that promotes that movement to the user. Game mechanics refers to the way points are scored or how the player advances or retreats. This are the rules that are the basis for each scenario or level.

Gameplay modes define different types of playing interaction. A primary gameplay mode should be defined, and additional modes can be added afterwards, as needed. Gameplay modes consist of a series of definitions. These include where the game takes place and what are the interaction and camera models used. The challenges, actions, and mechanics used should also be defined. Secondary gameplay modes should answer an extra question explaining the reason and necessity for that gameplay mode to be present.

2.3.3. *Internal economy*

The key resources in a game define the internal economy. Examples are money, points, ammunition, health, energy among other. Each resource should be identified, and the methods of generation and consumption should be defined (how can it be generated or its origin and how it is consumed or where it goes). The adjustment of these resources to manage the level of difficulty of the game is a possibility. All variations should be identified. Positive feedback is very important as a motivational factor, and usage should be identified in which conditions and what kind of feedback to use.

2.3.4. *How to win or lose*

The conditions leading to a victory or loss should be clearly stated. This is valid for every single stage of the game where applicable. If these conditions change in different scenarios or levels that should be also mapped. Depending on the game type there could be relations or dependencies with other game variables. These dependencies can be related to internal economy elements of the game such as points or money, exploring a set of areas in the game space, collecting a set of objects, strategic supremacy, or other context related conditions. The player should be aware of what is needed to win and be capable of judging its progress. What are the mechanisms that make the player perceive this? There are games in which there are no victory conditions. In this case, a clear definition of the objective is needed. What is the player’s goal and what is he/she trying to achieve?

2.3.5. *Level design*

One way of promoting engagement and motivation is the use of levels in game design. The sense of moving to a next level is perceived as a reward. The current user's level is a kind of rank and distinctive factor among users. The levels should be designed to alter some game variable in a way that keeps increasing the challenging factor of the game but always keeping focus on the therapeutic considerations of the specific game/exercise being performed.

2.3.6. *User interface design*

The UI can be more neutral or more integrated with the game world aesthetic. The level of integration and support must be defined. User interface relevant items include buttons, sliders, numbers, clocks, lights, dials, pointers, type fonts among others.

The hardware for the game must allow for all the actions required. All the actions in the game world must be mapped onto the hardware selected. The most important actions must be identified and a way of allowing them to be performed with the control devices must be devised. The user interface layout must be determined, focusing the major gameplay modes and scenarios assuring that the important elements are present and the communication works. The interaction hardware must be determined, and the supported input devices identified. Options vary from traditional keyboard and mouse, joysticks, game controllers, NUI devices, VR devices among others. All hardware and special devices requirements should be identified.

Mockups are a great way of planning screen layout. They are a representation of what the screen layout will look like. Gameplay modes and levels should be previewed in this way assuring the presence of key information for the player. Examples of key information are score, health, map, time. Mockups can be complemented with textual descriptions for relevant information about the scene.

2.3.7. *Art Direction*

A game should have a coherent aesthetic style. The Art Director must define the aesthetic style of the game and the definition of key visual elements. The styling of the game should take in consideration different elements and tools to try to maintain game visual identity. The type of presentation of a game defines the graphical level of a game (e.g., photorealistic, simple cartoon, complex cartoon, pixelized, anime). The aesthetic style of the game world defines the general ambiance (e.g., industrial, medieval, science fiction). The colors used have multiple meanings and can be dynamic. Mood can be created using only the color palette (e.g., green for nature, blue and black for night). Lighting can also be used to influence mood or gameplay. Lighting effects include darkness, fog, candle flames, moonlight, lamps among others.

Game characters presentation should reflect their role and personality. The way characters are presented in the game presented must be defined. For instance, if they are people-like, they can be young, old, athletic, heroic, sexy, or ugly. All characteristics should be described. The role of the characters should be well supported and justified through their appearances.

The designing of the game packaging (could be physical, as a box, or digital, as a logo or other kind of promotional items) defines the image of the game for market promotion.

Concept drawings are a tool that help preview relevant game elements. They can be from simple sketches to finished art pieces, but the point is to make the team visualize key elements of the game to better model those elements digitally. Examples of relevant elements are the player's avatar, other characters, locations, or key items.

2.4. Gamification and serious games

Gamification and serious games are tools available when designing solutions to support therapy interventions. These techniques use gaming elements to promote user interest, motivation, and engagement.

2.4.1. *Gamification*

The concept of gamification is based on the use of game elements in non-game tasks. Similar notions started to be researched in the context of HCI during the 1980s, at the first boom of computer games, about the application of gaming user interfaces, narratives, or cover stories to make activities more interesting and appealing. The relation between fun and ease of use also started to be researched during those days. Documented use of the gamification term exists since 2008 but more popular usage started by 2010. The term was interpreted and used in multiple ways leading to two distinct but related concepts. The first concept is the progressive move to the ubiquity of games in common life. The second concept is using game elements to promote user motivation and engagement. In 2011, Deterding et al. [5] researched on the subject and proposed a definition for gamification as "the use of game design elements in non-game contexts". The need for clarification of the term was suggested in [23] due to overlap of goals and means in gamification and marketing. The concepts of game elements and non-game contexts were considered subjective and prone to misinterpretation and another definition for gamification was proposed by Werbach, in 2014, stating that "gamification is the process of making activities more game-like" [24]. Gamification has now been applied in areas like education, work, health, well-being, sustainability, and civic participation. It is a field of study with an increasing interest on its uses and effects. The overall user experience is evaluated using these techniques.

Gamification techniques make use of people's natural desires. These include the need and will for social interaction, learning, self-improvement, competition, goal accomplishment, status, personal expression, or closure. Initial strategies used simple rewards on task completion to motivate users. Examples of rewards are points, trophies, level upgrades, progression bars or virtual currencies. Public display of such gatherings promotes competition. Competition can lead to unethical or uncooperative behavior and some systems tend to limit this approach. Other approaches to gamification include making the tasks more game-based with the use of concepts and techniques like meaningful choices, tutorials, challenge increase or adding contextual narrative. One way to achieve the gamification of tasks is implementing serious games.

Gamification in health has been used in assessing disease status, cognitive and motor rehabilitation and to promote motivation and engagement. In a literature overview by De Croon et al. [25], some game design guidelines and techniques are proposed, to adapt and personalize the gamification mechanics to individual users. A state-of-the-art overview in gamification and serious games for e-health is available in that article.

Studies on the efficacy and validity of gamification have been made. Hamari et al. have conducted a literature review in 2014 researching the effects of gamification [26]. The study concluded that gamification provides positive results but is directly dependent on usage context and the users that use it.

2.4.2. *Serious games*

There is no single definition of serious games. Serious games differ from conventional games as gaming elements are used but the focus is not pure entertainment. The focus of the serious game can be learning or training a specific area or skill. A serious game can be “any form of interactive computer-based game software for one or multiple players to be used on any platform and that has been developed with the intention to be more than entertainment”, as Ritterfeld et al. described in [27]. Serious games are used since several thousand years ago mainly as a military tool. Uses in education and business started from mid-20th century. Before 2000, the number of games that people with visual disabilities or motor disabilities could play with was extremely low, close to zero [28].

Serious games are used for rehabilitation. Motivation and engagement of patients is the main reason for using serious games. The rehabilitation process can be long and tedious. Some patients have difficulty in maintaining their therapy protocols. The most relevant causes associated with low levels of adherence to exercises are the severity of the disease, financial constraints preventing access to care, and lack of time and motivation. The use of computer games has been recognized as valuable tool in the promotion of engagement and motivation of patients in the fields of neurorehabilitation, physiotherapy, and occupational therapy.

Games modeling rehabilitation exercises are often referred as exergames. Serious games have been used for rehabilitation in aging, obesity and overweight, stroke, balance training, cerebral palsy, Parkinson’s disease, orthopedic, and with other pathologies including multiple sclerosis, fibromyalgia, systemic Lupus Erythematosus, schizophrenia, Down syndrome, mild cognitive impairment, Alzheimer’s disease, pulmonary diseases, diabetes, urinary incontinence, and pain management. Therefore, the main purpose in the use of serious games, in health-related applications, is to promote engagement and motivation. The use of games also promotes autonomous unsupervised home-based training meaning that patients can practice and perform exercises in their home environment without the presential supervision of therapists.

Smart serious games (SSG) are serious games that make use of IoT devices [29]. These IoT devices, sensors, or actuators, allow games to extend their interaction with users. Sensors gather extra information that can be used in real time and/or collected for later analysis. Actuators are used to promote more immersive gaming environments by controlling real world devices.

2.4.3. *Validation*

The validity of the usage of serious games has been the subject in several studies. Randomized controlled trials (RCT) are common tools for validation and testing of clinical interventions and can provide useful information for validation on the use of serious games. An RCT was setup in [6] with a population with cerebral palsy, aged between 6 and 18, with the ability to stand and no cognitive or language impairments. The RCT consisted of setting up 3 groups of patients from a significative population and applying a protocol to each group during a period. The first group had no serious games included in therapy, the second group had a mixed percentage of 50% serious games therapy and 50% regular therapy, and the third group only had serious games as therapy. Evaluations were performed in 5 different stages of the intervention: one before the intervention, to establish a baseline, two during the intervention, one immediately after, and one 3 months after the intervention, to determine the progress over time. The materials used included a physical rehabilitation platform for children with cerebral palsy, using specially designed and validated games and mini-games [30].

In a systematic review on serious games for children with chronic diseases, released in 2018, Holtz et al. [31] demonstrate the potential of the use of games for health improvement with young patients. The research team reviewed literature to assess if serious games impact health outcomes in non-adults (less than 18 years of age). 18 studies were reviewed. The targeted pathologies in these applications are cerebral palsy, asthma, diabetes, developmental coordination disorders and vision disorders. The variables measured to evaluate results were lung function, glycemic control, hospital visits, motor proficiency and visual acuity. Psychosocial variables measured were self-efficacy and knowledge. Global results included 6 studies with significant outcomes and 4 studies had changes on one psychosocial variable. 9 of the studies focused on motor proficiency, with 3 showing significant improvement, 1 with partial improvement, 4 with no improvement and 1 with decline. Mixed results obtained during the research revealed the need for better design and testing of the serious games.

2.4.4. *Serious game design*

Models for designing serious games focused on learning were conceptualized in several studies since the year 2000. Game-based learning game design models provide relevant guidelines, directly applicable or adaptable, for the design of serious games. Djaouti et al., in 2001, proposed a simple global model for the conception of serious games. Yusoff, in 2009, proposed a serious game conceptual framework. In 2012, Marfisi-Schottman, conceived a 7-

stage model for the conception of serious games. Each step in the process has a specific responsible actor. The MDA model, developed by Hunicke et al. in [32], is a game design framework. The model has 3 layers: mechanics, dynamics, and aesthetics, that are briefly described in Table 14. All layers are interconnected and get input and output from each other.

Table 14 - MDA framework model layers

Layer	Description
Mechanics	Game structural components (e.g., data model and algorithms)
Dynamics	Game interaction (e.g., runtime behaviors for mechanics components)
Aesthetics	Game ambiance. The aesthetics components include sensation, fantasy, narrative, challenge, fellowship, discovery, expression, and submission. Distinct gameplays value some areas more than others, providing different experiences and searching for a specific emotional response from the user.

Calleja et al. researched user immersion and immersive systems, proposing the concept of incorporation. The 6 dimensions of involvement of Calleja, discussed in [33], are used for modelling serious games. The model defines macro and micro involvement phases, progressing to incorporation. This set includes kinesthetic, spatial, shared, narrative, affective and ludic dimensions. An ontology for serious games was developed by Tang et al. in [34]. This tool gives an overview of the relations between the serious game components.

The systematic review of 2019, by Aguilar-lazcano et al. [35], focused on the interaction modalities used in serious games for upper limb rehabilitation. The research included a systematic search of literature, from 2007 to 2017, resulting in a set of 33 articles meeting the search criteria. Systems were categorized in 3 distinct interaction modalities: vision systems (42.4%), complementary vision systems (30.3%) and no-vision systems (27.2%). A significant 48.48% of the systems used the Kinect sensor for interaction. The most treated area of the body was the shoulder with 19% of the systems covering it. Limitations of vision systems are lighting conditions. No-vision systems present inaccurate measurements of the range of motion in angles of the body. Identified areas for further research are finger rehabilitation, trauma injuries rehabilitation, changing the game difficulty based on muscle strength and posture and multisensory data fusion.

An affordable way of developing games for rehabilitation is based on the concept of mini games. Mini games can provide more focused ways of interaction for children with special needs. These mini games make use of available generic game engines. Each mini game can be focused on a specific exercised or train a specific skill. The need for a specialized input device is considered a limitation on development and use. Selecting the right game for a specific patient is of extreme relevance. Therapists need tools to help them select the right games for specific therapy exercises and patient groups. There is a lack of configurable and adaptable games and the ones that exist are usually only partially configurable. The used sensor or sensors is a non-replaceable requirement. Some games do not provide tools for performance or clinical progression analysis.

2.4.5. *Serious games for children*

A set of 40 guidelines for designing serious games for children are identified and validated in a study by Valenza et al. [36], and are listed in Table 15.

Table 15 - Guidelines for designing serious games for children

Id	Description
G1	Simplify the use of the mouse
G2	Avoid differentiating between left and right
G3	Use efficient Interaction mechanisms with interface elements
G4	Allow spoken instructions
G5	Hide features of advanced levels
G6	Explore cooperative use
G7	Easy-to-read font type usage
G8	Relate interface metaphor to children world
G9	Make interaction elements ease to spot
G10	Use interaction time according to children’s age
G11	Use meaningful icon as a replacement or help to texts
G12	Prefer recognizing than remembering
G13	Use of visual interface mainly
G14	Provide accurate and fast feedback
G15	Clearly show the status of the system
G16	Prefer to use characters for interaction
G17	Present information to users according to their level of development
G18	Use interfaces and conventions that are already known by the users
G19	Layout must be rich in content with little empty spaces
G20	Present scoring and/or classification as clear as possible in screen
G21	Interface must look and behave as real as possible
G22	A great variety of themes/skins must be available
G23	Information must be presented in more than one way
G24	Reward the player
G25	Devise a way to emphasize clickable interface elements
G26	Documentation and help must be objective and easy to find
G27	Allow players to undo and correct mistakes
G28	Allow user to explore the system and build things
G29	Logically scaffold the content
G30	Fit vocabulary to target audience
G31	Fit the game to user’s level of experience
G32	Design interesting and challenging activities

G33	Teachers must be able to configure the game if the game is to be assisted by them
G34	Make the target content fit in naturally with its surrounding context
G35	Use narratives/stories to engage players
G36	Clearly define goals
G37	Avoid cognitive loads
G38	Allow many different levels of the game
G39	Show players' tracks (already visited places and contents)
G40	Teachers must be able to control the game

These are guidelines for children, not taking into account the specificities of special needs. Some items should really be avoided or adapted in some way. Most of these guidelines apply without any constraint but should be adapted to the specific context. For instance, there is absolute need to distinguish left from right for some rehabilitation exercises.

A study by Henschke et al. [37], implemented several serious game directed for children with cerebral palsy, and performing a pilot study for assessment of game relevance for the target audience.

2.5. Technology

Several technological components are used when designing solutions. The pervasive and ubiquitous computing paradigms suggested by Marc Weiser in 1991 [38] are now a reality. Life already happens in world of smart "invisible" sensors and actuators embedded in the environments surrounding us. Technology is everyday more present and at the same time more integrated becoming an intrinsic characteristic of devices. These devices have processing capabilities that make them smart, in the way that they can process data and make decisions. Everything leads us to believe that future trends will continue in the same direction, with devices becoming more and more smarter with the added capabilities. The use of these smart devices with an interconnection infrastructure allows for the implementation of smart environments where sensors and actuators respond to user interaction. This concept follows the Industry 4.0 vision of the use and integration of smart technology for continuous automation of industrial practices and processes.

2.5.1. *Design of interactive interfaces*

Human-Machine Interaction (HMI) is an area of study that originated at the start of the 20^{th} century. In the early days was used to research and study interaction between workers and machines in industrial environments. During the Second World War HMI focused on military issues. Human-Computer Interaction (HCI) studies the interaction between the human user and the computer. Child-oriented HCI is also known as Child-Computer Interaction (CCI).

The 10 usability heuristics from the Nielsen Norman Group, available in [39], are a set of guidelines for application usability. These guidelines highlight the relevance of the consistency of the interface and the use of standards for interaction solutions. Recognition should be favored to recall. Applications should be flexible and efficient in use, with effective aesthetics and a minimalistic design, providing. Error prevention should be implemented to allow fluid use and prevent user frustration. It is important to provide visibility of the system status. The user should be in control and have freedom of action in the application. Virtual and real environments should be matched, when applicable. Feedback is very important for action and behavior assessment. Help should be provided as needed and documentation should be available.

Schneiderman et al. provide 8 rules for interface design. The interface should be intuitive, consistent, and inclusive, with rapid and frequent feedback to users, allowing clear confirmation of actions. Error conditions should be avoided reducing user frustration and actions should be undoable. The user should be, and feel, in control of the system [40].

Interfaces should be inclusive, and when following a collaborative design methodology, users can participate in the interface design. Interfaces can be designed to be multiuser and allow real-time interaction between users. Patient and therapist in-game interaction can be implemented to promote patient motivation. Interaction between patients is also a possibility. Patients could be allowed to interact with each other in the game in some collaborative way to finish a task or accomplish a common goal. Another interaction method between patients is competition. Competition can be useful is some situations, but the overall feeling is that it sometimes promotes negative behaviors that are not what is sought. Therefore, the implementation of competition mechanics should be well thought of and the real benefits evaluated against the drawbacks.

Other interfaces are possible. For instance, Cerezo et al. researched the design and use of tangible interfaces for very young children and children with special needs, applying a children-centered design [41].

2.5.2. *Immersive technologies*

Immersion is a perception. Immersive technology enhances learning experiences, promotes participation in collaborative activity and increases creativity and engagement. Immersive technology has multiple definitions. Immersive systems make use of augmented, virtual, or mixed reality technologies. Virtual reality is usually immersive VR, meaning the complete immersion in a CGI environment, and with the use of computerized virtual objects. Augmented reality is the superimposition of CGI elements over the real-world imaging. Augmented virtuality (AV) is the use of real objects in virtual worlds. Mixed reality, defined by Milgram et al. in [42], is anything between the completely virtual and completely real. MR relates to the creation of hybrid environments where virtual and natural elements coexist and includes VR, AR, and AV. Validation of the use of virtual reality with patients with cerebral palsy has been researched by Chen et al. in [43].

2.5.3. *Smart environments*

Creating smart environments for therapy has specific challenges. Each patient or user has different interaction needs. A smart environment acquires data and applies knowledge, adapting to improve the user experience. Smart sensors and actuators can be combined to create smart environments [44]. Features of smart environments include remote control of devices, device communication, data acquisition and distribution, smart devices and enhanced services, prediction and decision-making capabilities, and a set of networking standards, protocols, and regulations. Smart devices have networking and data processing capabilities being able to provide enhanced services in the environment. These devices can access remote data sources and can intercommunicate being able to complement information with external data. Collaboration among smart devices to fulfill some specific tasks is a current research topic. Smart devices provide new solutions and specialized hardware.

The automation process, in the smart environment, is a continuous cycle based on sensors collecting data, data processing and control commands sent to actuators for actions to be executed. The automation and adaptation process of the environment allows the use of machine learning algorithms for prediction and decision-making capabilities. Device communication commonly uses the Internet or wireless communication protocols like Bluetooth.

A smart room is an indoor space that is setup to use several embedded services for the tracking and interaction of users. Multiple sensors can be used for tracking and measurement of user actions. Actuators are used for providing feedback to user interaction. Chen et al. define smart rooms [7] as indoor spaces configured to let users of the physical space invisibly interact with the virtual world of computers during their normal activities. Several existing services in the underlying networked computing infrastructure continually observe and track what users do using multiple sensor modalities, controlling actuators that modify the physical state of the room or provide other feedback to the occupants. Sensors used include NUI (Natural User Interface) sensors, that allow non-intrusive and more intuitive ways of interaction, biometric sensors and IMU (Inertial Measurement Unit) sensors. Speaking about smart environments for physiotherapy means that the interaction should happen taking in consideration the user pathology and therapeutic needs, as a natural extension to the therapeutic role of the applications. Therefore, there is a need for real time, precise and configurable control of actuation devices.

2.5.4. *IoT devices*

In an indoor open-space room for therapy or research laboratory there are usually multiple areas with users using distinct rehabilitation tools and applications. These applications track and measure user data using multiple sensors but feedback to user interaction is mostly limited to audio and screen visualization. To achieve an extra level of immersion, these applications could benefit from a way to expand their interaction to surrounding devices.

The smart gateway is a device used to act as an interface between sensors and actuators.

The gateway is a smart IoT device interfacing between applications and actuators available. Actuators are available as capabilities of actuation modules. Actuation modules are smart devices that provide means for controlling actuators. Actuators are electronic devices, specially designed or generic, that can be controlled when connected to the actuation module. The gateway and modules communicate using Bluetooth. The gateway scans for actuation modules. The modules acknowledge and respond announcing their presence and actuators provided. The gateway manages available resources and application requests.

Gateways allow applications to make use of surrounding devices that can extend the user experience and add a new level of immersion. The gateway acts as an interface that receives and interprets application commands controlling external devices. Gateways are IoT devices that search and identify surrounding compatible devices and provide applications with a description of a specific location capabilities and functionality. Depending on components used, the proximity ranges can vary but surrounding devices means the devices that are a few meters away, usually in the same room. Technically IoT means using the Internet. The gateways referred here do not necessarily use the Internet, sometimes using Bluetooth for local communication. Popular low-cost hardware devices used for rapid prototyping of IoT Gateways are the Arduino microcontroller and the Raspberry Pi.

2.5.5. *Sensors and actuators*

There are multiple types of HMI technologies depending on types and formats of the devices used when designing solutions. Traditional interfaces include the keyboard and mouse. These can sometimes present difficulties for some users due to pathologic limitations. Use of the keyboard and mouse can be available as a tool for the therapist or curator. The most common HMI technologies used when implementing this kind of interfaces can be visualized in Table 16.

Table 16 - HMI technologies

Technology	**Description**
Keys or Buttons	Pressing keys or clicking buttons
Mouse	Controlling a mouse and mouse buttons
Touch	Using the fingers or other body parts
Gesture recognition	Hands, fingers, motion, facial expressions
Voice recognition	Using the voice as input and/or output
Augmented Reality	Superimposing virtual CGI on real world
Virtual Reality	Immersive CGI environment
Brain Interfaces	Neurological interfaces

Voice recognition can be used both as a navigational tool and a therapeutic interaction tool. Voice commands can be used for navigation through the application views and interaction with the view’s elements. Voice input can also be used for speech therapy exercises.

2.5.6. *NUI Sensors*

Sensors are used for capturing data from user interactions. The sensors can be used for enabling the user interaction with the application UI and/or for measuring or capturing user interactions during exercises/gameplay. Sometimes users cannot use traditional interfaces.

Natural User Interfaces (NUI) are interfaces that provide ways of interaction that are more natural to the user [45]. NUI are an alternative to standard input interfaces and currently the obvious choice for implementing systems for children with special needs. NUI sensors can present a non-intrusive way of capturing user's interactions overcoming some of the limitations that traditional interfaces have. Using more natural ways of interfacing and interaction is the current trend, for some time, in almost every aspect of technology and is expected to continue in the future. The natural interfaces category includes the use of the hands and fingers but also the body itself. Voice, gestures, motion, and facial expressions can be used to interact. These interfaces are more intuitive to the user. Actions modeled using natural interfaces allow users lower learning curves.

Several types of NUI sensors are available (Figure 2). Each one of these sensors present their own characteristics making it the right option for a specific use. The choice of the right one depends on the specific domain of application and several other factors like price and availability. NUI sensors provide motion tracking and gesture recognition. Some sensors also provide speech recognition (e.g., Microsoft Kinect).

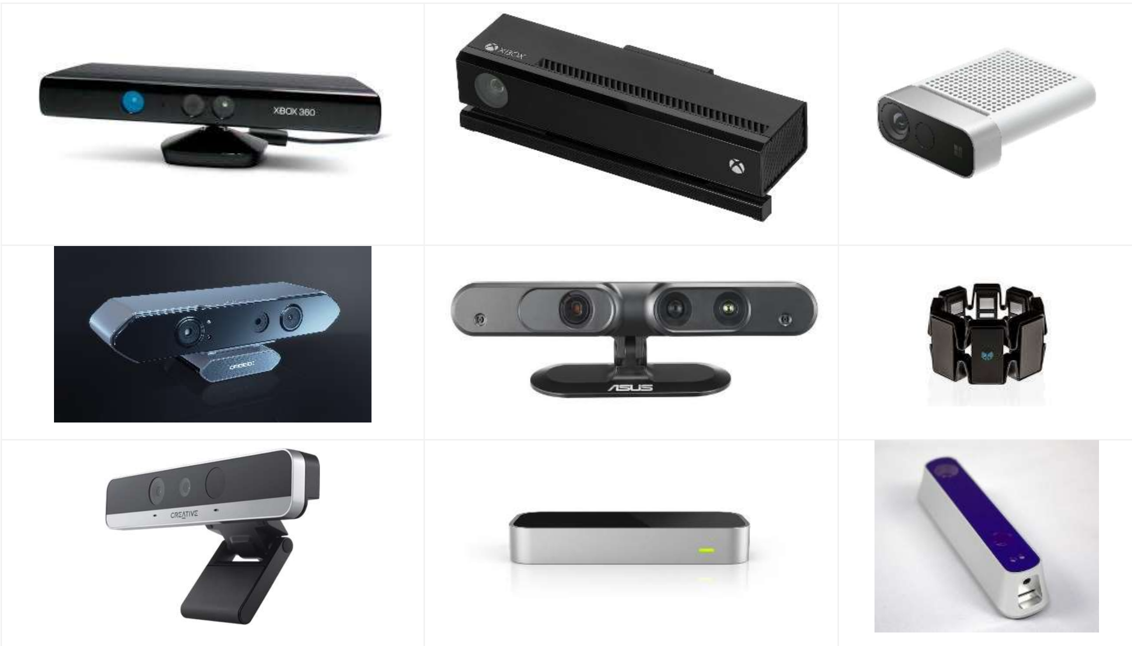

Figure 2 - NUI Sensors (from top left to bottom right: Microsoft Kinect v1, v2 and Azure; Orbbec Persee Astra; Asus Xtion; Myo; Real Sense D435; Leap Motion; and Structure)

Microsoft Kinect The Microsoft Kinect sensor was the leading sensor for real-time full body tracking. The Kinect has known several versions: V1, V2 and Azure. The Kinect V1 allows body tracking of 20 distinct skeleton joints and face tracking with 121 face points. The Kinect V2 has an RGB camera with a resolution of 1920 x 1080 pixels and an infrared (IR) camera and emitter. Is capable of face tracking with 1000 face points. Uses machine learning techniques are used to infer body positions. To build a skeleton, a depth map is built projecting a known shape into the scene and interpreting its deformations. Then, using computer vision techniques, like depth from focus and depth from stereo, the depth map is updated. Body parts are inferred using a randomized decision forest (decision tree with millions of nodes). The Azure Kinect DK is the latest version of the Kinect. It's a development kit for computer vision and speech models. The device has video camera, depth sensor, orientation sensor and spatial microphone array, in an integrated package. Several SDKs are available with the device: Sensor SDK, Body Tracking SDK and Speech Cognitive Services SDK. Cognitive Vision services can be used with the RGB camera. The Azure version uses a cloud service available through the Microsoft Azure platform.

The Kinect v1 tracks the body using 20 skeleton joints and face tracking is achieved using 121 distinct points as can be viewed in Figure 3.

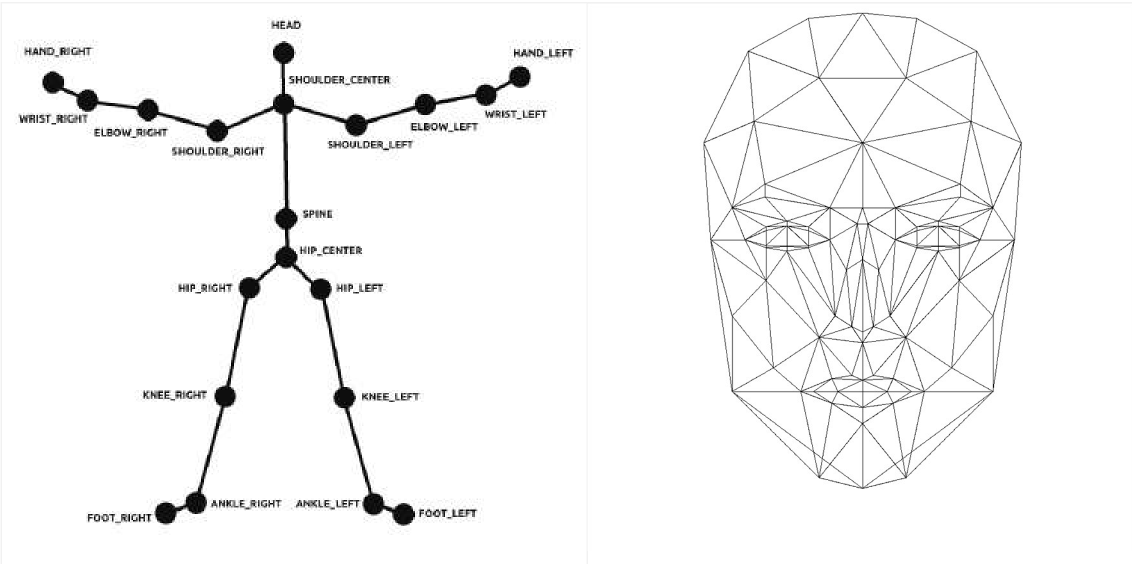

Figure 3 - Kinect v1 20 skeleton joints (left) and 121 face points (right)

The Kinect sensor has been evaluated for use in specific areas of rehabilitation such as upper extremity function evaluation. Both V1 and V2 versions of the Kinect sensor were evaluated and analyzed in multiple studies for validation of usage in rehabilitation solutions [46]. Naturally, the Kinect sensor V2 has been considered more reliable then the V1 for face tracking [47].

Leap Motion The Leap Motion sensor was launched in 2012. Focused on finer motion tracking. The sensor can recognize hand states, fingers, and joints. Uses two infrared cameras and 3 IR led. Can achieve 300 frames per second. The interaction area of the Leap Motion sensor is like an inverted pyramid starting at the device. Inside this area there's a bounding interaction box.

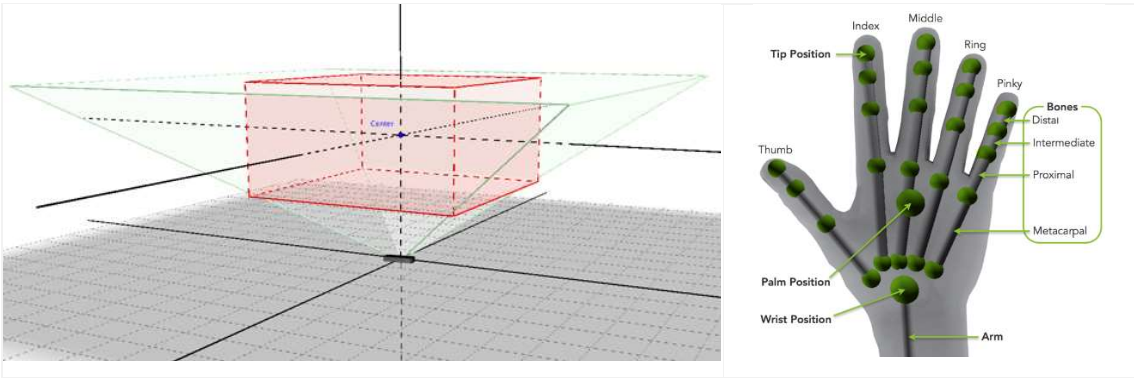

Figure 4 - Leap Motion interaction box (left) and hand hierarchy (right)

The Leap Motion SDK has 3 different versions. The V2 SDK supports standard and tool tracking. Gestures are no longer supported. The V3 SDK is the first Orion release, optimized for VR platforms. Preserves legacy APIs. The V4 SDK is the second generation of the Orion SDK. The SDK has a C-style API named LeapC for communication with the leap service. Engines for Unity and Unreal are built on the API. The LeapC API is directly available for C programs. The V5 SDK is only available for the SnapDragon platform. The Leap Motion tracks hand and finger joints for both hands. The hand joint hierarchy can be viewed in Figure 4 (right).

Ultraleap Stereo IR 170 Evaluation Kit The Ultraleap has stereo IR cameras capable of 90 fps. Using the same software as the Leap Motion, the Ultraleap offers wider field of view and a longer tracking range, having lower power consumption. Tracks hands within a 3D interactive zone, extending from the device, from 10cm to 75cm or more, on a 170x170° typical field of view

Structure The Structure sensor is manufactured by Occipital. It is very similar to the Microsoft Kinect sensor. Creates 3D models by 3D scanning of objects and environments and can be applied on an AR/VR headset and has 6-DoF positional tracking. Structure Core is another sensor by Occipital. Used for AR/VR SLAM, robot vision and embedded applications. Simultaneous Localization and Mapping (SLAM) is a deep learning algorithm that allows bidirectional translation of real-world data and virtual environments. It is currently being used in commercial automatic navigation systems like TESLA's and Volvo Autopilot products. Occipital is the maintainer of the OpenNI project, mentioned in the Software Frameworks section.

Intel Real Sense D415/D435 Depth Cameras The Real Sense D415 and D435 sensors are manufactured by Intel. These sensors are depth cameras. Regarding precision, the Real Sense is not as accurate as the Kinect and Leap Motion but can do both gross and finer motion tracking. Can do hand, finger, and face tracking. Has reliable voice recognition.

Orbbec Persee Astra The Persee Astra is a 3D camera. It is provided in 3 different models: ASTRA PRO, ASTRA S e ASTRA. Each one has a specific range, respectively 0.6-8m, 0.4-2m and 0.6-8m. All have VGA video, allowing a resolution of 1280 x 720 at 30 FPS, and a depth tracking image resolution of 640 x 480 at 30 FPS. Has 2 integrated microphones. Uses include face and gesture recognition, body tracking, 3D measurement, environment perception, and 3D map reconstruction. Compatible with Android, Linux, and Windows. Provides the Astra SDK and is compatible with OpenNI.

Asus Xtion The Asus Xtion PRO is a depth sensor, with an IR camera, and has a VGA resolution of 640x480 at 30 FPS. Detects motion in a range of 0.8 to 3.5 meters. Is compatible with OpenNI and provides an USB 2.0 interface. The Asus Xtion PRO Live has color RGB camera, depth IR camera and 2 microphones, being capable of a resolution of 1280x1024 (SXGA). Provides USB 2.0 and 3.0 compatibility.

2.5.7. *Pressure mats*

NUI sensors can be used to track both upper and lower limb joints. Legs and feet can be monitored using some of these sensors. Another kind of sensor for lower limb interaction are floor pads. Floor pads sense pressure when users step over them. Posture and gait can be assessed using these tools, that are used also to detect and try to prevent falls among elderly people. Examples are the Wii Dancing Mat, single- or double-person version, or the Cosiki Dance Pad (Figure 5). Some of these pads can be configured and used as game controllers.

Figure 5 - Wii Dance Mat (left); Wii Double Dance Mat (center); Cosiki Dance Pad (right)

2.5.8. *Wearables*

Wearables are specially designed devices that the user can wear and use for interaction. These devices use embedded sensors to monitor and collect user biometric data. Some devices use actuators to provide force feedback to users. Data like interaction statistics or patterns can be collected. Wearable devices can be classified considering 4 categories: Assistive, Workplace, Healthcare and Consumer Products.

IMU sensors The Shimmer is a IMU sensor. Has accelerometer, gyroscope, magnetometer, and pressure sensors and 2 digital EMG channels. Uses Bluetooth for communication. It’s very portable and has an elastic strap to attach it to an arm or leg. Needs 3 electrodes applied over the monitored muscle area. Allows detection of localized muscular activity.

Figure 6 - Shimmer sensor and sensor strapped on leg

The Valedo Motion is a commercial product used for back spine therapy. The solution uses 3 distinct IMU motion sensors, applied at the back and chest of the patient.

The Myo sensor was developed by Thalmic Labs and the market price was around $199. Myo used EMG technology to detect and measure muscle electrical activity from gestures and control other devices based on these signals. It had a great potential in prosthetics and was popular among developers. Some drawbacks were that it was made with only one size that was not a perfect fit for some people (most notably female users had more issues with the armband being loose). There were reports of being uncomfortable after long usage periods. Myo production has officially ended in 2018.

Haptic gloves Haptic technology applies forces to achieve results. The haptic glove is more intrusive than NUI sensors but offer different possibilities and challenges. NUI sensors suffer from the problem of occlusion of joints and bones. The glove and other wearable devices can overcome that limitation but providing some of their own. There is still need for sending the sensor data to a server. More hardware is needed for that logic and there is also need for a power supply or battery. Some research prototypes still suffer from generated heat from power supplies and other components. The Dexmo Glove [48], developed by Dexta Robotics, a chinese startup, was presented in 2016. The Teslasuit is a complete haptic force-feedback suit from Tesla [49]. The Teslasuit Glove is an haptic force-feedback glove from Tesla [50]. The force feedback system is like the one found in the 2016 Dexmo Exoskeleton glove. The Rapael Smart Glove was developed by Neofect and is advertised as a high-tech stroke rehab device. A study from 2018 identified 14 distinct commercial haptic gloves available. The gloves were classified as traditional gloves, thimbles, and exoskeletons. Traditional gloves are tissue-based wearables that fit the hand and fingers, allowing free movement, and using embedded sensors and actuators to measure flexion and provide force feedback. The thimble configuration has an actuator at the fingertip, in each finger. The exoskeletons are articulated structures wear over the hands, applying forces to the fingers.

Headsets Headsets are used mainly for phobia and post-traumatic stress disorders. In the 2018 literature analysis on the state of immersive technology research [51], Suh et all. make an overview of augmented and virtual reality systems used for education, marketing, business, and healthcare. The authors detect an increase of solutions applying these techniques and the need for some theory framing on the effects of immersion on user experiences and performance.

The Oculus Rift was the first VR headset available for end-users (Figure 7). The headset includes an integrated OLED display and headphones. A gyroscope is used for positional awareness, detecting the head position, and updating the virtual environment.

The Qualcomm Snapdragon XR2 5G is a state-of-the-art headset using 7 concurrent cameras and 5G. Uses AI to offer optimized mixed reality. Allows 60 fps 8K 360º video. Provides a voice UI and context awareness. The Leap Motion V5 SDK is only available for this platform.

Figure 7 - Oculus Rift Headset (left) and Leap Motion adaptation (right)

Neurological interfaces Brain-machine interface (BMI) is a device for directly interfacing with the brain electrical signals. BMI systems are being used in stroke rehabilitation [52]. There are many advantages related to BMI interfaces, as they become less intrusive, currently being one of the leading and most promising research fields. Neural data acquired from the brain is processed and analyzed to infer the user intent. It is then converted to an output generating relevant actions in the system.

2.5.9. *Multiple sensors and sensor fusion*

Multiple sensors can be used for data capturing. Using data captured from multiple sensors can provide more precise information. Combining data obtained from each sensor helps build better models for information extraction. Body Area Networks (BAN), a specific usage of Wireless Sensor Networks (WSN), use sensors to monitor the human body.

Multiple Kinects are used in [53] for building multi-surface environments. The approach used integrates multiple Microsoft Kinects to track users and their devices in larger rooms while overcoming occlusion issues to a substantial degree. An API provides the required functionality.

In [54], two Leap Motion sensors have been setup simultaneously for using the information from both sensors in a hand tracking device. The device is used for remote control of a robotic arm in tabletop object manipulation. Motivation for this multiple sensor usage is to overcome the limitations of a single sensor when finger occlusion occurs. The best physical setup of the two sensors is researched and an optimal scheme is proposed. Data fusion is achieved overcoming the single sensor limitations of the Leap Motion API. Improved gesture recognition is used to model robot movements.

Leap Motion and Kinect sensors data fusion is used by Penelle et al. [55] to improve hand tracking performances targeting upper limb amputees. The technique is capturing 3D images of the patient and applying a mirror effect before displaying them and creating the illusion of two arms.

Hadjidj et al. [56] discuss challenges and opportunities of using Wireless Sensor Networks (WSN) for rehabilitation applications. In this kind of sensor-based system, the patient wears several small nodes able to assess human movement without interfering with his natural behaviors. These nodes gather specific information such as position, motion, direction, and physiological state. This solution complements vision-based systems that have occlusion and line-of-sight constraints. Body Area Networks (BAN) are a specialization of WSN for healthcare rehabilitation.

2.5.10. *Actuators*

Actuators are system elements that allow applications to react to users' interactions activating devices in the physical world.

The application display device or screen is the most common actuator. Multiple screens can be used. Depending on the pathologies, graphical elements can be a distraction. Relevant factors are colors, element sizes, contrast, number of dimensions (2D or 3D).

Sound can be used as music or sound effects. Sound can be sampled or synthetic. Sound can help build a more immersive environment or set a mood. Depending on patient pathology, sound and sound effects can be distractive or effective. Sound therapy makes use of sound to promote rehabilitation.

Voice can be sampled or synthetic. Using sampled voice segments is a possibility but is less flexible than the synthesized variant. When using sampled voice segments the system is limited to the voicings recorded. An example is the Microsoft Speech Platform SDK, that can be used for speech synthesis and recognition. Male and female voices are available, and age can be configured. The JPK platform uses it for English, Portuguese, and Spanish while other languages are supported. Other voice synthesis frameworks are available depending on the targeted platforms.

IoT gateways are devices that control several actuators and usually have processing capabilities for local control of connected devices. The gateway acts as an interface, communicating with a remote server, using the Internet, and the local devices. Boards like the Arduino or ESP-32 can easily be configured for rapid prototyping of IoT devices. Light-weight communication protocols like MQTT are an option for IoT communication.

2.5.11. *Software frameworks*

Some software frameworks and libraries that work with multiple sensors are available and can be a valuable tool when designing solutions. The OpenNI is an open-source SDK developed and maintained by Occipital. Currently available as OpenNI 2. Provides APIs for the creation of middleware libraries and applications using 3D sensors. Supported on Windows, Linux, macOS and Android. The source code is available on GitHub. The Nuitrack SDK is a software development kit for implementing NUI-based applications. Compatible with most 3D sensors including Orbbec, LIPS, Intel RealSense, Asus, Kinect. Compatible with OpenNI SDK. Single cross-platform API that works for Android, Windows, and Linux. Provides full body skeletal tracking, user masks, 3d point clouds, gesture recognition and face tracking. Latest version works with Unity, Unreal Engine, C++, and C#. The Nuitrack SDK is a commercial product offering a limited free trial.

2.5.12. *Data mining and machine learning*

Artificial Intelligence (AI) provides tools that can and are actively being used also for health applications. Data analytics and machine learning (ML) tools are used to extract relevant information from data, using supervised or unsupervised methods, providing means for decision-making, and allowing platforms to adapt and personalize its behavior responding to user interaction.

Data mining (DM) is the process of pattern recognition and/or outlier detection, through data visualization or using data analysis tools. DM is used for problem solving by analyzing existing datasets, extracting implicit, previously unknown, and potentially useful, information. Pattern searching is done using database querying applications. Depending on the relevance of these pattern, they can be used for inferring accurate predictions of future data.

DM uses a combination of techniques such as pattern recognition, statistics, and machine learning. It is a complex and iterative process not requiring the previous setup of models. DM systems explore historical data to infer useful data in the future, detect data patterns, suggest new business rules, allow informed decision-making, and action planning. The data mining virtuous cycle consists of analysis, modeling, and evaluation, meaning that the business logic problem must be translated as a DM problem, select the data sources, know the data sources, correct problems with data, transform data for usage in methods, generate data models, evaluate those models, implement business models, and evaluate results.

Typical DM tasks include:

- interpreting the visual and iterative data exploration, through graphical matrices, trend lines, and bidimensional charts.
- clustering, grouping data with common features (previously unknown) through

hierarchical methods such as agglomerative and iterative K-means.

- classification, generation models organizing data in pre-determined classes, including methods such as decision trees and neural networks.
- association, finding frequent groups and association rules.

Knowledge representation uses decision tables, decision trees, classification rules (data categorization), association rules (hidden relation discovery), outliers (anomaly analysis), outlier and deviation detection, rules with relations, numeric inference trees, instance-based representation, regression (building a mathematical model or function from independent variables, for using in the inference of a dependent variable values), and clusters (splitting a big dataset in related subclasses). An existing set of algorithms can be applied to data, for inference of basic rules, statistical modeling, decision trees, rules, association rules, linear models, instance-based learning, and clustering.

Ethical questions must be taken into consideration, when preparing data analysis projects, specifically if using personal user data or if data relations allow user identification.

Machine learning (ML) is a process of using AI for extracting knowledge from data. The ML process has usually four stages: data harmonization, representation learning, model fitting and evaluation.

Deep learning (DL), an emerging technology, is a specification of ML that can be described as using deep neural networks in knowledge discovery. DL is the process of using neural networks with big data and optimized algorithms for extracting information. DL differs from traditional ML methods on how it learns from data. DL uses computational models composed of multiple processing layers, with each layer based on neural networks, allowing data representations with multiple levels of abstraction [57]. DL architectures used in healthcare are mostly based in convolutional neural networks (CNN), recurrent neural networks (RNN), restricted Boltzmann machines (RBM) and auto-encoders (AE). DL has been used for clinical imaging, electronic health records, and genomics. There are also still unsolved challenges such as the volume of data needed to train models, the quality of the available data, temporality, domain complexity and interpretability. Future work in this field looks very promising and presents opportunities for application in healthcare, such as feature enrichment, federated inference, model privacy, expert knowledge integration, and temporal and interpretable modeling. [58]

Several solutions are available for DM and ML, both commercial and free, allowing extensive exploration of data, for knowledge extraction. Some provide tools for integration.

Examples of proprietary commercial solutions available focusing on DM and ML are Microsoft Sharepoint, BOARD, TARGIT Decision Suite, Birst, Stratum, Dundas BI, Sage X3, SAP Business Intelligence, QlikView, IBM Cognos Analytics, Phocas Business Intelligence, SAS Analytics Pro. PowerBI is a set of tools for business analysis. Uses Key Performance Indicators (KPI), dashboards, parametrized reports, and Pivot tables. A Pivot table, or dynamic table, resumes data in another table, applying an operation, such as classification, or calculating a mean, or a sum of the data in the source table, usually including data grouping.

There are several very interesting free, and open source, DM solutions worth mentioning.

These all relate to DM, but there are others, more focused on data analytics or ML, also useful in the process of obtaining knowledge.

Weka is a DM tool, developed by the Waikato University, in New Zealand, containing several ML algorithms. These algorithms, that can be applied on datasets or used in implemented applications, allow pre-processing, classification, regression, clustering, association, and data visualization. Weka can be used with big volumes of data, as in big data, and provides the ability to allow the development of new ML algorithms. Weka has several main components, namely Knowledge Explorer, Experimenter, Knowledge Flow, Workbench, and Simple CLI. Weka is open source and available through the GNU General Public License.

MOA (Massive Online Analysis) is a Java framework for streaming DM. It includes machine learning algorithms such as classification, regression, grouping and anomaly detection, deviation detection and assessment tools. It realizes data mining of large data streams in real time and large-scale machine learning. New algorithms, flow generators and evaluation measures can be added.

RapidMiner is a predictive analysis platform. Open source and commercial version. It offers a library of data processing and machine learning algorithms. Recognized by Gartner as a market leader in advanced analytics platforms in 2016. It offers the usual methods of cleaning, filtering, data grouping and the use of templates, reproducible workflows, professional visualization environment. Integration with Python and R languages.

Orange is a Python library with data mining and machine learning algorithms for pre-processing, classification, modeling, regression, and data grouping. Includes a visual programming environment.

Knime is developed in Java and Eclipse. Free and commercial versions are available. Includes a graphical interface that allows you to manage the data flow and perform pre-processing, collection, analysis, modeling, and reports, and uses the Eclipse extension system to add functionality.

DataMelt, also known as Dmelt, is a calculation platform that provides statistics, numerical and symbolic calculations, scientific visualization, among other features. Among the data mining capabilities, it offers linear regression, approximations to curves, cluster analysis, neural networks, fuzzy algorithms, analytical calculations and interactive visualization of 2D / 3D graphs and histograms. It can be used by the IDE or through the Java API. Successor to the jHepWork and ScaVis applications, used in data analysis.

Apache Mahout is developed by Apache. Library of machine learning algorithms to create clusters, classifiers, and frequent pattern detection. It can be used in distributed mode integrating with Hadoop (framework for the processing of large data sets, using computer clusters, through simple programming models) to analyze large volumes of data.

ELKI (Environment for Developing KDD-Applications Supported by Index-Structures) was developed in Java. It focuses mainly on cluster analysis and anomaly detection. It provides a graphical interface for viewing the results of the application of the algorithms.

KEEL, an acronym for Knowledge Extraction for Evolutionary Learning. Java-based tool. It provides a graphical interface that allows you to import, export, edit and view data in different formats, and to apply data pre-processing, statistical libraries and some data mining and evolutionary learning algorithms. Uses computational intelligence to evaluate the behavior of the algorithms.

Rattle, whose name means ‘R Analytical Tool To Learn Easily’, was developed in the R language. Available for Linux, MacOS and Windows. With a graphical interface, it allows statistics, groupings, data modeling and visualization and the use of the R language. One of the main features is that all interactions with the user interface are recorded in R scripts so that they can be reused without the interface.

2.6. Related work

Relevant research has been made on serious games design. Technologies for the implementation of e-health solutions for children with special needs are available. Affordable solutions can be implemented to provide more inclusive e-health tools. Some hardware and software combinations allow building complex systems that can provide therapists and patients with better tools for achieving therapeutic results. Yet, projects providing solutions for therapy with children with special needs are not very common.

Systematic mapping study Designing serious games presents technical complexities that most therapists do not have. This solution could benefit from the inclusion of a Domain Specific Language (DSL). The DSL could be valuable tool for therapists and developers.

As a preliminary phase in the research of a DSL for the design of Serious Games (SG) for children with special needs, the previous and current research on the field must be assessed. This review is expected to summarize existing evidence, identify the gaps in current research, providing background and a framework for future research. To this effect, a study on available systematic reviews on a set of relevant domains was made, offering an overview of current contributions, and providing useful information for the setup of a systematic study. A search was made in selected data sources. A set of reviews [59] [60] [61] [62] [63] [64] [65] were considered relevant for inspection providing some insight into the explored domains and relevant information for creating a more inclusive query string. The systematic mapping study focused on works using a DSL for the design of serious games for children with special needs. This systematic review was expected to summarize existing evidence, identify the gaps in current research, providing background and a framework for future research. Other relevant work is also analysed.

Some work has been done on the application of Model-Driven Engineering (MDE) for creating rehabilitation platforms, models, and frameworks. MDE is a model-based software development approach where models are used instead of general-purpose programming languages. MDE usually follows one of two different approaches: using a Model-Driven Architecture (MDA) or using domain-specific languages (DSL). DSLs are, usually, small, and declarative languages for specific problem domains. A DSL can be a wrapper for more complex libraries, hiding details and easing integration. When analyzing DSLs, it's important to determine the language purpose (what the language need to describe from the domain), the language realization (how can it be implemented), the language content (structural, behavior and hybrid), the abstract (model of language concepts, relationships, and constraints) and concrete syntaxes (language representation).

Best practices and methodologies for DSL development have been identified by Mernik et al. [66]. Any GPL associated with an API can act as a DSL. The API provides a domain-specific vocabulary available to the GPL. This is usually the most cost-effective solution and is frequently used. While this can be enough in some scenarios, the implementation of a DSL provides several advantages that are identified:

- appropriate domain-specific notations
- appropriate domain-specific constructs and abstractions
- using a DSL allows analysis, verification, optimization, parallelization, and transformation (AVOPT)
- a DSL doesn't need to be executable

DSL usage promotes reuse of software artifacts like language grammars, source code, software designs, and domain abstractions. Development of a DSL is a complex task requiring both technical and domain-specific knowledge. There are more development techniques than for GPL to consider. Training, maintenance, and support for the DSL can become an issue depending on growth of the user community. The benefits and usefulness of implementing a DSL should be clear and evident. DSL development has a set of phases: decision, analysis, design, implementation, and deployment. It's an iterative process with options in each phase affecting other phases. Patterns have been identified for the decision, analysis, design, and implementation processes in DSL development. Decision patterns, that represent common usage with success implementations of DSL, include notation, domain specific AVOPT, task automation, product line, data structure representation, data structure traversal, system front-end, interaction, and GUI construction. Notation has two relevant sub-patterns: transformation of visual to text notation and adding a user-friendly notation to an existing API. Analysis patterns, used for problem domain identification and domain knowledge gathering, include informal, formal, and extract from code. Formal domain analysis outputs a domain model. The analysis usually provides domain-specific terminology and abstract semantics. An ontology can be useful in this process.

A well-defined DSL specifies a model with precise syntax and semantics. The semantics are related to what is represented in the models. Abstract syntaxes can be code-like models, game object specifications, directional graphs, state machine models, production rule systems, decision trees, or graph transformations. Two distinct techniques are used to create an abstract syntax: Modelware (meta-modeling) and Grammarware (using context-free grammars). Grammars use Backus-Nauer Form (BNF) extensions. Meta-modeling builds a meta-model, a model of models, for describing the model structure, using a meta-modeling language and Object Constraint Language (OCL). The concrete syntax creates a representation of the abstract syntax, and is usually a textual or graphical representation, or a combination of both. Concrete syntaxes can be graphical, textual, tree-view, or form-based view. Model trans-formations allow automation of model manipulations. Possible transformations are Model-to-Model (M2M), Model-to-Text (M2T), or in-place transformations, to change, restructure or optimize the model. Usual tools related to DSL are model editors, model interpreters, semantic validators, and code generators.

Systematic literature reviews (SLR) are evidence-based studies, consisting of the search and analysis of individual studies, to identify, evaluate, and interpret the current state of the research on a particular subject. Contributing studies are called primary studies and the SLR are considered secondary studies. Other types of studies include systematic mapping studies (used for finding evidence clusters and deserts) and tertiary studies (systematic reviews of systematic reviews). SLR are used to summarize existing evidence, identify gaps in current research, providing researchers to background and a framework for future research.

The study Planning the review includes the identification of research questions. Defining the research questions is the most important part of a systematic review, as this is the basis for all the review methodology. Research questions in software engineering try to address relevant domain issues as assessing the effect of a technology, the frequency or rate of development factors or project success or failure. Other addressed issues, when asking research questions, include the identification of associated cost and risk factors, the impact on the reliability, performance and cost, and a cost benefit analysis of the solutions application and usage.

Table 17 - PICOC analysis

Population	This research will review studies proposing models for the design of serious games for children with special needs and, specifically, providing modeling languages for the effect.
Intervention	A model-driven approach and the provision of a modeling language.
Comparison	Not applicable for this study
Outcome	Identification of existing models and modeling languages, and the characterization of existing modeling languages for the target domain
Context	Studies in both academic and industry environments are considered.

The PICOC criteria is a tool used to help frame research questions [2]. PICOC stands for Population, Intervention, Comparison, Outcome and Context. To be able to find the right research questions, a PICOC analysis was made (Table 17).

The research questions have been designed based on this preliminary analysis. These questions define the information that is expected to be retrieved from each primary study and are exposed in the following list:

1. What is the status of research in using modeling languages for serious games?
2. Which conferences and publications present studies on this domain?
3. What kind of modeling languages are used and with what purpose?
4. What problems are addressed with the use of existing modeling languages?
5. What are the existing modeling languages for serious games?
    a. Is the proposed solution theoretical or does it have a prototype?
    b. Has the proposed solution been validated?
    c. Is the proposed solution currently being used?
6. Which tools are available for DSL development?

A search string was derived from the research questions. The objective was to include the more suitable terms to find domain relevant research that helps answering the questions asked. For this study, we looked for model-driven frameworks for serious game design, and specifically, focusing on children with special needs and providing a domain-specific language.

The search strategy aims to detect as much of relevant literature as possible. Search was applied in relevant knowledge bases for the researched domain. Data repositories used include Dblp, Google Scholar, Web of Science, ACM, IEEE Xplore, Science Direct and Springer.

Inclusion and exclusion criteria were used to filter initial search results and determined which studies were selected for review. The search terms were in English and only studies in the English language were considered. Only peer-reviewed studies were considered. A date filter was setup to limit the study to only include studies since 2018. Filtering by publication context was also applied, considering only studies from topic related conferences and journals.

During data extraction and monitoring, data retrieved from each study was used for classification of the selected works. Works were checked for:

- What is the main domain of intervention?
- Is it model-based or does it provide a conceptual model?
- Does it provide a DSL?
- Does the study provide tools that help achieve the required functionality?
- Does the study provide a taxonomy?
- Is it ontology-based or does it provide an ontology?

Data synthesis was achieved by interpreting collected data.

Searching The search string was applied in each of the selected repositories. After applying the inclusion and exclusion criteria, and removing duplicate entries from the several sets, a final set of studies was obtained. Restricted access studies were discarded. From the remaining set of studies, the most relevant were selected. A process of snowball sampling was used for searching related research. A set of identified studies were explicitly included in the final selection set due to being considered relevant. The resulting selection of studies is the corpus for this study analysis (Table 18).

Table 18 - Search results

Data source	Search string	Inclusion criteria	Selected	Analyzed
Dblp	5	4	1	3
b-on	218	169	3	1
Web of Science	143	29	9	2
Springer	81103	123	35	3
ACM	7220	101	3	1
Science Direct (Elsevier)	610	79	2	0
IEEE Xplore	87	19	4	4
Google Scholar	449	162	11	0
Total	**89835**	**686**	**68**	**11**

Selected works analysis

An individual analysis of each selected work was performed to extract relevant information. A synthesis of gathered information can be seen in Table 19.

Table 19 - Selected work categorization

work	domain	mdd	dsl	tools	taxonomy	ontology	prototype	validated
Avila-Pesantez et al.	Learning	✓	●			●	✓	✓
De Lope et al.	Learning	✓	✓	✓		●	✓	✓
Khowaja et al.	ASD	✓	●			●	✓	✓
Aziz et al.	CPS	✓	✓			●	✓	✓
García et al.	Learning	✓	✓			●	✓	
Garcia et al.	Inclusion	✓	●	✓	✓	●		
Peñeñory et al.	Therapy	✓	●	●	●	●	●	✓
Gómez et al.	CPS	✓	✓			●		
Guan et al.	CPS	✓	✓			●	✓	✓
Da Silva et al.	SaSs	✓	✓			●	●	✓
Karim et al.	Therapy	●	●			✓	●	
Da Silva et al.	SaSs							
Pescador et al.	MDE		✓		✓			
Tang et al.	Learning	✓	✓	✓	✓	●	✓	

Avila-Pesantez et al. [67] provides a conceptual model for serious game design and discussion about model component relations. The model has 4 phases: Analysis, Design, Development, and Evaluation. The Analysis phase produces a requirements document based on several components including therapeutic goals, and user profile. The Design phase produces a game requirements document and has several components including environment, game mechanic, scenario, game objects, and technology. The Development phase produces a game prototype based on the previously produced documents. The Evaluation phase tests the prototype on several components including goals, and implementation, and validates the resulting game. The process is iterative until reaching the desired game for release. A serious game was developed to help children with learning disabilities. A case study analysis was made concluding that the game helped children in the level of concentration, hand-eye coordination, motor skills, and cognitive reinforcement.

De Lope et al. [68] provides a set of meta-models for adventure-based educational serious game design and a graphical notation for component representation, based on the UML standard. A conceptual model for designing serious games is proposed and is based in 6-layer components (identity, immersion, interactivity, increasing complexity, informed teaching and instructional). A prototype SG was developed, and validation tests were made.

Khowaja et al. [69] is a framework for the design of vocabulary-based serious games with children with Autism Spectrum Disorder (ASD). The conceptual model is based on 3 interconnecting layers (theory, content, and game design). A prototype SG was developed with expert validation.

Aziz et al. [70] propose a DSL for designing Cyber-Physical Systems (CPS). The DSL is used for modeling structural and behavioral aspects of CPS systems. Both abstract and concrete syntaxes are defined. The DSL was applied in two prototypes.

García et al. [71] uses a DSL for agile development of multi-platform educational serious games. The DSL allows definition of platform-independent information such as menus and levels of the game, and allows generation of a game model, based in XML files.

Garcia et al. [72] present a framework for collaborative game design. The framework is based in 3 pillars, a game meta-model, a platform for game design, and a collaborative work model. Users can design, share, and play the games using a collaborative and inclusive model. The game creation process has iterative phases: Conception, Conversion, Evaluation, Creation, Enrichment, Distribution, Use, and Conclusion. Prototypes were developed using the framework.

Peñeñory et al. [73] proposes a methodology for the design of serious games for psychomotor rehabilitation of children with hearing impairments and motor delays. The methodology provides guidelines, directed at developers, for designing therapeutic experiences. The method considers the identification and design of the formal elements, the exercises to perform and the interaction to achieve.

Gómez et al. [74] introduces a model-based approach for event-driven Internet of Things (IoT) CPS architectures using AsyncAPI, a DSL for automating architecture design. AsyncAPI is a message-driven REST API using the JSON standard. An API specification and JSON grammar are provided. The Eclipse Modelling Framework (EMF) was used for building the DSL. The XText Framework is used for defining both a concrete and an abstract syntax, and tool generation such as editor and parser. AsyncAPI uses an internal Java DSL for event-driven communication.

Guan et al. [75] propose a DSL for modeling stochastic and hybrid behaviors of CPS. The DSL allows the design, modeling, and simulation of CPS. Both abstract and concrete syntaxes for the DSL are described. A case study of DSL application is presented.

Da Silva et al. [76] propose a DSL for designing Self-adaptive Systems (SaSs) by extending the metamodel of the UML language. SaSs systems can adapt their behavior during runtime in response to contextual changes. The UML profile developed was analyzed by a focus group and considered more expressive than the UML standard for producing SaSs conceptual models.

Karim et al. [77] use an ontology-based approach for formal description of concepts related to special needs. This ontology describes special needs user's impairments, interface characteristics, and their relations, to provide customized interfaces.

Pescador et al. [78] propose a pattern-based approach for the construction of DSMLs, and providing a tool and a taxonomy.

Da Silva et al. [63] made a SLR of UML-based DSML for self-adaptive systems.

Van Rozen et al. [64] made a systematic study on languages of games and play, with a specific section on DSL usage, identifying five works that allow domain experts to help design games scenarios (mostly for learning and educational purposes).

Tang et al. [79] present work on a DSL for educational serious game design. This work is very relevant even if focusing on education. The DSL design process followed 4 phases:

1. Identifying abstractions and relationships
2. Specifying language concepts and rules (meta-model)
3. Create a visual representation (notation)
4. Definition of generators

Work has been done until phase 3 providing a conceptual framework describing abstractions and relationships of SG. The framework is used as basis for development of a DSML for use in SG MDD development environment and provides abstractions for assets and functionalities. The SG design conceptual framework follows a 3-tier architecture: flow, scenarios, and objects. An analysis of SW modelling languages identifies relevant options: Z-notation, Petri-nets, BPMN, IDEF, UML. The framework uses the Viable System Model (VSL) to fully represent in-game components. The DSML provides the semantic notation to model serious games components. UML class diagrams and extended state diagrams were used to model flow and in-game components. The serious game modelling is divided in data modelling (definition of objects, flows, and processes) and visual modelling (game components positioning, environment construction, and GUI layout).

Discussion The results obtained from this study should answer the research questions. So, assessing the status of research in using modeling languages for the design of serious games, evidence shows that research in the design of serious games has relevant contributions following model-driven approaches, with several proposals of conceptual models. Some studies propose DSL specifications for distinct purposes. Several analyzed works provide DSL designs in their contributions. Detected DSL and their purposes are described in Table 20.

Table 20 - DSL categorization

Work	purpose	realization
De Lope et al.	Game component modeling	UML graphical notation
Aziz et al.	System modeling	Graphical DSL
García et al.	Multiplatform game design	XML-based DSL
Gómez et al.	CPS architecture design	JSON-based REST API
Guan et al.	Model CPS behaviors	DSL
Da Silva et al.	Model representation	UML extension

Specific problems are addressed with the use of modeling languages. DSLs are being used to abstract model components, relations, and methodologies, allowing access to provided functionality to non-domain experts. Common usages include system and game modeling, and model component representation.

Assessing existing modeling languages used for serious games design, study results showed that UML is often used by extension or specialization. REST API is a common solution for DSL realization. Concrete syntaxes are often implemented through XML or JSON.

Jacome-Guerrero et al. [80] identify several tools available for MDE and DSL development, summarized in Table 21. The Eclipse EMF is a modeling framework providing tools that allow implementation of DSL solutions. Several frameworks integrate with Eclipse to build visual editors as plugins. GMF, Eugenia, Spray, or Sirius are model-based frameworks. The Graphiti framework also integrates with EMF to build visual editors, but uses a distinct approach, being based on a Java API and coding.

Conclusions of the systematic mapping study Some conclusions can be obtained from interpreting these results. DSL implementations are being used for serious game design. UML extensions and specializations, and REST APIs are technologies that provide viable solutions for DSL implementations. XML and JSON are viable options for designing the concrete syntax of a DSL. This study made an overview of the status of research in the development and usage of model-driven development and domain-specific languages for design of serious games for children with special needs and provided relevant information for future research.

Table 21 - DSL Tools

Category	Tools
IDE	OMG Architecture, EMF Architecture, MetaCase, Microsoft DSL Tools, Project TCS, Proprietary
Metamodeling languages (Abstract syntax)	MOF, Ecore/OCL, GOPRR, Domain Model, KM3, MetaDepth
Graphical Editors (Concrete syntax)	GMF, EuGENia, Graphiti, Spray, Sirius, DiaGen, Microsoft DSL Tools, MetaEdit+, Devil, Concrete, AToM

Textual Editors (Concrete syntax)	Xtext, EMFText, TCS
Transformation of models	ATL, QVT, RubyTL
Code generation	Acceleo, JET, MOFScript, Xpand, Mof2Text
DSL examples	SQL, HTML, CSS, Matlab, Dot, PlantUML, Sed

Other relevant works Relevant research has been made on serious games. Technologies for the implementation of e-health solutions for children with special needs are available. Affordable solutions can be implemented to provide more inclusive e-health tools. Some hardware and software combinations allow building complex systems that can provide therapists and patients with better tools for achieving therapeutic results. Yet, projects providing solutions for therapy with children with special needs are not very common.

The use of serious games, as a complement to conventional therapies, has been validated in several studies [81]. NUI sensors are used for capturing user interaction in a more natural non-intrusive way. The Microsoft Kinect and Leap Motion are examples of NUI sensors that have been extensively used for developing health-related projects with the implementation of successful prototypes [17] [82] [83] .

Wattanasoontorn et al. [84] made a survey on serious games for e-health proposing a classification system for e-health games, based in 3 main sections: serious game, health, and player. The serious game section classifies the game in purpose and functionality. The health section classifies by state of disease. The player section classifies by player category.

Remote systems can be used for training, as a complement to presential clinic therapy. It's possible to setup smart environments inside clinic spaces and even at patient's homes. Pervasive and ubiquitous computing are paradigms that consider a world of permanent interaction with technology. Technology is embedded in surrounding smart devices and is not perceptible to the user, providing tools to build smart environments. Smart environments are a subject of research and application in many areas of knowledge. A smart environment acquires data and applies knowledge, adapting to improve the user experience. Features of smart environments include remote control of devices, device communication, data acquisition and distribution, smart devices and enhanced services, prediction and decision-making capabilities, and networking standards, protocols, and regulations. Smart devices have networking and data processing capabilities being able to provide enhanced services in the environment. These devices can access remote data sources and can intercommunicate being able to complement information with external data. Collaboration among smart devices to fulfill some specific tasks is a current research topic. Smart devices provide new solutions and specialized hardware. The automation process is a continuous cycle based on sensors collecting data, data processing and control commands sent to actuators for actions to be executed. The automation and adaptation process of the environment allows the use of machine learning algorithms for prediction and decision-making capabilities. Device communication commonly uses the Internet or other wireless communication protocols like Bluetooth.

A smart room is an indoor space that is setup to use several embedded services for the tracking and interaction of users. Multiple sensors can be used for tracking and measurement of user actions. Actuators are used for providing feedback to user interaction. Chen et al. [7] define smart rooms as indoor spaces configured to let users of the physical space invisibly interact with the virtual world of computers during their normal activities. Several existing services in the underlying networked computing infrastructure continually observe and track what users do using multiple sensor modalities, controlling actuators that modify the physical state of the room or provide other feedback to the occupants.

Recommendation systems help the decision-making process, for instance, helping therapists and patients select the most adequate option for their goals, in the relevant context. A smart system using a recommendation system suggesting relevant rehabilitation exercises, based on user skills, was presented by González et al. [85]. The TANGO:H platform allows the design of accessible educational games using the Kinect sensor. The system provides personalized game modes based on the history and preferences of each user. A configurable level system allows more precise adaptation and personalization of exercises. The difficulty of exercises increases with user experience preventing exercises to become repetitive and boring. The platform has two main components: Game (focusing on the users and game interaction) and Designer (focusing on specialists, and exercise design and assessment). Exercises can be physical, cognitive, or hybrid. Physical exercises are for reduced mobility or physical impairments. Cognitive exercises promote improved perception, memory, learning and reasoning abilities. Hybrid exercises do not follow any pattern. Game modes are classified as single player and multiplayer. Gesture interaction and recognition is achieved using the Kinect sensor. For system simplification and ease of use, the 3D skeleton provided by the sensor is projected on a 2D plane, with game interaction using the 2D projection. This provides a less complex system logic and more accessible exercise design. The Data model associates each exercise with a set of pairs of skills and values, representing the mastery level needed, for each skill, when performing the exercise. The User model registers the skills gained and their respective values. The user history registers the performance for each exercise. Skills monitored include speed, equilibrium, coordination, and reflexes. The platform is implemented using .Net, WPF, C#, and XML.

The potential of sensorial stimulation to extend interactivity in games for children with CP was studied by Oliveira et al. [86]. The use of tangible user interfaces (TUI) has been explored for rehabilitation of CP patients with the objective of promoting motivation and engagement. Weightman et al. worked on a game and force feedback interface for children with cerebral palsy, for arm exercises [87]. Zirbel et al. [88] present VRehab, a low-cost mobile VR system for upper limb rehabilitation of post-stroke patients. The system provides a motion control interface (MCI), allowing collection and processing of EMG and other IMU data, using the Myo arm-band sensor. The MCI can be used to induce haptic feedback for user assessment of interaction success.

Every patient has a different combination of pathologies and distinct severity degrees for each one. For a system to be inclusive, adaptation is an essential part of the process. The system response should adapt, or be adaptable, to the user’s condition, to provide a similar experience among distinct users and allow patients with a more severe condition to be able to use it. Omelina et al. [30] discuss a specialized configurable system architecture for games focusing on neuro-muscular rehabilitation. The proposed architecture aims to increase the adaptability and usability of exergames both for patients and therapists. The solution is based on three layers: a game controls layer for hardware abstraction, a game configuration layer for game variables controls and the game logic layer. A prototype with four mini games is presented, also providing tools for progression analysis.

A model for designing serious games focused on children health is proposed by Kemble et al., in [89]. The model is based on the foundations of gameplay, learning, and usability and is structured in 6 sections, contributing 15 guidelines for designing a serious game: designed game experience (goals and limits, challenge and accommodation, facilitate flow), usability in games (agency and engagement, culture and collective play), usability principles (intuitive integration, communication and custom content), kinesthetic/active learning (reasoning and comprehension, realistic metaphors to transfer), learning through games (problem solving and processing, guided construction, behavior modeling and mentorship), and models of learning (flexible direction, competency and confidence).

A process framework for serious game development focused on motor rehabilitation is proposed, by Alcover et al. [90]. The PROGame framework follows an iterative approach based on two dimensions. The first dimension includes project setup, an iterative process flow consisting of 4 activities and a clinical trial. The process flow activities are planning and control, modelling, construction, and evaluation. The second dimension supports 3 incremental phases of the clinical trial: interaction mechanism, interaction elements, and serious game. Desirable features of rehabilitation serious games are identified. The framework methodology was applied for development of a serious game for improvement of balance and postural control of CP adult patients.

Bonnechère et al. [91] proposed standardization strategies to improve treatment comparison and serious games use in rehabilitation, for a more efficient analysis of results. The use of SG for rehabilitation with CP patients is validated and the relevance of highly configurable SG scenarios, to provide a more inclusive response, is highlighted. Games can be adapted or built from scratch. Adapted games must have configurable options, such as playing without background, to make them suitable for the distinct pathologies and pathology severity of each patient.

Cloud-based data analytics is used by Caggianese et al., in a serious game, for virtualization and personalization of rehabilitation treatments [92]. Authors describe and implement a telerehabilitation system based on serious games and in-cloud data analytics services. The system focuses on stroke patients, providing components for real-time acquisition of patient's motor data and a decision support service for their analysis.

Tang et al. [34] designed an ontology for documenting serious game design. An ontology is a tool that allows quick identification of entity relations and provides a reference standard for term usage. Game design languages and meta-models are formalized methods helping inexperienced game designers to develop correctly designed solutions.

Tang et al. [79] worked on a DSML for designing serious games. The DSML provides the semantic notation to model serious games components. UML class diagrams and extended state diagrams were used to model flow and in-game components. The serious game modelling is divided in data modelling (definition of objects, flows, and processes) and visual modelling (game components positioning, environment construction, and GUI layout).

Tang et al. [93] proposed a MDE framework for the design of educational serious games. The aim of the framework is to help domain experts, with no technical background, in the design of serious games. The framework identifies a set of requirements that should be met:

- embed best practice for SG design (gameplay, game structures, game content, motivation, and workflow)
- provide support for existing game software frameworks
- loose coupling of artefacts for maximum flexibility
- integration of external media and game functionality
- use assistive UI for encapsulating technical aspects of games

Ferreira et al. [94] researched lower limbs rehabilitation using Unity and the Kinect v2 sensor. Several body joints are detected and metrics such as distance between feet, left and right feet usage frequency, execution time, angles and velocities are recorded. The system provides data storage using a remote database.

Interaction with real life IoT devices has been explored by Kosmides et al. [95]. The project provides a gamification framework for educative serious games. Real world information is integrated into the game world for the creation of a pervasive and more immersive environment.

A methodology for the design of serious games for psychomotor rehabilitation of children with hearing impairments and motor delays is proposed by Peñeñory et al. [73]. The proposed methodology provides guidelines, directed at developers, for designing therapeutic experiences. The method considers the identification and design of the formal elements, the exercises to perform and the interaction to achieve.

Chapter 3

Concept

In this chapter the framework conceptual model and architecture are introduced and explained. The development of a serious game is a complex task. Developing a serious game focusing on children has specific challenges that make the development process differ from targeting older users. Children with special needs have extra requirements and constraints that must be taken in account. The adaptation and personalization of solutions is very important. The conception of a model for the development of serious games for children with special needs is a valuable tool that could help as a guideline for other projects. The framework conceptual model is presented, describing the architecture and main components. The expected contributions from this model are exposed.

3.1. Conceptual model

The conceptual model for this framework is based on the notion that children with special needs must perform physiotherapy exercises for rehabilitation and regaining functionality. Due to the repetitive nature of the rehabilitation and the characteristics of the population, serious games are used to motivate and promote engagement while exercising. The complexity and extreme variation of focused pathologies require flexibility and adaptation strategies from solutions. Data from user interactions must be collected and analyzed to be usable in decision-making processes. Therapists and patients can benefit from that acquired knowledge.

Web-based solutions provide almost universal access, using the internet, and have relative low-cost requirements for hardware and software setups in the client side. The framework provides a multiuser web-based platform for patients, therapists, and clinics. The platform provides tools for the several roles and allows data analysis on those multiple levels. The platform can be used by patients in clinical or home environments, in a supervised or autonomous way, providing tools for enabling the setup and integration of smart environments.

The platform knows patients by their platform profile. Therapists can manage and monitor patients and patient exercise schedules, consult and update the patients' profiles, and design of patient focused serious games, based on the patient profiles, allowing the personalization of exercises, with data collection of exercise results and sensor samples.

We could say, in our good will, that **PLAY** is a **P**latform for **L**earning **A**bout **Y**ou, referring to the patient. The platform provides a data collection repository and tools for data analytics. Data collected allows the use of machine learning techniques for retrieving relevant information. The concept includes the possibility of extending the platform by integrating with external applications and their data collection implementations. Sensors are used for collecting data to be analyzed and provide information about exercise performance and patient progression.

By providing a web services API, the platform is open to the possibility of expansion, integrating with applications and devices that consume it. Actuation devices expand the applications environment promoting more immersive experiences and should be low-cost IoT devices to be affordable both by clinics and domestic users.

The framework model is composed by 3 interconnected layers. The domain layer relates to the child patients and their needs, the therapies, and results analysis. The technology layer includes software and hardware solutions. The game layer relates to game design and serious game elements. The framework intersection relates to the game performance and profile progression analysis, by therapists and machine learning tools, producing knowledge. The high-level framework model is displayed in Figure 8.

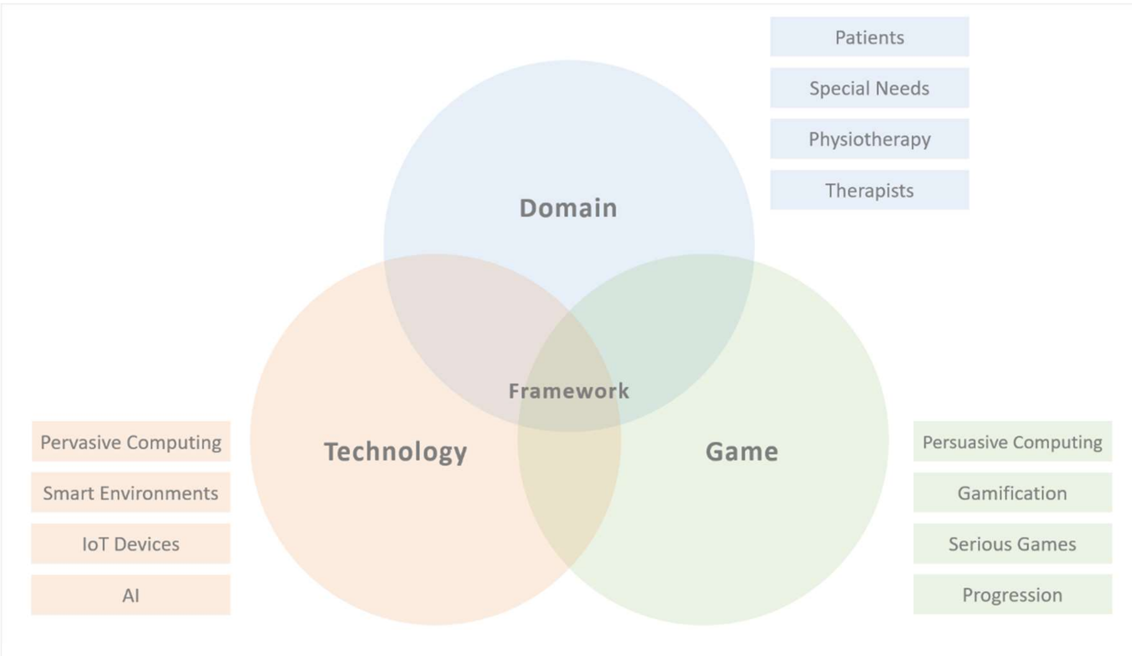

Figure 8 - Framework model

3.1.1. *Domain layer*

The domain layer relates to the child patients and their needs, the therapies, and results analysis. The current patient state and profile is defined by the therapeutic needs. The patient profile is built considering the therapies needed and metrics validated by therapists. The profile is dynamic, recording changes and taking in consideration the patient's platform usage history and game results. Sensor samples obtained during the exercises allow for exercise reconstruction and performance analysis.

3.1.2. *Technology layer*

The technology layer includes software and hardware solutions, abstracting the technological aspects of the framework. This layer includes technologies used for the platform and communication and control of sensors and actuators. External applications consuming the provided API integrate the framework providing added functionality by their own implementations of the games and supported sensors.

3.1.3. *Game layer*

The game layer relates to the gaming components of the framework including game design and serious game elements.

Actions are the building blocks for games. Parameters are used to define configurable elements for the platform, games, and actions. Games have a structure based on levels. Each level has one or more sequences of actions. The actions model distinct interactions with the system and can have multiple settings, configurable through action parameters. An action can require a specific device. Devices can be sensors and actuators. Sensors and actuators are external components. Game executions generate game results and samples. Samples allow movement reconstruction. Movement samples and game results provide data for analysis. Games can be configured through game parameters.

External applications implementing serious games should use the framework game model and guidelines to integrate with the platform.

3.1.4. *Framework*

The intersection of the layers provides the methodologies and tools for this framework. The framework model is based on a set of premises. Patients have therapeutic needs and require physiotherapy services. Physiotherapy sessions often require repetitive exercises. Serious games are used to promote engagement and motivation. A platform for assisting therapists implementing physiotherapy protocols using serious games is provided. The platform allows management of users and serious game therapy sessions. Users can have roles of patient, therapist, clinic, or administrator. Patients have a therapeutic profile. The patient profile is based on personal info, usage history, therapist analysis, and results.

Therapists configure games. Games are composed of sequences of actions organized in levels. Actions have parameters defining the action runtime behavior. Actions relate to sensor devices. Therapists schedule game executions in patients' exercise schedules. Games are recommended based on the patient profile. Patients play games. Games can be platform or external games. Games use local sensor and actuator devices. Games send results and samples to server. Therapists can monitor the game flow and adjust game parameters in real time.

The framework model integrates the components of each layer allowing several advantages. Each component contains a set of own internal models. This framework proposes and aims to provide:

- A model for a patient profile.
- A flexible model for games.
- A recommendation system for games based on the user profile.
- Repository of game executions, results, and samples.
- Data source for machine learning algorithms and other tools for obtaining knowledge.
- Abstraction of devices.
- Easy access to data and tools using the Internet.

3.2. Framework components

Designing serious games for special needs presents distinct challenges. Many interconnected components are used for this platform. The framework components have distinct specificities that need introduction and clarification. The user profile and game model are central parts of the framework model. Adaptation is a concept that must be available for better respond to users' needs. Data acquisition and analysis is vital for building a knowledge base allowing learning and adaptation.

3.2.1. *Users*

The user model used by this framework is based on 3 distinct user roles that follow a set hierarchy, that can be visualized in Figure 9. Clinics can have therapists and patients. Clinic therapists can follow the patients registered at that clinic.

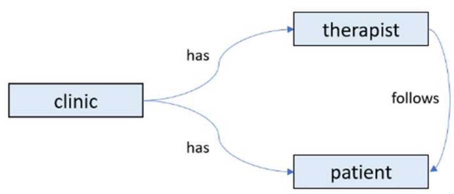

Figure 9 - User roles and relations

3.2.2. *Knowing the patient*

The patient profile needs to be able to model therapeutic needs of the patients. It is the data structure gathering information about skills and needs. Maintains the base information allowing the system a first iteration at personalization. Relevant patient profile areas can be observed in Figure 10. Each patient has a current profile. Profile changes are registered and timestamped.

The personal information refers to relevant patient identification data including name, birth date and gender. Biometric data record relevant patient data such as arm length or affected side.

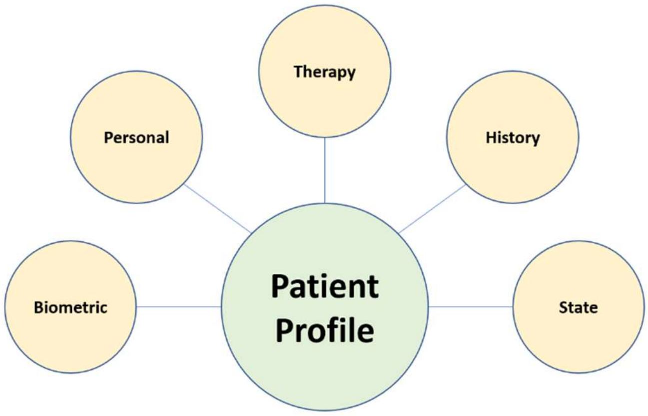

Figure 10 - Patient profile components

A preliminary analysis is required to identify the needs for each specific patient. Therapists can manage the current therapy categories needed by the patient. Four therapy categories are available: Physical, Speech, Music, and Cognitive. Patients can require interventions in one or multiple categories.

The platform usage history provides valuable information for the patient profile. Game results and sensor samples also provide data for assessment of user progression and profile updates.

A pseudo-Likert scale with 5 stages is used for assessing and keep track of 3 distinct metrics. The profile allows tracking and update of the current state of therapeutic severity. This value is managed by the therapist as a personal perception evaluation. The current emotional and motivational states are also available, also using a 5-stage scale. State changes are registered, and states progression can be assessed.

The user profile can provide information for external applications, allowing the ones that support it, an extra level of adaptation and personalization of the application’s contents and flow, for the current user state and therapeutic needs of that user.

3.2.3. *Modelling exercises*

Games must take in consideration the special needs of patients. To model the therapeutic exercises, the framework proposes a game model based on actions and parameters, for added flexibility. Each action can have a set of configurable parameters. These parameters define the configurable parts of the action when defining a game structure.

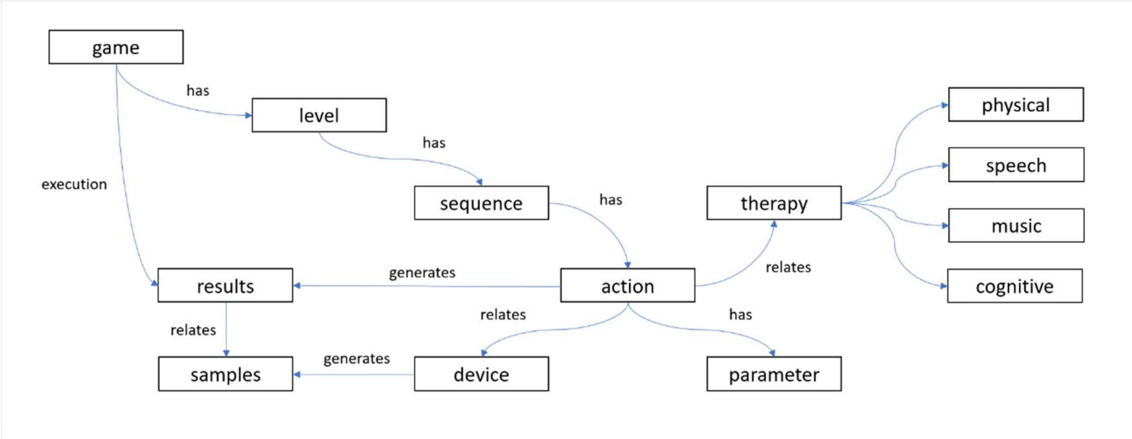

Figure 11 - Game and action relations

The game model consists of game structure and configuration. The game structure is based in actions, that can be organized in sequences. Sequences of actions can be grouped as levels. The game action model tries to provide a way to contextualize and frame actions to be modeled in the platform. Every action has a set of common attributes like time and score, and specific attributes such as angle or shape. Distinct actions are modeled for specific therapeutic interventions. Every game in the platform must follow the game model. The original games in the JPK platform can be modeled using this approach.

Actions and Parameters

Actions are the basic components for an exercise. Actions can have configurable parameters that define their variable parts. Actions are associated to therapeutic needs and related therapies. Actions can also have associations with specific devices (sensor or actuator). The actions concept provides a flexible way for designing sequences of actions focusing on multiple therapies. Action examples are listed in Table 22.

Table 22 - Actions and Parameters

Action	Verb	Parameters	Physical	Occupation	Speech	Music	Cognitive
reach an object	reach	angle, distance					
touch an object	touch	angle, distance					
drag object to target	drag	angle, distance, target					
grab object	grab	angle, distance					
follow path of objects	follow	shape					
say a phoneme or word	say	phoneme, word					
play a sequence of notes	play	notes					
identify two similar objects	associate						

Parameters can be discrete values (integer or float), or collection identifiers. Common parameters are time, for definition of a time limit for the action, and score, for defining a value to be gained when the action is successfully completed. Specific parameters depend on each action scope. Collection identifiers refer to existing datasets available in the platform.

Game API

Games communicate with the main server through a REST API. The API allows game applications to fetch the game structure and parameter configuration and use those settings in the implemented game engines. The API also provides tools to manage external application authentication.

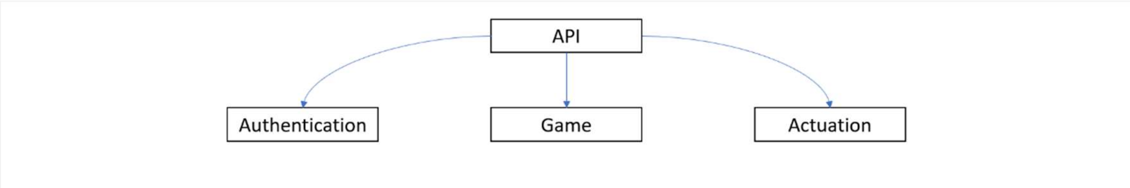

Figure 12 - API services

Main game loop

Games follow a specific game flow. The main game loop can be observed in Figure 13. After the pre-game actions, the game loop starts. A game is a sequence of actions. Each action invokes the game action module for the action execution. After conclusion of the game action, the result and samples are stored. The game state is then checked to verify if any stop condition is met. If the game finishes, post-game actions are performed.

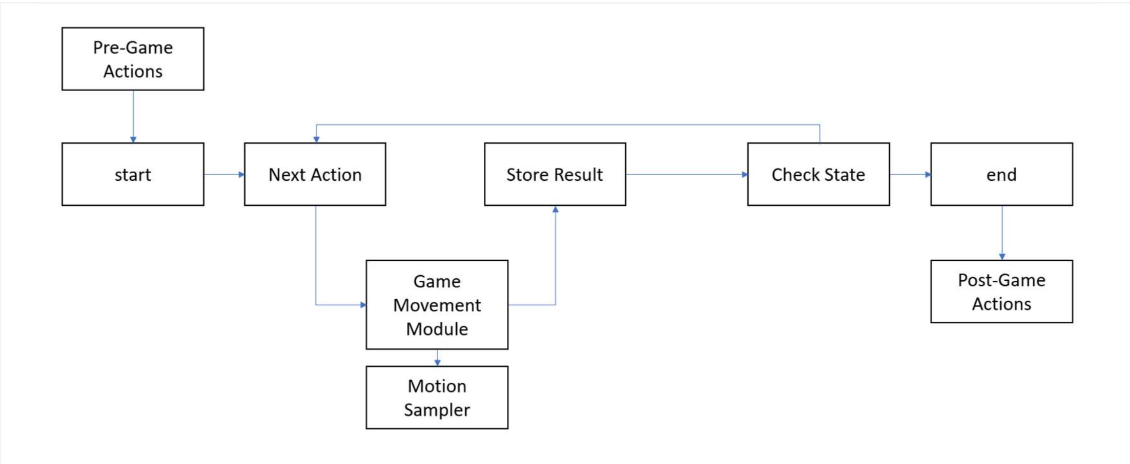

Figure 13 - Main game loop

To be able to detect if a game has reached the end, one or several stop conditions must be set. Possible stop conditions for a game can be a timeout, achieving some score, reaching several plays or other events like leaving the game while playing.

Game Mechanics

The games mechanics are defined by the composing actions. To make the game more dynamic and motivating different gameplay modes are available. Levels are used to promote a sense of achievement and progression. Levels can be used in the platform as continuous progressive stages or as a finite number of stages to reach. Badges and Rewards are used to promote user engagement and effort recognition. Every relevant move in the game is rewarded in some manner. The simplest form of reward is increasing the score. Badges are used to acknowledge user progression using the platform.

3.2.4. *Adaptation*

Adaptation is of vital importance for user inclusion. Due to the distinct pathologies and pathology combinations presented by patients, and the extreme variation of patient pathology severities, software platforms and applications must adapt to user's needs.

The patient profile registers the clinical, emotional, and motivational levels for each patient. These levels are based on a scale from 1 to 5, with distinct contextual meaning. Emotional and motivational levels best values are at the top of the scale while for the clinical level the 5 means the most severe range of pathology. These values are entered by the therapist and are based on their perception of the patient condition. The patient condition varies over sessions and the levels can be updated at any time, allowing the platform to better adjust to patient needs.

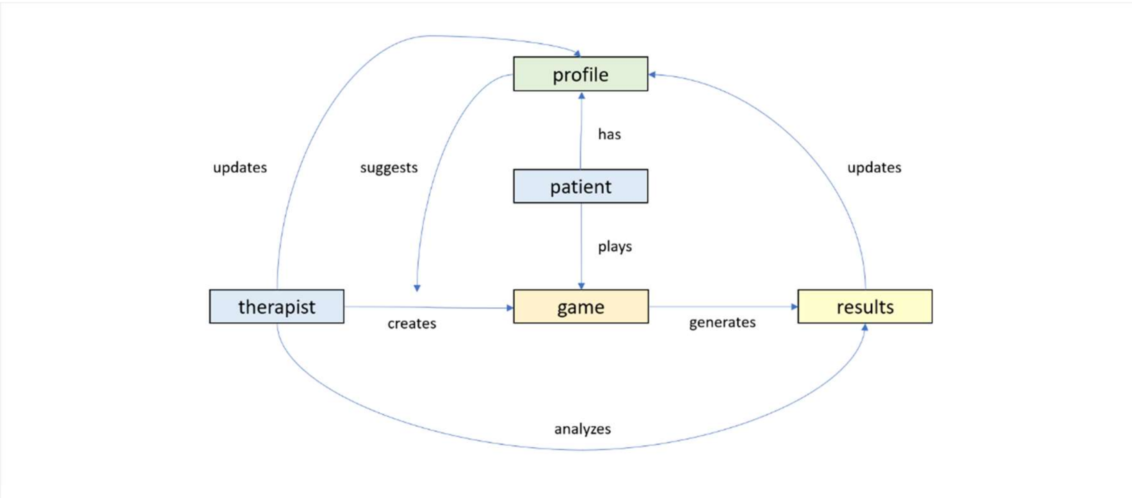

Figure 14 - Game interactions diagram

A recommendation system for the games, based on the patient profile and the game definition, allows therapists to quickly identify the most suitable games for a specific patient. Patient needs are the focus here. All patients are different and have distinct combinations and degrees of pathologies. The serious game tries to respond to the user need in the best way possible. Depending on the level of game configuration available that can be more efficiently achieved or not. Each patient has distinct impairments that limit their ability to play certain games. Depending on the patient limitation, some games requiring specific abilities or skills are not adequate for that patient. The platform can, based on this concept, provide therapists with the necessary insight for selecting the correct type of game for a specific patient with specific impairments. When creating a therapy protocol for a patient, the system should provide a personalized pre-configured protocol suggestion based on the patient profile. Games created by other therapists can also be suggested, if relevant for the patient current needs.

3.2.5. *Game results and samples*

Therapists need to evaluate the patient performance when executing the game. Every executed game produces game results, based on time and score, that can be compared and evaluated. To be able to compare results, a standard system should be used. Depending on the specific game and sensors used, the game can capture samples of the patient skeleton joints for posterior analysis. This data sampling allows movement reconstruction providing a tool for performance analysis. A patient profile report provides the current state of the patient.

3.2.6. *Performance analysis and progression assessment*

Data sampling allows the reconstruction of the played game actions. The full game can be replayed. Skeleton visualization is possible when using depth sensors with skeleton tracking. The captured skeleton joint positions contain the element coordinates for the positioning in the 3D space. This visualization mode allows therapists to observe the skeleton from an arbitrary position in the 3D space, for better understanding of action performance. Progression analysis is possible by evaluating metric evolution during a time frame. Cloud services can be used for the processing and analysis of results and sampled data.

Chapter 4

Platform

A computational platform supporting the proposed model has been implemented. The solution is analyzed, describing its functional and technical components. The framework instantiation happens using the tools provided by the computational platform and with the integration of external applications, extending the system to include those applications capabilities.

A first example of instantiation is defining and scheduling platform games for a specific patient. A practical example of the framework integration is presented, using the case study of a clinic, with therapists and patients. A therapist following a certain patient creates a game for execution of a set of exercises using the platform tools. After configuration, the game execution is scheduled in the patient timeline. The patient is notified of the scheduled game in real time, if logged in, or when he accesses the system. The patient performs the scheduled game and game execution related data is sent to the server. Data resulting from game executions and the patient progression is then analyzed by the therapist using the platform interface.

4.1. Architecture

A computational platform was implemented during this research to support the proposed model. Development setup was based on an ASUS ROG STRIX G G531GT with an Intel Core i7-9750H CPU at 2.6 GHz, and 16 GB of RAM, running 64-bit Windows 10 Pro. The IDE used was Visual Studio Code and the DBMS used was MySQL 8.0.21. Production hardware setup is based on a Linux CentOS 7 server. The deployment solution includes Nginx as a reverse-proxy for several Docker containers. The platform is available online, for testing purposes, at a restricted URL, as the server is currently serving multiple projects' applications. The system is based on a series of components with distinct roles and responsibilities.

The architecture is based on a web-based server providing a frontend and web service APIs for external applications and IoT devices integration. The server uses a DBMS for data persistence. The system architecture can be visualized in Figure 15.

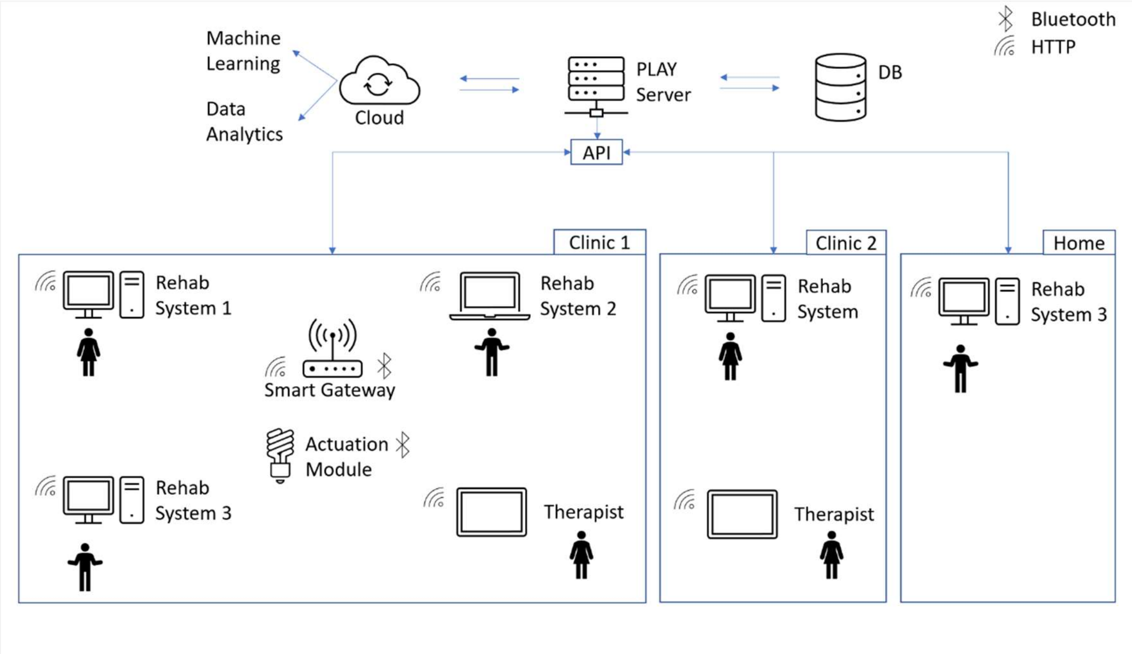

Figure 15 - System architecture

The platform architecture consists of a cloud-based server with 3 main roles, assuring the backend functionalities, a distinct frontend for each platform actor and providing a set of web services. The client applications can be web browsers, third-party applications, or devices. Users access the platform through a web browser or using a third-party application. Third-party applications integrate with the platform by consuming the provided API. Each client establishes a web socket connection with the server, providing a tool for real time bidirectional communication.

The server provides a set of web services for platform component applications interaction. The server was implemented using Node.JS technology and the Express framework. WebSocket's technology is used to establish connections with each client allowing real-time bidirectional communication. The DBMS used is MySQL.

The platform can be accessed using a web browser through the server frontend. Other applications can communicate with the server using the provided API. The API is the platform common interface for data access and management. Games consume the Game API services provided by the platform server. The server provides a two distinct APIs as the original JPK API has been integrated. The JPK Original API has 4 main routes: admin, physio, patient, and user. Each route provides service access points for specific user roles.

4.2. Backend

The central platform unit is the server. The platform server consists in a series of components providing backend support for the platform functionality, management logic, and data storage. A set of web services provide an API for games and devices interaction. The server also provides a frontend website with distinct user interface for each actor role (Administrator, Clinic, Therapist, Patient). All actors are users in the system. Clinics have therapists and patients. Therapists create, configure and schedule games for patients. The administrator manages and has complete access to all system components. The information displayed is visualized according to the user privilege.

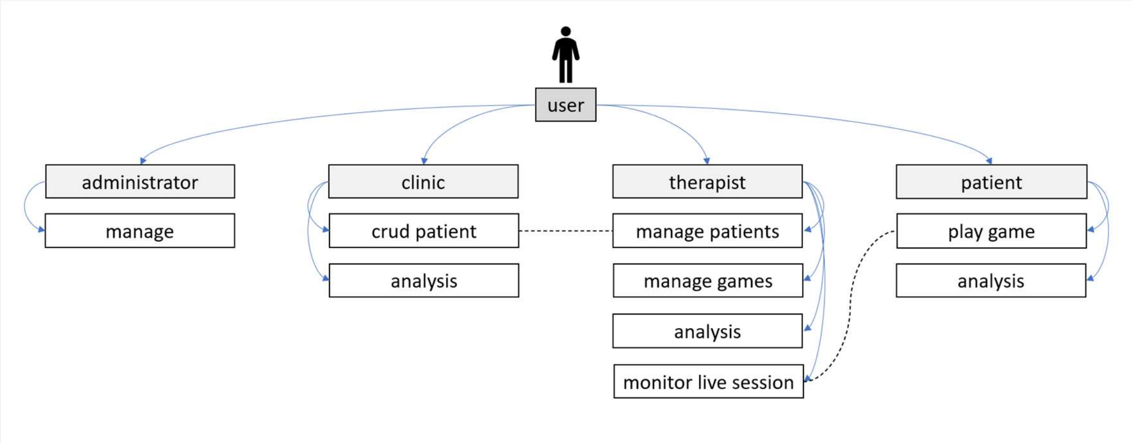

Figure 16 - Platform user roles

The Administrator can manage users, games, devices, and rewards as well as platform languages and global UI settings, among other systems components. The Clinic manages therapists and patients. When a clinic creates a new patient, this patient becomes available in the clinic therapist's patient list. Therapists can manage patients, access their profile and progression records and schedule exercises. Therapists can design exercises/games specifically suited for a specific patient. Patients can perform exercises, both scheduled and autonomous, and access their profile and the obtained rewards regarding game and platform usage.

Table 23 - Platform functionality by user role

Functionality	Administrator	Clinic	Therapist	Patient
View User	●	●	●	
Add User	●	●		
Remove User	●	●		
Notify User	●	●	●	●
Follow/Unfollow User			●	
View Game	●	●	●	●
Create Game	●		●	
Schedule Game	●		●	
Play Game				●

View Badge	●		●	●
Manage Badge	●		●	
Manage Language	●			
Manage Translatable	●			
Manage Translation	●			
Change Language	●	●	●	●

4.2.1. *Server*

The server provides a set of web services for platform component applications interaction. The server was implemented using Node.JS technology and the Express framework. WebSocket's technology is used to establish connections with each client allowing real-time bidirectional communication. The DBMS used is MySQL.

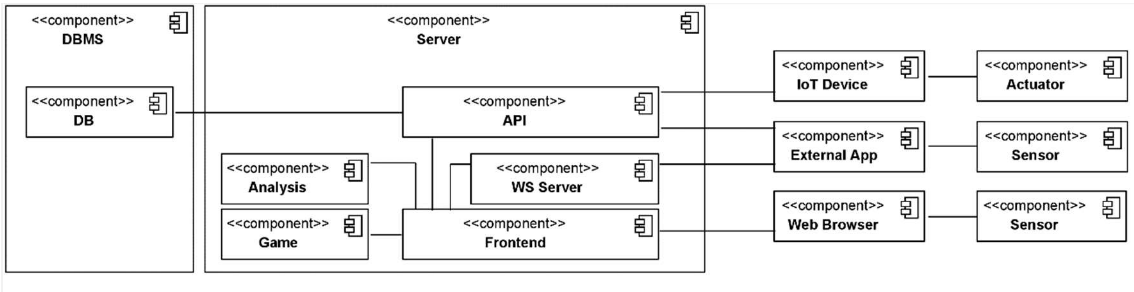

Figure 17 - Server architecture

A set of main components provide the needed functionalities. The platform can be accessed using a web browser through the server frontend. Other applications can communicate with the server using the provided API. The API is the platform common interface for data access and management. Games consume the Game API services provided by the platform server. The server provides a two distinct APIs as the original JPK API has been integrated. The original server supporting the JPK platform had some issues and wasn't always available. That led to the need to implement a module that replaced that server for this research. The implemented module provides support for the JPK REST API calls for all JPK components and a new set of web services for added functionality. The JPK Original API has 4 main routes: admin, physio, patient, and user. Each route provides service access points for specific user roles.

4.2.2. *Data Model*

A data model for the platform requirements has been designed and a database implemented. The database maintains platform content and the recording of platform usage, game results and samples. Interaction with data is provided by using the platform API. The data model is displayed in Figure 18.

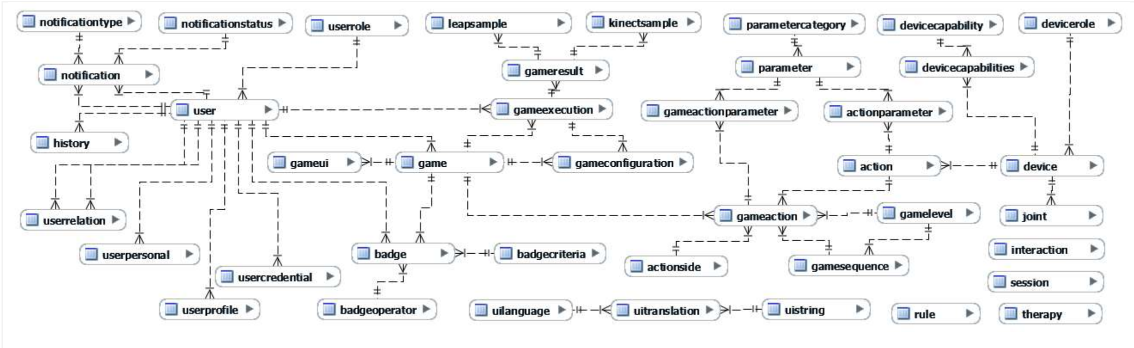

Figure 18 - Data Model

4.2.3. *Design*

The platform system is composed by a set of main classes and modules. These classes model the required entities and functionality needed for system operation. Modules encapsulate distinct functionalities in the system. The system class diagram can be visualized in Figure 19.

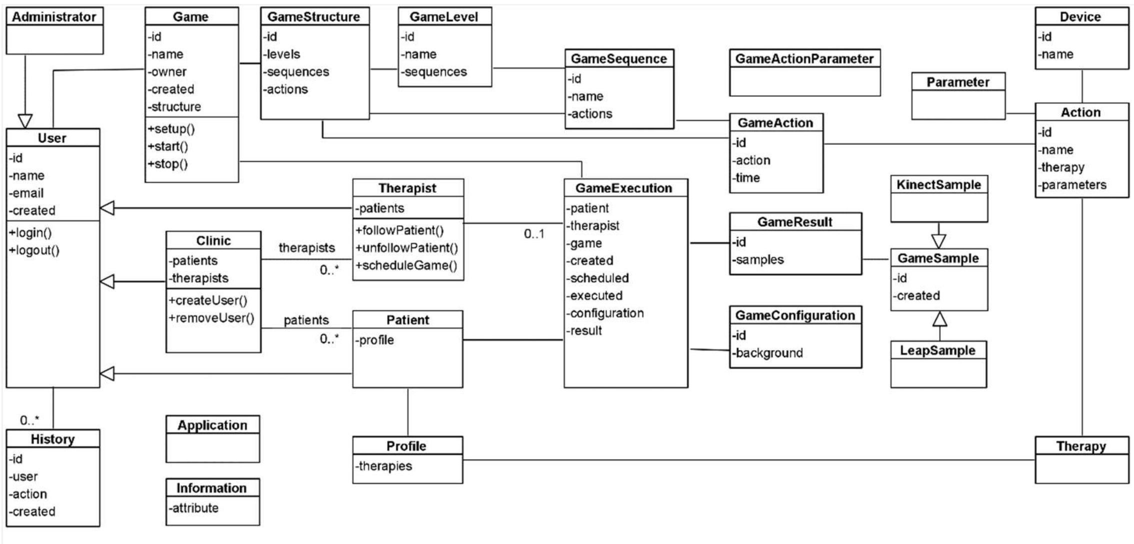

Figure 19 - Class Diagram

The User class models the system actor roles: Administrator, Clinic, Therapist and Patient. This class implements common user methods like login and logout. There are also 4 specialization classes for Administrator, Clinic, Therapist and Patient.

The Game class objects model games and have an id and name for reference. When created, the game stores the creation date and creator id as owner. The game owner is the only one allowed to make changes to the game structure and parameters.

The Badge class is the core of the reward system. The reward system provides a way of valuing the platform usage by a specific user and recognizing it through an award. Each badge has an attribution criterion, based on user platform usage or game metric variables, to automate badge generation. Badges can have a visual representation uploaded.

External applications are modeled in the Application class. Each application has an id and name. External applications can take advantage of the provided API to integrate the system. A client web socket connection must also be established to the platform server.

The platform uses sensors and actuators to gather data and expand the interaction with the real world. The Device class models the sensor and actuator devices. Each device has an id and name.

The notification and messaging system allows for message interchange within the platform. Messages can belong to 3 distinct categories: Message, Notification, and System.

The system can be available in several languages. Languages can be added by the administrator. The localization system works by using a distinct identification for each localizable string. The identifications used are then also added to the database allowing reference when adding translations for the languages created. Translations created for the selected language are then used in the elements with the referred identification. This system allows for real-time update of the localizable user interface strings.

4.2.4. *API*

The API is available as a web service and is the platform access point to data management, provided functionality and interface for external applications. The API uses JSON as data format and follows a RESTful approach. The API provides registration and authentication services for the applications and users. Game services are provided for exchanging game information as structure and actions, results, and samples. Device services allow actuation devices registration and authentication. Interaction services provide device management and control.

External games consume the API provided by the PLAY platform for gaming services. The process flow is displayed in Figure 20. The API provides services for user authentication. After an authentication request from an external game, the server returns a token for validating subsequent communication. The user profile and scheduled games can be retrieved.

Before the patient initiates playing the game, the game structure and configuration parameters are requested from the server. The JPK application implements the necessary game logic for game execution.

During the playing of the game, the game flow variables, such as time and score, are sent to the server for real time monitoring. Any game parameter updates, or user notifications can be sent to the game in real time. Any action in the game structure that has not been played yet can have its parameters updated and they will be sent to the server for real time update of the game action. This feature allows therapists to correct and adjust game parameters if they detect that need during the game. Sometimes increasing the remaining time for completing a required action is enough to prevent patient frustration. The game results and samples are sent to the server after game completion

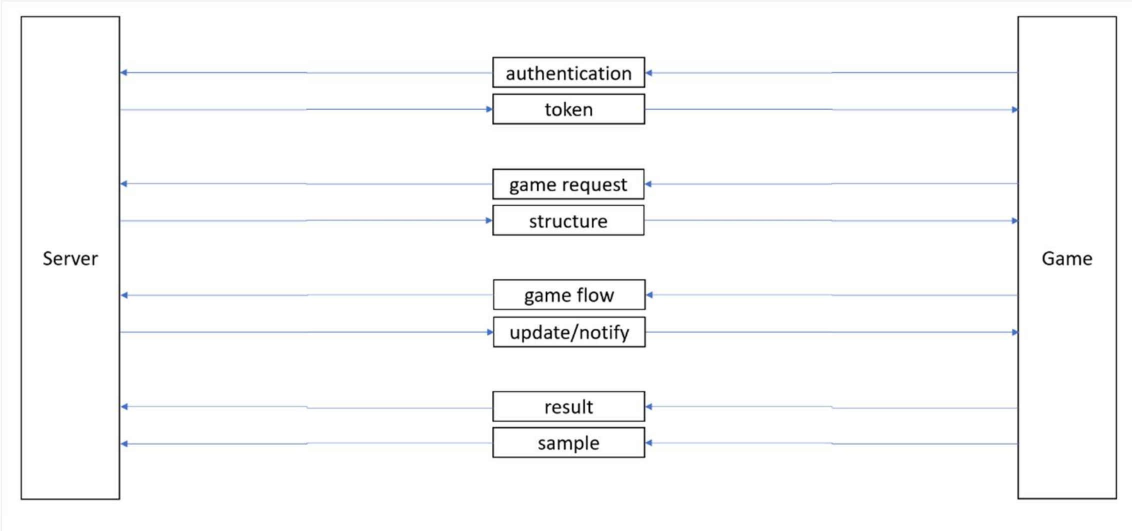

Figure 20 - API interactions

The API also provides actuation services for communication and control of external IoT devices. Devices must register with the PLAY server and use the provided token for authentication of subsequent communication. Games can use the API to invoke actuation requirements from the PLAY server. The server routes the actuation requests to available actuation devices.

The PLAY API is detailed in Appendix III.

4.3. Frontend

The frontend provides distinct functionalities for each user role in the system. The frontend is available as a responsive web application with a similar UI for all user roles. The Administrator frontend provides an overview dashboard, users, games, rewards, devices, languages, and translations management. The Administrator dashboard displays the user count and game count, a platform usage chart, that can be viewed as list, and a chart of user counts by category. The Clinic user frontend provides an overview dashboard and provides users management. The Therapist user frontend provides an overview dashboard, and patient, games, and rewards management. The Patient user frontend provides an overview dashboard, the patient schedule, and games and rewards views.

The user interface (UI) designed for the frontend layout uses a consistent interface with similar elements and procedures for all users but offering distinct functionality for each user role. Bootstrap and jQuery are among the technologies used. The basic layout consists of a collapsible sidebar with the platform logo and navigation buttons for the platform specific areas. A top bar displays, on the left, the icon and title for the current area location, and, on the right, the notification and message counts and user session tool buttons. The sidebar is the main navigational control and provides access to the main platform areas. The sidebar navigation depends on the user role and can be visualized in Table 24.

Table 24 - Sidebar navigation

Platform Area	Administrator	Clinic	Therapist	Patient
Dashboard	●	●	●	●
Users	●	●	●	
Now	●		●	
Games	●		●	●
Actions	●			
Parameters	●			
Badges	●		●	●
Devices	●			
Languages	●			
Database	●			
Statistics	●	●	●	
Schedule				●

Popup dialogs and tooltips are used. List items are clicked for selection and double-clicked for edition. Icons are used for most UI button (e.g., the minus and plus signs in the interface are buttons to create or remove the current context relevant entity).

Home

The main access page shows the platform logo and provides login, and registration buttons. Users can register as Clinic, Therapist or Patient. After registration, users can initiate sessions using the registration credentials. The homepage and login views are displayed in Figure 21.

Figure 21 - Homepage (left) and Login (right) views

All users share some platform functionality such as a personal profile, messages, history, settings, and logout. These options are accessible through the fixed top right toolbar button, displaying the username, and allow navigation to each area. Next to it are the alerts and notification counters. Clicking the counters displays a popup with unread notifications. The left side of the top toolbar displays the current area title and icon. Each area has a distinct icon.

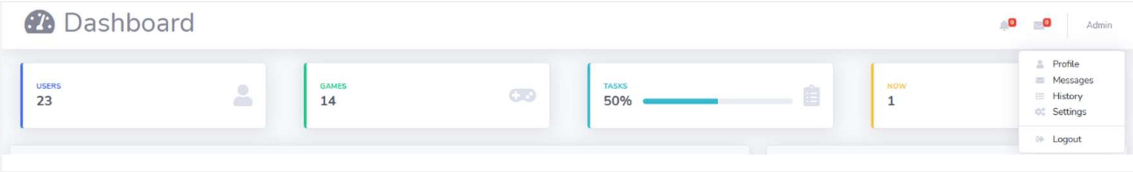

Figure 22 - Current user personal menu

Dashboard

Every user category shares a dashboard view displaying relevant information and quick-access buttons. The Administrator and Therapist dashboards displays the 4 top cards with the user and game counts, tasks progression and current live sessions count. The top cards can be clicked leading to each specific area. Bottom charts are displayed on the Administrator and Therapist dashboards. The Administrators' dashboard has a line chart displaying the combined platform usage of all users through time, and a user count by category pie chart.

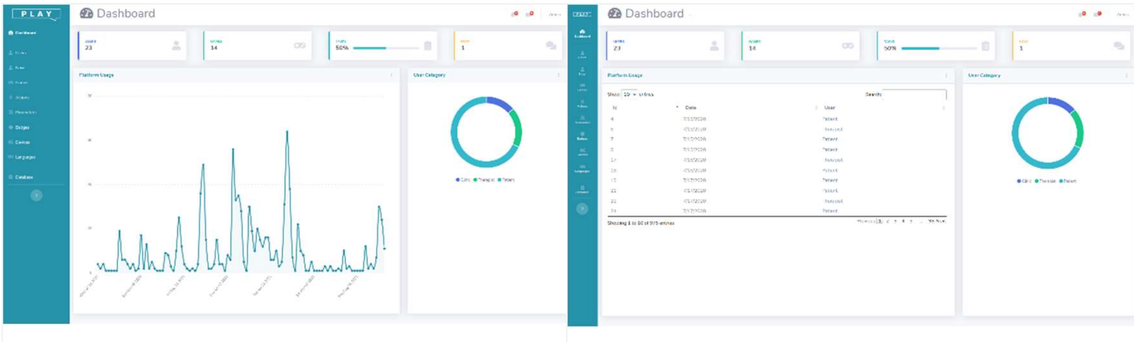

Figure 23 - Administrator dashboard with chart view (left) and list view (right)

The Therapists' dashboard displays a full width bottom line chart, of platform usage, with individual lines for each patient. Charts can be visualized in a list format, by using the relevant option, available in each chart top right button.

4.3.1. *Administrator*

The Administrator has distinct responsibilities, and the platform provides the needed tools. Actions, parameters, devices, and languages management are done by the administrator. These vital elements allow greater platform management flexibility.

Actions

Actions and parameters are managed by the Administrator, in the Actions view. Parameters can be managed in the Parameters tab. The Actions view is displayed in Figure 24.

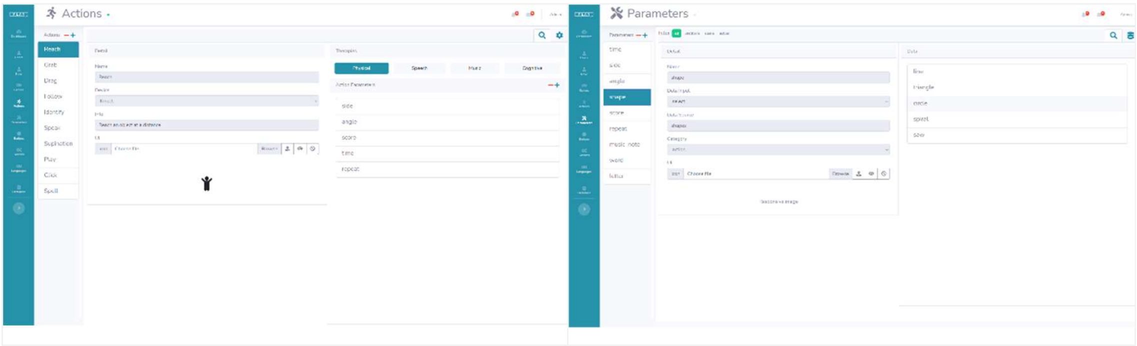

Figure 24 - Actions (left) and Parameters (right)

Actions and parameters have a configurable visual representation in the platform by displaying the name and an associated icon. Currently stock icons have been used but customized icons will be designed and integrated, for a better perception of the provided functionalities.

Parameters

Parameters are the building blocks of actions. A parameter can be shared among actions. Parameters represent common properties and patterns in action construction as well as specific one-time usage properties of actions. A parameter like time or repeat can be used in almost every action, and a parameter like angle only has meaning for reaching actions.

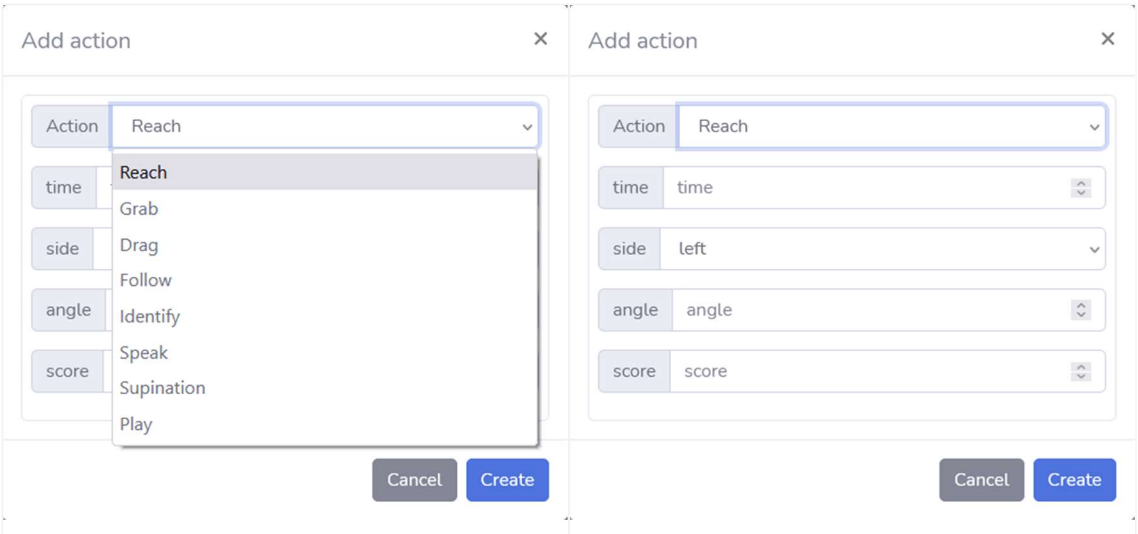

Figure 25 – Add action (left) and Reach action (right)

Parameter’s data source property allows definition of data collections as input. For instance, a follow action uses geometric shapes to provide guidance that users must try to replicate. A data repository of common shapes is provided, and therapists can choose the relevant shape for inclusion as parameter setting of the game action. In a similar way, the speak action uses a word repository as source for the word parameter. When adding a speak action to the game structure, therapists can select the relevant word from the word repository. When the game is executed, the vocalization of that specific word will be requested during the game.

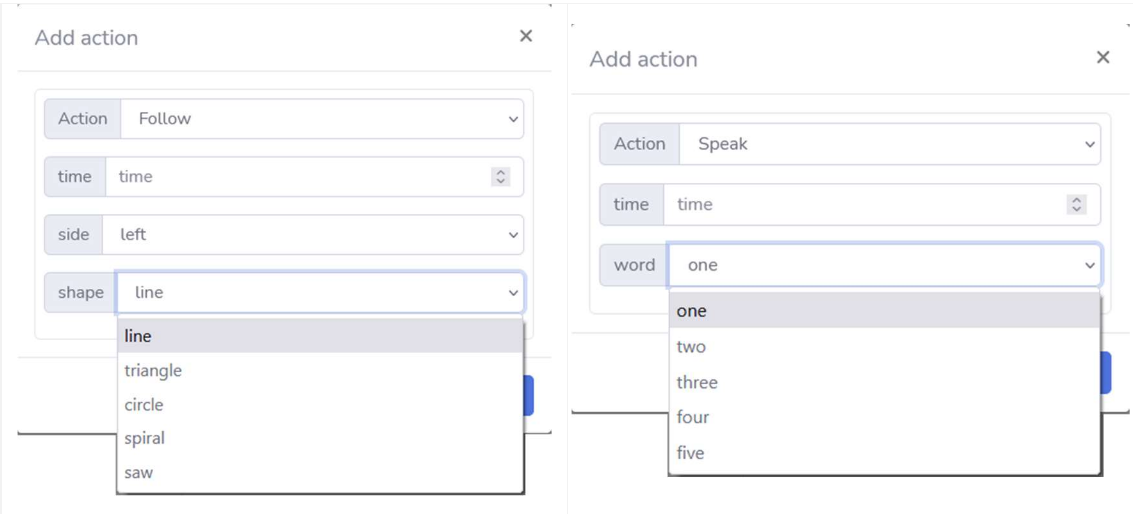

Figure 26 – Follow action (left) and Speak action (right)

Languages

The Language view allows the Administrator to manage the available languages, language strings and translations for the platform UI. Each localizable item in the platform has an id and can be referenced as a language string. Each language can have a translation for each language string. Missing translations revert to the default language. All users can change the platform interface language through the settings menu. The selected language will be active, for that user, after login. The Languages interface can be viewed in Figure 27.

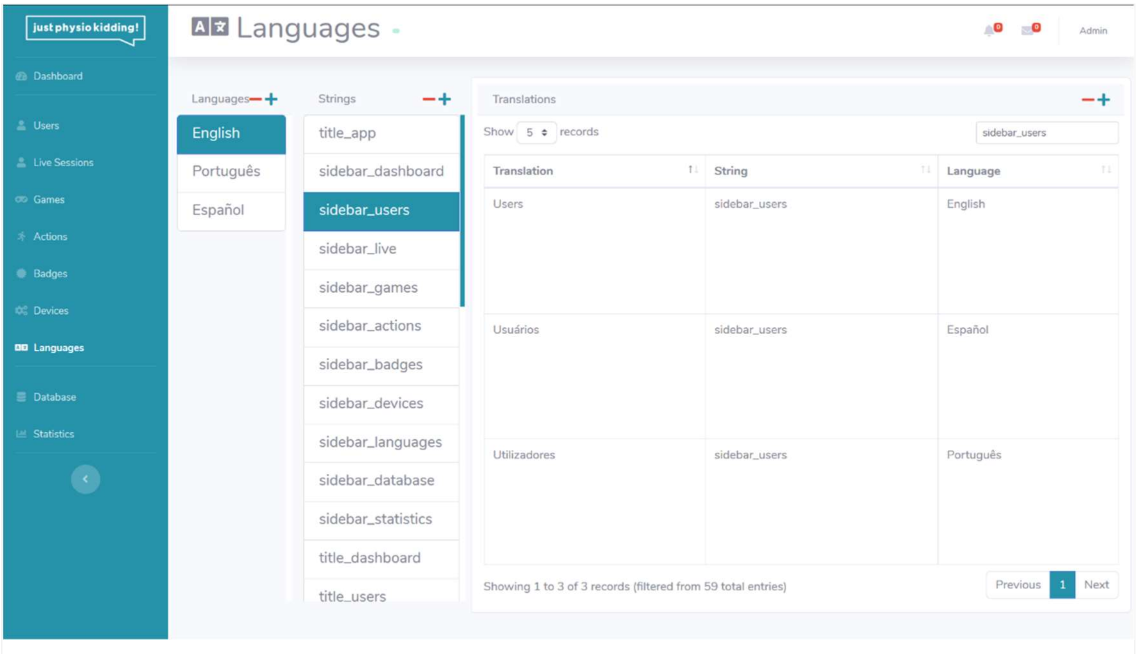

Figure 27 - Languages, language strings and translations

The Database area allows database inspection providing a record count for each table.

4.3.2. *Clinic*

Clinics can manage therapists and patients. The Clinic user has an overview of platform usage by its therapists and patients.

Users

The user view provides a list of the users of the platform. The Administrator and Clinics can create users. Users can be added and removed using the top header minus and plus signs. Double-clicking the username opens a popup dialog, allowing the editing and update of user data. The top row of the user profile provides 4 clickable buttons to toggle the status of the patient therapeutic needs (physical, speech, music, or cognitive therapies). The user profile also shows a set of metrics (clinical, emotional, and motivational status), displaying the current patient status and allowing update of each metric, and a progression line chart for the select metrics. All lines can be visible simultaneously. Each line can be made visible or hidden, and the chart updates to show the selected metrics progression. The profile view is displayed in Figure 28.

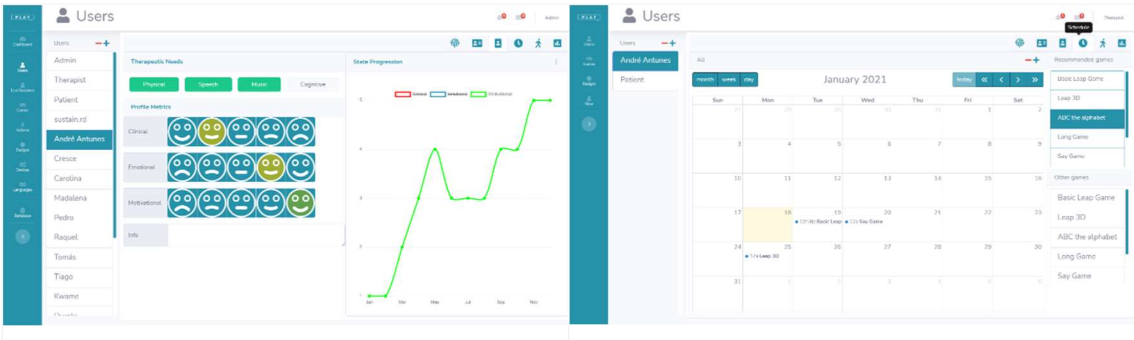

Figure 28 - User profile (left) and schedule (right)

Each patient has a schedule. Exercises can be scheduled by therapists using the platform tools. When an exercise is scheduled, is inserted in the patient schedule, and can be visualized on this page. Games can be scheduled by dragging from the Recommended or Other games' lists. Scheduled games appear in the patient schedule for posterior execution. The user schedule displays the current exercise schedule for each patient. After showing up in the patients' schedule, the exercise can be selected for execution. The schedule calendar can be consulted at multiple levels providing month, week, and day views and contextual forward and back navigation. The schedule view can be visualized in Figure 28. The progression view displays the user platform usage. Data can be visualized as a line chart or a table list of game executions, by selecting the charts' relevant top-right menu option. User results can be consulted individually.

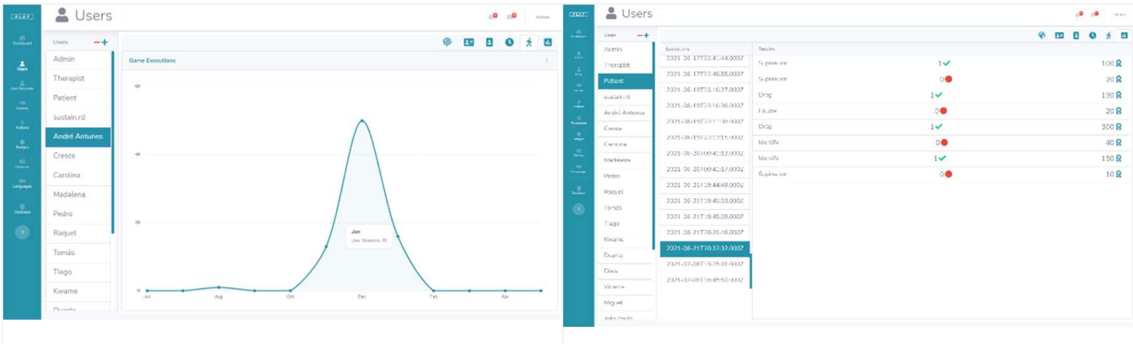

Figure 29 - User progression (left) and results (right)

4.3.3. *Therapist*

Therapists can manage patients, games, and badges. The dashboard displays the information relating followed patients and owned games and badges.

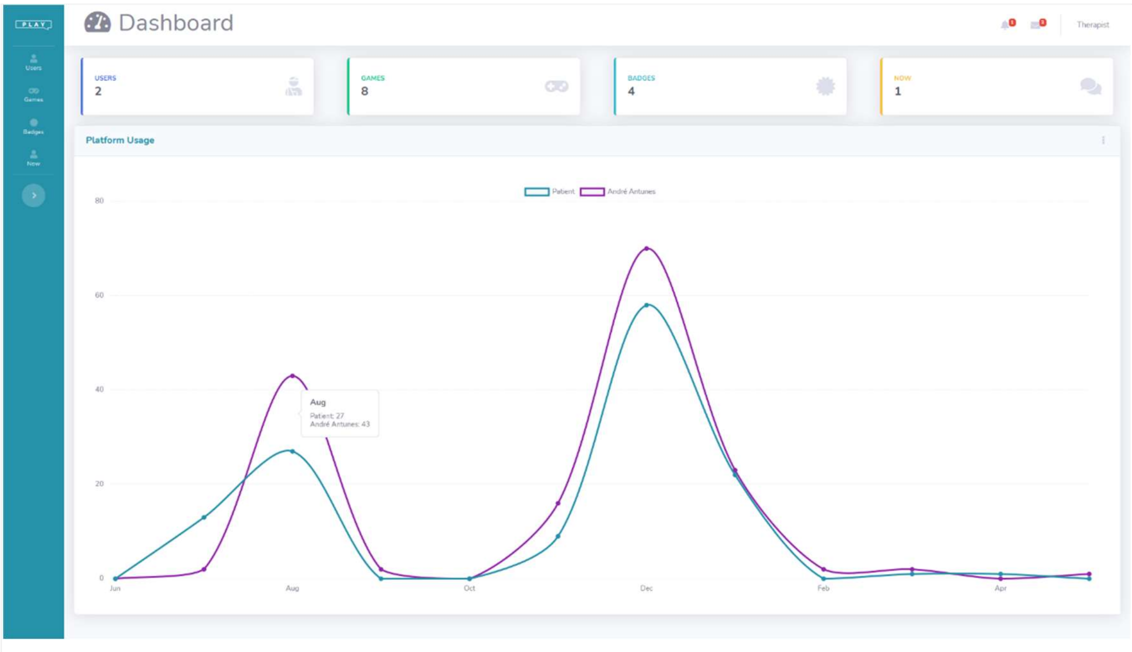

Figure 30 - Therapist dashboard

Games

The Games view displays a list of available games. Games can be added and removed using the top header minus and plus signs.

The top toolbar provides access to several game-related tabs: Detail, Structure, Configuration, Theme and Plays. The Detail tab provides an overview of the game. Structure, Configuration and Theme allow for setup of game variables, while the Plays tab provides a relation of game executions.

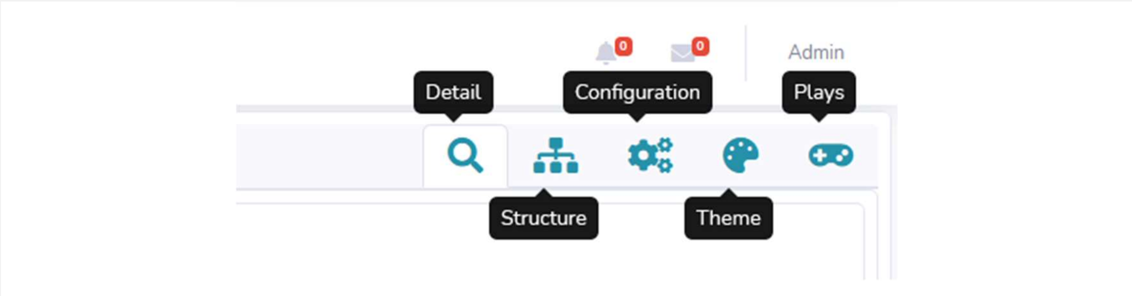

Figure 31 - Games top toolbar

The game name and info can be edited by double-clicking the game name. The game logo image is displayed, if defined. The game structure is displayed in a tree format. Each tree node can be hovered, displaying a tooltip for expanded information.

The game structure can be configured by adding levels, sequences, and actions, as can be visualized in Figure 33. Levels, sequences, and actions can be added and removed using their respective top header minus and plus signs and edited by double-clicking each item.

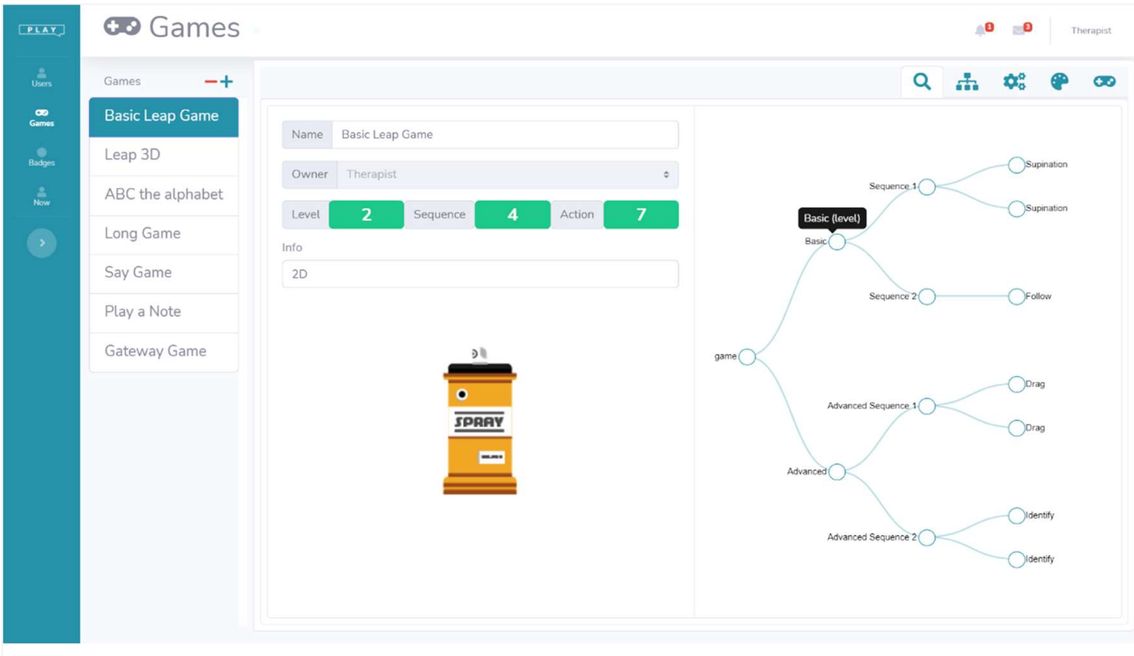

Figure 32 - Game details

The game theme elements, for the platform representation of the game, can be setup using the interface, as displayed in Figure 33. The background color for backdrops can be selected and setup. The game logo and splash screen images can be uploaded and previewed. These images are used for the game representation in the platform, before starting.

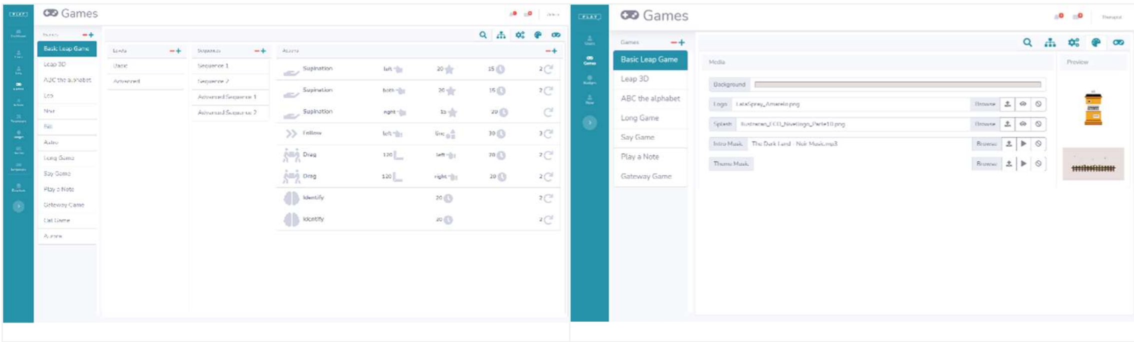

Figure 33 - Game structure (left) and theme (right)

Rewards

The reward system allows the creation of rewards to be provided to patients based on a previously setup criteria. Criteria used for reward setup include, currently, the game usage, time, and score. A set of comparison operators together with a reference value allow to define the conditions for attribution of a specific reward.

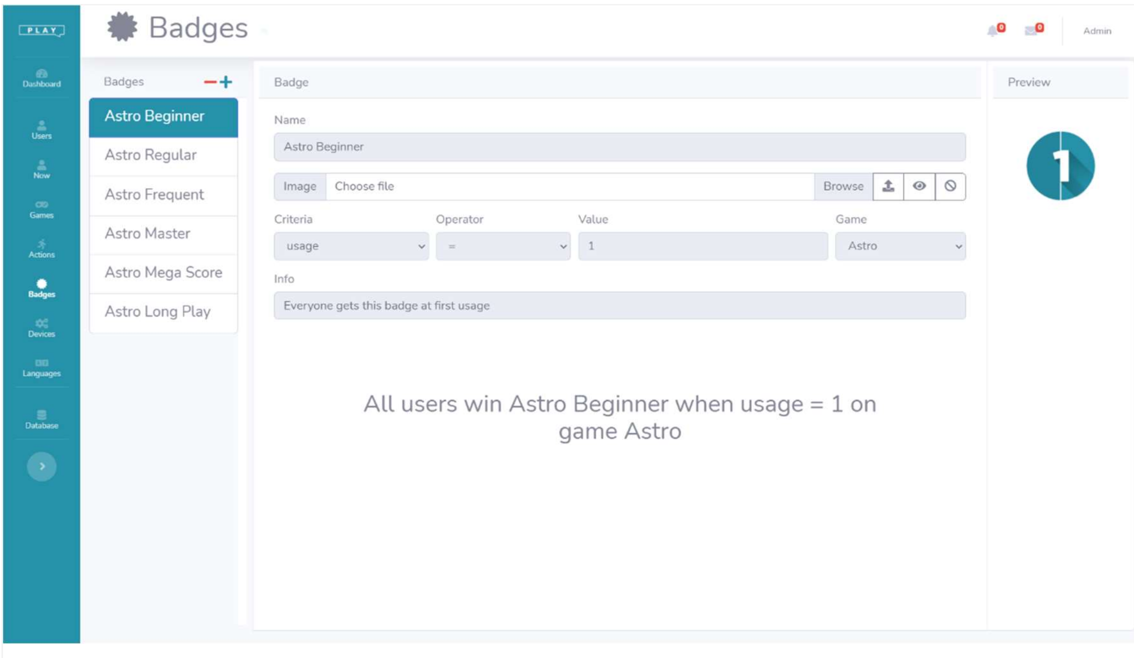

Figure 34 - Rewards

4.3.4. *Patient*

Patients can play games. Games can be played autonomously or scheduled by the therapist. The schedule can be consulted as well as the earned badges and rewards.

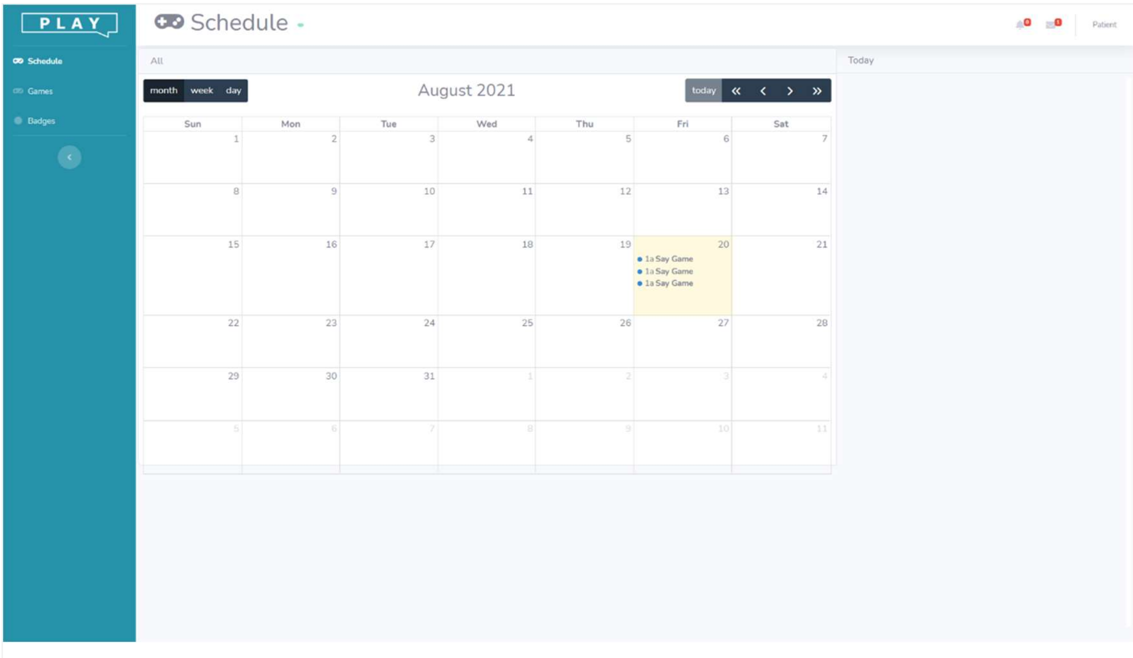

Figure 35 - Patient schedule

4.4. Platform games

The platform provides a game builder interface for therapists. Patients have client access to the generated games. Platform games and games created and configured by the therapist are displayed and selectable in the patient games menu. Patients can access these games using the provided interface, as displayed in Figure 36. Selecting a game activates full screen mode for game execution.

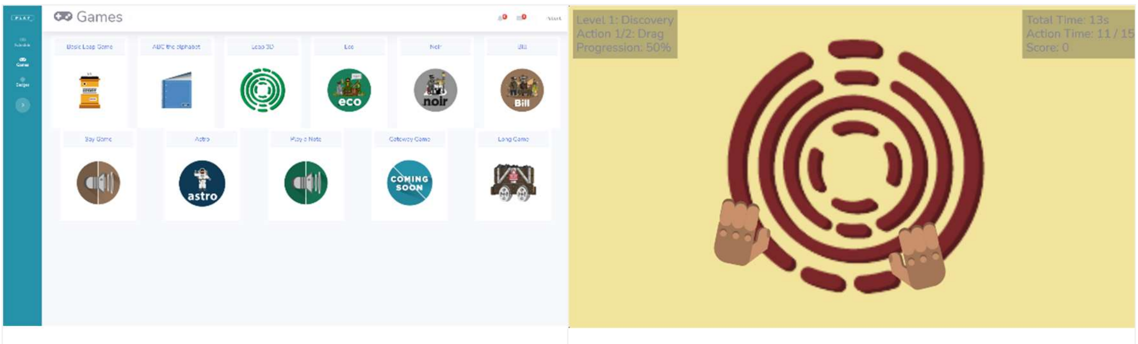

Figure 36 - Games menu (left) and game execution (right)

4.4.1. *Modules*

The games client available in the platform follows a modular approach, is composed of several modules, and has the possibility of adding extra modules for expanded functionality. An initial analysis allowed the design and implementation of the platform games main modules.

Table 25 - Platform game modules

Module	Description
3d	The 3D module encapsulates the methods for 3D interaction
api	The API module encapsulates communication with the platform server
app	The App module is the main app controller
audio	The Audio module manages audio subsystem for music and sounds
basic	The Basic module has setup and startup responsibilities
game	The Game module encapsulates the game logic
geo	The Geo module is used for geolocalization
sensor	The sensor module encapsulates sensor specific logic
speech	The Speech module manages speech and uses the Speech API
storage	The Storage module manages local storage interaction
ui	The UI module manages user interface components
websocket	The WebSocket module manages Web Sockets communication with the server
webworker	The WebWorker module encapsulates web workers logic

4.4.2. *Game mechanics*

The game is played in a 2D environment. The game mechanics requires users to make a supination action to achieve success and progress to the next action. The action must be performed during a limited time. If no supination action is performed until reaching the time limit, the resulting score is 0 and the game progresses to the next action.

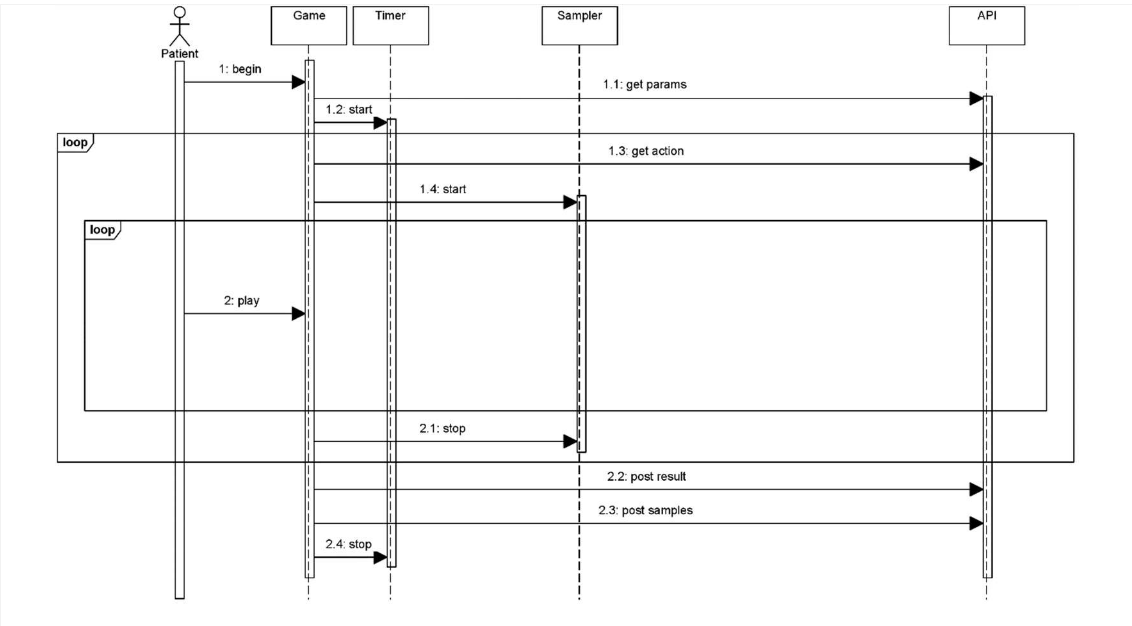

Figure 37 - Game loop sequence diagram

Gameplay modes

To make the game more dynamic and motivating different gameplay modes will be available.

Table 26 - Gameplay modes

Name	Sensor	Description
Fingers	Leap Motion	Finger stretching, joining, spreading
Hands	Leap Motion	Supination, pronation, grab, release
Shapes	Leap Motion	Follow shape path. Line, Circle, Triangle, Saw, Spiral
Pipes	Leap Motion	navigate inside delimited path
Speech	Mic	say or repeat a specific word or sentence
Reach	Kinect	reach a specific location or object
Walk	Kinect	walk to a specific location

Levels are used to promote a sense of achievement and progression. Each game can have multiple levels, defined in each game structure. To promote user engagement and effort recognition, every relevant move in the game should be rewarded in some manner. The simplest form of reward is increasing the score. Badges are used to acknowledge patient progression using the platform or achieving a milestone in the game. Platform games can be monitored in real time by therapists and game parameters updated during patient game play, if needed.

4.4.3. *Instantiation*

A platform game implementation, using the Leap Motion sensor, was developed. The game is available through the Games menu of the Patient frontend. The application consumes the API services for integration with the platform. The game detects if the Leap Motion is available in the user system and notifies the users of the missing requirement if not detected.

Case study 1

A certain clinic provides physiotherapy services for children with special needs. The clinic patient population is composed mostly of cerebral palsy patients. The clinic promotes the use of serious games as a complement to regular therapy protocols trying to engage and motivate patients to perform the needed therapies. Following this principle, the clinic registers a user account in the PLAY platform. After registration, the clinic user can access the platform functionalities and manage users and games. The clinic user can create user accounts for the therapists and patients that will use the platform. As patients and therapists can create accounts on their own, if any of the users are already registered in the platform, their accounts can be associated with the clinic.

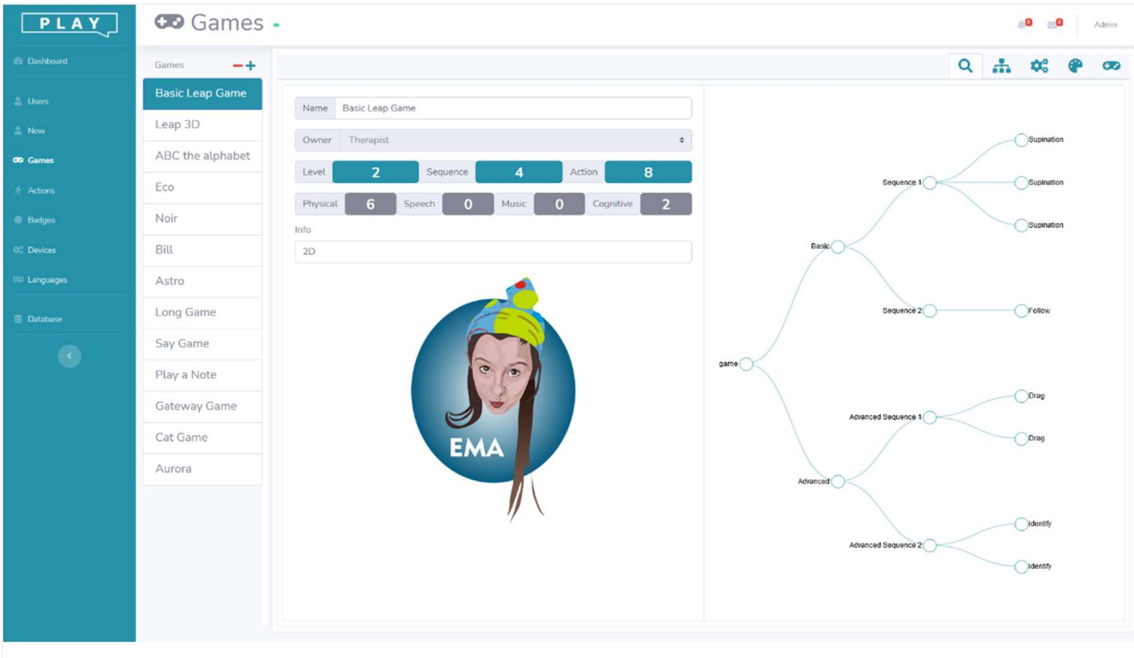

Figure 38 - Personalized platform game (example 1)

Therapists in the clinic use the platform for promoting motivation and engagement among patients. For this effect, therapists have a list of patients, where they can associate patient accounts to be managed and monitored, using the platform tools. A therapist from the clinic adds a patient, with a cerebral palsy condition, and with need of physical rehabilitation, to the patients list. The monitored patient has a profile in the platform that should be updated. The therapist manages the patient profile, activating the Physical Therapy setting. After setting the needed therapies, an initial assessment of the patient therapeutic severity is made, using a 5-stage scale, and set in the platform. The assessment is based on the therapist perception and evaluation. This patient presents a mild condition, so the therapist sets a value of 2, with 5 being the worst possible. The emotional and motivational states can also be set, using a similar scale but with 1 being the worst value. This patient is not very motivated, and a bit emotionally instable, so the therapist sets 2 for the motivational state and 2 for the emotional state. This initial setup allows the platform to gather some initial information about the patient.

The therapist creates a game for execution of a set of exercises using the platform tools. Actions available for selection are filtered, based on the patient profile settings, and only relevant actions for that patient needs are displayed. Relevant actions for physical therapy are, for instance, touch, reach, drag or follow movements.

The created game remains available in the platform. The game can then be configured for execution, and after configuration, the game execution can be scheduled in the patient timeline. The notification system in the platform is used for notifying the patient of a new scheduled game. The patient is notified of the scheduled game in real time, if logged in, or when he accesses the system. The patient performs the scheduled game, executing each action available in the game structure. Game execution related data is sent to the server. After game completion, results and sensor samples are also sent to the server. Data resulting from game executions and the patient progression is then analyzed by the therapist using the platform interface.

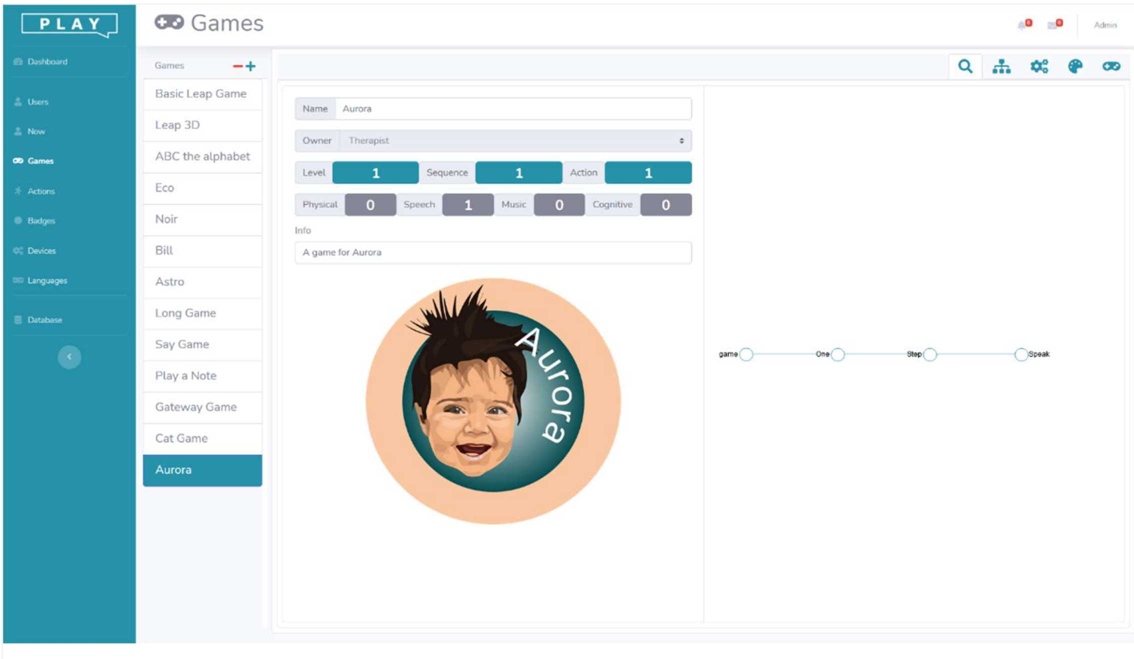

Figure 39 - Personalized platform game (example 2)

Case study 2

Patient A is 4 years old, very cooperative and has no cognitive impairments. He has a mild condition on his right arm that requires intervention. This patient requires exercises that promote grabbing actions for the right arm, on the right side, but also on the top and bottom areas of the left side. The patient is already registered and in the therapist patients list.

The therapist can access the patient profile to verify and update the current state of the profile components. The profile contains the identification and age of the patient. The profile setting for physical therapy is activated indicating that the patient requires actions involving exercises for motor rehabilitation. The therapist creates a game structure for the exercises required by Patient A.

4.5. Web Services

The API provides tools for external applications and devices to integrate with PLAY. External applications can be, for example, game engines interpreting and applying the game structure and configuration or game editors for designing game structures.

External devices expand the platform scope and the user experience. Devices can be sensors or actuators for capturing that or expand the interaction.

4.5.1. *External applications*

External applications can integrate the system by consuming the provided web services.

These can be games, game editors, DM, or ML applications or IoT devices. When implementing games, the applications receive the game structure, defined by a set of actions. The actions define the type of action to implement and the expected behavior. The actual game implementation is specific to each application but must follow the proposed structure of actions. After game execution, the application sends results and samples to the server.

A C# implementation of a new game using the Kinect sensor was designed and integrated in the JPK platform. The game theme is Space. Astro (short for Astronaut) is the protagonist and must travel through space collecting space related items. The items to be collected are positioned at positions promoting therapeutic exercise movements. The needed JPK modules were developed and included in the JPK platform. The resulting integration of the two platforms is detailed in Chapter 5.

4.5.2. *Actuation devices*

External devices can integrate the system by consuming the provided web services. These IoT devices use the HTTP protocol for communication with the platform server. Bluetooth is used for low range communication between applications and actuation modules.

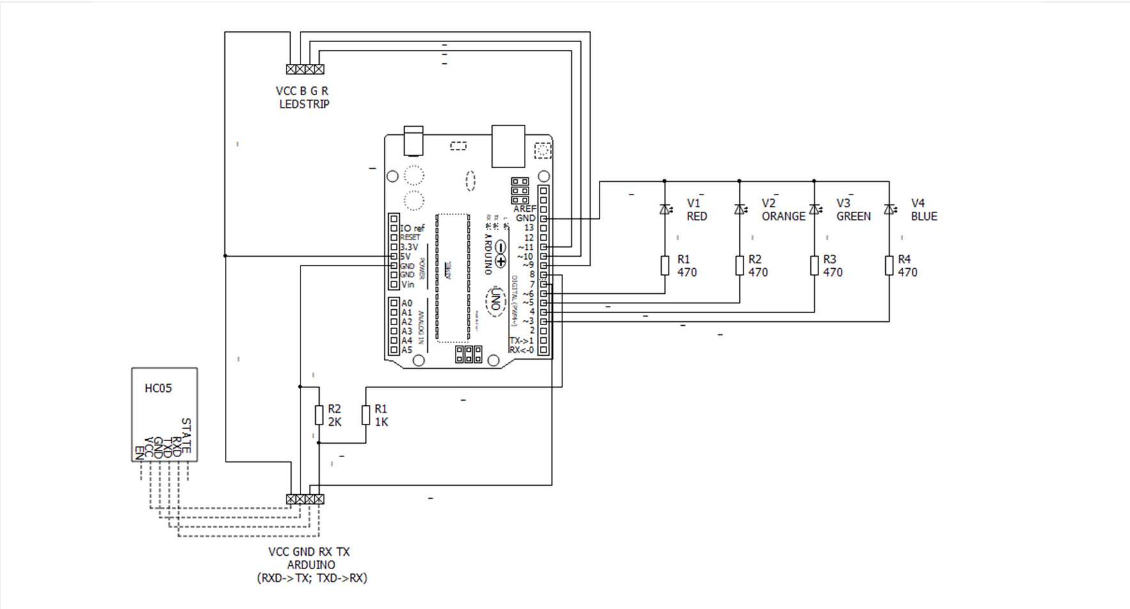

Figure 40 - Actuation module schematic

A prototype for an actuation module has been designed and developed for testing the actuation interactions. This prototype is an update to the actuation module designed for the JPK platform. The hardware prototype for an actuation module was implemented using an Arduino One Board with an HC-05 Bluetooth communications module and some supporting electronics. The hardware setup was expanded to include an ESP-32 board, and use its WI-FI connection, for direct server HTTP communication. Internal module software was programmed using C++ and has communication, data processing and actuator control logic. The device is powered using a mainstream 220V to 12 V transformer. The actuators included are provided as a proof of concept and very simple. This actuation module has preset actuation capabilities, four colored leds (red, green, yellow, and blue) and a tricolor led strip allowing independent global control of the elements of each color.

The smart device understands the server protocol and manages the execution of commands to actuators. Internal logic provides the execution of smooth fading transitions and easing variations for analog and digital outputs, controlling the connected actuators. Available commands for actuators can be visualized in Table 27.

Table 27 - Actuator commands

Command	Description
analog	Discrete analog value [0 - 255]
digital	Discrete digital value [0 or 255]
fadein-reverse	Reversed fadein (255 to 0)
fadeout-reverse	Reversed fadeout (0 to 255)
fadein	Fade in (0 to 255)
fadeout	Fade out (255 to 0)
hello	Hello sequence
all-off	Turn all actuators off
led-off	Turn all leds off
strip-off	Turn all ledstrip off
test	Test sequence
strip-fadein-mask	Masked ledstrip fadein
strip-fadeout-mask	Masked ledstrip fadeout
led-fadein-mask	Masked all leds fadein
led-fadeout-mask	Masked all leds fadeout
led-on-mask	Masked turn all leds on
led-off-mask	Masked turn all leds off

The designed system and prototype were successfully demonstrated with the integration of API calls in the JPK platform. The JPK system is a rehabilitation platform using serious games and NUI sensors. If the gateway is available, the platform establishes a session and uses the gateway services for enhancing exercises and games and the overall user experience.

The JPK patient application uses the leds and led strip available as actuators during the exercises with colored leds indicating the status of the performed moves, with the green led being activated when the movement is successfully achieved, and the yellow and red leds being used for partial and unsuccessful completions. The led strip is used for creating blue-based ambient lighting effects during UI navigation (blue is the main JPK UI color) and specific lighting effects related to the game theme colors after game theme selection. Audio and speech resources are also used during game execution for reinforcement and motivation.

The solution worked very well for the created smart environment. Applications register and establish sessions with the gateway and extend their interaction using the actuation modules provided.

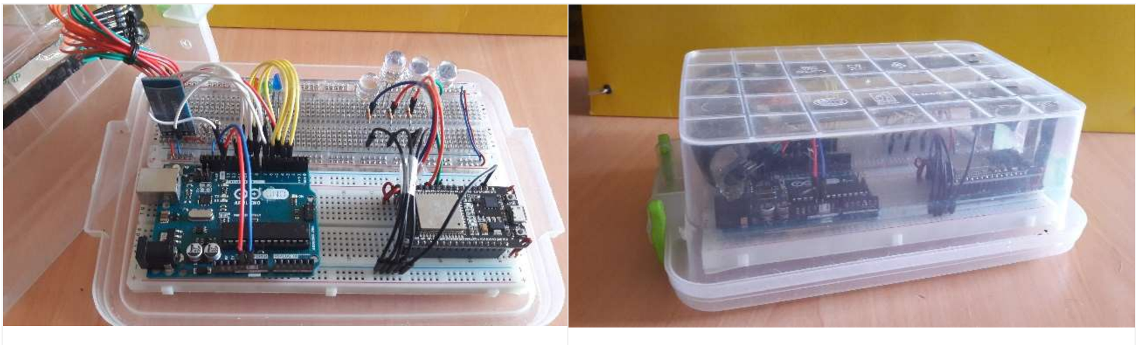

Figure 41 - Actuation module prototype (left) and actuation example (right)

The proposed solution has potential for improvement in several directions. New gateway functionalities can be added with new components and algorithms. The actuation modules hardware can be further explored and enhanced. Future work includes improvement of the control protocols allowing control of mainstream devices.

4.6. Data collection

Sensor data is sampled and can be analyzed and used for action reconstruction. The platform can sample data from several distinct sensors. Depth sensors allow tracking of the position of skeleton joints and bones. Each sensor sample model gathers distinct relevant data that allows element positioning when reconstructing the scene. For instance, the Kinect sensor sample data frame consists of 4 sections: sample generic information such as the sample id, timestamp, order, and associated result id, and the 3D coordinates for the upper torso skeleton joints, for the middle, left and right sides.

Kinect Sample

Id		ShoulderLeft	X Y Z	ShoulderRight	X Y Z
Order		ElbowLeft	X Y Z	ElbowRight	X Y Z
Created		WristLeft	X Y Z	WristRight	X Y Z
ResultId		HandLeft	X Y Z	HandRight	X Y Z
Position	X Y Z	HipLeft	X Y Z	HipRight	X Y Z
Hipcenter	X Y Z	KneeLeft	X Y Z	KneeRight	X Y Z
Spine	X Y Z	AnkleLeft	X Y Z	AnkleRight	X Y Z
ShoulderCenter	X Y Z	FootLeft	X Y Z	FootRight	X Y Z
Head	X Y Z				

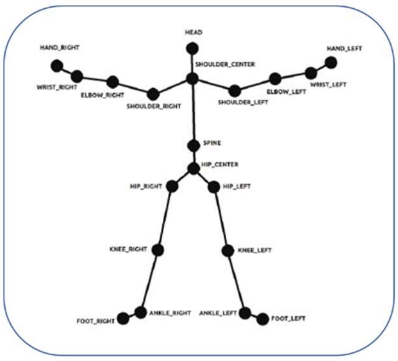

Figure 42 - Data frame for a Kinect sensor sample

Sampled data includes the 3D coordinates of monitored patient skeleton joints. This data allows the analysis and extraction of information, as can be seen in Figure 43, where the maximum reach angles, for the left and right arms, in a sequence of actions, are displayed. The maximum reached left angles are displayed as blue squares and the maximum reached right angles as red dots. In the visualized chart, we can see results from 3 distinct games. The patient played a game named “JPK #1”, on the 29th October 2018. The action sequence is composed of 5 actions. Each action displays the respective values for the maximum lateral reach angle for both arms.

Figure 43 - Data analysis of captured data (using an external tool)

A visualization module provides tools for 3D reconstruction and replay of movements, when applicable. Sensor samples providing joint coordinates allow for display of the composed skeleton. Actions can be navigated through the captured frames.

A visualization variant allows the simultaneous display of the action frames allowing a comparative sequence analysis of the movement. All action frames are displayed in 2D, using a frontal projection, with a small x-axis delta allowing an overview of the captured movement. The left and right hands joints are colored, using blue and red colors, for easier detection and tracking (Figure 44).

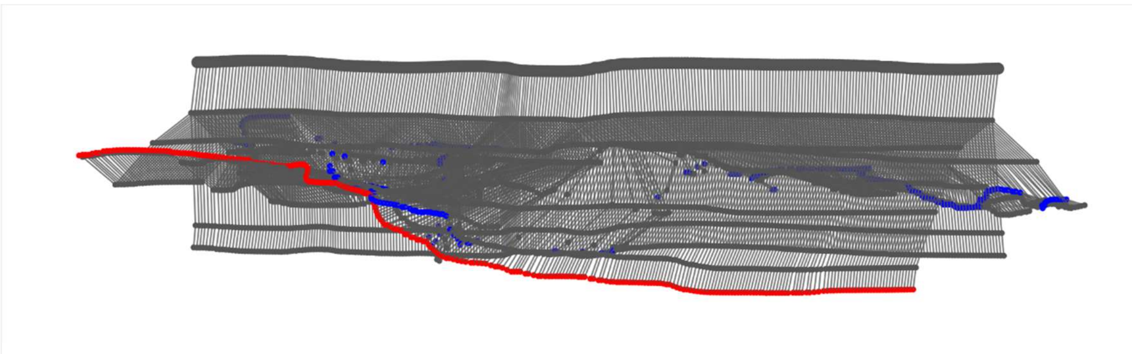

Figure 44 - Captured motion from action using Kinect sensor

Data mining study on platform collected data

An exploratory study was performed using data collected using the platform. The main goal in applying this technology is to find data patterns through time. The dataset is based in data from a computational platform to support motor rehabilitation for patients with cerebral palsy (JPK). The dataset includes user personal and biometric data, game and exercise data, game execution associated data and results, and Kinect sensor samples. Posterior data explorations will include datasets from multiple system setups allowing a more wide and thorough analysis. The Weka Workbench, that allows the integration of all other Weka tools, was used to explore this dataset.

Research question

The research question was formulated: Can the final status of a game movement be inferred from the target angle and movement type? The possible final states are Lost, Captured or Won.

Phase One

The number of patients in this phase was decided to be limited to 6. The resulting query had 509 rows. Several algorithms were applied.

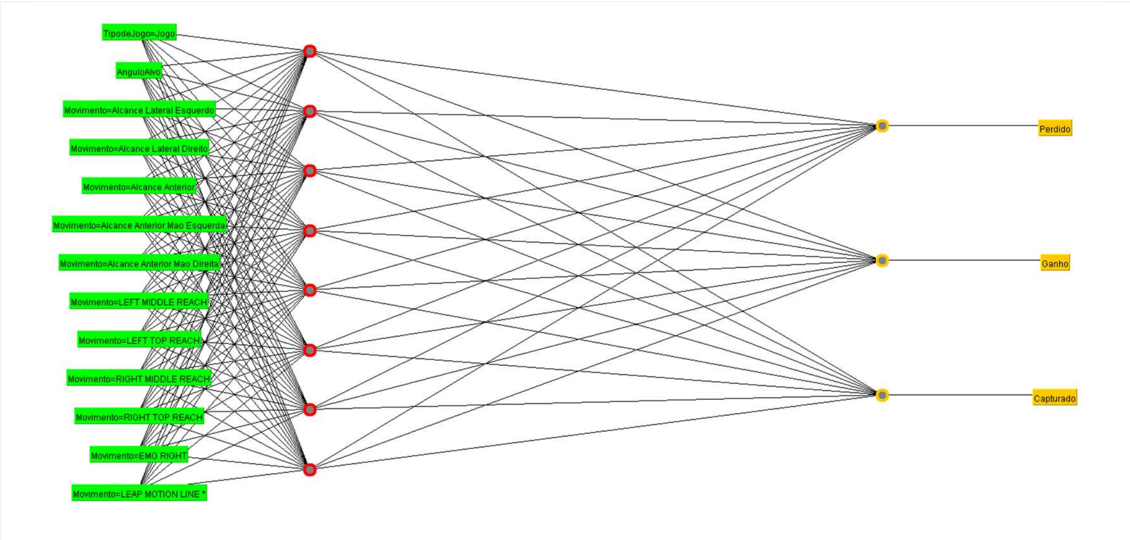

Figure 45 - Neural network

As a final test, for phase one, the Apriori classifier was applied on the previously tested model, to evaluate results of frequent actions clusters. The selected features were Tipo de Jogo, Ângulo de Jogo e Estado de Jogo. Tested sub-features were:

Table 28 - Tested features and sub-features

Features	Tipo de Jogo, Ângulo de Jogo e Estado de Jogo
Sub-features	@attribute TipodeJogo {Rotina,Jogo} @attribute AnguloAlvo {'\'(-inf-52.5]\'','\'(52.5-65]\'','\'(65-77.5]\'','\'(77.5-95]\'','\'(95-125]\'','\'(125-135]\'','\'(135-210]\'','\'(210-inf)\''} @attribute Movimento {'Alcance Lateral Esquerdo','Alcance Lateral Direito','Alcance Anterior','Alcance Anterior Mao Esquerda','Alcance Anterior Mao Direita','LEFT MIDDLE REACH','LEFT TOP REACH','RIGHT MIDDLE REACH','RIGHT TOP REACH','EMG RIGHT','LEAP MOTION LINE *'} @attribute 'Estado DeJogo' {Perdido,Ganho,Capturado}

Minimum support used was 0.1 (51 instances) with minimum confidence of 0.9 and 18 epochs. The 10 best rules found are displayed in Table 29.

The legend for the elements [conf: confidence, lift, lev: leverage, conv: conviction]

Table 29 - 10 best rules

#	Rule
1	AnguloAlvo='(-inf-52.5]' EstadoDeJogo=Ganho 134 ==> TipodeJogo=Jogo 134 <conf:(1)> lift:(1.61) lev:(0.1) [50] conv:(50.81)
2	AnguloAlvo='(-inf-52.5]' Movimento=Alcance Lateral Direito 82 ==> TipodeJogo=Jogo 82 <conf:(1)> lift:(1.61) lev:(0.06) [31] conv:(31.09)
3	AnguloAlvo='(65-77.5]' Movimento=Alcance Lateral Esquerdo 70 ==> TipodeJogo=Rotina 70 <conf:(1)> lift:(2.64) lev:(0.09) [43] conv:(43.46)
4	AnguloAlvo='(65-77.5]' Movimento=Alcance Lateral Direito 67 ==> TipodeJogo=Rotina 67 <conf:(1)> lift:(2.64) lev:(0.08) [41] conv:(41.6)
5	AnguloAlvo='(-inf-52.5]' Movimento=Alcance Lateral Esquerdo Estado DeJogo=Ganho 65 ==> TipodeJogo=Jogo 65 <conf:(1)> lift:(1.61) lev:(0.05) [24] conv:(24.65)
6	AnguloAlvo='(65-77.5]' Estado DeJogo=Perdido 89 ==> TipodeJogo=Rotina 87 <conf:(0.98)> lift:(2.58) lev:(0.1) [53] conv:(18.42)
7	AnguloAlvo='(65-77.5]' 141 ==> TipodeJogo=Rotina 137 <conf:(0.97)> lift:(2.56) lev:(0.16) [83] conv:(17.51)
8	AnguloAlvo='(-inf-52.5]' 255 ==> TipodeJogo=Jogo 247 <conf:(0.97)> lift:(1.56) lev:(0.17) [88] conv:(10.74)
9	AnguloAlvo='(-inf-52.5]' Movimento=Alcance Lateral Esquerdo 127 ==> TipodeJogo=Jogo 123 <conf:(0.97)> lift:(1.56) lev:(0.09) [44] conv:(9.63)
10	TipodeJogo=Jogo Movimento=Alcance Lateral Esquerdo Estado DeJogo=Ganho 68 ==> AnguloAlvo='(-inf-52.5]' 65 <conf:(0.96)> lift:(1.91) lev:(0.06) [30] conv:(8.48)

Analysing the generated rules

The features, sorted by occurrence, in descending order, in the extracted rules are displayed in Table 30.

Table 30 - Sorted features from extracted rules

AnguloAlvo [-inf-52.5] (6), TipodeJogo =Jogo (5), TipodeJogo=Rotina (4), AnguloAlvo [65-77.5](4), Movimento=Alcance Lateral Esquerdo (4), Estado DeJogo=Ganho (3), Movimento=Alcance Lateral Direito (2), Estado DeJogo=Perdido (1)

Rules inferred from results are displayed in Table 31.

Table 31 - Rules inferred from results

Category	Rule
Useful	O TipodeJogo=Rotina está associada ao conjunto de ângulos [65-77.5] O TipodeJogo=Jogo está associado ao conjunto de ângulos [-inf-52.5] do Movimento=Alcance Lateral Esquerdo com Estado DeJogo=Ganho
Trivial	O Movimento=Alcance Lateral Direito e Movimento=Alcance Lateral Esquerdo foi associado ao tipo de jogo, Rotina e Jogo e ambos aos ângulos de [-inf-52.5] e [65-77.5]
Inexplicable	TipodeJogo=Rotina foi associado ao ângulo [65-77.5] e a Estado DeJogo=Perdido
Other	O conjunto Estado DeJogo=Ganho e o AnguloAlvo [-inf-52.5] ocorre 3 vezes. O conjunto Movimento=Alcance Lateral Esquerdo e o AnguloAlvo [-inf-52.5] ocorre 3 vezes. O TipodeJogo =Jogo e o AnguloAlvo [-inf-52.5] ocorre 5 vezes. O AnguloAlvo [65-77.5] e o TipodeJogo=Rotina ocorre 4 vezes.

Hence, we conclude that the most tested and most common attribute in 3 different associations was AnguloAlvo [-inf-52.5], and that TipodeJogo = Rotina and AnguloAlvo [65-77.5] form an association in 3 rules. Other conclusions are that TipodeJogo = Rotina is associated with movements from higher angles and that TipodeJogo = Jogo at lower angles, these rules are trivial because we know that TipodeJogo = Jogo has interactable graphics at the bottom of the screen and TipodeJogo = Rotina is related to high movements on the screen. Therefore, the general set of rules collected should be viewed as Trivial, as they were already known to everyone involved in the project.

Phase Two

On the second phase of this study, the same question was searched, but the dataset size and the number of features analyzed were increased, to search for other relations.

The Apriori classifier was applied to the dataset of 351168 instances and 73 unnormalized features, for comparison with the tested dataset, and assessing if a small sample could be used as analysis model for the complete dataset. Unfortunately, the dataset with 70 features did not return consistent results with the previous ones, reflecting the missing data normalization. The best rules found asserted the existence of a Y-axis movement for any X-axis movement (trivial rule), and the remaining rules were classified as inexplicable (e.g., associating gender to generic movements).

ETL process

Extract, transform, and load (ETL) is a process that allows data analysis and discovery, and is divided in 3 steps. A first step for data setup, a second step for visual analysis, and the third step for an advanced analysis using specific tools. The ETL process requires knowledge of the data relations and data modeling. Data was imported by connecting Excel to the database. Making some changes to specific columns, improved the semantic level and clarified the discovered relations. An updated comma-separated values (CSV) file was exported from Excel for use in Weka.

Data Mining

Several algorithms were applied in the existing data exploration. To determine the relevance of the determined attributes, it was decided to apply the algorithm InfoGainAttributeLevel, combined with the Ranker search method, for assessing which attributes generate a bigger gain in relation to the selected class (in this example was the attribute GameStatus). The evaluation result can be observed in the next table.

Table 32 - Attribute evaluation

average merit	average rank	attribute
1.019 +- 0	1 +- 0	16 GamePoints
0.845 +- 0.005	2.4 +- 0.49	39 kinectJHCZ
0.84 +- 0.008	3.7 +- 1.79	42 kinectJSZ
...	...	...
0 +- 0	70.6 +- 0.92	28 GameMovementLocationY
0 +- 0	70.7 +- 0.46	27 GameMovementLocationX
0 +- 0	71.7 +- 0.46	26 GameMovementBarrierAngle

A rule-based decision tree was generated, applying the J48 classifier. J48 is an implementation of the C4.5 algorithm, that is capable of building decision trees from a set of training data. The method is like the one used by ID3, using the concept of information entropy, but with significant improvements including handling both continuous and discrete attributes, training data with missing values, attributes with different costs, and pruning trees after generation. The nodes and rules, generated through the application of the algorithm, can be visualized in the decision tree of Figure 46.

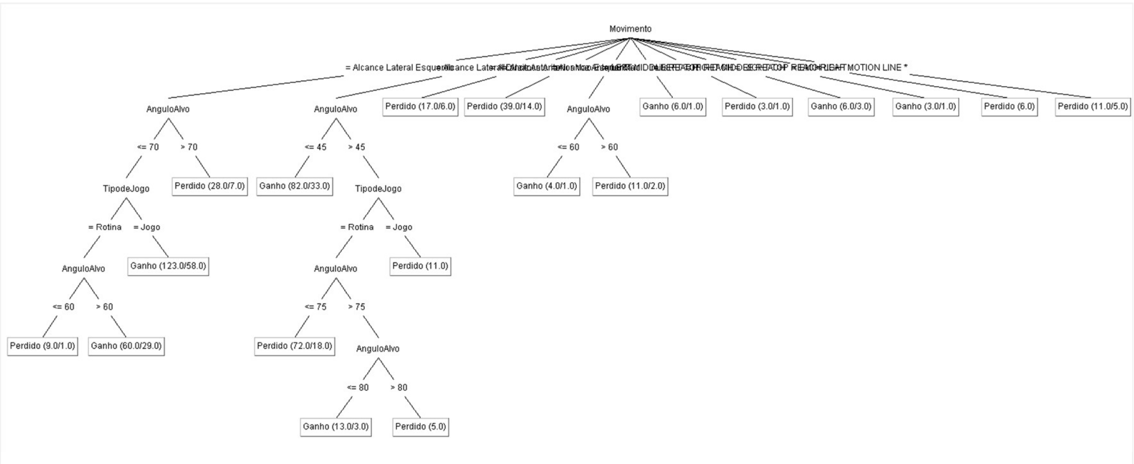

Figure 46 - Decision tree

The results obtained, by training this decision tree with 66% of the records, and using the remaining 33% for inference, allowed the metrics displayed in the next table.

Table 33 - Classification metrics

Correctly Classified Instances	119395 (99.9983 %)
Incorrectly Classified Instances	2 (0.0017 %)
Kappa statistic	1
Mean absolute error	0
Root mean squared error	0.0033
Relative absolute error	0.0046 %
Root relative squared error	0.8413 %
Total Number of Instances	119397

The rules from the decision tree can be seen next in Table 34.

Table 34 - Decision tree rules

```
GamePoints <= 0
|  Gender = F: Captured (340.0)
|  Gender = M: Wrong (230359.0)
GamePoints > 0
|  GamePoints <= 10
|  |  Gender = F
|  |  |  kinectJWRY <= -0.200539
|  |  |  |  kinectJHCZ <= 1.82302
|  |  |  |  |  GameTime <= 7
|  |  |  |  |  |  kinectZ <= 1.46032
|  |  |  |  |  |  |  GameTime <= 5: Wrong (6.0)
|  |  |  |  |  |  |  GameTime > 5: Right (5.0)
|  |  |  |  |  |  kinectZ > 1.46032: Right (253.0)
|  |  |  |  |  GameTime > 7
|  |  |  |  |  |  kinectX <= -0.024304: Right (44.0)
|  |  |  |  |  |  kinectX > -0.024304: Wrong (39.0/1.0)
|  |  |  |  kinectJHCZ > 1.82302: Wrong (29.0)
|  |  |  kinectJWRY > -0.200539
|  |  |  |  kinectX <= -0.068569: Right (12.0/1.0)
|  |  |  |  kinectX > -0.068569: Wrong (83.0)
|  |  Gender = M: Captured (6378.0)
|  GamePoints > 10
|  |  Gender = F
|  |  |  GameTime <= 15
```

```
| | | | MovementLeftArmCheck <= 0
| | | | | kinectJSCZ <= 1.47727: Wrong (2.0)
| | | | | kinectJSCZ > 1.47727: Right (56.0)
| | | | MovementLeftArmCheck > 0
| | | | | kinectJHCY <= -0.10626: Wrong (18.0)
| | | | | kinectJHCY > -0.10626: Right (5.0)
| | | GameTime > 15: Wrong (17.0)
| | Gender = M: Right (113522.0)
```

The confusion matrix generated can be seen in Table 35.

Table 35 - Confusion matrix

a	b	c	**<-- classified as**
38807	0	1	**a = Right**
0	2311	0	**b = Captured**
1	0	78277	**c = Wrong**

The application of the SimpleKMeans clustering algorithm, where samples are grouped in clusters by the closest mean value, generated a cluster set, but the analysis of the application of this algorithm did not produce relevant information.

Applying the MultiLayer Perceptron classifier, a simple neural network was generated. The full visual representation of the neural network was not possible due to the number of attributes.

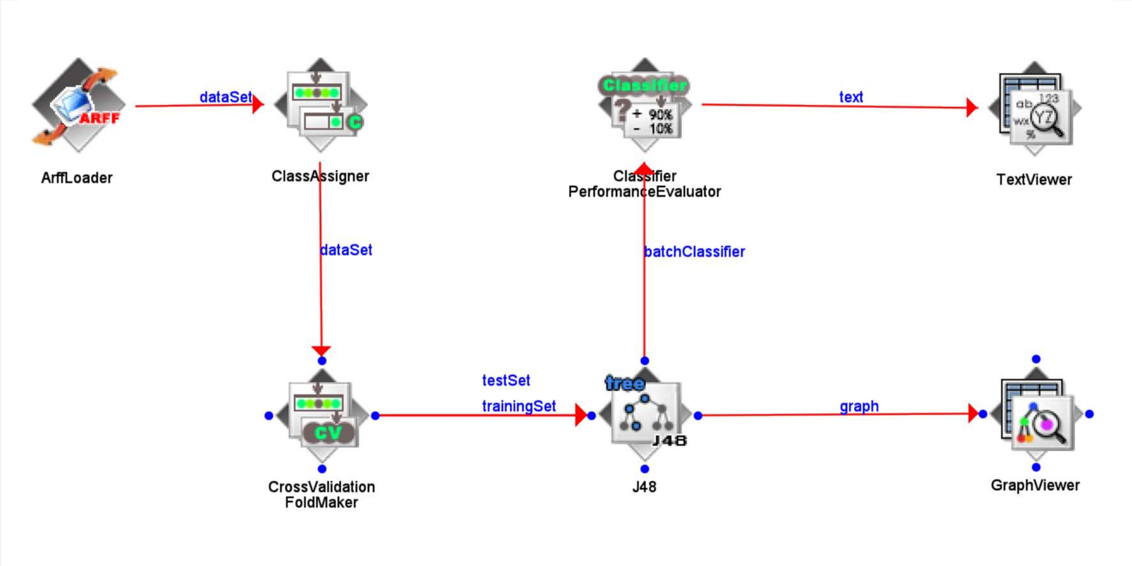

Figure 47 - Weka Knowledge Flow display

Learning

The results from the first phase were very modest, with an inference error rate of 40%. The second phase produced better results, with an inference error rate of 0.002%. The major constraint in working with big data is the processing time, that exponentially grows with data.

Conclusions and future work

The main conclusion was the evidence of the relevance of data quality and that the obtained results were directly dependent on this factor. Some difficulties felt during this study derived from the missing normalization of the data, and were corrected, as possible, during the several phases. The application of algorithms needs more research, for better understanding of parameter optimization, for results improvement. Anyway, this study provided better understanding and knowledge of the involved data. Acquired knowledge can be used for improving the related applications and processes. The data will be further explored, and transformed, in search of new ways for interpretation. This study provides a model that can be used for future inferences and analysis for this platform. Further uses and analysis possibilities with the captured data are still being researched. Future work in this area looks very interesting and promising.

Chapter 5

Results

Several results were achieved during this research. A result of this research is the instantiation of the model by integrating external applications.

The provided API allows external applications and devices to integrate with PLAY. For instance, an external application, JPK, consumes the API for integration with the PLAY platform. A game played by the patient, in the external application, communicates with the platform sending the relevant data during interaction. IoT devices can be used by platform games and external applications. Available actuation devices act in response to the platform requests, activating relevant actuators during game execution and platform usage. A new game has been designed and integrated in JPK. The Astro game provides new game mechanics and gameplay options, expanding JPK capabilities, and is also a case study for integration with the PLAY platform.

Endless Runner Game (ERG) is another example of an external game consuming the API for integration with PLAY. ERG is a master's dissertation prototype currently being developed.

Tests are required at distinct stages of development and for distinct purposes. A study with children was made at a partnering clinic for assessing platform validation when used as a complement to conventional therapy. The study settings, results and conclusions are presented.

Validation tests, with therapists and curators, are designed. A questionnaire was prepared, and tests are planned to happen soon. The focus will be in assessing the framework model relevance and validity as a tool for meeting therapeutic response requirements.

5.1. API

An API is provided as a web service and is the platform access point to data management, provided functionality and interface for external applications. Using JSON as data format and following a RESTful approach, the API provides registration and authentication services for the applications and users. Game services are provided for exchanging game information as structure and actions, results, and samples. These game services allow applications to create their own game editors and game engines. Device services allow actuation devices registration and authentication. Interaction services provide device management and control.

5.2. External applications

External applications can consume the provided platform API to integrate the PLAY environment. So, as proof-of-concept, the JPK platform patient application integrates with the PLAY framework platform prototype. The PLAY server provides the web services needed by the JPK platform components. It also provides the gateway services, used in the JPK platform, answering requests for actuator interaction by JPK applications. The JPK platform web client component (JPKM) has been updated to forward the requests to the PLAY server. This integration allows the PLAY framework to use the Microsoft Kinect sensor through the JPK game implementations.

5.2.1. *JustPhysioKidding*

Preliminary work included the implementation of a rehabilitation serious game, Just Physio Kidding (JPK) [83], in co-creation with Cresce Com Amor (CCA), a clinic providing physiotherapy services to patients with SN. JPK was designed according to the presented model, initially developed as a stand-alone application, but then connected to PLAY using the provided API to test the platform and benefit from all its features. This way, JPK will be available to be used in the context of other clinics that have a PLAY account. Sensors used include the Microsoft Kinect, Leap Motion and Shimmer.

CCA is working with 31 children (7 girls and 24 boys). Some patients have different pathologies at distinct levels, with the most frequent being CP. The motivational factor is extremely important and therapists at the clinic complement their interventions with patients by using PLAY's SG, being JPK the primary one.

Therapists reported, and recorded on video, a patient that couldn’t usually stand on his own but, when playing the game, he got so engaged and focused that he got motivated to overcome his own impairment and stood up while playing, as can be observed in Figure 48. That is very motivating for the research team too.

A study was conducted, at the clinic, for testing the current state of the platform, assessing potential development directions, and shortcomings for participation in the games.

Figure 48 - Patient playing the JPK platform serious games at the clinic

Population characteristics and therapeutic needs

The therapy center is working with 31 children (7 girls and 24 boys). Some patients have different pathologies and distinct levels of pathology. The most frequent pathology among the clinic patients is cerebral palsy.

Preliminary interview sessions with the therapists allowed the identification of some needed functionality, that should be available in future platform interventions. During several meeting sessions, held with Cresce therapists, patient and therapist needs for the platform, and the need for therapy solutions, were discussed. Some inquiries were also provided for therapists to fill. The inquiries are provided as Appendix II. The clinic therapists characterized each patient, identifying the individual skills and areas of interest for intervention. This characterization chart is displayed in Figure 49.

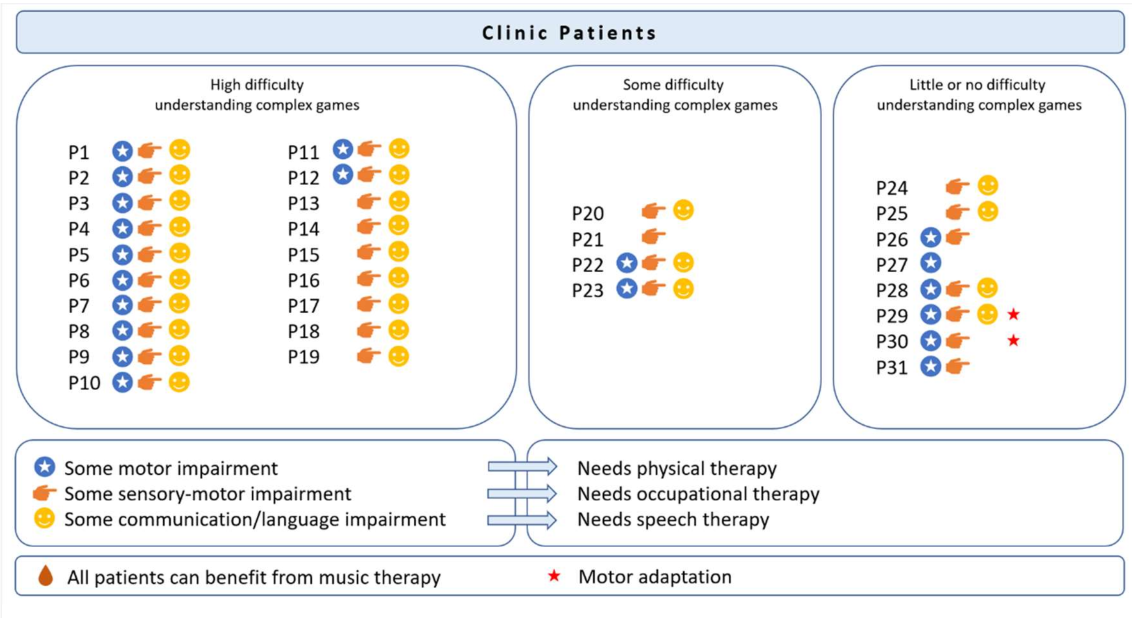

Figure 49 - Clinic patient’s characterization

Distinct types of impairments are considered: motor, sensory-motor, communication, and language. Motor impairments require physical therapy, with two patients needing motor adaptation to be able to participate. Sensory-motor impairments require occupational therapy. Communication and language impairments require speech therapy. All children can benefit with some aspects of music therapy.

One of the ideas that emerged during the team meetings is a game like super Mario bros where action and characters move through different scenarios involving pipes that interconnect them. Kinect could be used as the main interface for controlling characters when outside the pipes. The Leap Motion could be used to control finer movements inside the pipes. The patient could be asked to perform specific tasks to train fine motor skills. These tasks should be configurable to be adapted to specific patient conditions.

Another idea is joining several aspects of therapy, like motor, focus, cognitive, speech, in the same games or exercises. This promotes the role of the platform as an aggregate therapy tool for multiple areas. This idea is even more valid as we understand that most of the children need to work in multiple impairment areas. Most of the patient cannot afford to get therapy on all areas they need. A tool like this would make it easier to allow therapies on multiple areas promoting more accessible healthcare. Another valid though is the one that, aggregating several aspects of therapy, patients keep working some therapy areas that are not the focus of a specific exercise.

Some patients, with stronger cognitive impairments, cannot fully understand that they are controlling the game. Integration of the image or video of the child playing the game could help the child understand the he or she is in control and interacting. That image or video could help patients get a sense of immersion or placement in the game. Some pathology conditions lead the patient to not fully understand his own body in a spatial context. Patients at the clinic have distinct levels of cognitive impairment. Motivation is hard to accomplish with children with strong cognitive impairments. Understanding progression during the game is not an easy concept for some patients. Some users need strong and meaningful rewards after exercise completion to keep motivated. Patients with lighter degrees of cognitive impairment are easier to motivate as they understand progression. A progressive level of difficulty should be implemented to keep them motivated.

Patients in need of physical rehabilitation can benefit from exercises focusing gross and fine motor skills. Gross motor rehabilitation refers to movement and functionality of the upper and lower limbs. A set of relevant movements for gross motor rehabilitation can be seen in Table 36.

Table 36 - Essential gross motor movements

Movement	**Description**
Weight transfer	transfer of center of mass
Lateral reach	reach a lateral location with the hand with varying angles
Anterior reach	torso back rotation with varying angles
Torso rotation	left or right rotation of the torso with varying rotation angles

Posterior reach	torso frontal rotation with varying angles
Cross-arm reach	left or right reach of location with the opposite arm

Fine motor movements are captured using the Leap Motion sensor. Fine motor skills refer to movement and functionality of the hands and fingers. Examples of movements for fine motor rehabilitation can be seen in Table 37.

Table 37 - Essential fine motor movements

Movement	**Description**
Finger stretching (each finger)	stretch the relevant finger
Finger reach	reach and touch with the relevant finger
Hand movement tracking	following a predetermined path with the hand
Gestures	make a specific gesture with the hand
Hand open/close grab/release	complete opening or closing of the hand with or without grabbing objects
Hand pronation	frontal (up/down) hand rotation
Hand supination	lateral (left/right) hand rotation

Some patients cannot stand on their own. The system should determine if the patient is standing, sitting, or lying on the floor. The system should then act reflecting the user's limitations and adapting its game logic. For this to happen, the location and position of the patient should be assessed. Therapists suggested that spatial reference elements should be included in the captured samples, for better understanding of scene and actors positioning, and efficient scene replay and analysis. The location of the used sensors and the patient is a must have. The ground location is also a requirement. The location of the exercise goal target is also a needed reference point. These reference coordinates help the system to effectively provide some relevant information. Therapists identified the need for the information of the hand or limb that executed the exercise (if using the left or right hand). Data capture from both hands must be a requirement. Even therapies that only focus on one side of the body need to monitor if there is a compensatory movement on the other side. Therapists identified relevant angle measurements for shoulder joint movements like extension, flexion, and rotation. The movements and angle ranges are described in Table 38, Table 39 and Table 40. These movements and angles relate to both left and right limbs.

Table 38 - Shoulder Joint Measurements

Description	**Movement A**	**Movement B**
sagittal	flexion 0-180º	extension 0-60º
frontal	abduction 0-180º	adduction 0-40º
horizontal	horizontal abduction 0-130º	horizontal adduction 0-60º
vertical	internal rotation 0-80º	external rotation 0-80º

Table 39 - Elbow Joint Measurements

Description	Movement A	Movement B
sagittal	flexion 0-170º	extension 0-10º
axial	supination 0-85º	pronation 0-85º

Table 40 - Wrist Joint Measurements

Description	Movement A	Movement B
sagittal	palmar flexion 0-80º	dorsal flexion 0-70º
frontal	radial deviation	ulnar deviation

The clinic provides several therapeutic services for patients. Physiotherapy or physical therapy promotes the rehabilitation of motor skills. Occupational therapy is used for training daily activities skills. Speech therapy is applied on rehabilitation of speech and communication disorders. Music therapy is the application of music in rehabilitation. Therapeutic application of music in rehabilitation can be used for improving strength, range of motion, balance, communication, and cognition [18]. Therapists could benefit from a game model that could promote multiple ways of interaction, such as movement and joint tracking, voice control, touch and drag actions, force platforms for posture and weight distribution analysis, eye tracking, among other possibilities. This game could combine these various action types for a wider range of intervention. This kind of game would be a valuable tool for assessment of several metrics in the multiple therapy areas, these being physical, speech, music, and occupational therapies. A patient profile would be the tool to gather and keep track of these metrics. This kind of game model would allow interventions in each domain, in an isolated way or acting in multiple areas simultaneously. Therapists could, using this kind of tool, create personalized games with multiple therapeutic goals.

Exercise performance assessment and evaluation

One of the challenges faced when designing computer-assisted exercises is how to evaluate patient performance during the exercise. Physiotherapy with children with special needs is much based on the personal sensibility of the therapist. Using technology efficiently on health tools require developing ways to measure the user performance in a therapeutic context. Metrics and ranges must be identified and validated. Some tools exist that help therapists evaluate exercise execution. The evaluation tools used at the clinic for assessing patient performance of exercises are QUEST, SPCM / LSS and MACS.

JPK Platform

Just Physio Kidding (JPK) is a rehabilitation platform developed in partnership with Cresce Com Amor clinic. The prototype was originally directed for stroke patients in wheelchairs. The

JPK platform was developed as the final project for a BSc in Software Engineering and improved during a following research grant (TailorPhy). The platform consists of a serious games patient desktop application and an analysis tool for therapists in desktop and web. [96] Developed in C#, has 3 main components (JPK, JPKT, JPKW) and 3 secondary components that provide common models and a restful API (JPKREST, JPKDATA, JPKM).

The platform UI was created from scratch integrating best practices for UI design. The user experience and functionalities were subject to extensive analysis and refactoring during the progression of the project. The UI was subject to refinement with added functionalities, providing better usability and multiple ways of interaction. It was tested in several sessions at the clinic with a set of patients. Primary movements modeled in the system, displayed in Table 41, make use of two of the sensors that the platform uses, Microsoft Kinect and Leap Motion.

Table 41 - JPK Primary movements

Sensor	**Movements**
Kinect	Weight Transfer; Side, Anterior and Later reach; Torso rotation
Leap	Finger stretching, thumb opposition, path following (line, circle, triangle, saw, spiral)

The third sensor used in platform is the Shimmer sensor. The Shimmer sensor can, if available, measure electromyography (EMG) signals during exercises. EMG signals allow detection of muscular activity in a monitored body area, usually arms or legs.

The platform uses an internal adaptation module providing configuration settings for the gaming component, that can be set a priori by the therapist or at runtime by the patient, and consumes the services of the external P^{2}Muca API [97], using the available clustering algorithms to create user stages where patients are placed based on platform usage and game performance. User game settings are changed automatically based on cluster placement promoting a more personalized game adaptation to patient therapeutic needs and a better user experience. The therapist tool (JPKT) is used to create and manage games and exercises and review patient progression, performance, and results. The execution of the exercises can be replayed displaying a 3D skeleton view. To increase the provided tools for user immersion, the platform integrates with a smart IoT gateway, for providing external devices control.

Research and development for the JPK platform was a valuable source of knowledge that was applied in this research. The patient application of the JPK platform is used, in this research, as a case study for the instantiation of the proposed model.

5.2.2. *Astro game*

Following the steps of Eco, the famous (!) character from JPK platform, in his quest for recycling garbage, we venture the ways of sustainability and present a **Journey for a Sustainable World**. In this series of games, we travel in search of possible contributions for making the world a better place to live, reducing the usage of resources, reusing used ones, and promoting local and global sustainability. Noir and Bill are two of Eco's friends. Together they decided to do something about the current state of the world. All exercises are modeled using this paradigm and contextualized to promote this concept.

A new character, Astro, an interplanetary explorer astronaut, joins the sustainability effort and promotes recollection of space garbage and related items. Let's promote sustainability at a universal level. The character has been designed from scratch, following the previous implemented JPK aesthetics, integrating the JPK game characters set.

Figure 50 - JPK game characters (Eco, Noir, Bill, Astro)

The Astro game can be accessed through the JPK main game menu, as displayed in Figure 51, for autonomous play. Games can also be scheduled and will appear in the patient schedule as a requested execution. The JPK engine only captures sensor samples when playing a scheduled game, and when that happens, the sampled data is sent asynchronously to the PLAY server after the game finishes.

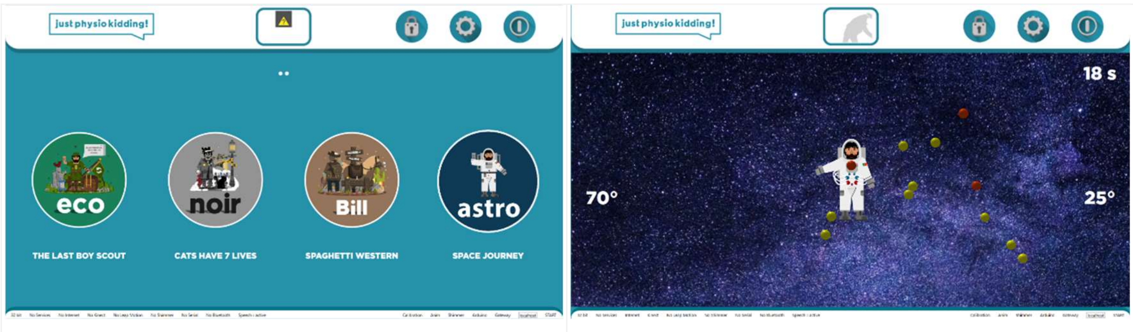

Figure 51 - JPK Game menu (left) and Astro game (right)

The Astro game uses the same mechanics as the other JPK games. The game structure is defined by the therapist, using the game creation tools in the PLAY platform and requested from the PLAY server. The game structure is composed of sequences of actions, composed by the therapist for that specific patient. The Astro game logic prepares the needed data structures for game execution based on the received instructions.

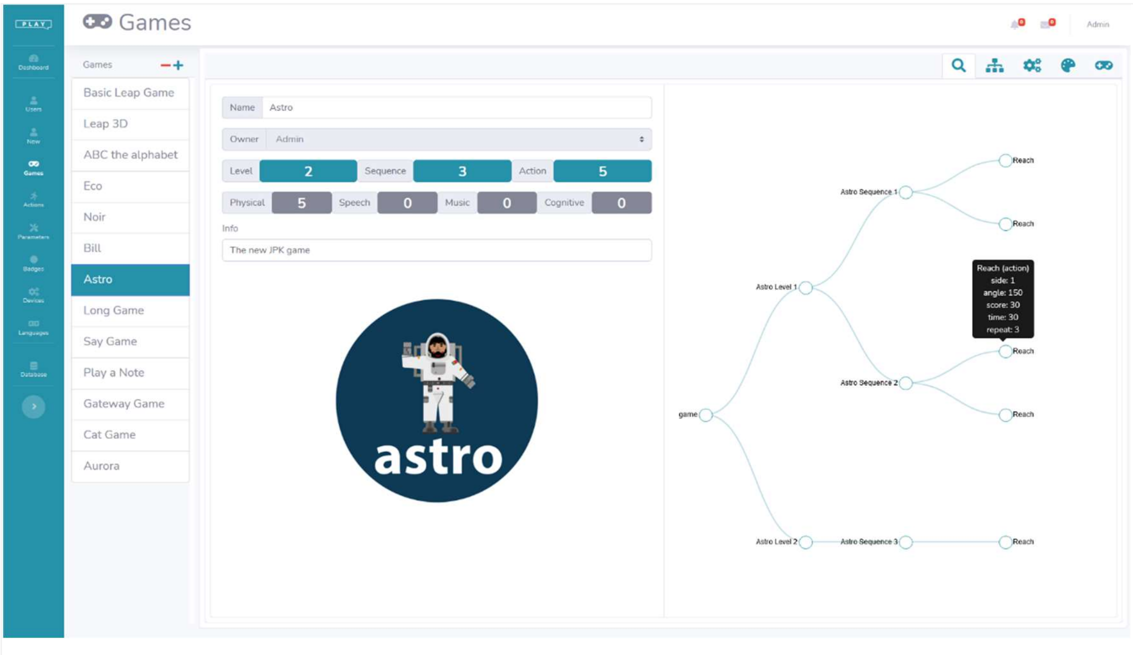

Figure 52 - Astro game in PLAY

The Astro game definition used as example has 2 levels, the first one with 2 sequences of 2 reach actions, and the second one with a sequence of a single reach action. The reach actions are associated to the Kinect sensor. Each action is configured with specific side, angle, score, time, and repeat parameters.

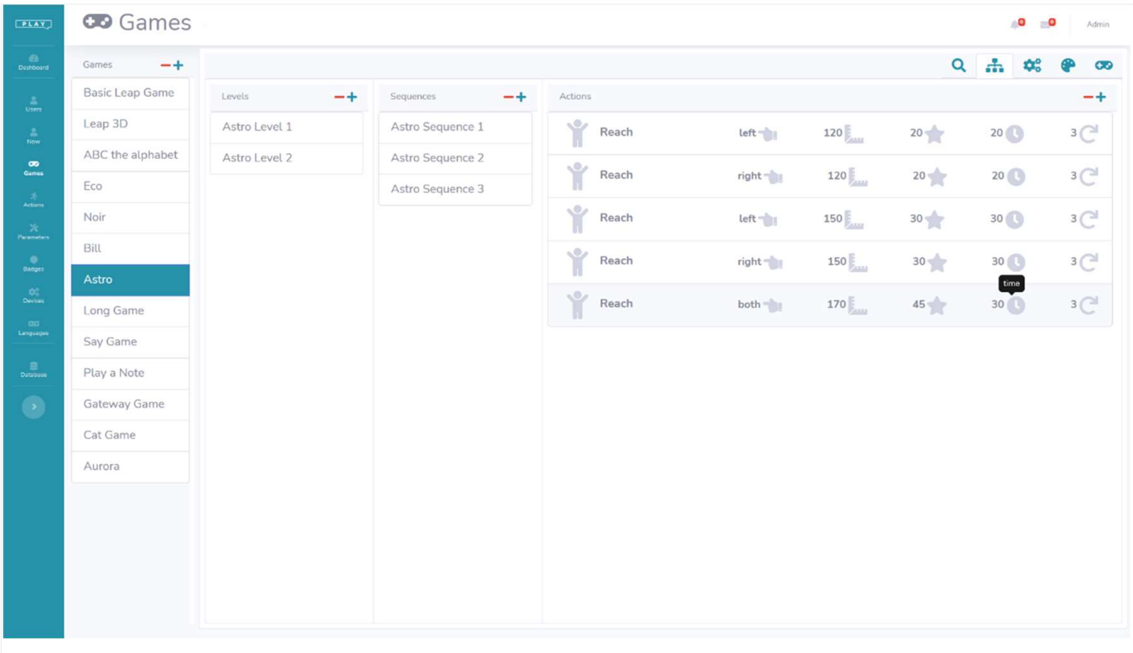

Figure 53 - Astro game actions in PLAY

5.2.3. *Consuming the API*

The JPK application consumes the API provided by the PLAY platform. The API provides services for user authentication. After an authentication request from an external game, the server returns a token for validating subsequent communication. The user profile and scheduled games can be retrieved.

Before the patient initiates playing the game, the game structure and configuration parameters are requested from the server. The JPK application implements the necessary game logic for game execution.

The game character, used as patient avatar during the game, has movable arms. This allows lateral opening and closing of the arms for reaching certain angles between selected skeleton joints. Target objects are placed in specific screen locations promoting the requested joints angle.

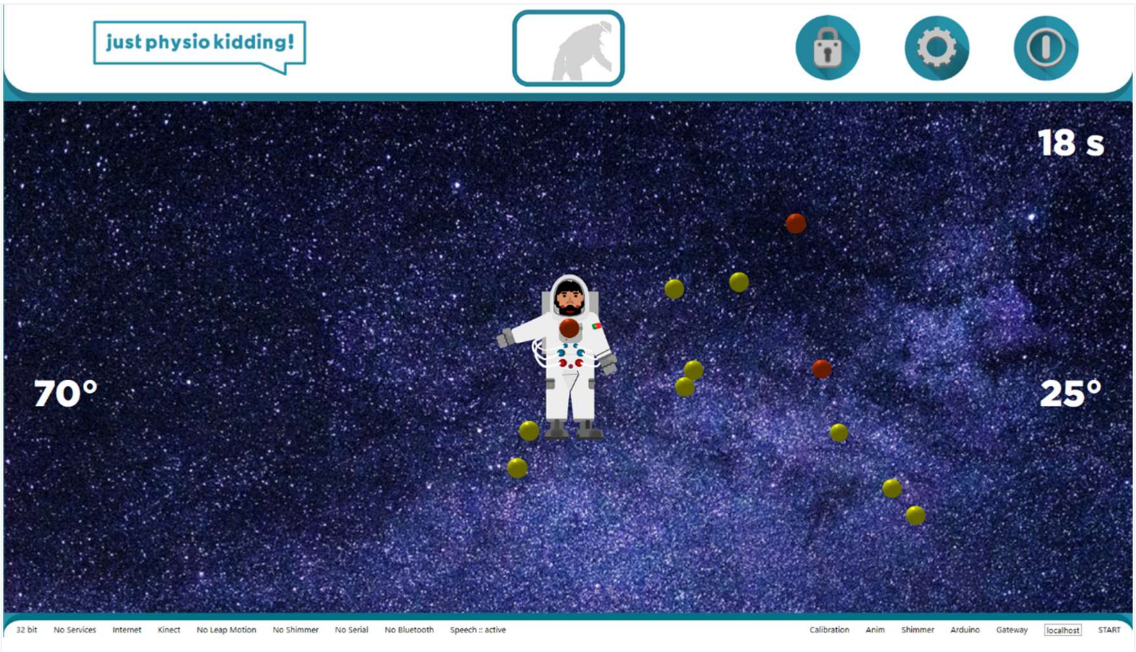

Figure 54 - Astro game

During the playing of the game, the game flow variables, such as time and score, are sent to the server for real time monitoring. Any game parameter updates, or user notifications can be sent to the game in real time. Any action in the game structure that has not been played yet can have its parameters updated and they will be sent to the server for real time update of the game action. This feature allows therapists to correct and adjust game parameters if they detect that need during the game. Sometimes increasing the remaining time for completing a required action is enough to prevent patient frustration. The game results and samples are sent to the server after game completion.

5.2.4. *Real time monitoring*

The game can be monitored, by the therapist, in the framework frontend. The current game action is highlighted, and the current action time is updated in real time.

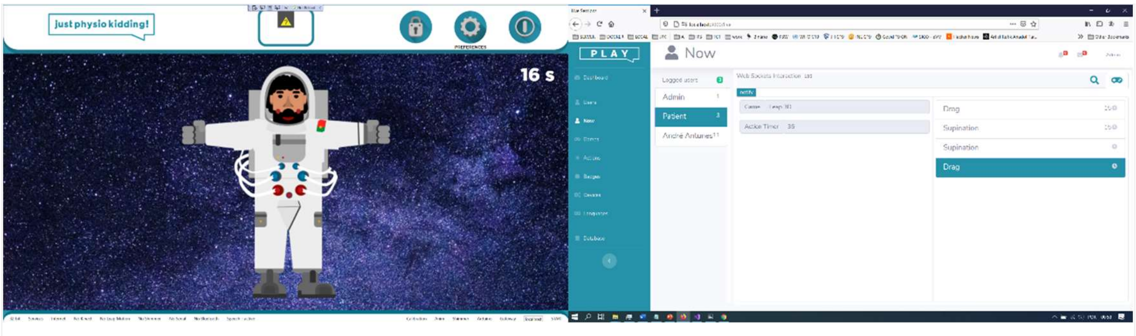

Figure 55 - Live patient game interaction displayed in therapist session

The therapist can use the provided tools for real time interaction with the user, by sending notifications. Game actions parameters can be altered with real time update in the patient session. Results of performed games can be consulted and replayed using the frontend tools.

5.2.5. *Actuation resources*

The actuation module receives the interaction requests from the JPK application, and triggers the relevant actuation device, providing an extra level of environment awareness, to promote greater immersion and ultimately boost interest and motivation.

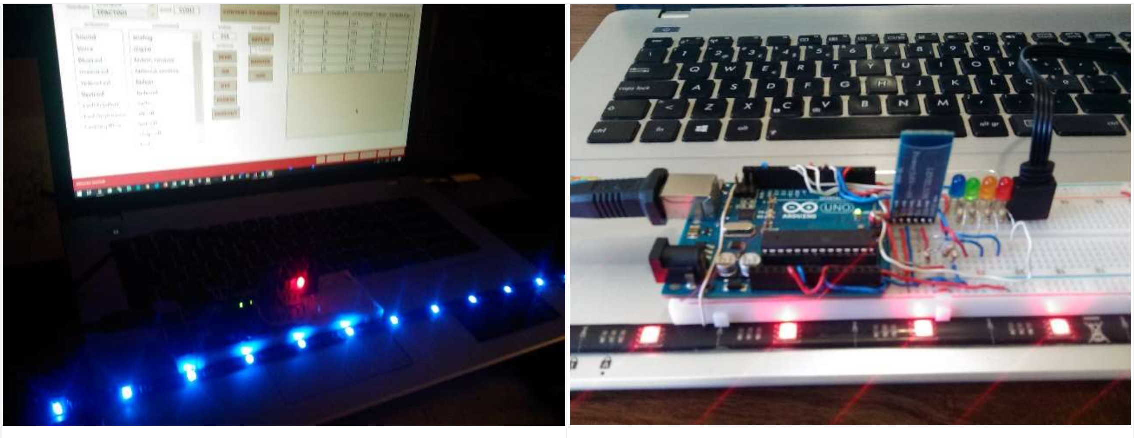

Figure 56 - Live actuation session

5.2.6. *Endless Runner Game*

Endless Runner Game (ERG) is a master's dissertation project from Gonçalo Roxo, a master's student of Computer Science at Faculdade de Ciências e Tecnologia of Universidade

Nova de Lisboa (FCT-UNL). The solution presents a 3D game environment with a character avatar representation and is based in Unity and C#.

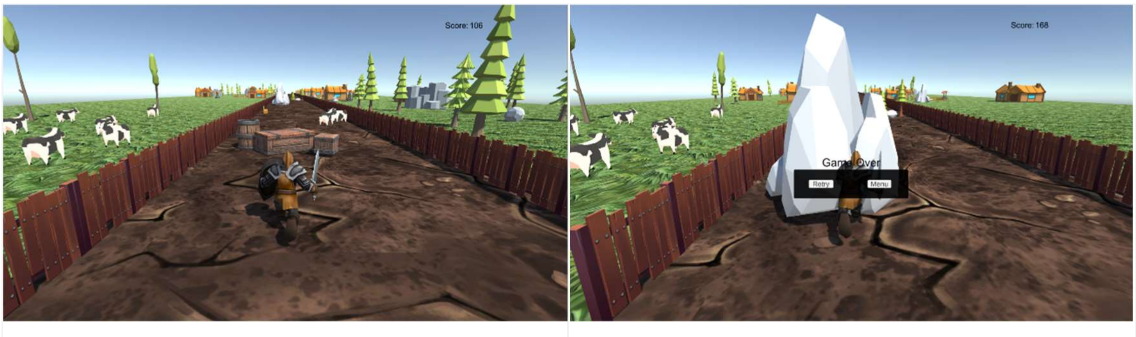

Figure 57 - ERG in-game (left) and game over (right)

The game mechanics determines that patients must run through an endless road, avoiding and/or collecting items during a time frame (Figure 57). Several sensor devices can be used as input for controlling the character actions. Integration with a Cosiki Dance Pad is being implemented for using lower limbs postural exercises.

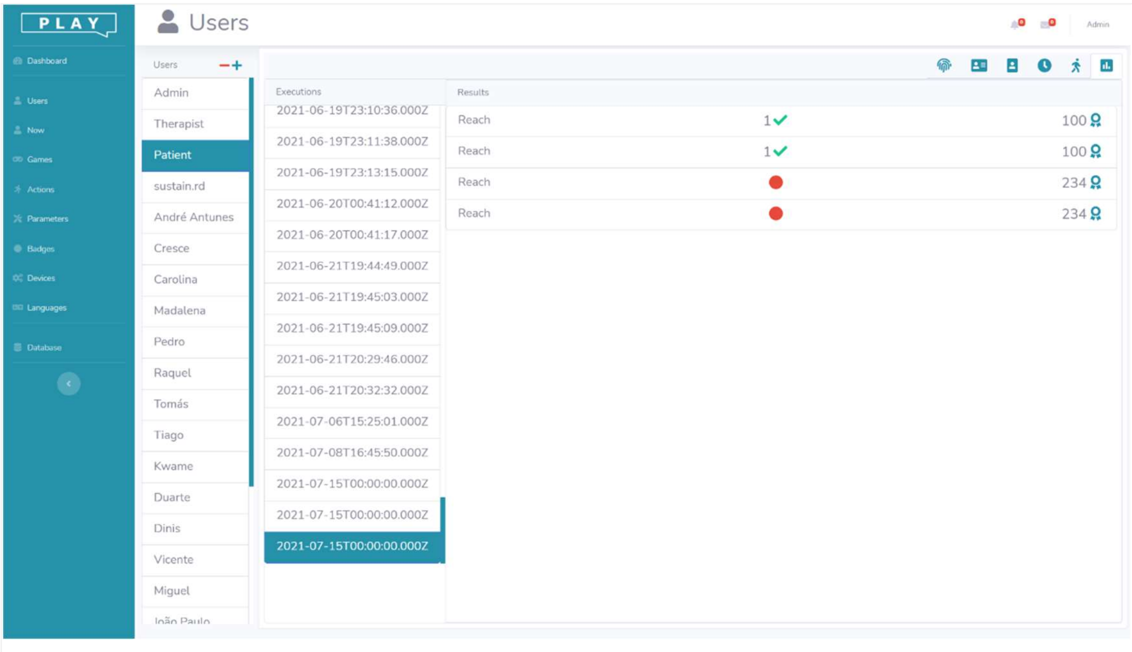

Figure 58 - ERG game results in PLAY

Results obtained using ERG game are sent to the PLAY server (Figure 58). The game execution results can be consulted, allowing assessment of user progression.

5.3. Tests with patients

A study was conducted for testing the current state of the platform, assessing potential development directions, and limiting factors of participation in the games [98]. A therapy session protocol was conducted at the clinic, with clinical patients, using JPK games as a complement to conventional activities. The patient population included eight children and adolescents. The inclusion and exclusion criteria used when building the test group can be seen in Table 42. All participants and their parents, or legal guardians, were informed and provided written consent before inclusion in the study. Therapy sessions were recorded on video.

Table 42 - Inclusion and exclusion criteria

Inclusion	Exclusion
age 5-25 years capability to understand and follow simple instructions LSS level V-VIII	severe visual and auditory impairments severe cognitive and/or behavior impairment LSS level I-IV

Each child patient had his own intervention plan with objectives and goals. Sometimes goals and objectives were shared among patients. Some other variables like motivation and behavior can also be correlated between patients. Each patient was evaluated individually and in each exercise component. A global evaluation was also made by the therapist. Patient population characteristics are displayed in Figure 59. Patients present mild or severe motor impairment including unilateral and bilateral CP, plexus brachial injury, spinal muscle atrophy type II, and learning difficulty.

	Age	Gender	Diagnosis	GMFCS	BFMF	LSS	Cognitive imp.
Case 1	5	M	Hemiparesis	I	II	VIII	-
Case 2	7	F	Diplegia	II	I	VIII	+
Case 3	7	M	Diplegia	II	I	VIII	-
Case 4	7	M	Tetraparesis	III	III	VI	-
Case 5	24	M	Ataxic CP	III	II	VIII	+
Case 6	6	M	PBI	-	I	VIII	-
Case 7	8	M	SMA II.	-	III	V	-
Case 8	7	F	LD	-	I	VIII	+

CP: cerebral paresis; PBI: plexus brachial injury; SMA: spinal muscle atrophy; LD: Learning difficulty; GMFCS: Gross Motor Function Classification System; BFMF: Bimanual Fine Motor Function; LSS: Level of Sitting Scale; Cognitive imp: Cognitive impairment

Figure 59 - Patient population characteristics

5.3.1. *Settings*

Patients used the platform between 3 and 5 times, in sessions of 10 minutes, as a complement to conventional therapy. The conventional therapy applied included stretching and mobilization, strengthening, proprioceptive and balance training, and motor planning education.

At the beginning and end of the program the motor function and postural control at the sitting position were evaluated. Functional classification systems were applied once, for each participant, at the beginning of the study:

- GMFCS was used for characterization of motor functions of CP patients.
- BFMF was used for upper limb functions classification.
- LSS was used to characterize the sitting ability level.

Individual games were designed, for each patient, based on personalized therapeutic goals. Game actions used were gross motor reach and drag actions. Patients had to grab items at specific screen locations, and drag them to the corresponding containers, promoting upper limb movements and achieving a specific angle between relevant skeleton joints. The Kinect sensor was used for this interaction.

For every session a report was made, registering the motivational level, using a five-grade Likert-scale, at start and end of the game. Therapists also registered notes about the worn orthoses, position of the participants and compensatory movements.

5.3.2. *Results*

A 1-point improvement was registered, for two children, in the function section of SPCM. No other changes were registered for the other participants. In the QUEST, all patients except one, registered a slight improvement from baseline. Results for SPCM and QUEST are displayed in Figure 60.

	QUEST		SPCM Alignment		SPCM Function		
	Pre-treatment	Post-treatment	Pre-treatment	Post-treatment	Pre-treatment	Post-treatment	Number of session
Case 1	67,14	67,53	87	87	48	48	3
Case 2	70,46	71,39	86	86	48	48	3
Case 3	84,44	87,22	78	78	48	48	5
Case 4	33,57	38,19	74	74	40	41	4
Case 5	90,10	91,49	84	84	46	46	5
Case 6	81,77	84,66	85	85	48	48	5
Case 7	20,77	24,19	82	82	36	37	3
Case 8	88,42	88,42	87	87	47	47	3

QUEST: Quality of Upper Extremity Skills Test; SPCM: Seated Postural Control Measure

Figure 60 - QUEST and SPCM results

The patient group was very heterogeneous both in terms of diagnosis and age. The number of interventions was very short (3-5), due to distinct reasons for each patient. These limitations prevented obtaining enough and quality data for the application of statistical analysis.

	Session 1		Session 2		Session 3		Session 4		Session 5	
	Start	End	Start	End	Start	End	Start	End	Start	End
Case 1	5	4	5	4	4	4	-	-	-	-
Case 2	5	2	4	3	4	3	-	-	-	-
Case 3	5	5	5	5	5	4	4	4	3	4
Case 4	5	5	5	5	5	5	4	5	-	-
Case 5	4	3	3	2	3	2	2	2	3	3
Case 6	5	5	5	5	5	5	4	5	4	4
Case 7	5	4	5	4	3	4	-	-	-	-
Case 8	5	3	5	3	4	3	-	-	-	-

Figure 61 - Motivation results

Patient 1 is a boy aged 5 years old, diagnosed with left side hemiparesis. After 3 sessions, he could dissociate his hands much faster and with less failure, improving 1 point in the QUEST domain of dissociated movement. SPCM scores were unchanged.

Patient 2 is a girl aged 7 years old, diagnosed with spastic diplegia. She was always very excited to play the game, causing behavioral and attentions problems, and resulting in the need to interrupt the game every time. QUEST and SPCM did not show significant improvement.

Patient 3 is a boy aged 7 years old, diagnosed with spastic diplegia. Always very engaged and motivated to play the game. Achieved great improvement in left and right arm discrimination. QUEST scores improvement and SPCM remained unchanged.

Patient 4 is a boy aged 7 years old, diagnosed with spastic tetraplegia. Played 4 sessions. Always very motivated to play the game. Improvement in left and right arms discrimination. Improvement for QUEST scores on all domains, and SPCM function section.

Patient 5 is a boy aged 24 years old, diagnosed with ataxic CP. Played 5 times in the sitting position. Showed a high level of motivation, at the start of the sessions, but lost it by the end of each session. Improvement in matching objects by color with less improvement in arms dissociation. Improvement in QUEST domain of grasping. SPCM scores remained unchanged.

Patient 6 is a boy aged 6 years old, diagnosed with right side plexus brachial injury. Since this patient does not have CP, GMFCS was not applicable, but he is totally independent. Very engaged, with high level of motivation, both at start and end of sessions. Great improvement at mastering the game and decreasing the time required for exercise completion but requiring compensatory movements. Improvement in QUEST scores, and SPCM scores remained unchanged.

Patient 7 is a boy aged 7 years old, diagnosed with spinal muscle atrophy (SMA). Played the game sitting on the floor, with the elbows supported on a bench. Showed interest in the game but was not very motivated. Improvement in QUEST domains of dissociated movement and grasping, and in SPCM function section.

Patient 8 is a girl aged 7 years old, was a premature, born in the 26th week. She had low motivation to play the game and a few times became distracted but showed rapid and huge improvement in the intervention areas. QUEST and SPCM scores remained unchanged.

The study concluded that the platform has great potential but could benefit from further development in the gaming component. Therapists felt the need for games focusing a specific age, gender, and cognitive level, which would promote engagement and motivation for longer periods. More complex motion options and the application of other sensors are also recommended. The platform was validated, by the therapists, for distinct therapeutic goals (in motor and cognitive areas), as discriminated in Table 43.

Table 43 - JPK validated therapeutic goals

Motor functions	Cognitive functions
Increasing the active ROM of the upper limbs	Right-left discrimination
Strengthening of the muscles of the upper limbs	Improve the spatial orientation
Improve the balance and the proprioception	Recognition and matching colors
Improve the motor coordination	Improve the multichannel attention
Dissociate the upper limb movements	Enhance the eye-hand coordination
Train the transverse movements through the body midline	

5.4. Tests with therapists

All software products need testing to determine its validity and conformance to user's needs. Supervised validation tests with therapists are planned. This prototype will be subject to different tests to assess framework validation from therapists and curators. The System Usability Scale (SUS) [99] test will be used as a first approach to assess the prototype usability.

5.4.1. *Tests script*

A test script has been prepared to help maintaining the test sequence and execution of the required actions. Tests will be supervised by an assistant. After a brief introduction to the concept and platform functionality, therapists must follow the script and execute a sequence of actions. These actions cover and demonstrate key concepts of the framework model that are implemented in the platform.

Required actions include the creation and configuration of a game for a specific patient. The game structure must be defined, and the composing actions added.

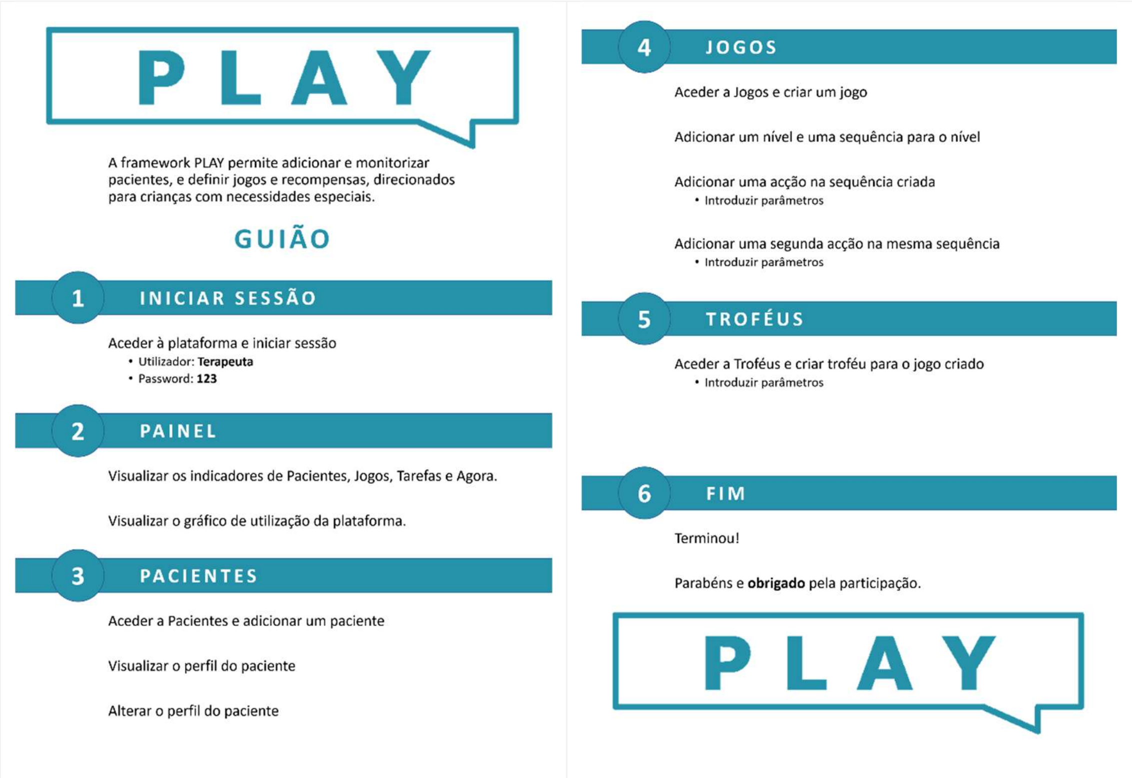

Figure 62 - Test script

5.4.2. *Questionnaire*

A questionnaire has been prepared for assessing the validity of the proposed model and platform usability. Therapists are welcome to fill this questionnaire after the testing session. Questions are designed to understand therapists' feelings and level of acceptance towards model and platform. An initial introduction provides information about the questionnaire's context and objectives.

The first set of questions is used for knowing about the therapist answering the questionnaire and his/her characterization.

A second set of questions refers to platform friendliness and efficacy for managing patients and designing games.

The third section of the questionnaire is intended for measuring the global usability of the applications, so it was based on the System Usability Scale (SUS), which is the most used questionnaire for measuring perceptions of usability, being technology independent. The SUS is used for measuring the global usability of the platform. Since the questionnaire is provided in Portuguese, this section is based on the Portuguese version of the SUS [99]. Therefore, we applied the 10-item questionnaire with seven response options instead of the usual five options (thus using a seven-point Likert-type scale, 1 – strongly disagree; 7 – strongly agree). Participants should complete the questionnaire right after having used the prototype and before any debriefing or another discussion, following the advice of Brooke [100]. The fourth section focuses on assessing the relevance of the conceptual model and platform existing functionality for meeting therapeutic requirements.

Net Promoter Score assesses the probability of therapists recommending the framework to their peers. Finally, additional feedback can be entered allowing therapists to express whatever they wish and feel relevant about the experience.

The questionnaire is available as Appendix I (Questionnaire B).

Chapter 6

Conclusions

In this chapter the conclusions of the research are presented and discussed. Relevant results were achieved and are also described. The current state of the platform and future work are evaluated.

6.1. Research

A sustainable world is a world of inclusion. There is urgent need of tools that help make that goal a reality. The aim of this work is to provide tools to promote a more sustainable world. Social sustainability can only be achieved by providing tools for all and not only for those who can adapt to it. This is a challenging research topic but very rewarding.

It was not easy to setup the goals for this research. Multiple and interconnected research areas make this an interdisciplinary project that requires a global overview and broader understanding of the framework components relations.

This research tried to respond to a set of research questions.

RQ1: *Can this framework provide a more intuitive tool, abstracting technology requirements, allowing therapists to design serious games, by using a specific modeling language?*

This research provides a model for the design of serious games focusing on children with special needs. A platform for clinics, therapists and patients was implemented. The platform allows therapists to design personalized serious games addressing specific patient's needs. External developers can integrate their applications with PLAY. The REST API is tool allowing the design and execution of serious games by external applications.

A DSL provides a more natural language for domain experts usage of the PLAY technology. Using the visual tools provided, therapists and curators can design games by inserting and configuring the composing actions. Future work includes the continuity of research and improvement of a DSL, including update of metamodel and semantics as needed, and research for more appealing and efficient GML notations for serious game design.

For RQ1 (and other RQs) to be fully answered, validation with users is needed. Tests and inquiries with therapists are designed and planned to happen and will help assess the validity and usability of the proposed solutions.

RQ2: *Does the model of action and parameters flexible enough to allow representation of therapeutic exercises?*

The action and parameters model provides a tool for the platform to adapt and evolve by allowing the design of distinct therapeutic exercises. Exercises have been designed using the proposed model for use in the platform and with external applications. Modeling more and distinct types of exercises will test the limits of this system. Anyway, the action model was successfully used to design therapeutic exercises so the answer to RQ2 is yes.

RQ3: *Can this framework help therapists design serious games for children with special needs, by suggesting relevant actions, based on the patient profile?*

The currently implemented suggestion system uses a very simple algorithm but can suggest the relevant actions for a game and relevant games for a specific patient. Future work is needed for designing a more robust algorithm and possibly integrating ML tools for a more efficient suggestion based on all relevant available information. The answer to RQ3 requires tests with therapists.

RQ4: *With this platform, can therapists benefit from being able to compare results and exercise samples between patients?*

The ability to compare results and samples is always theoretically beneficial. The question is not that simple. What is meant is if relevant knowledge can be extracted by using the provided tools. This question has not yet been answered as the tools are not fully implemented. Relevant data analysis methodologies and processes are currently subject of research and should be subject to future work. Tests will be needed for assessing the answer to RQ4.

RQ5: *Can developers allow their applications to integrate the platform by consuming an API?*

The PLAY API is used by the internal platform components and games. It has been successfully used for integrating the external JPK and ERG games. This leads to conclude that the API is allowing applications to integrate the platform, answering affirmatively to RQ5. The integration with more and more diverse applications will provide more certainty. The API will be updated if needed.

6.2. Results

This research produced a set of relevant results worth mentioning.

The framework model is conceptualized, providing:

- A model for a patient profile.
- A model for games.
- Abstraction of devices.

A platform prototype is implemented, providing:

- Repository of game executions, results, and samples.
- Data source for machine learning algorithms.
- Tools for obtaining knowledge.
- Easy access to data and tools using the Internet.
- A first iteration in the creation process of a recommender system for games based on the user profile.

A study with patients was made, at the clinic, with children using the platform, and results have been presented and analyzed. A full paper has been submitted to an international conference for dissemination of this work and those results. However, the number of interventions for each patient was very short and, since this is a work-in-progress, we need to conduct further tests to obtain enough data for a solid statistical analysis. Once the final form of the platform is completed, a high-quality clinical trial will be needed.

6.3. Future work

The framework and platform are stable as starting point for continuous progress. The very nature of its composition is prepared to easily adapt and grow as needed. The model can be refined and improved. The platform will integrate new functionalities and update the existing tools as needed.

Possible platform improvements include modelling and integration of new action types in the system allowing a wider range of possible interventions.

Increase the hardware support for new sensors and actuation devices. The inclusion of new sensors will provide extended possibilities of interaction during games and exercises.

Increase the data analytics tools in the platform for better analysis capabilities.

Research and implementation of machine learning algorithms for further optimization of adaptation and personalization.

The game model provides a flexible tool for game design, allowing the creation of new games focusing distinct therapies, for a more inclusive therapeutic response.

The API is a valuable tool for integration with other applications and platforms and could be further explored and expanded.

The development of a set of domain-specific modelling languages (DSML) is already being researched.

Further testing the framework with therapists, curators, and patients is needed as mean for assessment of research validity.

Bibliography

[1] WHO, *International classification of functioning, disability and health : children & youth version : ICF-CY*. World Health Organization, 2007.

[2] J. Guerrero-Garcia, J. M. González-Calleros, J. Muñoz-Arteaga, and C. A. Collazos, Eds., *HCI for Children with Disabilities*. Cham: Springer International Publishing, 2017.

[3] P. Rosenbaum *et al.*, "A report: The definition and classification of cerebral palsy April 2006," *Dev. Med. Child Neurol.*, 2007, doi: 10.1111/j.1469-8749.2007.tb12610.x.

[4] J. C. Read and M. M. Bekker, "The Nature of Child Computer Interaction," *BCS-HCI '11 Proc. 25th BCS Conf. Human-Computer Interact.*, no. 1994, pp. 163–170, 2011.

[5] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: Defining 'gamification,'" *Proc. 15th Int. Acad. MindTrek Conf. Envisioning Futur. Media Environ. MindTrek 2011*, pp. 9–15, 2011, doi: 10.1145/2181037.2181040.

[6] B. Bonnechère, *Serious Games in Physical Rehabilitation*. 2018.

[7] A. Chen, R. Muntz, and M. Srivastava, "Smart Rooms," in *Smart Environments*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005, pp. 295–322.

[8] J. Piaget, "Part I: Cognitive development in children: Piaget development and learning," *J. Res. Sci. Teach.*, vol. 2, no. 3, pp. 176–186, 1964, doi: 10.1002/tea.3660020306.

[9] L. D. Rodić and A. Granić, "Tangible interfaces in early years' education: a systematic review," *Pers. Ubiquitous Comput.*, May 2021, doi: 10.1007/s00779-021-01556-x.

[10] GPO, "eCFR — Code of Federal Regulations," *Government Publishing Office (US)*, 2020. https://www.ecfr.gov.

[11] W. J. Little, *On the Nature and Treatment of the Deformities of the Human Frame*, no. 1. 1853.

[12] V. Pavone and G. Testa, "Classifications of Cerebral Palsy," in *Orthopedic Management of Children with Cerebral Palsy*, Nova Science Publishers, Inc., 2015.

[13] R. J. Palisano, P. Rosenbaum, D. Bartlett, and M. H. Livingston, "Content validity of the expanded and revised Gross Motor Function Classification System," *Dev. Med. Child Neurol.*, vol. 50, no. 10, pp. 744–750, 2008, doi: 10.1111/j.1469-8749.2008.03089.x.

[14] L. Krumlinde-sundholm, "Mini-Manual Ability Classification System for children with cerebral palsy 1 - 4 years of age."

[15] S. E. Fife, L. A. Roxborough, R. W. Armstrong, S. R. Harris, J. L. Gregson, and D. Field, "Development of a clinical measure of postural control for assessment of adaptive seating in children with neuromotor disabilities," *Phys. Ther.*, vol. 71, no. 12, pp. 981–993, 1991,

doi: 10.1093/ptj/71.12.981.

[16] D. A. Field and L. A. Roxborough, “Responsiveness of the seated postural control measure and the level of sitting scale in children with neuromotor disorders,” *Disabil. Rehabil. Assist. Technol.*, vol. 6, no. 6, pp. 473–482, 2011, doi: 10.3109/17483107.2010.532285.

[17] A. Da Gama, P. Fallavollita, V. Teichrieb, and N. Navab, “Motor Rehabilitation Using Kinect: A Systematic Review,” *Games Health J.*, vol. 4, no. 2, pp. 123–135, Apr. 2015, doi: 10.1089/g4h.2014.0047.

[18] S. Paul and D. Ramsey, “Music therapy in physical medicine and rehabilitation,” *Aust. Occup. Ther. J.*, vol. 47, no. 3, pp. 111–118, 2000, doi: 10.1046/j.1440-1630.2000.00215.x.

[19] R. T. Hadley, W. H. Hadley, V. Dickens, and E. G. Jordon, “Music therapy: A treatment modality for special-needs populations,” *Int. J. Adv. Couns.*, vol. 23, no. 3, pp. 215–221, 2001, doi: 10.1023/A:1013107909160.

[20] K. Shikako-Thomas, A. Majnemer, M. Law, and L. Lach, “Determinants of participation in leisure activities in children and youth with cerebral palsy: Systematic review,” *Phys. Occup. Ther. Pediatr.*, vol. 28, no. 2, pp. 155–169, 2008, doi: 10.1080/01942630802031834.

[21] R. A. Georges, E. M. Avedon, and B. Sutton-Smith, “The Study of Games,” *West. Folk.*, vol. 34, no. 2, p. 155, Apr. 1975, doi: 10.2307/1499097.

[22] K. Salen and E. Zimmerman, *Rules of play : game design fundamentals*. 2004.

[23] K. Huotari, “Defining Gamification - A Service Marketing Perspective,” pp. 17–22, 2012.

[24] K. Werbach, “(Re)Defining Gamification: A Process Approach,” in *Persuasive Technology*, 2014, pp. 266–272.

[25] R. De Croon, D. Wildemeersch, J. Wille, K. Verbert, and V. Vanden Abeele, “Gamification and serious games in a healthcare informatics context,” *Proc. - 2018 IEEE Int. Conf. Healthc. Informatics, ICHI 2018*, no. 110067, pp. 53–63, 2018, doi: 10.1109/ICHI.2018.00014.

[26] J. Hamari, J. Koivisto, and H. Sarsa, “Does gamification work? - A literature review of empirical studies on gamification,” *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, pp. 3025–3034, 2014, doi: 10.1109/HICSS.2014.377.

[27] U. Ritterfeld, M. Cody, and P. Vorderer, *Serious Games: Mechanisms and Effects*, 1st ed. 2009.

[28] D. Archambault and J. Dupire, “Digital games accessibility: Introduction to the special thematic session,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, vol. 10896

LNCS, pp. 241–244, doi: 10.1007/978-3-319-94277-3_39.

[29] D. Sharma, M. Favorskaya, L. C. Jain, and R. J. Howlett, *Fusion of Smart, Multimedia and Computer Gaming Technologies*. Springer, 2015.

[30] L. Omelina, B. Jansen, B. Bonnechère, S. Van, S. Jan, and J. Cornelis, *Serious games for physical rehabilitation: designing highly configurable and adaptable games*. 2012.

[31] B. E. Holtz, K. Murray, and T. Park, "Serious Games for Children with Chronic Diseases: A Systematic Review," *Games Health J.*, vol. 7, no. 5, pp. 291–301, Oct. 2018, doi: 10.1089/g4h.2018.0024.

[32] R. Hunicke, M. LeBlanc, and R. Zubek, "MDA: A Formal Approach to Game Design and Game Research," *Comput. Sci.*, 2004.

[33] G. Calleja, "Revising immersion: A conceptual model for the analysis of digital game involvement," *3rd Digit. Games Res. Assoc. Int. Conf. "Situated Play. DiGRA 2007*, pp. 83–90, 2007.

[34] S. Tang and M. Hanneghan, "Game content model: An ontology for documenting serious game design," *Proc. - 4th Int. Conf. Dev. eSystems Eng. DeSE 2011*, pp. 431–436, 2011, doi: 10.1109/DeSE.2011.68.

[35] C. A. Aguilar-lazcano, E. J. Rechy-ramirez, H. Hu, H. V. Rios-figueroa, and A. Marin-hernandez, "Interaction Modalities Used on Serious Games for Upper Limb Rehabilitation : A Systematic Review," vol. 8, no. 5, 2019, doi: 10.1089/g4h.2018.0129.

[36] M. V Valenza, I. Gasparini, and M. S. Hounsell, "Serious Game Design for Children : A Set of Guidelines and their Validation," in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, 2019, vol. 22, pp. 19–31, doi: 10.1109/ICALT.2019.00034.

[37] M. Henschke, D. Hobbs, and B. Wilkinson, "Developing serious games for children with cerebral palsy: Case study and pilot trial," *Proc. 24th Aust. Comput. Interact. Conf. OzCHI 2012*, pp. 212–221, 2012, doi: 10.1145/2414536.2414574.

[38] M. Weiser, "The Computer for the 21st Century," *Scientific American*, 1991.

[39] J. Nielsen, "Usability inspection methods," *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 1994-April, pp. 413–414, 1994, doi: 10.1145/259963.260531.

[40] Schneiderman, Plaisant, Cohen, Jacobs, and Elmqvist, *Designing the User Interface*, 6th ed. Pearson, 2016.

[41] A. Nijholt, *More Playful User Interfaces. Interfaces that Invite Social and Physical Interaction*, vol. II. 2015.

[42] P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays," *IEICE Trans. Inf. Syst.*, vol. E77-D, pp. 1321–1329, 1994.

[43] Y. Chen, H. C. D. Fanchiang, and A. Howard, “Effectiveness of virtual reality in children with cerebral palsy: A systematic review and meta-analysis of randomized controlled trials,” *Physical Therapy*, vol. 98, no. 1. Oxford University Press, pp. 63–77, Jan. 2018, doi: 10.1093/ptj/pzx107.

[44] D. Cook and S. K. Das, Eds., *Smart Environments: Technologies, Protocols and Applications*. 2005.

[45] D. Wigdor and D. Wixon, *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann, 2011.

[46] K. Khoshelham, “Accuracy Analysis of Kinect Depth Data,” *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XXXVIII-5/, no. August, pp. 133–138, 2012, doi: 10.5194/isprsarchives-xxxviii-5-w12-133-2011.

[47] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, “Evaluation of pose tracking accuracy in the first and second generations of microsoft Kinect,” *Proc. - 2015 IEEE Int. Conf. Healthc. Informatics, ICHI 2015*, pp. 380–389, 2015, doi: 10.1109/ICHI.2015.54.

[48] R. Lai, “Dexmo exoskeleton glove lets you touch and feel in VR,” *Endgadget*, 2016. .

[49] J. Rigg, “Teslasuit does full-body haptic feedback for VR,” *Endgadget*, 2016. .

[50] A. Robertson, “Teslasuit’s new VR gloves let you feel virtual objects and track your pulse,” *The Verge*, 2019. .

[51] A. Suh and J. Prophet, “The state of immersive technology research: A literature analysis,” *Comput. Human Behav.*, vol. 86, 2018, doi: 10.1016/j.chb.2018.04.019.

[52] A. Venkatakrishnan, G. E. Francisco, and J. L. Contreras-Vidal, “Applications of Brain–Machine Interface Systems in Stroke Recovery and Rehabilitation,” *Curr. Phys. Med. Rehabil. Reports*, vol. 2, no. 2, pp. 93–105, Jun. 2014, doi: 10.1007/s40141-014-0051-4.

[53] A. Azazi, T. Seyed, F. Botros, and D. Sabourin, “Using Multiple Kinects to Build Larger Multi-Surface Environments,” *Proc. Work. Collab. Meets Interact. Surfaces, ITS*, pp. 1–8, 2013.

[54] H. Jin, Q. Chen, Z. Chen, Y. Hu, and J. Zhang, “Multi-LeapMotion sensor based demonstration for robotic refine tabletop object manipulation task,” *CAAI Trans. Intell. Technol.*, vol. 1, no. 1, pp. 104–113, 2016, doi: 10.1016/j.trit.2016.03.010.

[55] B. Penelle and O. Debeir, “Multi-sensor data fusion for hand tracking using Kinect and Leap Motion,” in *Proceedings of the 2014 Virtual Reality International Conference on - VRIC ’14*, 2014, no. April, pp. 1–7, doi: 10.1145/2617841.2620710.

[56] A. Hadjidj, M. Souil, A. Bouabdallah, Y. Challal, and H. Owen, “Wireless sensor networks for rehabilitation applications: Challenges and opportunities,” *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 1–15, 2013, doi: 10.1016/j.jnca.2012.10.002.

[57] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–

444, 2015, doi: 10.1038/nature14539.

[58] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," *Brief. Bioinform.*, vol. 19, no. 6, pp. 1236–1246, 2017, doi: 10.1093/bib/bbx044.

[59] B. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.

[60] M. Mernik, "Domain-specific languages: A systematic mapping study," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10139 LNCS, pp. 464–472, 2017, doi: 10.1007/978-3-319-51963-0_36.

[61] B. E. Holtz, K. Murray, and T. Park, "Serious Games for Children with Chronic Diseases: A Systematic Review," *Games Health J.*, vol. 7, no. 5, pp. 291–301, 2018, doi: 10.1089/g4h.2018.0024.

[62] A. Suh and J. Prophet, "The state of immersive technology research: A literature analysis," *Comput. Human Behav.*, vol. 86, pp. 77–90, 2018, doi: 10.1016/j.chb.2018.04.019.

[63] J. P. S. Da Silva, M. Ecar, M. S. Pimenta, G. T. A. Guedes, L. P. Franz, and L. Marchezan, "A systematic literature review of UML-based domain-specific modeling languages for self-adaptive systems," *Proc. - Int. Conf. Softw. Eng.*, pp. 87–93, 2018, doi: 10.1145/3194133.3194136.

[64] R. Van Rozen, "Languages of Games and Play: A Systematic Mapping Study," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–38, 2021, doi: 10.1145/3412843.

[65] A. Wortmann, O. Barais, B. Combemale, and M. Wimmer, "Modeling languages in Industry 4.0: an extended systematic mapping study," *Softw. Syst. Model.*, vol. 19, no. 1, pp. 67–94, 2020, doi: 10.1007/s10270-019-00757-6.

[66] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.*, vol. 37, no. 4, pp. 316–344, 2005, doi: 10.1145/1118890.1118892.

[67] D. Avila-Pesantez, R. Delgadillo, and L. A. Rivera, "Proposal of a Conceptual Model for Serious Games Design: A Case Study in Children with Learning Disabilities," *IEEE Access*, vol. 7, pp. 161017–161033, 2019, doi: 10.1109/ACCESS.2019.2951380.

[68] R. P. De Lope, N. Medina-Medina, M. Urbieta, A. B. Lliteras, and A. Mora García, "A novel UML-based methodology for modeling adventure-based educational games," *Entertain. Comput.*, vol. 38, no. April, p. 100429, 2021, doi: 10.1016/j.entcom.2021.100429.

[69] K. Khowaja and S. S. Salim, *A framework to design vocabulary-based serious games for children with autism spectrum disorder (ASD)*, vol. 19, no. 4. Springer Berlin Heidelberg, 2020.

[70] M. W. Aziz and M. Rashid, "Domain specific modeling language for cyber physical

systems," *Proc. - 2016 Int. Conf. Inf. Syst. Eng. ICISE 2016*, no. 1, pp. 29–33, 2016, doi: 10.1109/ICISE.2016.12.

[71] C. González García, E. R. Núñez-Valdez, P. Moreno-Ger, R. González Crespo, B. C. Pelayo G-Bustelo, and J. M. Cueva Lovelle, "Agile development of multiplatform educational video games using a Domain-Specific Language," *Univers. Access Inf. Soc.*, vol. 18, no. 3, pp. 599–614, 2019, doi: 10.1007/s10209-019-00681-y.

[72] F. E. Garcia and V. P. de Almeida Neris, "A framework for tailorable games: toward inclusive end-user development of inclusive games," *Univers. Access Inf. Soc.*, Nov. 2020, doi: 10.1007/s10209-020-00779-8.

[73] V. M. Peñeñory, C. A. Collazos, Á. F. Bacca, C. Manresa-Yee, S. P. Cano, and H. M. Fadoun, "APRehab: a methodology for serious games design oriented to psychomotor rehabilitation in children with hearing impairments," *Univers. Access Inf. Soc.*, vol. 20, no. 2, pp. 255–264, 2020, doi: 10.1007/s10209-020-00728-5.

[74] A. Gómez, M. Iglesias-Urkia, A. Urbieta, and J. Cabot, "A model-based approach for developing event-driven architectures with AsyncAPI," *Proc. - 23rd ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Model. 2020*, pp. 121–131, 2020, doi: 10.1145/3365438.3410948.

[75] C. Guan, Y. Ao, D. Du, and F. Mallet, "XSHS: An Executable Domain-Specific Modeling Language for Modeling Stochastic and Hybrid Behaviors of Cyber-Physical Systems," *Proc. - Asia-Pacific Softw. Eng. Conf. APSEC*, vol. 2018-Decem, pp. 683–687, 2018, doi: 10.1109/APSEC.2018.00090.

[76] J. P. S. Da Silva, M. Ecar, M. S. Pimenta, G. T. A. Guedes, and E. M. Rodrigues, "Towards a domain-specific modeling language for self-adaptive systems conceptual modeling," *ACM Int. Conf. Proceeding Ser.*, pp. 208–213, 2018, doi: 10.1145/3266237.3266244.

[77] S. Karim and A. M. Tjoa, "Towards the use of ontologies for improving user interaction for people with special needs," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4061 LNCS, pp. 77–84, 2006, doi: 10.1007/11788713_12.

[78] A. Pescador, A. Garmendia, E. Guerra, J. S. Sánchez Cuadrado, and J. De Lara, "Pattern-based development of Domain-Specific Modelling Languages," *2015 ACM/IEEE 18th Int. Conf. Model Driven Eng. Lang. Syst. Model. 2015 - Proc.*, pp. 166–175, 2015, doi: 10.1109/MODELS.2015.7338247.

[79] S. Tang and M. Hanneghan, "Towards a Domain Specific Modelling Language for Serious Game Design," *6th Int. Game Des. Technol. Work.*, pp. 43–52, 2008.

[80] S. P. Jacome-Guerrero, M. Ferreira, and A. Corral, "Software Development Tools in Model-Driven Engineering," *Proc. - 2017 5th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2017*, vol. 2018-Janua, pp. 140–148, 2018, doi: 10.1109/CONISOFT.2017.00024.

[81] D. Tarakci, B. Ersoz Huseyinsinoglu, E. Tarakci, and A. Razak Ozdincler, “Effects of Nintendo Wii-Fit® video games on balance in children with mild cerebral palsy,” *Pediatr. Int.*, vol. 58, no. 10, pp. 1042–1050, 2016, doi: 10.1111/ped.12942.

[82] R. N. Madeira, A. Antunes, O. Postolache, and N. Correia, *Serious…ly! just kidding in personalised therapy through natural interactions with games*, vol. 10714 LNCS. 2018.

[83] R. N. Madeira, A. Antunes, and O. Postolache, *Just physio kidding - NUI and gamification based therapeutic intervention for children with special needs*, vol. 207. 2018.

[84] V. Wattanasoontorn, R. J. G. Hernández, and M. Sbert, “Serious Games for e-Health Care,” in *Simulations, Serious Games and Their Applications*, 2014, pp. 127–146.

[85] C. S. González-González, P. A. Toledo-Delgado, V. Muñoz-Cruz, and P. V. Torres-Carrion, “Serious games for rehabilitation: Gestural interaction in personalized gamified exercises through a recommender system,” *J. Biomed. Inform.*, vol. 97, 2019, doi: 10.1016/j.jbi.2019.103266.

[86] E. Oliveira, G. Sousa, T. A. Tavares, and P. Tanner, “Sensory stimuli in gaming interaction: The potential of games in the intervention for children with cerebral palsy,” 2014, doi: 10.1109/GEM.2014.7243758.

[87] A. P. H. Weightman, N. Preston, R. Holt, M. Allsop, M. Levesley, and B. Bhakta, “Engaging children in healthcare technology design: developing rehabilitation technology for children with cerebral palsy,” *J. Eng. Des.*, vol. 21, no. 5, pp. 579–600, Oct. 2010, doi: 10.1080/09544820802441092.

[88] C. Zirbel, X. Zhang, and C. Hughes, “The VRehab System: A Low-Cost Mobile Virtual Reality System for Post-Stroke Upper Limb Rehabilitation for Medically Underserved Populations,” *GHTC 2018 - IEEE Glob. Humanit. Technol. Conf. Proc.*, no. October 2018, pp. 1–8, 2019, doi: 10.1109/GHTC.2018.8601885.

[89] W. R. Kemble, “A proposed model for designing children’s health-focused serious games,” *ProQuest Diss. Theses*, p. 160, 2014.

[90] E. Amengual Alcover, A. Jaume-I-Capó, and B. Moyà-Alcover, “PROGame: A process framework for serious game development for motor rehabilitation therapy,” *PLoS One*, vol. 13, no. 5, pp. 1–19, 2018, doi: 10.1371/journal.pone.0197383.

[91] B. Bonnechère *et al.*, “Can serious games be incorporated with conventional treatment of children with cerebral palsy? A review,” *Res. Dev. Disabil.*, vol. 35, no. 8, pp. 1899–1913, Aug. 2014, doi: 10.1016/j.ridd.2014.04.016.

[92] G. Caggianese *et al.*, “Serious Games and In-Cloud Data Analytics for the Virtualization and Personalization of Rehabilitation Treatments,” *IEEE Trans. Ind. Informatics*, vol. 15, no. 1, pp. 517–526, 2019, doi: 10.1109/TII.2018.2856097.

[93] S. Tang and M. Hanneghan, “A model-driven framework to support development of

serious games for game-based learning," *Proc. - 3rd Int. Conf. Dev. eSystems Eng. DeSE 2010*, pp. 95–100, 2010, doi: 10.1109/DeSE.2010.23.

[94] D. Ferreira, R. Oliveira, and O. Postolache, "Physical rehabilitation based on kinect serious games," *Proc. Int. Conf. Sens. Technol. ICST*, vol. 2017-Decem, pp. 1–6, 2018, doi: 10.1109/ICSensT.2017.8304512.

[95] P. Kosmides, E. Adamopoulou, N. Koutsouris, and V. De Luca, "InLife : Combining Real Life with Serious Games using IoT," 2018, doi: 10.1109/CIG.2018.8490434.

[96] R. N. Madeira, A. Antunes, and O. Postolache, "just Physio kidding - NUI and Gamification based Therapeutic Intervention for Children with Special Needs," 2018, pp. 56–61.

[97] R. N. Madeira, P. A. Santos, and N. Correia, "Building a platform for pervasive personalization in a ubiquitous computing world," *MobiQuitous 2014 - 11th Int. Conf. Mob. Ubiquitous Syst. Comput. Netw. Serv.*, no. December, pp. 345–346, 2014, doi: 10.4108/icst.mobiquitous.2014.258158.

[98] I. Tóth and A. C. Bernardo, "A preliminary study of the application of just Physio kidding software in the pediatric rehabilitation . A case series," 2018.

[99] A. Isabel, A. Filipa, A. Queirós, and A. Silva, "European Portuguese validation of the System Usability Scale (SUS)," *Procedia - Procedia Comput. Sci.*, vol. 67, no. Dsai, pp. 293–300, 2015, doi: 10.1016/j.procs.2015.09.273.

[100] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," in *Usability Evaluation In Industry*, CRC Press, 1996, pp. 207–212.

Appendix I

Questionnaires

This appendix contains questionnaires provided for therapists and used on information retrieval.

QUESTIONNAIRE A

Questionário Clínica Cresce

Identificação de necessidades

Tendo como referência o sistema JPK, que apresenta a possibilidade de monitorizar certos movimentos durante a execução dos exercícios, responda às seguintes questões.

1. Existe algum movimento ou movimentos que não façam sentido? Quais?
2. Que novos movimentos deveriam ser incluídos na plataforma?
3. Que tipo de métricas, como por exemplo distância ou ângulo, estão associadas aos novos movimentos identificados?
4. Como se pode avaliar a qualidade da execução desses movimentos?
5. Relativamente à mecânica de jogo, que neste momento consiste em alcançar e arrastar objectos, o que acha que poderia ser melhorado ou acrescentado? Consegue identificar alguma outra mecânica de jogo que fosse aplicável neste contexto?
6. Como poderia o reconhecimento de voz existente na plataforma ser utilizado em contexto de terapia da fala?
7. Que aspectos de musicoterapia poderiam ser incluídos na aplicação? Como poderiam as músicas e elementos sonoros da aplicação ser melhorados nesse sentido?
8. Que tipo de objectivos de jogo fariam sentido no contexto dos exercícios a realizar e dos pacientes envolvidos?
9. Que funcionalidade acha que deveria ser incluída na plataforma, de forma a permitir uma mais eficaz utilização por parte de pacientes e terapeutas?
10. A visualização dos resultados e amostras do movimento são extremamente importantes para transmitir a percepção da execução do exercício. Como se poderia melhorar a visualização dos resultados e das amostras do movimento de forma a tornar mais eficaz a transmissão da informação relevante ao terapeuta?

Obrigado.

QUESTIONNAIRE B

INTRODUÇÃO

Estamos interessados em saber a sua opinião sobre a sua experiência com a framework PLAY, que é uma aplicação Web que permite criar e gerir intervenções terapêuticas utilizando jogos sérios para crianças com necessidades especiais. A informação que recolhemos sobre si é anónima e será apenas utilizada para o nosso uso interno no âmbito da nossa investigação. Não será partilhada com terceiros.

IDENTIFICAÇÃO

- Qual é a sua idade (insira um número em anos)?
- Com que género é que se identifica?
- Qual é a sua área profissional?
- Qual a sua área de especialização?
- Qual a população com quem tem trabalhado nos últimos 5 anos?
- Descrever a população....
- Normalmente, utiliza jogos (p.e., plataforma Wii, em tablet) em complemento à intervenção?
- Caso tenha respondido afirmativamente à questão anterior, são jogos sérios desenvolvidos especificamente para o objetivo da intervenção?

PLATAFORMA

- Foi fácil selecionar pacientes.
- Foi fácil adicionar, alterar ou remover pacientes na aplicação.
- Foi fácil selecionar jogos na aplicação.
- Foi fácil adicionar, alterar ou remover jogos na aplicação.
- Foi fácil visualizar a estrutura do jogo na aplicação.

USABILIDADE DE PLAY

- Acho que gostaria de utilizar a framework PLAY com frequência.
- Considerei a framework PLAY mais complexa do que necessário.
- Achei a framework PLAY fácil de utilizar.
- Acho que necessitaria de ajuda de um técnico para conseguir utilizar a framework PLAY.
- Considerei que as várias funcionalidades da framework PLAY estavam bem integradas.
- Achei que a framework PLAY tinha muitas inconsistências.
- Suponho que a maioria das pessoas aprenderia a utilizar rapidamente a framework PLAY.
- Considerei a framework PLAY muito complicada de utilizar.
- Senti-me muito confiante a utilizar a framework PLAY.

- Tive que aprender muito antes de conseguir lidar com a framework PLAY.

CONCEITO

- Acho interessante o modelo de jogo baseado em níveis, sequências e acções.
- Acho interessante a possibilidade de agregar resultados e amostras de outras aplicações de jogos sérios na framework PLAY.
- Acho interessante que a plataforma PLAY possa vir a recomendar jogos para determinadas intervenções com base no histórico e perfil do paciente

NET PROMOTER SCORE

- Qual é a probabilidade de recomendar a framework PLAY a um colega?

FEEDBACK

- Tem algum feedback adicional sobre a sua experiência com a framework PLAY?

Appendix II
JPK API

The platform implementation of the original JPK API. This API uses Stored Procedures.

/admin/		
Verb	**Endpoint**	**Stored Procedure**
GET	getHistory	Admin_GetHistory
GET	getLog	Admin_GetLog
GET	getMovements	Admin_GetMovements
GET	getPatients	Admin_GetPatients
GET	getPhysiotherapists	Admin_GetPhysiotherapists
GET	getUsers	Admin_GetUsers
POST	editMovement	Admin_EditMovement
POST	editUser	Admin_EditUser
POST	insertMovement	Admin_InsertMovement
POST	removeMovement/:movId	Admin_RemoveMovement
POST	removePatient/:patId	Admin_RemovePatient
POST	removePhysiotherapist/:physioId	Admin_RemovePhysiotherapist

/physio/		
Verb	**Endpoint**	**Stored Procedure**
GET	exportKinectSamples/:patId/:execId	Phys_ExportKinectSamples
GET	exportLeapSamples/:patId/:execId	Phys_ExportLeapSamples
GET	exportProgression/:patId	Phys_ExportProgression
GET	exportShimmerSamples/:patId/:execId	Phys_ExportShimmerSamples
GET	getBadges/:userId	Phys_GetBadges
GET	getChallenges	Phys_GetChallenges
GET	getGameConfiguration/:gameId	Phys_GetGameConfiguration
GET	getGameExecutions	Phys_GetGameExecutions
GET	getGameLevels/:gameId	Phys_GetGameLevels
GET	getGameMovement/:gameMovId	Phys_GetGameMovement
GET	getGameMovements/:gameId	Phys_GetGameMovements
GET	getGames	Phys_GetGames
GET	getMovements	Phys_GetMovements
GET	getPatientChallenges/:patientId	Phys_GetPatientChallenges
GET	getPatientGame/:patientId	Phys_GetPatientGame
GET	getPatientRoutines/:patientId	Phys_GetPatientRoutines
GET	getPatientSchedule/:patientId	Phys_GetPatientSchedule
GET	getPhysiotherapistNewPatients/:physioId	Phys_GetPhysiotherapistNewPatients
GET	getPhysiotherapistPatients/:physioId	Phys_GetPhysiotherapistPatients
GET	getResultKinectSamples/:resultId	Phys_GetResultKinectSamples
GET	getResultLeapImage/:resultId	Phys_GetResultLeapImage

GET	getResultLeapPoints/:resultId/:pointType	Phys_GetResultLeapPoints
GET	getResultLeapSample/:resultId	Phys_GetResultLeapSample
GET	getResults/:gameExecId	Phys_GetResults
GET	getRoutines	Phys_GetRoutines
GET	getShimmerSample/:resultId	Phys_GetShimmerSample
POST	editConfiguration	Phys_EditConfiguration
PUT	editGame	Phys_EditGame
POST	editGameExecution	Phys_EditGameExecution
POST	editGameMovement	Phys_EditGameMovement
POST	editPatientProfile	Phys_EditPatientProfile
POST	insertConfiguration	Phys_InsertConfiguration
POST	insertExecution	Phys_InsertExecution
POST	insertGame	Phys_InsertGame
POST	insertGameMovement	Phys_InsertGameMovement
POST	removeExecution/:gameExecId	Phys_RemoveExecution
POST	removeGameMovement/:gameId	Phys_RemoveGameMovement
POST	removePhysiotherapistPatient/:physioId/:patientId	Phys_RemovePhysiotherapistPatient
POST	setPhysiotherapistPatient/:physioId/:patientId	Phys_SetPhysiotherapistPatient

/patient/		
Verb	**Endpoint**	**Stored Procedure**
GET	getMovementSensors/:movId	Patient_GetMovementSensors
GET	getPatientRanking/:gameId	Patient_GetPatientRanking
POST	insertBadge	Patient_InsertBadge
POST	insertExecution	Patient_InsertExecution
POST	insertKinectSamples	Patient_InsertKinectSamples
POST	insertLeapImage	Patient_InsertLeapImage
POST	insertLeapPoints	Patient_InsertLeapPoints
POST	insertLeapSamples	Patient_InsertLeapSamples
POST	insertPersonalization	Patient_InsertPersonalization
POST	insertResult	Patient_InsertResult
POST	insertResultExtended	Patient_InsertResultExtended
POST	insertSample	Patient_InsertSample
POST	insertSampleBone	Patient_InsertSampleBone
POST	insertSampleJoint	Patient_InsertSampleJoint
POST	insertSessionEntry	Patient_InsertSessionEntry
POST	insertShimmerSamples	Patient_InsertShimmerSamples
POST	performGame/:gameId/:date	Patient_PerformGame
POST	updatePersonalization/:userId/:name/:value	Patient_UpdatePersonalization
POST	updateResult	Patient_UpdateResult

/user/		
Verb	**Endpoint**	**Stored Procedure**
GET	getBadge/:badgeId	User_GetBadge
GET	getFace/:userId	User_GetFace
GET	getFaceImage/:userId	User_GetFaceImage
GET	getPatient/:userId	User_GetPatient

GET	getPersona/:userId	User_GetPersona
GET	getPersonalizations/:userId	User_GetPersonalizations
GET	getPhysiotherapist/:email	User_GetPhysiotherapist
GET	login/:pin	User_Login
GET	logout/:pin	User_Logout
POST	createPhysiotherapist/:email	User_CreatePhysiotherapist
POST	insertFaceImage	User_InsertFaceImage
POST	insertFacePoint	User_InsertFacePoint
POST	insertPhoto/:filename	User_InsertPhoto
POST	loginHistory/:userId	User_LoginHistory
POST	logoutHistory/:historyId	User_LogoutHistory
POST	registration/:armSize/:torsoSize	User_Registration
POST	updatePhoto/:pin/:photo	User_UpdatePhoto

Appendix III

PLAY API

Element	Fields	Datatype
action	id	integer
	ordering	integer
	timelimit	integer
	sideid	integer
	actionid	integer
	sequenceid	integer
	levelid	integer
	gameid	integer

HTTP Verb	Endpoint	Description
POST	action	insert a new action
PUT	action	update existing action
DELETE	action	delete existing action
GET	action	get all actions
GET	action/count	get the number of total actions
GET	action/:id	get a specific action

Element	Fields	Datatype
badge	id	integer
	name	string
	info	string
	criteriaid	integer
	operatorid	integer
	criteriavalue	integer
	ui	integer
	gameid	integer
	userid	integer

HTTP Verb	Endpoint	Description
POST	badge	insert a new badge
PUT	badge	update existing badge
DELETE	badge	delete existing badge

GET	badge	get all badges
GET	badge/count	get the number of total badges
GET	badge/:id	get a specific badge

Element	Fields	Datatype
device	id	integer
	name	string
	info	string
	roleid	integer

HTTP Verb	Endpoint	Description
POST	device	insert a new device
PUT	device	update existing device
DELETE	device	delete existing device
GET	device	get all devices
GET	device/count	get the number of total devices
GET	device/:id	get a specific device
GET	device/:id/joint	get device joints
GET	device/:id/capability	get device capabilities
GET	device/:id/action	get device-related actions

Element	Fields	Datatype
game	id	integer
	name	string
	info	string
	created	datetime
	updated	datetime
	removed	datetime
	userid	integer

HTTP Verb	Endpoint	Description
POST	game	insert a new game
PUT	game	update existing game
DELETE	game	delete existing game
GET	game	get all games
GET	game/count	get the number of total games
GET	game/:id	get a specific game
GET	game/:id/action	get the game actions
GET	game/:id/level	get the game levels

GET	game/:id/sequence	get the game sequences
GET	game/:id/ui	get the game ui

Element	Fields	Datatype
history	id	integer
	login	datetime
	logout	datetime
	userid	integer

HTTP Verb	Endpoint	Description
POST	history	insert a new history
PUT	history	update existing history
DELETE	history	delete existing history
GET	history	get all history
GET	history/count	get the number of total history
GET	history/:id	get a specific history

Element	Fields	Datatype
notification	id	integer
	message	string
	typeid	integer
	statusid	integer
	sourceid	integer
	targetid	integer
	created	datetime

HTTP Verb	Endpoint	Description
POST	notification	insert a new notification
PUT	notification	update existing notification
DELETE	notification	delete existing notification
GET	notification	get all notifications
GET	notification/count	get the number of total notifications
GET	notification/:id	get a specific notification

HTTP Verb	Endpoint	Description
GET	patient/:id/game	get all patient games
GET	patient/:id/badge	get all patient badges

Element	Fields	Datatype
parameter	id	integer
	name	string
	datainput	string
	datasource	string
	categoryid	integer

HTTP Verb	Endpoint	Description
POST	parameter	insert a new parameter
PUT	parameter	update existing parameter
DELETE	parameter	delete existing parameter
GET	parameter	get all parameters
GET	parameter/count	get the number of total parameters
GET	parameter/:id	get a specific parameter

Element	Fields	Datatype
uilanguage	id	integer
	name	string

HTTP Verb	Endpoint	Description
POST	uilanguage	insert a new uilanguage
PUT	uilanguage	update existing uilanguage
DELETE	uilanguage	delete existing uilanguage
GET	uilanguage	get all uilanguages
GET	uilanguage/count	get the number of total uilanguages
GET	uilanguage/:id	get a specific uilanguage

Element	Fields	Datatype
uistring	id	integer
	name	string

HTTP Verb	Endpoint	Description
POST	uistring	insert a new uistring
PUT	uistring	update existing uistring
DELETE	uistring	delete existing uistring
GET	uistring	get all uistrings
GET	uistring/count	get the number of total uistrings
GET	uistring/:id	get a specific uistring

Element	Fields	Datatype
uitranslation	id	integer
	name	string
	stringid	integer
	languageid	integer

HTTP Verb	Endpoint	Description
POST	uitranslation	insert a new uitranslation
PUT	uitranslation	update existing uitranslation
DELETE	uitranslation	delete existing uitranslation
GET	uitranslation	get all uitranslations
GET	uitranslation/count	get the number of total uitranslations
GET	uitranslation/:id	get a specific uitranslation

Element	Fields	Datatype
user	id	integer
	name	string
	email	string
	status	integer
	roleid	integer
	created	datetime
	updated	datetime
	removed	datetime

HTTP Verb	Endpoint	Description
POST	user	insert a new user
PUT	user	update existing user
DELETE	user	delete existing user
GET	user	get all users
GET	user/count	get the number of total users
GET	user/:id	get a specific user
GET	user/:id/execution	get the user executions
GET	user/:id/history	get the user history
GET	user/:id/personal	get the user personal
GET	user/:id/profile	get the user profile
GET	user/:id/notification	get the user notifications
GET	user/:id/relation	get the user relations
GET	user/:id/schedule	get the user schedule