

Article

Proteo: A Framework for Serious Games in Telerehabilitation

Giuseppe Bernava¹, Andrea Nucita^{2,*}, Giancarlo Iannizzotto², Tindara Capri³ and Rosa Angela Fabio³

¹ Institute for Biomedical Research and Innovation (IRIB-CNR), Via Torre Bianca, Mortelle, Istituto Marino, 98164 Messina, Italy; giuseppe.bernava@irib.cnr.it

² Department of Cognitive Sciences, Psychological, Educational and Cultural Studies, University of Messina, 98122, Messina, Italy; ianni@unime.it (G.I.)

³ Department of Clinical and Experimental Medicine, University of Messina, 98121 Messina, Italy; tcapri@unime.it (T.C.); rafabio@unime.it (R.A.F.)

* Correspondence: anucita@unime.it

Abstract: Background: Telerehabilitation has grown significantly in the past years, especially in 2020 when it has been a crucial tool for supporting patients during the COVID-19 pandemic. Within the context of telerehabilitation, serious games have a significant role but realizing software for serious games is resource demanding. Methods: we present Proteo, a modular and open-source framework for developing serious games from scratch. We also present two serious game implementation examples with analysis of end user's and therapists/researchers' satisfaction. Results: by involving a group of 11 specialized therapists and 9 end users we analyzed the Proteo user's satisfaction. We found that both groups scored high level of involvement, and the therapists scored also high level of suitability. More in depth, both groups showed significant differences between positive and negative feeling, with positive feeling scoring higher than negative ones. Finally, users' level of suitability was reported as high while the difficulty of the system and the difficulty of the task were reported as low. Conclusions: Proteo has been proved to be a useful tool to develop serious games for telerehabilitation and has been well accepted by the users we involved in the evaluation tests.

Keywords: telerehabilitation; serious games; human-computer interaction

1. Introduction

Telerehabilitation (TR) can be defined as delivering rehabilitation services to the patient's home through a remote connection with the therapist, possibly enhanced by additional multimedia technologies [1,2]. TR can be provided using a variety of technological tools and ways, for examples: real-time visits with audio, video or both, asynchronous e-visits, telephone assessment; closed-circuit television to video conference; internet-based communication software (e.g., teleconference platform) with headsets, webcams, speakers, handheld cameras, microphones, and internet; serious games; or remote evaluations of recorded videos or images [1].

Several systematic reviews have argued that TR is an effective tool for patients with different disabilities: from musculoskeletal conditions, multiple sclerosis, neurological disorders, stroke to autism and genetic syndromes [4-9]. Moreover, it was found that TR services are effective both for children and adults with disabilities, and professionals also have reported high levels of satisfaction and acceptance of TR services. [5,10-12]. However, it is expected that advancements in technology can increase the usability and effectiveness of TR, but current TR systems do not usually allow the therapists to customize or adapt them to characteristics of patients and setting, and they are limited so specific patient condition [13-15]. This can be due to different factors, such as the complexity of the process, the technical aspects of TR services development, and the heterogeneity of patients with disabilities. Thus, it is necessary to develop and test the feasibility, safety, and effectiveness of TR modalities that can be customized by the therapist across subgroups of patients.

The last year, deeply impacted by the COVID-19 pandemic, has led to a significant increase in the use of remotely guided rehabilitation and therapy interventions [16-18]. However, patient conditions vary significantly and a need for a framework to define customized TR systems that can improve engagement of involved patients arises, with the aim of facing the enormous variability in patient specificity and needs. Moreover, such a framework should have a modular structure, such that a therapist can add support for specific hardware devices (i.e., eye tracker) or software capabilities (i.e., computer vision, machine learning) useful to let patients interact with the system and to acquire data for further analyses. Consequently, the implementation of TR leads to a double challenge: motivating patients and ensuring an effective system [19-21].

In this scenario, we propose Proteo, a framework designed to give therapists and researchers, without specific skills in software developing, the capability of defining TR service customization, precisely serious games, starting from game templates. Moreover, the framework can be applied in rehabilitation centers that could easily develop suites of remotely administered serious games, focusing only on a simplified game implementation, exploiting the framework modules for client/server communication, user management, human-computer interaction or deep learning techniques.

This framework aims at: a) the creation of personalized serious games for TR systems; b) monitoring of the therapeutic goals of the rehabilitation process; c) simplifying the interaction schemes between the patients and serious games. Moreover, Proteo is open-source and cross-platform, and is distributed under the MIT license.

In the next sections, after presenting Proteo, we discuss two use cases to analyze the practicability of the serious games creation process and to discuss the user satisfaction.

2. Materials and Methods

Several libraries and frameworks have been proposed to facilitate the programming of visual/artistic applications (e.g., openFrameworks.cc) and games (e.g., stencyl.com, defold.com, www.yoyogames.com/en/gamemaker, <https://gdevelop-app.com/>). Libraries such as [openFrameworks](http://openFrameworks.cc) can benefit from C++ language power, and plenty of modules that are specifically designed for artistic purposes, even with computer vision support, that is a crucial tool when dealing with serious games. However, [openFrameworks](http://openFrameworks.cc) is intended for C++ developers, which can be a barrier to those that have no skill about programming. On the other hand, applications, and libraries for gaming development (even with a simplified block programming interface) miss significant tools in the field of tele-rehabilitation, such as computer vision tools (pose estimation, face detection, facial expression recognition etc.).

The aim of the proposed platform, Proteo, is hence to give therapists and researchers, without specific skills in software developing, the capability of defining serious game customizations starting from game templates. Given a game template, which defines the game characteristics, therapists/researchers should be able to define the game logic and the user interaction in a simplified manner, focusing only on what she/he wants to analyze, overcoming technological barriers and concentrating only on defining tasks and collecting data.

Moreover, the framework can be applied in rehabilitation or research centers with development skills, that could easily develop suites of remotely administered serious games, focusing only on a simplified game implementation, exploiting the framework modules for client/server communication, user management, database for collecting data, human-computer interaction, or deep learning techniques, thus potentially gathering large amount of data that can enable exploitation of machine learning techniques [22].

In the following we present Proteo, an open-source and cross-platform scriptable platform, highly modular and with built-in support for many useful modules for serious games implementation. Moreover, we analyze two use cases, testing Proteo with two different kind of users: therapist/researchers and end users. In the first case, we ask the therapists/researchers group to customize a serious game given a game template by means of

the platform. In the latter, we ask a group of end user to play a serious game developed with Proteo. In both cases we collected data about user satisfaction, that we will discuss in the results section.

2.1. Proteo Architecture

Developing a cross-platform, scriptable solution for multimedia applications even with computer vision and deep learning support is a challenging task, since many different components have to be connected and tuned to reach a robust yet performative system. To tackle the complexity of needed features, Proteo has a highly modular structure, in which every aspect can be easily and independently modified (**Figure 1**).

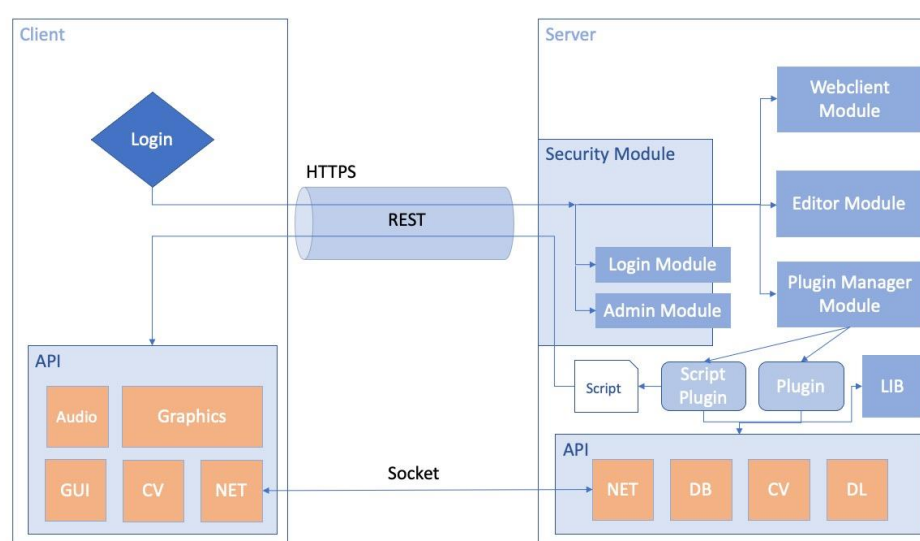


Figure 1. Proteo Architecture.

Proteo consists of two main applications, Proteo Server and Proteo Client, both written in C using libraries that can be ported to the main hardware and software platforms. At the heart of the system lies a Lua interpreter (LuaJIT.org) with an API designed specifically for Proteo. The use of pure C and LuaJIT allows considerable execution speed. Furthermore, on the server side, it is possible to extend the functionality of the system by installing Lua modules through package managers such as LuaRocks (LuaRocks.org).

Each client-side application consists of a script, which may require various libraries that are also loaded remotely. The script connects to the server via one or more plugins. Both scripts and plugins are written in Lua.

Proteo architecture supports both multi-client and multi-server configuration. The multi-server system has a star architecture, with a main server as the center of the star and several ancillary servers that are periodically queried by the main server. When a client logs in, the main server checks what resources are required and responds to the client by specifying which server (including itself if necessary) to address for each request. In order to manage the workload independently, the main server uses a simple ticketing system: when querying the secondary servers, the main server queries each server for available tickets. By considering which server has the higher value of available tickets and the load distribution, the main server assigns the resource request to a secondary server and sends to the client the address of that server. Due to the scriptable nature of Proteo, the ticketing system can be easily modified, if a more complex algorithm is needed.

2.1.1. Server Application

Each Proteo server exposes a REST interface, with base endpoints (login/admin/edit module), and specific endpoints, depending on the active plugins. Each plugin operates independently, but requires at least two GET endpoints: `plugin_name/info` and `plugin_name/permissions`.

The first provides service information: plugin version, whether the plugin is working and the number of tickets it offers.

The second returns an array of strings, each string being a permission that the plugin exposes. The plugin computation is transparent to Proteo, which will only take care of assigning permissions to individual users. The server application is composed of several modules that will be briefly described in the following.

Security module. Proteo implements a security mechanism in every REST call, that consists of two components:

- a call-based authentication system, based on keyed-hash message authentication codes (HMAC) and tokens
- a plugin-oriented permission management

During the login process, the client generates an HMAC using the password entered by the user, the client tries to generate the same HMAC using the password stored locally; if the two HMACs match, the server generates a token and sends it to the client. All subsequent client requests must contain an HMAC generated with a common key known to both client and server, and the token.

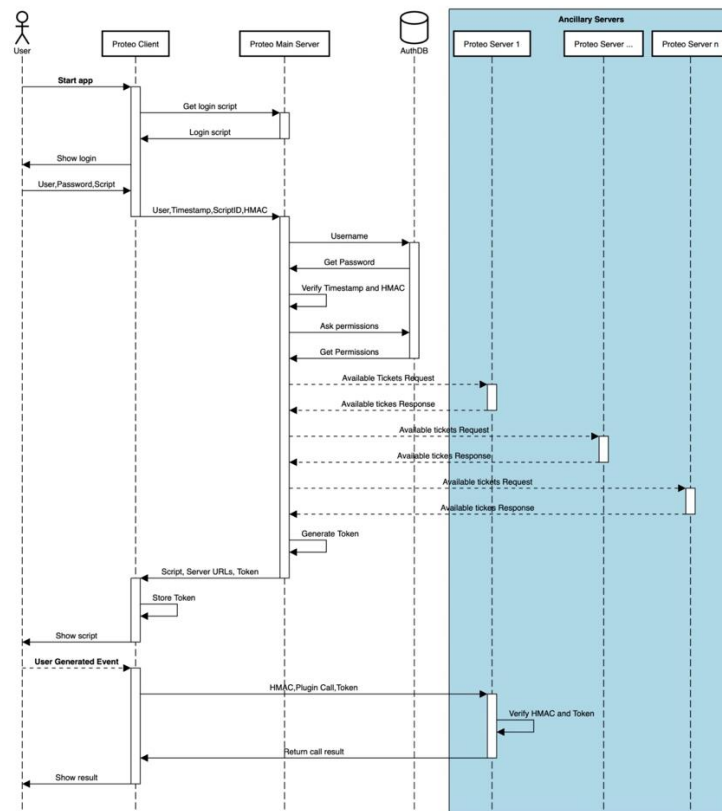


Figure 2. Proteo login sequence diagram.

The token is generated by encrypting (with a key owned by the server) information on the user identity and the token expiring date, so that the client can connect to several servers without storing copies of the user's authorization to each server.

Because of the scriptable nature of server-side plugins, it was necessary to develop a flexible structure for permissions. In practice, each plugin exposes a list of permissions to the server. These permissions are transparent to the server, but are assigned to users via

the Admin module. At each REST call, the server passes to the plugin the user id and the list of permissions the user has. The plugin will then act appropriately according to the received permissions.

Login module. A login request from the client is made via the [POST /login] call, this call can have two results:

- if the credentials are missing (or incorrect), a Lua script is returned that allows a login page to be displayed with the possibility of registering or recovering a lost password. This interface is written using the client's graphical API and can be modified to adapt it graphically and functionally to requirements.
- the login is successful, and the requested script is returned (together with the token, see Security Module)

Admin module. An administration interface can be accessed through a [GET/admin] call or through a login request. This script-based interface can also be modified, but to keep the main part of the implementation constant the administration functions are part of a library. In addition, the library makes it transparent to the plugin which method is used to save the administration information. In the current version, the administration database is stored via SQLite. Through the administration module it is possible:

- activate or deactivate plugins that the server loads at start-up
- create and deactivate users
- change users' password
- edit user permissions

During installation, there is only one user 'admin' with a password 'admin' who can access the script 'admin'. So, after the first installation it is possible to add users and manage permissions.

Plugin manager module. Each plugin consists of several access endpoints like the one shown in **Figure 3**:

```

proteo.route.get("plugin_name/helloworld/hi/:name",
  function(username, permission, data, param)
    local response={}
    response ["hello"]="Hi "..username.."and"..param["name"]
    return json.encode(response)
  end
)

```

Figure 3. Proteo endpoint example.

The routing system is very simple. Each endpoint is defined by a verb (in the example GET), a path and a function. The path can contain variables (in the example :name). The function arguments are the username, the user permissions, the data passed during the function call, and the parameters passed by the path. Conventionally, the function returns a JSON string, but any string is allowed.

Script plugin. A Proteo standard plugin is the plugin that manages the sending of scripts to clients. The main access points it exposes are:

- [GET proteo/permissions]: which returns the permissions needed to access the individual scripts
- [GET proteo/script/:script]: which checks whether the user has permission to access the script and if so, returns it encapsulated with JSON
- [GET proteo/lib/:lib]: which returns the requested library to the client. Libraries are not handled with permissions but there is an array in the script that identifies which libraries a client can request.

Editor. There are three types of script files in Proteo: plugin, script and lib. Every type of script is written in Lua and stored in specific folders within the server root folder. Given the aim of making Proteo easy to program, in addition to being able to edit scripts manually, it is possible to act at a higher level of abstraction through visual programming.

Using Blockly [23], it is possible, for instance, to translate the function in the previous figure (**Figure 4**) into a block in the following figure:

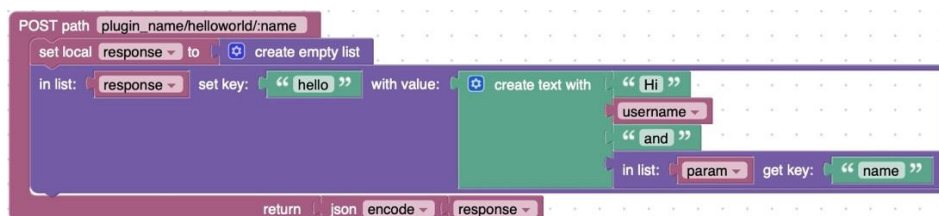


Figure 4. Blockly example for a simple function.

Proteo is equipped with an internal web-based editor to create or edit scripts and plugins. Through the endpoint [GET /admin] a user can authenticate (with editor permission) and access the files for which he/she has reading/writing permissions. The script is immediately available once saved. To the contrary, saving a Plugin restarts the server, so that it can be reloaded. Access and modification of script files is also managed by a server plugin.

All API blocks have been implemented but, in order to optimize visual programming, the aim is to further abstract the programming level through the use of libraries.

For example, to estimate the pose from a frame using the Blazepose algorithm [24], an appropriate library has been created, making the blocks extremely readable and easy to implement, as shown in **Figure 5**:

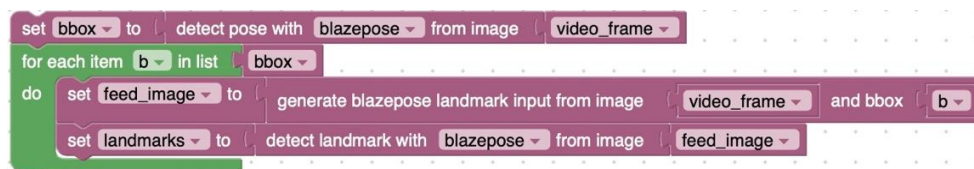


Figure 5. Blockly example for pose estimation using Blazepose.

Even if Blockly is capable of reproducing each of the Lua language constructs, it is specifically intended for users with no developing skill. In the context of tele-rehabilitation, we considered therapists or researchers that often need to set up task batteries (in the form of serious games), which is typically unfeasible without support from developers. With Blockly, starting from a serious game template, we give them the possibility to overcome technical skill barriers, and to reach autonomy. In Section 2.2. we present some examples of the process of building a serious game starting from a template.

API. Proteo API works as a general-purpose glue and wraps together several libraries making all parts of memory management, garbage collection and data interconnections transparent to the developer. There are system functions for accessing the disk, using timers, etc., which we will not go into detail about here, but we will concentrate on listing the parts of the library related to the development of serious games.

Net. Through the networking functions it is possible to make REST calls to specific addresses, and it is also possible to carry atomic messages across TCP/UDP, with patterns like fan-out, pub-sub, and request-reply. To do this, Proteo incorporates the functionalities

of the most popular networking libraries with a focus on portability: OpenSSL and ZeroMQ are just some of the libraries integrated into the Proteo API (**Figure 6**):

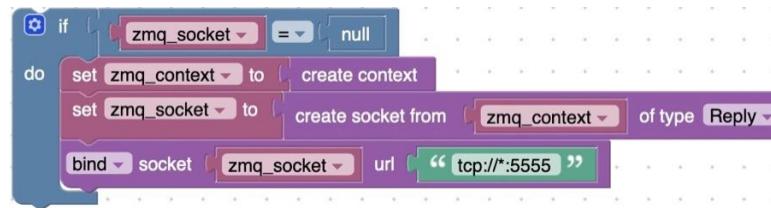


Figure 6. Blockly example for networking functions.

Communication between client and server is conventionally via JSON-encoded strings, even if this is not an implementation constraint.

In the context of serious games for tele-rehabilitation, network support is crucial, for instance for user authentication, multi-user support, and data collecting. Moreover, a tele-rehabilitation session may be intended as an autonomous activity by the patient, that follows a specific protocol, and which can generate data about users that are further analyzed by therapists. In this case, network support is useful for user authentication and data collection. Nevertheless, another kind of tele-rehabilitation session may involve the therapist, which monitors the ongoing user activities and can react in some given circumstances. Proteo is designed to give support to both the scenarios.

Computer Vision. In the field of computer vision, the reference libraries are OpenCV and FFMPEG. The use of Blockly simplifies coding (**Figure 7**).

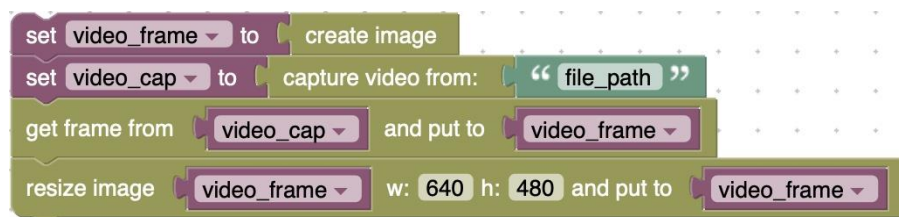


Figure 7. Blockly example for computer vision support.

There is a module in OpenCV that implements forward pass (inferencing) with deep networks, pre-trained using some popular deep learning frameworks, such as Caffe. FFMPEG is mainly used for codec functions.

The client can send a video stream to the server. This stream can be managed to create a group video chat, to be analyzed with a deep learning algorithm or both at the same time. The support for computer vision can be both client and server side.

Database. Proteo integrates two database systems, namely SQLite and EJDB2, so that it is straightforward to add support for data storage and manipulation in a serious game. However, support for other DBMS can be provided by adding specific modules.

Deep learning. One of the central aspects of Proteo is the possibility of using deep learning techniques in a simple way. For this reason, in addition to OpenCV, there is also a section of the API that allows inference via Tensorflow Lite. This is done using an extremely limited number of functions, for loading a model from a file and starting the execution of the model. Moreover, other useful functions allow interaction with other functions of the API. Thus, for example, it is possible to use an image outputted from OpenCV as input to the TensorFlow network (see **Figure 7**):

2.1.2. Client Application

Proteo is designed following the cloud computing paradigm. A Proteo client can be an empty container: the interface, multimedia and computational resources lie on the server. During login the client receives a “task”. This task may depend on the request, but also on user’s permissions or other conditions. In addition, the client also receives the addresses of the secondary servers from the main server, thus transparently distributing the workload in the cloud. APIs that require the presence of a media file also check for the presence of the file locally and if not found, download it from the server transparently.

However, even if Proteo relies on the cloud paradigm, some elaborations can be demanded to the client, if needed. This is particularly useful, for instance, when serious games need human-computer interaction based on computer vision algorithms. In general, client elaboration can be used when it can give a more reactive interaction experience. **Client – API.** Following the logic of cloud computing, the client-side API and libraries are oriented towards the human-machine interface. Net and Computer Vision APIs are also present with the same implementation to make the dialogue between Client and Server simple.

Client – GUI. Proteo has a set of standard GUI elements: label, text field, checkbox, button, list, dropdown, form and container. The API is designed to be simple and efficient to use, each GUI element can be created using a single function.

Client – Graphics. Proteo is oriented towards the development of serious games, which is why in addition to the standard elements of a graphic environment, such as images, lines, rectangles, ellipses and text, the system also has more complex elements: Sprites, Shapes and Skeletons.

Sprites are used for 2D animations created through a sequence of frames. During creation, a sprite sheet is specified from which subsequent frames will be cut out. Sprites are a valid technique for simple animations but for more complex animations a skeletal animation is required.

Shape element is a vector structure made up of a combination of two elementary objects: Polygon and Bezier, the first for drawing figures composed of straight lines, the second for drawing curves. A Shape object can also be imported from an SVG file, allowing Proteo to manage vector graphics.

By connecting a Skeleton object with a Shape object, Proteo automatically calculates the connection weights between the skeleton joints and the points representing the figure. After the weights are calculated, a modification to the skeleton results in a weighted modification of the figure.

Client – Audio. A first feature of the Audio module of the Proteo platform is the management of an internal software mixer that allows polyphonic playback (and possibly recording). From a logical point of view, the Audio API is divided into two sections: one section allowing the processing of audio files in recording and playback, and another section oriented towards streaming. The management of audio files is fundamental for serious games in which a sound compartment made up of music and polyphonic effects can foster user’s engagement. For the transmission of audio streams and, if necessary, for their analysis by the server, functions are required which allow the creation of a stream, the possibility of encoding it, packaging it, and sending it via a messaging service (see API-Network).

After succinctly presented the key components of the Proteo framework, in the following section, we discuss two serious game examples developed with Proteo. We adopted these serious game examples to analyze user satisfaction data about Proteo usage, which will be discussed in the results section.

2.2. Serious game implementation – two examples

In order to test Proteo and to collect preliminary data about the usage of the framework, we prepared two serious game examples:

- GymTetris
- Problem Solving Task

Each one of games, that will be described in the following, has been developed in three steps. The first step was the game ideation and involved both therapists and developers. During this step, therapists defined the characteristics of the game in terms of possible user interactions, data to collect, user management and so on.

In the second step, developers implemented the game templates, based on the requisites and constraints given by therapists. As you can see in the following, some of the game components are inaccessible to therapists in the third phase, while they have access at a higher level only on those components they can customize to adapt the game to the user needs.

The third phase was demanded to the therapists and regarded the game customization in terms of game logic, user interaction and choice of data to collect.

2.2.1 Sample and recruitment

A convenience sample of specialized therapist was recruited in the Department of Medicine and in the Hospital health units of the University of Messina and from the San Raffaele Hospital of Milan to test the problem-solving example for both involvement and suitability. The inclusion criteria were to have at least 3 years of experience with rehabilitation training, no attention was paid to previous experience in the use of computer or of computer apps. Twenty potential participants were approached personally or by a phone call. Nine of them refused to participate, and 11 were included. The final sample is composed by 11 specialized therapists in cognitive and motor training for people with disability. They have a mean age of 31.48 with a range of 26–43; ten participants were female and 1 was a male. Another convenience sample of end users was also recruited to test the Proteo modular framework for testing GymTretis. The 9 end users have a mean age of 32.08 with a range of 25–44; four participants were female and five were male.

2.2.2. GymTretis

GymTretis is based on the popular Tetris game (<https://en.wikipedia.org/wiki/Tetris>) and is conceived to encourage physical exercise. The serious game follows the classical game dynamic, but the user moves or rotates the pieces by means of body exercises. A scheme of the Proteo modules used to develop the game is shown in **Figure 8**.

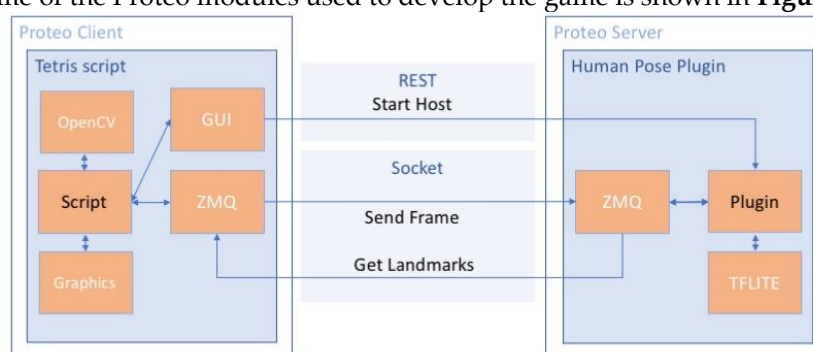


Figure 8. GymTretis scheme.

As previously discussed, the game template provides a basis that the therapist can customize, considering the needs of the user to which the game will be deployed. In the case of GymTretis, the game can be customized in terms of user interaction and data collection: the therapist can choose which are the body movements that the user has to perform to interact with the game. Moreover, the therapist can decide the user data to collect. **Figure 9** shows a screenshot of the game (you can find also the game implementation on GitHub: <https://massimobernava.github.io/proteo/>).

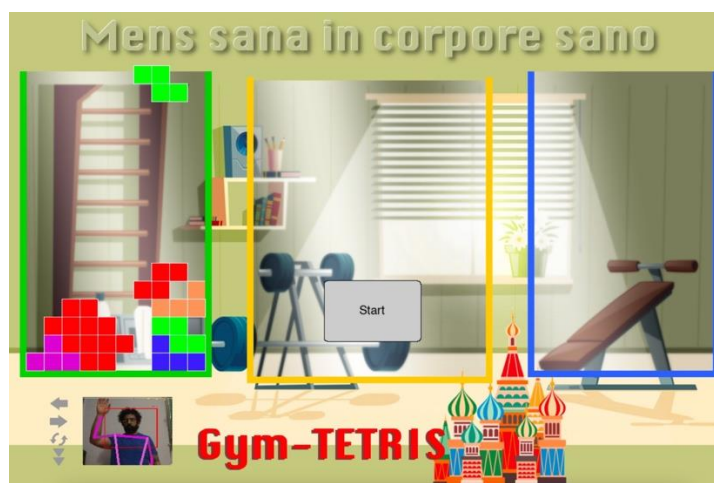
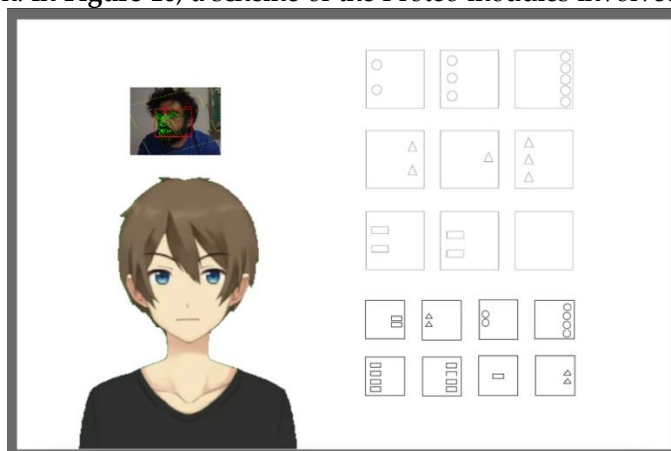


Figure 9. GymTetris game screenshot. User can move blocks by moving her/its body. To detect user movements the game include the pose estimation module.

2.2.3. Problem Solving Task

Problem Solving Task is a game in which the user has to solve cognitive tasks, such a series completion. Moreover, the user can interact with an avatar represented by an animated character that speaks to foster user's attention. In **Figure 10**, a scheme of the Proteo modules involved in



the game is shown, while in figure

Figure 11 a game screenshot is presented.

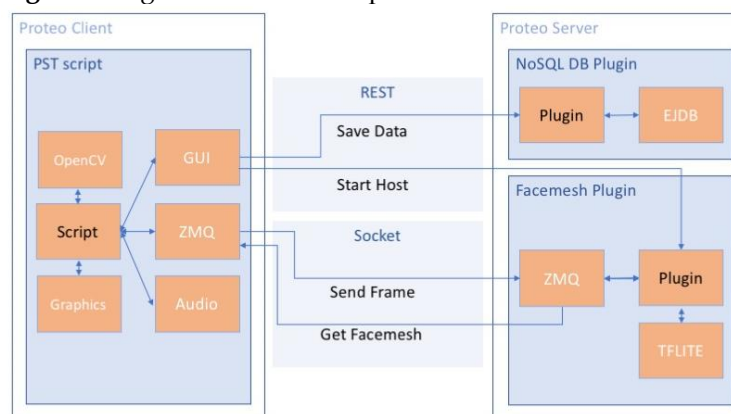


Figure 10. Problem solving task game scheme.

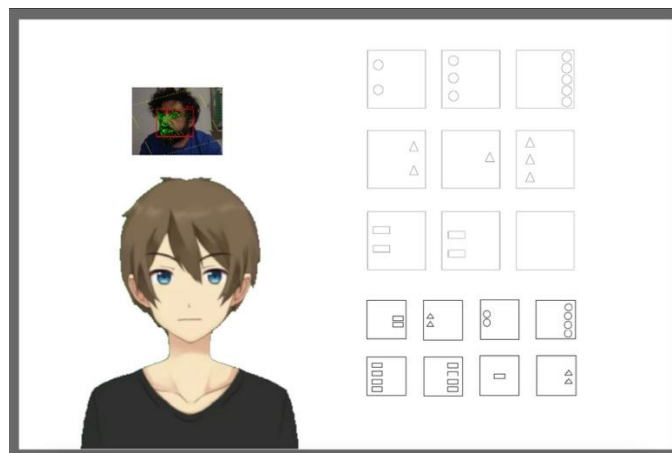


Figure 11. Problem solving task game screenshot. The avatar interacts depending on the user's attention. The user's attention is monitored through a face detection algorithm.

In this case, the therapist can choose the images needed for a specific task, to analyze user performance and collect data, but can also decide how and if the avatar interact with the user. Here, user interaction to choose the correct image to complete series is conventionally demanded to the mouse, but the actions of the avatar can be activated by user's level of attention. This kind of avatar interactivity, in facts, has been proved to be very important to enhance human-computer [25], especially for users with attention deficit [26].

In the following section, we present our findings about User Satisfaction Evaluation Proteo Questionnaire, both for a group of therapists and end users.

2.3. Outcome measures

In this study the User Satisfaction Evaluation Proteo Questionnaire was used, it is based on USEQ [27], but in this contribution USEPQ (see **Table 1**), is especially designed to test Proteo system. Basically, USEPQ extends USEQ, including questions to get responses for therapists in specific items related with Proteo. The Core Module measures the users' previous knowledge and experience with apps like Proteo through different components (see **Table 1**), the level of involvement components (with negative affect and positive affect) and the level of suitability of the system and of the tasks in a total of 18 items.

USEQP for therapists includes 18 questions, 16 of them with a response graded on a 5-point Likert Scale (from 1= "not at all" to 5 "very much", and a last open question. Following the USEQP scheme, the first six questions measure the users' previous knowledge and experience with apps like Proteo, with the computer, with the knowledge of serious game, with the capacity to create and program. The second four questions measure the involvement of the user (if he enjoys using Proteo, if he feels involved, if he feels bored or tired). The third eight evaluate the level of suitability of Proteo, the first evaluates the successful in the use of Proteo, the level of control (Q2), the clearness (Q3), the level of helpful (Q4), the perceived difficulty of the task and of the system (Q5 and Q6), the discomfort (Q7) and a final explanation open question (Q8). The global score of the first subscale ranges from 5 (poor previous knowledge) to 25 (excellent previous knowledge). Before to calculate the global score of the second subscale the Q9 and Q10 items have to be reversed, and the global score ranges from 4 (poor level of involvement) to 20 (high level of involvement). Before to calculate the global score of the third subscale the Q15, Q16 and Q17 items have to be reversed, and the global score ranges from 7 (poor suitability) to 35 (excellent suitability).

USEQP for end users includes only 9 questions (Q1-Q4, Q6-Q10).

Table 1. The USEQP (User Satisfaction Evaluation Proteo Questionnaire).

Question	Response				
	Not at all – very much				
Previous knowledge/experience					
Q1. Please indicate your level of computer experience	1	2	3	4	5
Q2. Indicate your level of experience with the apps on your computer	1	2	3	4	5
Q3. Have you ever used a system similar to Proteo?	No		Yes		
Q4. Indicate your level of knowledge of serious games	1	2	3	4	5
Q5. Indicate your level of knowledge in creating serious games	1	2	3	4	5
Q6. Indicate your level of knowledge of the security / privacy	1	2	3	4	5
Level of involvement					
Q7. I enjoyed trying Proteo	1	2	3	4	5
Q8. I felt involved	1	2	3	4	5
Q9. I felt tired	1	2	3	4	5
Q10. I felt bored	1	2	3	4	5
Level of adequacy/suitability					
Q11. Were you successful using the system?	1	2	3	4	5
Q12. Were you able to control the system?	1	2	3	4	5
Q13. Is the information provided by the system clear?	1	2	3	4	5
Q14. Do you think that this system will be helpful for tele-rehabilitation?	1	2	3	4	5
Q15. How difficult did you find the system?	1	2	3	4	5
Q16. How difficult did you find the task?	1	2	3	4	5
Q17. Did you feel discomfort during your experience with the system?	1	2	3	4	5
Q18. If yes, why?					

2.4. Data Analysis

Because of the ordinal nature of the measures, non-parametric statistical tests were used for data analyses. Hence, the median was used as a measure of central tendency and the interquartile range (IQR) for dispersion. To test for differences across conditions, we used the Wilcoxon signed-rank test was used for pairwise comparisons, with significance values adjusted by the Bonferroni correction. Data were analyzed using IBM Statistics for Mac, Version 20.0 (Armonk, NY: IBM Corp).

3. Results

Results are discussed before with reference to the therapists and after with reference to the end users. The three sections of Proteo related to the users' previous knowledge, the involvement of the user and the level of suitability were analyzed for therapists and the two sections on previous knowledge and their involvement were analyzed for end users.

3.1. Users' previous knowledge

Concerning Users' previous knowledge (**Table 2** and **Figure 12**), all conditions (Computer experience, Apps experience, Serious game knowledge, Security rules knowledge) were reported as presenting an intermediate level of Knowledge

(out of 3 points). Their level of Serious game creation was considered low. Pairwise comparisons with the Wilcoxon signed-rank test revealed that Serious game creation was significantly lower than Computer experience ($p < .011$), Apps experience ($p < .01$), Serious game knowledge ($p < .05$) and Security rules knowledge ($p < .05$).

Table 2. Medians and interquartile range in the Proteo experience questionnaire components

Previous experience	Median	Interquartile range
Computer experience	3	2-4
Apps experience	3	2-4
Serious game knowledge	2	1-3
Serious game creation	1	1-2
Security rules knowledge	3	2-4

Level of involvement		
Level of enjoyment	4	4-5
Level of involvement	4	4-5
Level of tiredness	2	1-3
Level of boring	1	1-2

Level of suitability		
Successfulness	4	3-4
Ability to control	3	3-4
Clarity	4	3-4
Helpfulness	4	3-5
Difficulties with the system	3	2-3
Difficulty with the task	2	2-3

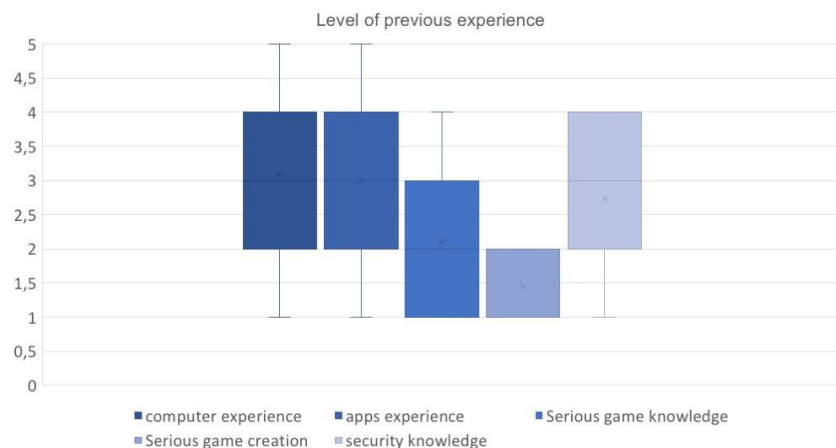


Figure 12. Boxplots of the components of the level of previous experience

3.2. Involvement of the therapist users

Regarding the user's level of involvement (**Figure 13**) the conditions of positive feeling (level of enjoyment and level of involvement) were reported as high (out of 4 points) while the conditions of negative feeling were reported as low (1 or 2). Concerning engagement (**Table 1**. The USEQP (User Satisfaction Evaluation Proteo Questionnaire). **Table 1**), in all conditions Pairwise comparisons revealed that Enjoyment and Involvement were significantly higher than Tiredness and Boring (Enjoyment vs Tiredness:

$p < .01$; enjoyment vs boring: $p < .01$; Involvement vs Tiredness: $p < .01$; Involvement vs boring: $p < .01$).



Figure 13. Boxplots of the components of the level of user's involvement.

3.3. Level of suitability

Concerning Users' level of suitability (**Figure 14**), the condition of successfulness, ability to control, clarity and helpfulness were reported as high (between 3 and 5), while the difficulty of the system and the difficulty of the task were reported as low. Pairwise comparisons with the Wilcoxon signed-rank test revealed the first four conditions were significantly lower than the last two at $p < .01$.

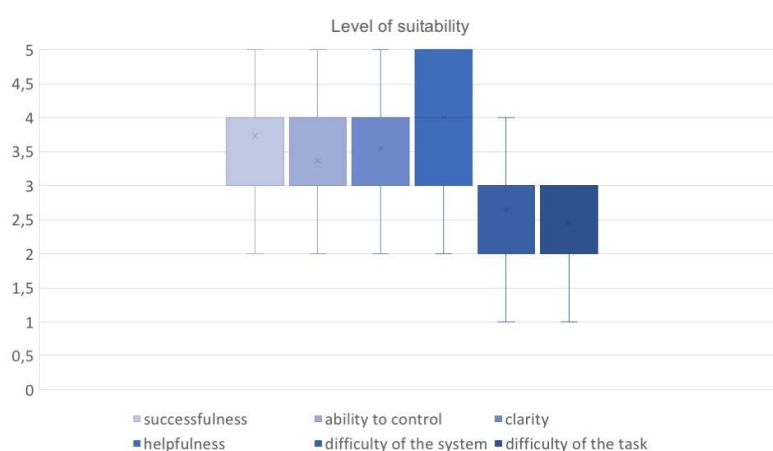


Figure 14. Boxplots of the components of the level of user's level of suitability.

3.4. End users' previous knowledge

Concerning Users' previous knowledge (**Table 3** and **Figure 15**), all conditions (Computer experience, Apps experience, Serious game knowledge, Security rules knowledge) were reported as presenting an high level of Knowledge (out of 4/ 5 points). Their level of Serious game knowledge was low (2 points). Pairwise comparisons with the Wilcoxon signed-rank test revealed that Serious game knowledge was significantly lower than Computer experience ($< .01$) and Apps experience ($p < .01$).

Table 3. Medians and interquartile range in the Proteo experience questionnaire components.

Previous experience	Median	Interquartile range
Computer experience	5	3-5
Apps experience	4	1-5
Serious game knowledge	3	1-5

Security rules knowledge	2	3-5
--------------------------	---	-----

Level of involvement		
Level of enjoyment	4	4-5
Level of involvement	3	3-5
Level of tiredness	1	1-1
Level of boring	2	1-3

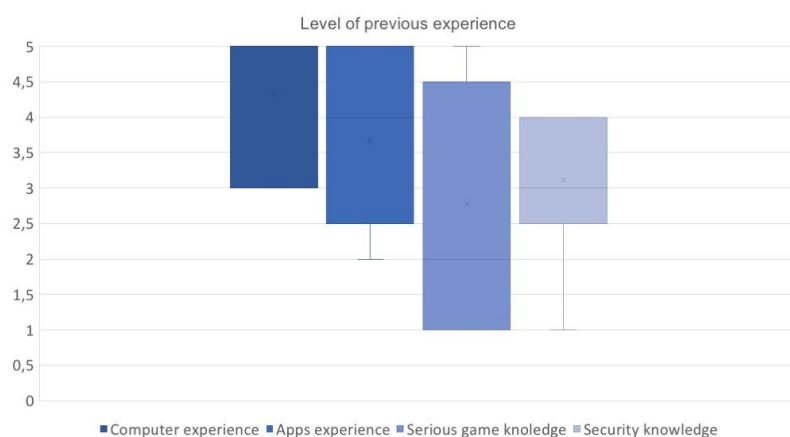


Figure 15. Boxplots of the components of the level of the user's experience.

3.2. Involvement of the user

Regarding the user's level of involvement (**Table 3** and **Figure 16**) the conditions of positive feeling (level of enjoyment and level of involvement) were reported as high (out of 4 points) while the conditions of negative feeling (boring and tiredness) were reported as low (1 or 2). Concerning engagement (**Table 2**), in all conditions Pairwise comparisons revealed that Enjoyment and Involvement were significantly higher than Tiredness and Boring (Enjoyment vs Tiredness: $p < .01$; enjoyment vs boring: $p < .01$; Involvement vs Tiredness: $p < .01$; Involvement vs boring: $p < .01$). The level of tiredness was 0.



Figure 16. Boxplots of the components of the end user's level of suitability.

4. Discussion

Telerehabilitation systems have been widely used in the past years, showing their crucial importance during the COVID-19 pandemic. In the field of telerehabilitation, serious games play an important role, since they can foster users' engagement and, hence, enhance rehabilitation outcomes. However, telerehabilitation systems based on serious games often are suited for specific patient conditions, so that implementing new games or even customizing existing ones is resource demanding and requires software development skills. Moreover, several libraries exist to ease game development, but they lack in offering advanced user interaction methods, as those that use computer vision algorithms or eye-tracking devices.

In this paper we present Proteo an open source, modular framework for developing serious games from scratch, providing a high-level interface for serious game development and customization. Proteo provides many built-in modules to cover most common features that a serious game may need, such as client/server communication, user management, data management, advanced human-computer interaction, computer vision or deep learning techniques.

After presenting the Proteo architecture and its base modules, we realized two serious games developed with Proteo. We then tested the process of game customization by therapists and analyzed the levels of experience, involvement and suitability for both therapists and end users. Data from the different components of the User Satisfaction Evaluation Proteo Questionnaire assessed the impact that the two games experience had on participants' engagement. Overall, and irrespective of the game mode, participants reported low feelings of boring and tiredness, i.e., of negative affect and high level of positive affect (involvement and enjoyment). Irrespective of the level of previous experience also the suitability with the condition of successfulness, ability to control, clarity and helpfulness were reported as high.

This study has some limitations that should be acknowledged. Regarding the data analysis, our sample size is relatively small for some of the statistical analyses performed, and results should be considered with caution. As for future work, we plan to conduct more extensive tests involving a larger group of therapists or researchers. Another possible next step is to explore other variants of game, such as complex task units (as with math problems). At last, we consider important to understand if these results are similar in the case of the group size increases, but also to better understand the possibility to apply it in different settings such as family, school and hospital.

Supplementary Materials: Proteo is available on GitHub under the MIT License at <https://massimobernava.github.io/proteo/>.

Author Contributions: Conceptualization, G.B. and A.N.; methodology, G.B., A.N., R.A.F. and G.I.; software, G.B.; validation, G.B., A.N., R.A.F. and G.I.; writing—original draft preparation, G.B., A.N., R.A.F. and T.C.; writing—review and editing, A.N., G.B., R.A.F. and T.C.; visualization, G.B. and A.N.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Putrino, D. Telerehabilitation and emerging virtual reality approaches to stroke rehabilitation. *Curr. Opin. Neurol.*, **2014**, *27*, 6, 631–636. <https://doi.org/10.1097/wco.0000000000000152>
2. Dovigo L., Capri T., Semino M., Nucita A., Giannatiempo S., Iannizzotto G., Fabio R.A. (in press). School's out. Interactive tele-education for Rett patients at the time of COVID. *Professional Psychology: Research and Practice*.
3. Brennan, D.M.; Mawson, S.; Brownsell, S. Telerehabilitation: enabling the remote delivery of healthcare, rehabilitation, and self management. *Studies in Health Technology and Informatics* **2009**, *145*, 231–248.
4. Agostini M.; Moja L.; Banzi R. et al. Telerehabilitation and recovery of motor function: a systematic review and meta-analysis. *J Telemed Telecare* **2015**, *21*, 202–213. <https://doi.org/10.1177/1357633x15572201>
5. Capri, T.; Fabio, R.A.; Iannizzotto, G.; Nucita A. The TCTRS Project: A Holistic Approach for Telerehabilitation in Rett Syndrome. *Electronics*, **2020**, *9*, 491. <http://dx.doi.org/10.3390/electronics9030491>

6. Capri, T.; Nucita, A.; Iannizzotto, G.; Stasolla, F.; Romano, A.; Fabio, R.A. Telerehabilitation for Improving Adaptive Skills of Children and Young Adults with Multiple Disabilities: a Systematic Review. *Review Journal of Autism and Developmental Disorders*, **2020**, *8*, 244-252. <https://doi.org/10.1007/s40489-020-00214-x>
7. Cottrell M.A.; Galea O.A.; O'Leary S.P.; Hill A.J.; Russell T.G.; Real-time telerehabilitation for the treatment of musculoskeletal conditions is effective and comparable to standard practice: a systematic review and meta-analysis. *Clin Rehabil.*, **2017**, *31*, 625–638. <https://doi.org/10.1177/0269215516645148>
8. Grona S.L.; Bath B.; Busch A.; Rotter T.; Trask C.; Harrison E. Use of videoconferencing for physical therapy in people with musculoskeletal conditions: a systematic review. *J Telemed Telecare*, **2018**, *24*, 341–355. <https://doi.org/10.1177/1357633x17700781>
9. Knutsen, J.; Wolfe, A.; Burke, B. L.; Hepburn, S.; Hepburn, S.; Lindgren, S.; Coury, D. A systematic review of telemedicine in autism spectrum disorders. *Review Journal of Autism and Developmental Disorders*, **2016**, *3*(4), 330–344. <https://doi.org/10.1007/s40489-016-0086-9>
10. Benham, S.; Gibbs, V. Exploration of the effects of telerehabilitation in a school-based setting for at-risk youth. *International Journal of Telerehabilitation*, **2017**, *9*, 39–46. <https://dx.doi.org/10.5195%2Fijtr.2017.6215>
11. Golomb, M. R.; McDonald, B. C.; Warden, S. J.; Yonkman, J.; Saykin, mA. J.; Shirley, B.; et al. In-home virtual reality videogame telerehabilitation in adolescents with hemiplegic cerebral palsy. *Archives of Physical Medicine and Rehabilitation*, **2010**, *91*, 1–8. <https://doi.org/10.1016/j.apmr.2009.08.153>
12. Huber, M.; Rabin, B.; Docan, C.; Burdea, G.; Nwosu, M. E.; Abdelbaky, M.; Golomb, M. R. PlayStation 3-based tele-rehabilitation for children with hemiplegia. Proceedings of Virtual Rehabilitation, Vancouver, BC, Canada, 25-27 August 2008. <https://doi.org/10.1109/ICVR.2008.4625145>
13. Vallejo D.; Gmez-Portes C.; Albusac J.; Glez-Morcillo C.; Castro-Schez J.J. Personalized Exergames Language: A Novel Approach to the Automatic Generation of Personalized Exergames for Stroke Patients. *Applied Sciences*, **2020**, *10*(20), 7378. <https://doi.org/10.3390/app10207378>
14. Pereria, F.; Bermudez I Badia, S.; Jorge, C.; Cameirao M.S. The use of game modes to promote engagement and social involvement in multi-user serious games: a within-person randomized trial with stroke survivors. *J NeuroEngineering Rehabil*, **2021**, *18*, 62. <https://doi.org/10.1186/s12984-021-00853-z>
15. Antón, D.; Berges, I.; Bermudez, J.; Goni, A.; Illarramendi, A. A Telerehabilitation System for the Selection, Evaluation and Remote Management of Therapies. *Sensors*, **2018**, *18*, 1459. <https://doi.org/10.3390/s18051459>
16. Aderonmu, J.A. Emerging challenges in meeting physiotherapy needs during COVID-19 through telerehabilitation. *Bull Fac Phys Ther*, **2020**, *25*, 16. <https://dx.doi.org/10.1186%2Fs43161-020-00018-4>
17. Prvu Bettger, J.; Resnik, L. J.; Telerehabilitation in the Age of COVID-19: An Opportunity for Learning Health System Research. *Physical Therapy*, **2020**, *100*(11), 1913–1916. <https://doi.org/10.1093/ptj/pzaa151>.
18. Christy B.; Keeffe J. Telerehabilitation during COVID-19: Experiences in service delivery from South India. *Indian J Ophthalmol.*, **2020**, *68*(7), 1489-1490. https://doi.org/10.4103/ijoo.IJO_1197_20
19. Castelli, I.; Antonietti, A.; Fabio, R. A.; Lucchini, B.; Marchetti, A. Do Rett syndrome persons possess Theory of Mind? some evidence from not-treated girls. *Life Span and Disability*, **2013**, *16*(2), 157-168.
20. Fabio, R. A.; Magaouda, C.; Capri, T.; Towey, G. E.; Martino, G. Choice behavior in Rett syndrome: The consistency parameter. *Life Span and Disability*, **2018**, *21*(1), 47-62.
21. Gangemi, A.; Capri, T.; Fabio, R. A.; Puggioni, P.; Falzone, A. M.; Martino, G. Transcranial direct current stimulation (TDCS) and cognitive empowerment for the functional recovery of diseases with chronic impairment and genetic etiopathogenesis. *Adv. Genet. Res.*, **2018**, *18*, 179-196.
22. Nucita, A.; Bernava, G.M.; Giglio, P.; Peroni, M.; Bartolo, M.; Orlando, S.; Marazzi, M.C.; Palombi, L. A Markov chain based model to predict HIV/AIDS epidemiological trends. *Lect. Notes Comput. Sci.*, **2013**, *8216*, 225–236. https://doi.org/10.1007/978-3-642-41366-7_19
23. Blockly. Available online: <https://developers.google.com/blockly> (accessed on 1st of May 2021).
24. Google AI Blog - On-device, Real-time Body Pose Tracking with MediaPipe BlazePose. Available online: <https://ai.google-blog.com/2020/08/on-device-real-time-body-pose-tracking.html> (accessed on 1st of May 2021).
25. Iannizzotto, G.; Lo Bello, L.; Nucita, A.; Grasso, G.M. A vision and speech enabled, customizable, virtual assistant for smart environments. In Proceedings of the 2018 11th International Conference on Human System Interaction, Gdansk, Poland, 4–6 July 2018; Article number 8431232. pp. 50–56. <https://doi.org/10.1109/HSI.2018.8431232>
26. Fabio, R.A.; Capri, T.; Iannizzotto, G.; Nucita, A.; Mohammadhasani, N. Interactive avatar boosts the performances of children with attention deficit hyperactivity disorder in dynamic measures of intelligence. *Cyberpsychol. Behav. Soc. Netw.*, **2019**, *22*, 9. <https://doi.org/10.1089/cyber.2018.0711>
27. Gil-Gomez, J. A.; Manzano-Hernández, P.; Albiol-Pérez, S.; Aula-Valero, C.; Gil-Gómez, E.; Lozano-Quilis, J.A. USEQ: A Short Questionnaire for Satisfaction Evaluation of Virtual Rehabilitation Systems. *Sensors*, **2017**, *17*(7), 1589. <https://dx.doi.org/10.3390%2Fs17071589>