# Integration of Cognitive Tablet Games for Rehabilitation with Socially Assistive Robotics

Carmen Díaz-de-Mera[1], Fernando Fernández[1], and José Carlos Pulido[2]

[1] Dpto. de Informática, Universidad Carlos III de Madrid, Av. de la Universidad 30, 28911 Leganés, Spain
[2] Inrobics Social Robotics, S.L.L., Av. Gregorio Peces Barba 1, 28919 Leganés, Spain

**Abstract.** The Inrobics Rehab platform aims to apply Social Assistive Robotics into rehabilitation and stimulation processes, using a social robot as a co-therapist in rehabilitation sessions configured by a clinical expert. The platform is composed of different exercises and games where the robot asks the patient to perform different poses and checks whether they are doing them correctly using a 3D sensor. In this paper we propose to extend the architecture of the platform by using as an additional element, a tablet, increasing its interaction capabilities and permitting a wider variety of games and exercises, including cognitive activities. We present how the architecture has been modified to include this new component in different robotic platforms and evaluate it with two new games that use the tablet, describing how this new paradigm of interaction affects the flow of the therapy session.

**Keywords:** Social Assistive Robotics · Cognitive Rehabilitation · Control Architecture · Human-Robot Interaction.

## 1 Introduction

Socially Assistive Robotics (SAR) is born from the intersection between Social Robotics and Assistive Robotics, where a robot helps a user through the execution of a task by means of social interaction [4]. The Inrobics Rehab platform was conceived as a way to apply SAR to rehabilitation, where a social assistive robot performs as a co-therapist in an autonomous way [16]. The platform integrates a robot, a 3D sensor, a control architecture based on Automated Planning (AP) and an android application where therapists can configure and execute sessions. As it is designed, the platform is independent of the robot used, so it can be deployed with any humanoid robot. Currently, the software has been fully integrated with the NAO [3] and Pepper[4] robots from Aldebaran Robotics, as well as with a virtual agent representing the NAO robot[5].

[3] `https://www.aldebaran.com/en/nao`
[4] `https://www.aldebaran.com/en/pepper`
[5] `https://drive.google.com/file/d/1L-BQEiyuWcYQ1Fa4Tthy5k3zkJDnz8nZ/view?usp=drive_link`

The platform consists on several exercises and games, where the patient is asked to perform different tasks, such as upper limb strength training, activities of daily living or symbolic association. Even if the main goal of the exercises and the difficulty and focus of the poses are different, most activities follow roughly the same flow: after an explanation of the exercise, the robot presents each routine that can be a pose or a sequence of poses, and which is then checked by the 3D sensor to see if the patient executed it correctly. After the check is finished, positive or encouraging feedback is offered by the robot.

In this paper we propose a new interaction paradigm for the platform, using the application not only for session configuration and control but also to play games with the robot during the session. In the case of robots with an integrated tablet such as the Pepper robot, the game will be played using that tablet instead of the separate device needed for the configuration. This new form of interaction requires a new element, the tablet, to be integrated in the existing control architecture of the platform.

To validate the new paradigm, two new use cases have been implemented as two new tablet games that have been added to the game catalogue of the Inrobics Rehab platform. These two use cases aim to help therapists perform cognitive rehabilitation with patients. In this paper we describe the two new use cases and how they are executed in a therapy session with the updated control architecture, demonstrating a correct performance and an open set of interaction possibilities.

In the following sections we present the current control architecture of the platform and how the tablet can be added as a new component to it. We also describe the two new games and how they are executed in a robot-assisted therapy session with the Inrobics Rehab platform.

## 2 Background

The Inrobics Rehab platform was born from the NAOTherapist project [15], where they used the NAO robot for upper-limb rehabilitation using a pose mirroring game. The robot is capable of carrying out the therapy session completely autonomously by means of the control architecture Mlaras (Multi-layered architecture for Autonomous Systems) [7].

The Mlaras architecture integrates planning, execution, monitoring, replanning and learning in different layers of information abstraction. In the Inrobics Rehab platform two layers of abstraction are used: a high-level layer for deliberation and a low-level layer for the information that the robot and the rest of elements of the architecture such as the 3D sensor can directly work with.

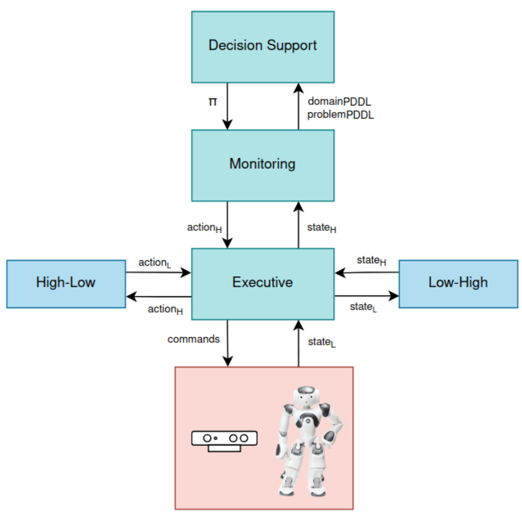
The high-level deliberation is performed using Automated Planning (AP) [6], an Artificial Intelligence technique which consists of obtaining a plan from a given initial state into a state where a set of goals defined are achieved, by applying deterministic actions. Formally, a classical planning task can be defined as a tuple $\Pi = \langle S, A, I, G \rangle$, where $S$ is the set of states defined as predicates, $A$ is the set of actions, $I \subseteq S$ defines the initial state and $G \subseteq S$ specifies the goals to achieve.

Actions $a \in A$ are tuples $\{pre_a, add_a, del_a\}$. For $a_i$ to be applicable in a state $s_i \in S$, $pre(a_i) \subseteq s_i$ and $s_{i+1} = s_i \cup add(a) \setminus del(a)$. A plan $\pi = \{a_1, a_2, \ldots, a_n\}$ is a solution of $\Pi$ if it is applicable in $I$ and results in a state $s_n$ such as $G \subseteq s_n$.

Classical automated planning assumes that an action's effects are deterministic and always known, without any external events interrupting the plan. However, in real environments such as the clinical environment with real patients where the NAOTherapist project operates, actions may fail so the expected state can change and the original plan may no longer be viable. Although there are AP paradigms that consider non-deterministic actions, a common approach is to tackle the inherent world uncertainty by replanning; if the state changes and the plan cannot longer be applied, a new plan is created from the current state [19].

The planning, replanning and monitoring of the plan is handled by the Decision Support and Monitoring modules, as shown in Figure 1, which use the standard Planning Domain Definition Language (PDDL) [13] to define the domain (actions and predicates) and problems (initial state and goals) needed for the planner. The planner used to solve the problems is the Metric-FF planner [9].

Fig. 1: Mlaras control architecture




The high-level PDDL actions obtained for the plan are sent to the Executive component, which performs the translation to the low-level execution layer using the high to low module. The low level actions are sent to the actuators of the architecture, which in the Inrobics Rehab platform are the robot and the 3D sensor. These components also act as sensors and send the Executive component their low-level state, which is translated to high-level PDDL predicates using the low to high module. The monitoring component checks that the state received matches the expected state, and triggers a replanning otherwise.

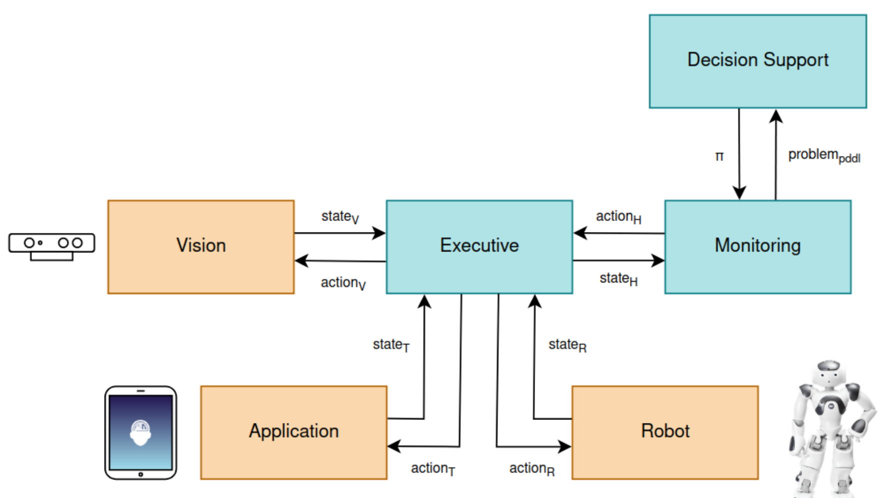
This architecture allows the Inrobics Rehab platform to function fully autonomously, once the therapist has configured the session, from where an initial plan can be generated. However, the only external elements that it connects to the high-level deliberation layer are the robot and the 3D sensor, so in the next section we will present how the architecture can be expanded by adding a tablet as a new element.

## 3 Control architecture

As described in the previous session, the control architecture used by the Inrobics Rehab platform [7] divides the execution among several components, where the Executive component centralizes the information received from the rest of the components to control the therapy session. It reasons mainly with the actions to execute and the state of the session and containing the High to Low and Low to High sub-modules.

In Figure 2 a simplified picture of the control architecture is shown, omitting the modules necessary for the translation of the layers, with the main addition from this paper, the tablet application, as a new component. This new component connects to the architecture through the Executive component, in the same way as the Vision and Robot components, which use the Zeroc Ice framework [6].

Fig. 2: Instance of control architecture with NAO




[6] https://zeroc.com/ice

The Executive component receives the high-level actions from the deliberation layer, using the High to Low module to decompose them to low-level commands. Each high-level action can be decomposed into several sets of commands that can be executed in parallel, and every set of parallel commands will be executed in sequence, with a translation language definition [8] similar to Behavior Trees [1].

These commands are sent to the rest of the components in the architecture to execute. As shown, the actuators of the architecture are now the robot, the 3D sensor and the tablet, so every command has to be sent to one of these devices ($action_R$ for the robot, $action_V$ for the sensor and $action_T$ for the tablet). The robot would for instance get commands indicating what to do or what to say, the 3D sensor would get the commands to check a routine or the tablet would get a command to show a specific interface.

Likewise, all these components of the architecture would send the information of their state to the Executive component ($state_R$, $state_V$ and $state_T$), such as the status of the robot (its battery, temperature level, execution status...), whether the patient is in front of the sensor or not or whether the patient has chosen the correct option on the tablet. This information is converted from a low-level state to high-level PDDL predicates that the Monitoring and Decision Support modules can reason with using the Low to High module.

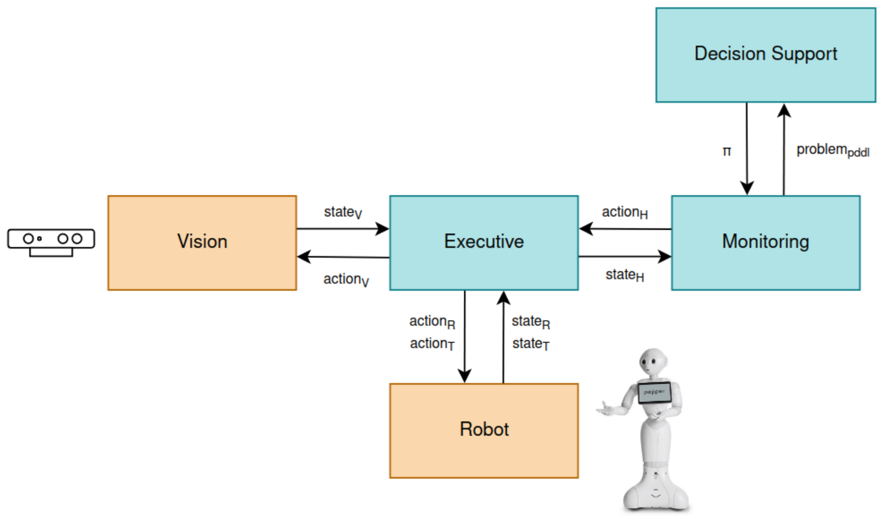
In Figure 2, the architecture was shown with the tablet as a separate device, which would be the case in robots with no tablet integrated. In the Pepper robot and similar, the architecture would be the one shown in Figure 3, where the actions related to the execution of the tablet ($action_T$) would also be sent to the robot instead of a separate device. In the same way, the state of the tablet ($state_T$) will be sent from the robot to the Executive component to reason with. Therefore, the only change needed is in the Executive component, which needs to know the type of devices connected to perform the communications accurately.

## 4 Use cases

The games included in the Inrobics Rehab platform all make use the 3D sensor to compare the poses done by the patient against the expected ones. This allows to have a wide rage of games and exercises to improve different capabilities of the patient such as their range of joint motion, their proprioception (their ability to know where each of their body parts is), their attention span or their memory, among others. Still all the exercises require physical movement, which can be limiting when defining new games to work more specific capabilities.

To increase the catalogue of the platform in this paper we propose a new paradigm of interaction, where instead of using the 3D sensor to allow the robot to offer feedback and guide the session, the main interaction between the patient and the robot is done through a tablet application, which has been added to the control architecture as described in the previous section. The 3D sensor will still be used so that the architecture knows that the patient is still in front of the robot and the session can continue as normal.

Fig. 3: Instance of control architecture with Pepper

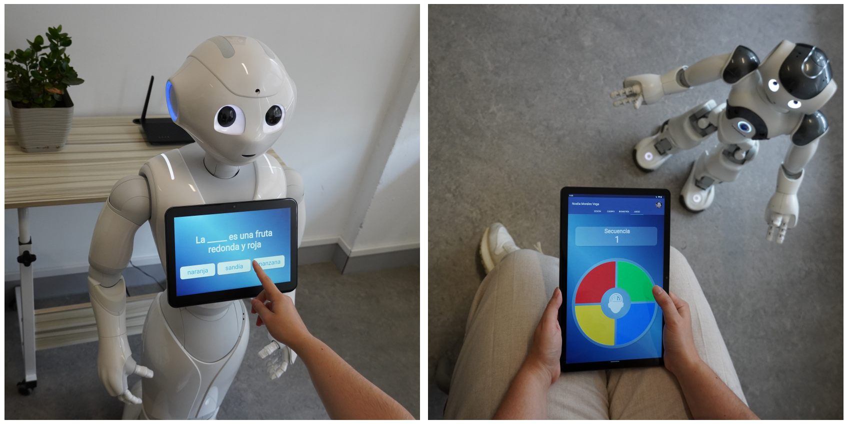



This new paradigm would also introduce a change of scenario in the new games. Normally the patient would be standing or sitting about 1.5 meters in front of the robot and the sensor, with their arms free to do any movement. When introducing the tablet as an interaction device in the therapy session, depending on which robot is being used the change of scenario would be different. As shown in Figure 4, the patient will have to get closer to the robot in the case of robots with integrated tablets such as Pepper, so that they can comfortably reach the device. In the case of robots without tablets such as NAO, the patient will be handed an external device to play the games, but keeping the same distance and position regarding the robot.

In this paper we propose two new games which make use of the tablet to interact with the robot and which will be integrated in the catalogue of the platform. The design of the use cases was done with the aid of the clinical advisor of Inrobics, an occupational therapist.

- The Gaps game consists on several sentences shown to the patient, which contain a gap to be filled. Every time a new sentence is shown the robot will read it aloud. The patient has to choose the correct word that fits the context of the sentence out of three options. If the patient chooses the correct option the robot will congratulate them; if not, the robot will offer the correct choice. If the patient takes too much time to answer, the robot will offer help by reading the sentence with the different options. This game aims to improve the vocabulary of the patient for patients with cognitive impairment.
- Colors is a memory game where the robot shows a sequence of colors with its eyes, which the patient has to remember and then repeat on the tablet.

Fig. 4: Use case scenarios




(a) Gaps game with Pepper

(b) Colors game with NAO

> The sequence is increased each turn until a maximum number of colors is reached or the patient makes three mistakes. If the patient gets a sequence wrong, the robot will repeat it, giving them another chance to get it correct. If they take too long to input the sequence, the robot will give some clues about the next color. Patients of every age can be benefited from this game.

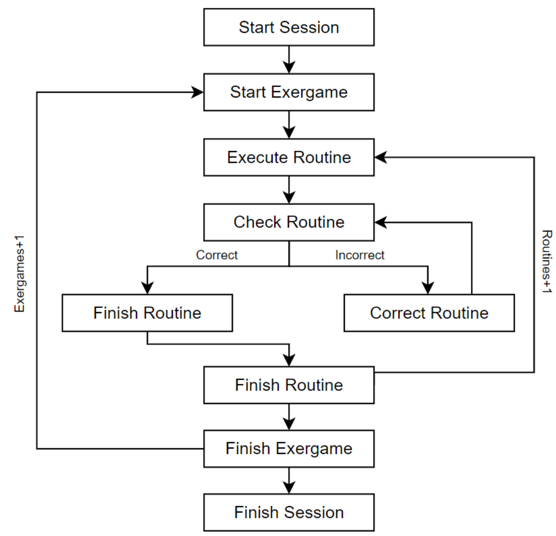
Therefore, in both use cases, interaction with the user is performed both with the robot and with the tablet, maintaining the social connection of the robot with the user that is key in social assistive robotics.

## 5 Session flow

As the catalogue of games of the platform has increased, a therapist will now be able to schedule tablet games alongside games that require the 3D sensor for a rehabilitation session. The typical flow of a robot-assisted therapy session with the Inrobics Rehab platform is shown in Figure 5, where after starting the session and welcoming the patient, each exergame is executed sequentially, first introducing and explaining the concept and what the patient and the robot will do.

Each exergame executed in the session is composed of routines, which previously were exclusively sequence of poses. The robot executes the routine to show the patient, and then the system checked them using the 3D sensor. If the routine is performed correctly the robot congratulates the patient and passes on to the next routine. If on the other hand the patient fails to execute the routine the robot offers corrective feedback. There is a maximum number of attempts per routine, if it is reached, the robot will skip the routine and continue the

Fig. 5: Flow of a therapy session




session. After finishing all exergames the robot will dance with the patient as a reward and say goodbye.

For the new two games the definition of routine has been broadened, instead of being a sequence of poses, a routine is defined as the atomic element of an exergame which can be executed and checked. In the gaps game, each sentence shown to the patient with a gap to be filled is a routine, where checking that routine consists on comparing the option chosen by the patient with the correct option. On the other hand. in the colors game each turn where the robot shows the sequence of colors to the patient is a routine. The first routine consists of only one color but each subsequent routine a new color is added. To check the routine the sequence input by the patient on the tablet is compared to the sequence shown.

The actions shown in Figure 5 roughly match the high-level actions defined in the PDDL domain, allowing to reuse this domain for all the games in the platform. The main changes between different games are therefore the routines and other parameters such as the number of attempts allowed for each routine. To generalize this domain the high to low decomposition of the control architecture is used, implementing a different set of low-level actions for the same high-level action depending on the context of the session. An example of this decomposition is shown in Figure 7, where the execution of a routine of the Gaps game, consisting of showing the sentence and reading it aloud, is shown. Each action can be executed by a different part of the control architecture, be it the robot or the tablet, as described in section 3. A wider description of language used in the High to Low module in MLARAS can be found in [8]).

```
High: execute-routine(routine, exergame), $exergame is gaps
Lows: saySpeechRoutine(routine)
      setInterface("gaps_sentence", routine)
```

Fig. 6: High to Low declaration of high-level action

```
If: $tablet_result < 0
    delete(correct-routine $routine_to_check, state_pddl)
```

Fig. 7: High to Low declaration of high-level action

## 6 Execution

To verify the correct implementation of the system and the new games, we have carried out several trials in a controlled laboratory environment, using the Inrobics application to configure therapy sessions with the two games. The validation was purely technical, checking that the flow of the session is correct and the platform works in the different situations that can arise. In this validation participated the clinical advisor of Inrobics, who checked that the interaction with the robot is suitable for patients.

### 6.1 Colors

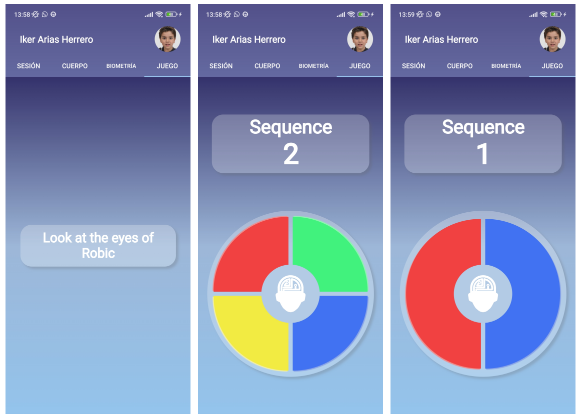
To configure the Colors game the therapist will have to define the maximum number of colors the sequence will reach and the difficulty of the game, between two or four possible colors, as shown in Figure 8.

The flow of this game consists on the patient first looking at the robot to concentrate on the colors being shown with its eyes and then using the tablet to input the colors. To guide the patient in this interaction, the robot indicates when to look at it and when it is the turn of the patient to play, with the help of the tablet interfaces. When the patient inputs the colors, the tablet will give sound feedback to each touch, indicating whether the color touched was correct or incorrect.

If the patient makes a mistake, thar architecture triggers a replanning and the robot offers a new opportunity, repeating the sequence. There are two ways to end the game: if the patient makes three mistakes, in which case the robot gives encouraging feedback and comment on the number of colors reached; or if otherwise the patient reaches the maximum number of colors, in which case the robot comments on it and congratulates them.

The other case that can occur in execution is that the patient does not interact with the tablet or touch any color. In this case the robot can give clues about the next color, and if a timeout is reached, it will be considered a mistake and it will repeat the sequence.

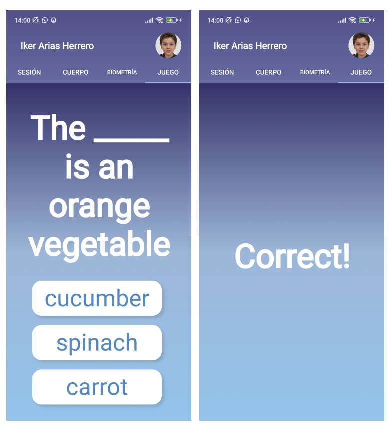
Fig. 8: Colors game interfaces




### 6.2 Gaps

For the Gaps game, therapists will be able to define the number of sentences that the patient will have to solve and the topics to include in the sentences (for example animals, food, colors...).

Fig. 9: Gaps game interfaces




The flow of this game is simpler, as it only consists on the robot reading the sentence aloud as it is shown on the tablet. When the patient chooses an option, a sound feedback will be given to the patient, the robot will congratulate them or show them the correct option while the interface shows the result too, as shown in Figure 9. The game ends when the patient has answered all questions.

As in the Colors game, there is a time limit for the patient to answer. If they take too long to answer, the robot tries to help them by reading the sentence with the different possibilities.

### 6.3 Replanning events

During the execution of the games we have also considered two possible external events that can interrupt the session and therefore will trigger a replanning event:

- Lost connection with the tablet: In the case of using a external device to play the games, the architecture could lose connection with it, for example if it has no battery and it gets turned off. In this case the robot will interrupt the session to try to reconnect. In the case it manages to reestablish connection, it will resume the session repeating the routine which was interrupted. If after three retries it does no manage to connect, the robot will skip the game and continue the session.
- Patient leaves the training zone: If the sensor does not detect the patient in front of it and the robot, it will notify the architecture. When replanning, the robot will ask the patient where they are and if they can get in front of the robot. When the patient is found, the session will continue as normal.

There can be many other situations in which a replanning could be necessary, such as the robot needing to charge, overheating or falling down. For each of these situations the robot will inform its state to the architecture, which will launch the recovery actions needed after replanning.

## 7 Related Work

Social Assistive Robotics show great potential in therapy and rehabilitation. They have many different applications with children such as post-stroke rehabilitation [12], rehabilitation for cerebral palsy [11] or therapy for children with autism [5]. They are shown to increase engagement and therapy adherence, as well as improving the emotional state of the patients. There is also research into applying SAR for the care of the elderly population. They have been used as exercise coaches [3] and companions [2], among others.

While a lot of use cases for this technology focus on physical training where the robot acts as a coach showing the exercises, there are also several studies that look into the applications of SAR for memory training [10] and cognitive rehabilitation [14]. Language training for patients with language impairment is also one of the uses that have been studied for social robots [17], which can also act as teacher assistants for language learning [18].

## 8 Conclusions and Future Work

In this paper we have have augmented the capacities of the control architecture of the Inrobics Rehab platform to include the control of a tablet and its interfaces during the therapy session. To make use of this new addition we have increased the game catalogue of the Inrobics Rehab platform with two new games: Colors and Gaps, both of which need a tablet to be played. We have described how the typical scenario of the Inrobics robot-assisted therapy can be adapted to the execution of these new games and how they behave in a real world therapy session with the Inrobics Rehab platform.

As future work, the new games implemented should be verified for their use with real users, patients that can benefit from the Inrobics Rehab platform. Both games are suitable for populations of all ages with cognitive impairment, so these several usability studies could be done with different populations to evaluate the impact the games can have in their rehabilitation process and daily lives. Apart from the clinical validation of the exercises implemented, an accessibility study could be done, specially with the elderly population, to evaluate if the interaction is appropriate for users that can have sensory difficulties such as being hard of hearing.

In general, the addition to the control architecture presented in this paper shows great potential for creating many new games to increase the catalogue of the Inrobics Rehab platform, improving its capacity to aid patients of all ages and clinical diagnostics.

## Acknowledgments

This work was partially funded by grants PID2021-127647NB-C21 and PDC2022-133597-C43 from MCIN/AEI/10.13039/501100011033, by the ERDF "A way of making Europe" and Next Generation EU/ PRTR and by the Madrid Government under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17) in the context of the V PRICIT (Regional Programme of Research and Technological Innovation) and by grant PID2021-127647NB-C21 from MCIN/AEI/10.13039/501100011033, by the ERDF "A way of making Europe", and by the Madrid Government under the Multiannual Agreement with UC3M in the line of Excellence of University Professors (EPUC3M17) in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

## References

1. Colledanchise, M., Ögren, P.: Behavior trees in robotics and AI: An introduction. CRC Press (2018)
2. Cooper, S., Di Fava, A., Vivas, C., Marchionni, L., Ferro, F.: Ari: The social assistive robot and companion. In: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). pp. 745–751. IEEE (2020)
3. Fasola, J., Mataric, M.J.: Robot exercise instructor: A socially assistive robot system to monitor and encourage physical exercise for the elderly. In: 19th International Symposium in Robot and Human Interactive Communication. pp. 416–421. IEEE (2010)

4. Feil-Seifer, D., Mataric, M.: Defining socially assistive robotics. In: 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005. pp. 465–468 (2005)
5. Feil-Seifer, D., Matarić, M.J.: Toward socially assistive robotics for augmenting interventions for children with autism spectrum disorders. In: Experimental Robotics: The Eleventh International Symposium. pp. 201–210. Springer (2009)
6. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory & Practice. Elsevier (2004)
7. González, J.C., Pulido, J.C., Fernández, F.: A three-layer planning architecture for the autonomous control of rehabilitation therapies based on social robots. Cogn. Syst. Res. **43**, 232–249 (2017)
8. González, J.C., Garcia, J., Fuentetaja, R., Olaya, A., Fernández, F.: From high to low level and vice-versa: A new language for the translation between abstraction levels in robot control architectures. In: 3rd Workshop on Semantic Policy and Action Representations for Autonomous Robots (SPAR) (2018)
9. Hoffmann, J.: FF: the fast-forward planning system. AI Magazine **22**(3), 57–62 (2001)
10. Louie, W.Y.G., McColl, D., Nejat, G.: Playing a memory game with a socially assistive robot: A case study at a long-term care facility. In: 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication. pp. 345–350 (2012). https://doi.org/10.1109/ROMAN.2012.6343777
11. Malik, N.A., Hanapiah, F.A., Rahman, R.A.A., Yussof, H.: Emergence of socially assistive robotics in rehabilitation for children with cerebral palsy: a review. International Journal of Advanced Robotic Systems **13**(3), 135 (2016)
12. Matarić, M.J., Eriksson, J., Feil-Seifer, D.J., Winstein, C.J.: Socially assistive robotics for post-stroke rehabilitation. Journal of neuroengineering and rehabilitation **4**, 1–9 (2007)
13. McDermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., Wilkins, D.: PDDL - The planning domain definition language (1998)
14. Nault, E., Baillie, L., Broz, F.: Investigating the usability of a socially assistive robotic cognitive training task with augmented sensory feedback modalities for older adults. In: 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). pp. 735–742 (2022). https://doi.org/10.1109/RO-MAN53752.2022.9900825
15. Pulido, J.C., González, J.C., Suarez-Mejias, C., Bandera, A., Bustos, P., Fernández, F.: Evaluating the child-robot interaction of the naotherapist platform in pediatric rehabilitation. Int. J. Soc. Robotics **9**(3), 343–358 (2017)
16. Pulido, J.C., Suarez-Mejias, C., González, J.C., Dueñas-Ruiz, A., Ferri, P.F., Sahuquillo, M.E.M., Vargas, C.E.R.D., Infante-Cossio, P., Calderón, C.L.P., Fernández, F.: A socially assistive robotic platform for upper-limb rehabilitation: A longitudinal study with pediatric patients. IEEE Robotics Autom. Mag. **26**(2), 24–39 (2019)
17. Spitale, M., Silleresi, S., Garzotto, F., Matarić, M.J.: Using socially assistive robots in speech-language therapy for children with language impairments. International Journal of Social Robotics pp. 1–18 (2023)
18. Westlund, J.K., Gordon, G., Spaulding, S., Lee, J.J., Plummer, L., Martinez, M., Das, M., Breazeal, C.: Learning a second language with a socially assistive robot. Almere, The Netherlands (2015)
19. Yoon, S.W., Fern, A., Givan, R.: Ff-replan: A baseline for probabilistic planning. In: ICAPS. p. 352. AAAI (2007)