

A Planning-Based Serious Game for Cognitive Rehabilitation in Multiple Sclerosis

Daniele Baschieri^{*}
University of Bologna
Bologna, Italy
daniele.baschieri@gmail.com

Mauro Gaspari
University of Bologna
Bologna, Italy
mauro.gaspari@unibo.it

Floriano Zini
University of Bologna
Bologna, Italy
floriano.zini@unibo.it

ABSTRACT

Many patients suffering from multiple sclerosis encounter cognitive impairment. The use of computerized serious games could play an essential role in their cognitive rehabilitation, especially when these games are designed to train the patient with intellectual disability on tasks typically required in their daily activities. In this paper, we present a serious game for the rehabilitation of executive functions, high-level cognitive processes that people activate to achieve their everyday goals. The game consists in executing, via a dedicated graphical interface, the actions that are necessary to complete a set of common tasks. Automated planning techniques are used to generate a potentially unlimited number of game scenarios with parametrized difficulty, and to verify if the patient's game strategy is leading to the correct completion of the tasks. An evaluation of the game usability is also presented.

CCS CONCEPTS

- **Applied computing** → **Health care information systems**;
- **Human-centered computing** → *Empirical studies in HCI*;
- **Computing methodologies** → *Planning and scheduling*;

KEYWORDS

Cognitive rehabilitation, serious games, planning, usability

ACM Reference Format:

Daniele Baschieri, Mauro Gaspari, and Floriano Zini. 2018. A Planning-Based Serious Game for Cognitive Rehabilitation in Multiple Sclerosis. In *International Conference on Smart Objects and Technologies for Social Good (Goodtechs '18)*, November 28–30, 2018, Bologna, Italy. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3284869.3284916>

1 INTRODUCTION

Multiple sclerosis (MS) is a disease that is not easily categorizable from the point of view of the cognitive disorders that it determines.

^{*} This research is an elaboration of Baschieri's master thesis at the Department of Computer Science and Engineering, University of Bologna. Baschieri is now at Blue Reply, Milano, Italy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Goodtechs '18, November 28–30, 2018, Bologna, Italy

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-6581-9/18/11...\$15.00

<https://doi.org/10.1145/3284869.3284916>

Therefore, the cognitive rehabilitation for a MS patient should be a personalized process carefully planned by his/her rehabilitation team according to his/her specific characteristics and impairment. MS-rehab [4] is a computerized system for cognitive rehabilitation specifically designed for MS. It allows the rehabilitation team to set up a rehabilitation program tailored to the patient's goals and cognitive profile, and supports both the patients and the clinicians in its execution.

In this paper, we extend MS-rehab by including a serious game for the rehabilitation of executive functions, high-level cognitive processes that people activate to achieve their goals. Being the executive functions of vital importance in the daily life of patients, we designed our game in order to be *ecological* (the tasks the patient executes during the game resemble everyday life activities) and to satisfy other requirements for a computerized exercise for cognitive rehabilitation.

In the game, called "Day of Commitments", the patient is presented a scenario including a set of tasks to be done during the day, a map of the city where the tasks have to be completed, and a set of temporal constraints for the tasks. The patient uses a dedicated graphical user interface to move on the map and execute the actions he/she thinks are necessary to carry out all the tasks without violating the constraints.

The implementation of "Day of Commitments" is based on *PDDL4J (Planning Domain Description Library for Java)* [9], an open source Java toolkit that provides state-of-the-art planners based on the *Planning Domain Definition Language (PDDL)*. In "Day of Commitments", we exploited PDDL4J in two ways: to implement a *Scenario Generator*, able to generate an almost unlimited number of solvable exercises, and to implement a *Game Engine*, able to verify, after any action executed by the patient, if a plan to solve the scenario still exists. In order to make the game *auto-adaptive* to the patient's solving capabilities, we also defined a function that calculates the patient's performance on executed exercises. It considers how much the (partial) plan found by the patients to solve an exercise is "close" to the optimal plan.

The rest of the paper is structured as follows: Section 2 presents related work; the serious game we have realized is described in Section 3, by focusing on its requirements, GUI, and implementation; an evaluation of the game usability is given in Section 4; Section 5 concludes the paper.

2 RELATED WORK

The idea to use planning for rehabilitation is not new. A first example is the work presented in [10], where an assistive technology based on planning is applied to personal reminder systems,

which can help people continue satisfactory performance of routine activities despite their cognitive decline. In another research [11], Classical and Hierarchical Task Network Planning are used and compared for the automatic generation of therapy plans for patients suffering obstetric brachial plexus palsy, in response to given therapeutic objectives.

Both these researches do not use a gamified approach to rehabilitation, which is used instead in [7, 12]. As a matter of fact, the adoption of serious games for cognitive rehabilitation is more and more fostered by therapists [1]. The work described in [12] introduces a web based Serious Games platform for Cognitive Rehabilitation that includes natural and multimodal user interfaces and social features, which can contribute to augment the motivation of patients during the rehabilitation process. The platform was tested with healthy subjects and the evaluation showed their motivation and interest while playing the games. In [7], the focus is on a serious game that presents 3D virtual environments to patients, with the goal of aiding them in the development of compensatory navigation strategies.

A general approach for the use of planning to automatically generate scenarios for serious games dedicated to training people is investigated in [2]. A prototype to train the First Aid treatment of burns was evaluated with experienced instructors: they concluded that the prototype produced scenarios were at least as good as those defined by laymen.

To the best of our knowledge, our proposal is the first attempt to apply automated planning to the development of ecological serious games for cognitive rehabilitation.

3 THE GAME

3.1 Requirements

The design of the serious game we propose is based on some of the requirements for a MS rehabilitation computerized system we elicited by analyzing the rehabilitation procedures adopted in specialized MS centers [4]. In particular, our game satisfies the following requirements:

- (1) the game is an *ecological exercise*, i.e., game instances are *representative* (the tasks the patient executes during the game resemble everyday life contexts) and *generalizable* (the tasks relate to behaviors outside the rehabilitation environment);
- (2) the patient is always offered new rehabilitative challenges by providing variegated game scenarios that involve him/her in a crescendo of difficulties that stimulate more and more his/her executive functions;
- (3) the patient plays games whose initial difficulty is tailored to his/her cognitive profile, which includes detailed information about clinical and neuropsychological conditions;
- (4) the game is auto-adaptive, i.e., the difficulty of the next instance proposed to the patient varies depending on his/her performance in the previous ones; the rehabilitation treatment is at the *threshold level*, i.e., games are always difficult enough to improve the executive functions, but not too difficult, and thus impossible to solve.

3.2 Game Description

The design of the game is inspired by the “Plan-a-day” approach [3], whose ecological validity to test planning ability has been proven [6]. In addition, in order to further improve the ecological features of the game (Requirement 1), the design was contributed by an expert psychologist.

When a patient plays “Day of Commitments”, he/she is first presented a set of common tasks to be executed during the day, a set of time constraints related to the places included in the game scenario, and a key for the city map describing the scenario. For example, Figure 1 in the upper part shows three time constraints and two tasks to be accomplished, while the lower part is dedicated to the key.

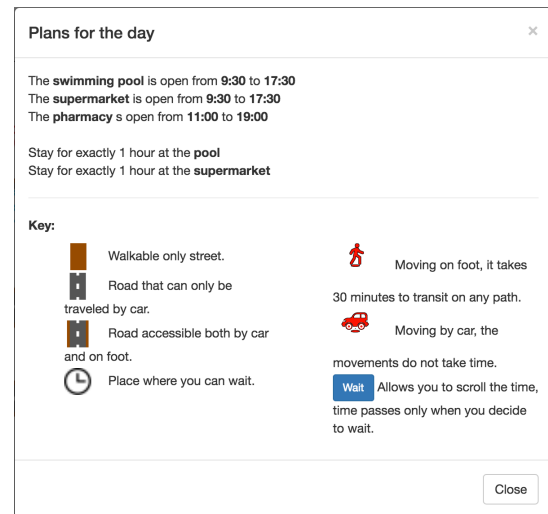


Figure 1: Tasks, constraints, and key.

After reading and understanding the tasks, the patient can start the game. The map representing the game scenario is shown (see Figure 2) and the patient can move from one place to another along the connecting roads, either on foot or by car (if he/she has it). Near each place, its opening hours are shown. At any location the patients can:

- pick one of the items at the place;
- leave an item he/she has;
- exchange an item with another one at the place;
- wait (only if a clock sign is shown).

For example, Figure 2 shows a game situation where the current time is 9:30, and the patient is at the swimming pool. The pool is open, the patient has already picked the diving mask from the pool (this is shown in the central column of the figure), and can pick the other items shown on the right side of the figure. The patient can also wait in this location by clicking on the blue “Wait” button and by choosing how long to wait. At any time, the patient can also see again the tasks for the day by clicking on the blue “Tasks to do” button on the top left corner of the interface.

The game continues until the patient is able to complete all the tasks before the end of the day without violating the given constraints, or he/she fails to find a solution. At the end of the game,

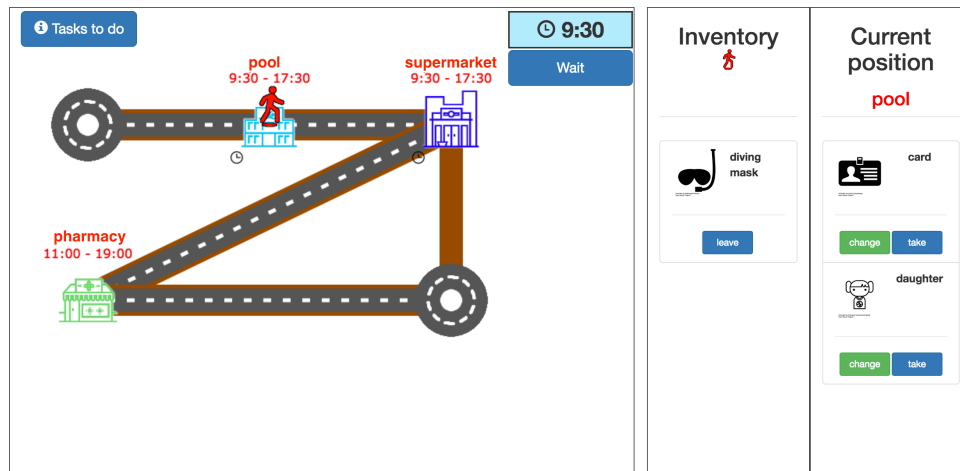


Figure 2: City map and game state.

the patient is shown his/her performance (see Section 3.6). If the performance¹ overcomes 80% of the maximum for two consecutive games, the difficulty is automatically incremented. The system is therefore *auto-adaptive* (Requirement 4): it is able to automatically change the difficulty of a game according to how the patient performs.

The interaction between the actors and “Day of Commitments” is formalized in Figure 3. In addition to the patient-system interaction described above, the sequence diagram shows the interaction of the clinician with the system. The clinician can assign the game to the patient, choosing a difficulty according to his/her cognitive profile (Requirement 3). The database includes a potentially unlimited number of game instances, categorized by various levels of difficulty. When the patient starts the game, a game instance of the selected difficulty is returned.

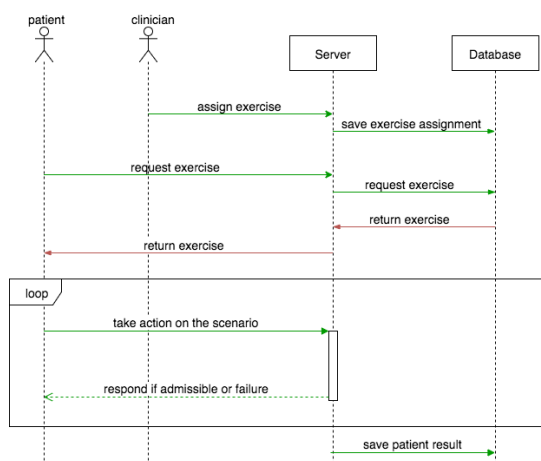


Figure 3: Interaction with “Day of Commitments”.

¹Details on how performance is calculated are given on page 5.

In order to obtain the system behavior described in Figure 3, we implemented two components of the Server: a *Game Engine* and a *Scenario Generator*. Before detailing these components, we first describe how PDDL has been used to model “Day of Commitments”.

3.3 PDDL Modeling

The *Planning Domain Definition Language (PDDL)* separates the model of the planning problem in two major parts: the description of the *domain* and the description of the *problem*. All aspects common to a class of problems (as the types of the objects, the predicates on the objects, or the legal actions that change a problem state into another state) are formalized in the domain. The problem includes the description of the specific objects, the initial state, and one or more goal states (states that have to be reached in order to solve the problem). A plan is a sequence of actions that leads from the initial state to one of the final states.

Figure 4 partially describes the domain for “Day of Commitments”. Three actions are specified: the action *move-to* is used to move on foot from one place to another; the action *take-item-at* is used to pick an item from a place; the action *wait-1unit* is used to wait for one time unit (i.e., 30 minutes). More in details, the action *move-to* has four parameters: a source and a destination (of type *place*) and two times. The patient can execute the action if the preconditions are satisfied, i.e., he/she is at the source place, source and destination are connected, t_0 is the current time, and t_1 is the next future time. The effects caused by the execution of *move-to* are that t_1 is now the current time and at that time the patient is not anymore at the source place but he/she is at the destination. Parameters, preconditions, and effects of the other actions are self-explanatory.

Figure 5 is the problem description of the “Day of Commitments” instance illustrated in Figure 2. First, the objects of the game are listed: the 48 time unit of the day; the places of the city (some of them are places where the patient can wait); the car; and the items to be possibly picked. Then, the *:init* keyword introduces the current state of the game: the game time (the p_{19} time unit corresponds to 9:30); the topology of the city, with indication whether

```

(define (domain day_of_committments)
  .....
  (: action move-to
   :parameters (?src - place ?dst - place
                ?t0 - time ?t1 - time)
   :precondition (and
                  (be ?src) (path ?src ?dst) (now ?t0) (next ?t0 ?t1)
                  (future ?t1)
                  )
   :effect (and
            (not (be ?src)) (be ?dst) (not (now ?t0))
            (not (future ?t1)) (now ?t1) (past ?t0)
            (be-at ?dst ?t1)
            )
   )
  (: action take-item-at
   :parameters (?pos - place ?item - item
                ?tp - time ?t - time ?tc - time ?tnow - time)
   :precondition (and
                  (be ?pos) (have ?pos ?item) (open-hour ?pos ?t ?tc)
                  (past ?tp) (next ?tp ?t) (now ?tnow) (future ?tc)
                  )
   :effect (and
            (not (have ?pos ?item)) (got ?item)
            (got-at ?item ?pos ?tnow)
            )
   )
  (: action wait-1unit
   :parameters (?pos - waitplace ?t0 - time ?t1 - time)
   :precondition (and
                  (now ?t0) (next ?t0 ?t1) (future ?t1) (be ?pos)
                  )
   :effect (and
            (not (now ?t0)) (not (future ?t1)) (now ?t1)
            (past ?t0) (stand-1unit ?pos) (be-at ?pos ?t1)
            )
   )
  .....
)

```

Figure 4: “Day of Commitments”: domain partial view.

roads between places are only walkable or can also be traveled by car; the opening hours of the places and the items located at them; and the current and past positions of the patient, as well as the items in his/her hands. Finally, the tasks to be accomplished: since none of them belongs to the current state, both have still to be executed.

3.4 Scenario Generator

Serious games for cognitive rehabilitation usually have a small number of game scenarios. To overcome this limitation and provide innumerable ecological exercises (Requirement 2) we developed a flexible *Scenario Generator*. The scenario generator consists of two modules:

- a *Map Generator* creates a realistic environment with buildings, roundabouts, and items;
- a *Problem Generator*, given a map, generates a solvable problem for that map.

The generators work on two pre-defined lists. The first includes about 10 places (e.g., *bank*, *home*, *station*) that are potentially part a scenario; for each place, we specified a set of attributes, such as the

```

(define (problem day)
  (: domain day_of_committments)
  (: objects
   p0 ... p48 - Time
   roundabout00 - place pool - waitplace
   supermarket - waitplace pharmacy - place
   ...
   car - vehicle
   syrup - item eyewash - item divingmask - item
   vegetables - item daughter - item fruit - item
   )
  (: init
   (first-number p1)
   (next p0 p1) ... (next p47 p48)
   (past p0) ... (past p18) (now p19)
   (future p20) ... (future p48)

   (path roundabout00 pool) (path pool roundabout00)
   (path pool supermarket) (path supermarket pool)
   (road pool supermarket) (road supermarket pool)
   ...

   (open-hour pool p19 p35)
   (have pool card) (have pool daughter)
   (open-hour pharmacy p22 p38)
   (have pharmacy pills) (have pharmacy syrup)

   (be pool)
   (be-at pool p17) (be-at pool p18) (be-at pool p19)
   (got divingmask) (got-at divingmask pool p19)
   )
  (: goal (and
           (stand-2unit pool) (stand-2unit supermarket)
           )
   )
)

```

Figure 5: “Day of Commitments”: problem description.

opening hours, the items that can be taken or left, and the actions the patient can execute. The second list includes about 25 items; for each item we specified the actions the patient can execute on it.

The map generator initially selects a random number of places, connects them with main roads (to assure a fully connected graph), and adds further random connections. Then, the generator adds to each place a random number of items (among those associated with it). This approach ensures randomness in the structure of the game without renouncing consistency.

Subsequently, the problem generator creates a set of PDDL goals, i.e., the tasks to be executed by the patient. Goals are related to staying (e.g., *be-at*, *stand*) or to obtaining and releasing items (e.g., *got-at*, *drop-at*). Each goal is created by randomly selecting a PDDL predicate and populating its fields coherently with the previously generated map. For example, if the selected predicate is *got-at*, the generator filters out among all the items included in the map those that can be collected, and then, on the basis of the place where the item is located, chooses a time to collect it within the opening hours times of the place. The created goals are added to the problem one by one, until the problem has reached the required difficulty. Each time a goal is added, the obtained (partial)

problem is checked for solvability using the Heuristic Search Planner (HSP) provided by PDDL4J, with Fast Forward heuristics to speed up solvability.

The exercise difficulty is empirically defined as $d = steps - min_steps + (place_num/3)$ where $steps$ is the minimum number of actions necessary to solve the exercise, min_steps is the number of actions needed to solve the easiest exercise, and $place_num$ is used to increase the difficulty with the size of the map, as simple problems may seem more complicated on larger maps.

3.5 Game Engine

A generated problem is solvable by construction and has an optimal solution². While the patient is playing “Day of Commitments”, the *Game Engine* is invoked after each action (see the loop in Figure 3) to check if he/she is proceeding towards the optimal or an alternative (sub-optimal) solution. More in details, the *Game Engine* receives the patient’s action as input and creates a new problem according to the effects of the executed action (see Figure 4). Then, the engine checks if the problem is still solvable. In this case, the control returns to the patient, which executes the next action. Otherwise, the patient is notified that the game no longer has a solution.

3.6 Patient Performance

A game can terminate with success (i.e., the patient is able to find a sequence of actions for the execution of all the given tasks) or with failure. In the second case, a patient could have been able to find a partial solution that is “close” to the optimal one. We have defined a function to calculate the patient performance that formally captures this closeness.

The patient performance is a combination of three variables characterizing an exercise: pC (number of correct actions performed); pW (number of incorrect actions performed); pM (number of actions the patient should have performed, but have not). To define the formula for the performance, we introduce *precision* ($p = \frac{pC}{pC+pW} \in [0, 1]$) and *recall* ($r = \frac{pC}{pC+pM} \in [0, 1]$). The function to calculate the performance *per* of a patient is nothing but an *F-measure*:

$$per = (1 + \beta^2) * \frac{p * r}{\beta^2 * p + r} \quad (1)$$

The parameter $\beta > 0$ weights precision and recall in the calculation of the performance. We set $\beta = 1$, a typical value which gives equal importance to precision and recall.

The values of pC , pW , and pM can be calculated as follows. Each exercise E has an optimal solution O , a sequence of actions of cardinality $|O|$ that allows all the goals of E to be satisfied. If the patient is able to complete the exercise successfully, he/she will produce a solution S that can be equal to O (and therefore optimal), different from O but equally optimal (if $|S| = |O|$) or different from O and not optimal (if $|S| > |O|$). In all the three cases, we will have that $pC = |O|$, $pW = |S| - |O|$, and $pM = 0$.

If instead the patient is not able to complete the exercise, then we consider the *partial solution* S_p to which he/she had arrived

²Even though this solution is not assured to have the minimum number of actions (as the Fast Forward heuristics does not always find optimal plans) in practice it is close to the optimum and for simplicity we call it optimal.

before doing the action a_f that led to failure. We also consider the *remaining solution* S_r , i.e., the sequence of actions that, chained to S_p , allows to solve the problem. If $|S_p \circ S_r| = |O|$, then all the actions in S_p are correct, those in S_r should have been executed to get an optimal solution, and the only wrong action is a_f ; therefore, we will have

$$pC = |S_p| \quad pW = 1 \quad pM = |S_r| \quad (2)$$

If $|S_p \circ S_r| > |O|$, then:

- in S_p , the sequence of performed actions, there are some correct and some incorrect actions, that is, $0 < pC < |S_p|$, $0 < pW < |S_p|$, and $pC + pW - 1 = |S_p|$, (there is at least one wrong action, namely a_f);
- in S_r , the remaining solution, there are some actions that should have been performed to achieve all the goals (but they were not done) and some actions that, if they had been done, they would have been wrong; omitting these last ones (which were not executed and cannot be counted as incorrect), we have that $0 < pM < |S_r|$.

To estimate pC , pW , and pM we can use the following formulas:

$$pC = \left\lfloor \frac{|O| * |S_p|}{|S_p \circ S_r|} \right\rfloor \quad pW = |S_p| - pC + 1 \quad pM = \left\lfloor \frac{|O| * |S_r|}{|S_p \circ S_r|} \right\rfloor \quad (3)$$

where $\lfloor x \rfloor$ indicates the integer closest to x . The idea behind these formulas is that the correct actions are included in the same proportion in S_p and S_r . The three formulas correspond to the previous ones (Equation 2) if $|S_p \circ S_r| = |O|$.

4 GAME USABILITY

For an application aimed at the neurocognitive rehabilitation of the executive functions in patients with MS, it is essential to carry out some usability tests [5]. We chose a user testing protocol called *Discount usability testing*, better known as Guerrilla [8]. This type of test is suitable when one wants to quickly evaluate a software without going too much into details.

We involved seven users in our test, chosen via convenience sampling in our acquaintances. Since MS principally affects people over 50, we privileged people in that age range. We did not enroll real patients as it would have been too onerous for a quick evaluation. The participants were convened in a appropriate environment, and they were briefly explained the test. Then, they were given a document describing the test tasks to be executed, and a tablet PC ready for the exercise execution. We adopted a *Thinking Aloud* methodology, i.e., we asked the users to reason out loud about what was happening on the device. Moreover, a researcher observed the user and classified possible errors into three categories: [*Catastrophic*] the user does not conclude the task; [*Serious*] the user concludes the task with significant slowdowns; and [*Cosmetic*] the user identifies small problems in carrying out the task.

Table 1 summarizes the results of our evaluation. The first column is the list of the test tasks, the second records the average satisfaction of the users for executing the task with the user interface (they expressed their satisfaction on a 1 to 5 Likert scale), and the third summarized the occurred errors. The overall satisfaction about the user experience with “Day of Commitments” was 4.0, all

| Task | Satisfaction | Notes |
|--------------------------------------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Understanding the text of an exercise | 4.0 | All users understood the text of the exercise and were able to understand how to do it correctly; nevertheless, at first glance, many of them felt confused by the lack of order in the goals to be achieved, and because they did not know whether or not it was necessary to memorize the objectives. |
| Starting an exercise | 5.0 | All users started the exercise without difficulty. |
| Moving to a new place | 4.0 | A rather common <i>[Cosmetic]</i> mistake was the attempt to touch the destination to move there; after this first attempt the majority of users managed to understand that to move the cursor it was necessary to drag it along the screen. |
| Collect an item | 5.0 | All users have easily managed to collect an item in the course of the exercise. |
| Seeing the list of tasks to be done | 5.0 | All users have readily understood how to visualize the list of tasks. |
| Waiting in a place for the required time | 3.3 | This was the most frustrating and difficult task compromising the usability of the software with both <i>[Cosmetic]</i> and <i>[Serious]</i> errors; some users pressed on the clock and not on the button below, others have wasted several seconds wondering how to execute the task. |
| Identifying the opening hours of the shops | 5.0 | All users found it extremely easy to identify opening hours. |
| Completing an exercise successfully | 4.3 | All users successfully completed the exercise. |
| Stopping an exercise | 5.0 | All users managed to stop an exercise. |

Table 1: Evaluation results.

users were able to play the exercise without any explanation on how it works.

After the usability test we ameliorated the GUI of the game. In particular we:

- (1) better explained how the player can perform a wait action in the instructions;
- (2) modified the box containing the current time and the blue “Wait” button (see Figure 2) by increasing its size, adding a clock icon near the time, and making clickable also the clock display and not just the button below it;
- (3) explained in the key (Figure 1) that moving on foot between two contiguous places takes 30 minutes, while the shift is immediate if done by car;
- (4) changed the icon used to identify the player on the game board, as several testers complained about it: we adopted a human figure when the player moves on foot, and a car figure when the player drives a car.

5 CONCLUSIONS

In this paper, we proposed the use of automated planning for the design and implementation of “Day of Commitments”, a rehabilitative serious game dedicated to patients suffering from MS. The game was designed to train the executive functions of the patients on ecological scenarios that resemble everyday life contexts. The usability evaluation we performed suggested that the game is suitable to be used by MS patients.

Other rehabilitative serious games are currently being implemented. They exploit automated planning to train the long term and visual planning abilities of the patients, and to provide system features that the patient can use to receive hints about actions to be executed, or verify if the exercise is still solvable.

ACKNOWLEDGMENTS

We would like to thank Dr. Debora Castellano for her contribution in the design of our serious game. We also thank the anonymous reviewers for their useful suggestions and comments. This research

was partially supported by Fondazione Del Monte and Delia Genari ONLUS.

REFERENCES

- [1] Ahmed Mohammed Elaklouk, Nor Azan Mat Zin, and Azrulhizam Shapii. 2015. Investigating therapists’ intention to use serious games for acquired brain injury cognitive rehabilitation. *Journal of King Saud University - Computer and Information Sciences* 27, 2 (2015), 160 – 169.
- [2] Gwen R. Ferdinandus, Marieke Peeters, Karel van den Bosch, and John-Jules Ch. Meyer. 2013. Automated Scenario Generation - Coupling Planning Techniques with Smart Objects. In *Proceedings of the 5th International Conference on Computer Supported Education - Volume 1: CSEDU*. INSTICC, SciTePress, 76–81. <https://doi.org/10.5220/0004354600760081>
- [3] J. Funke and T. Krüger. 1995. Plan-A-Day: Konzeption eines modifizierbaren Instruments zur Führungskräfte-Auswahl sowie erste empirische Befunde [Plan-A-Day: Design of a modifiable instrument for selecting managerial personnel and first empirical results]. In *Neue Konzepte und Instrumente zur Planungsdiagnostik*, J. Funke, T. Krüger, and A. Fritz (Eds.). Deutscher Psychologen Verlag, 97–120.
- [4] Mauro Gaspari, Floriano Zini, Debora Castellano, Federica Pinardi, and Sergio Stecchi. 2017. An advanced system to support cognitive rehabilitation in multiple sclerosis. In *Proceeding of IEEE RTSI 2017, 3rd International Forum on Research and Technologies for Society and Industry*. IEEE, Modena, Italy.
- [5] Fátima González-Palau, Manuel Franco, José M Toribio, Raquel Losada, Esther Parra, and Panagiotis Bamidis. 2013. Designing a Computer-based Rehabilitation Solution for Older Adults: The Importance of Testing Usability. *PsychNology Journal* 11, 2 (2013).
- [6] D. V. Holt, K. Rodewald, M. Rentrop, J. Funke, M. Weisbrod, and S. Kaiser. 2011. The Plan-a-Day approach to measuring planning ability in patients with schizophrenia. *J Int Neuropsychol Soc* 17, 2 (Mar 2011), 327–335.
- [7] Milan N. A. van der Kuil, Johanna M. A. Visser-Meily, Andrea W. M. Evers, and Ineke J. M. van der Ham. 2018. A Usability Study of a Serious Game in Cognitive Rehabilitation: A Compensatory Navigation Training in Acquired Brain Injury Patients. *Frontiers in Psychology* 9 (2018).
- [8] Jakob Nielsen. 1994. Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier. In *Cost-justifying Usability*, Randolph G. Bias and Deborah J. Mayhew (Eds.). Academic Press, Inc., Orlando, FL, USA, 245–272.
- [9] Damien Pellier and Humbert Fiorino. 2017. PDDL4J: a planning domain description library for java. (12 2017), 1–34.
- [10] Martha E Pollack. 2002. Planning Technology for Intelligent Cognitive Orthotics. In *Proceedings of the 6th International Conference on Artificial Intelligence Planning Systems*. AAAI, 322–332.
- [11] José Carlos Pulido, José Carlos González, Arturo González-Ferrer, J García, Fernando Fernández, Antonio Bandera, Pablo Bustos, and C Suárez. 2014. Goal-directed Generation of Exercise Sets for Upper-Limb Rehabilitation Assisted by Humanoid Robots. In *Proceedings of the 5th Workshop on Knowledge Engineering for Planning and Scheduling*, 38–45.
- [12] Paula Alexandra Rego, Rui Rocha, Brígida Mónica Faria, Luís Paulo Reis, and Pedro Miguel Moreira. 2016. A Serious Games Platform for Cognitive Rehabilitation with Preliminary Evaluation. *Journal of Medical Systems* 41, 1 (2016).