Exploring intelligent agents for controlling user navigation in 3D games for cognitive stimulation

R M E M Costa, D S Souza, I Mendonça

Departamento de Informática e Ciência da Computação
Universidade do Estado do Rio de Janeiro – UERJ,
Rua S. Francisco Xavier 524 – 6º B
20550-013 – Rio de Janeiro, BRASIL

rcosta@ime.uerj.br, diodss@hotmail.com, israel_mendonca@yahoo.com.br

ABSTRACT

Despite the many research efforts addressing the development of environments for Virtual Rehabilitation processes, only few of them consider a modeling step before system implementation. Our previous experience in developing virtual environments for stimulating cognitive functions stresses the necessity of adopting some Software Engineering practices. These open new possibilities to extend or integrate the system with others applications. The objective of this paper is to introduce some technical aspects related to modeling and implementing a multi-agent game for training memory and attention. We explore the integration of multi-agent model methodologies and present initial results of an experiment made with two specific languages for building three-dimensional virtual environments.

1. INTRODUCTION

Over the past few years, there has been an increasing amount of Virtual Rehabilitation research. In general, patients use these environments in sessions under the control of a therapist. However, due to growing demand, there is a need for environments that can be used by patients in their own home. At-home therapy allow in principle, better customization, more timely patient evaluation, and more access to care.. These environments must have an automatic control of the patients' evolution, simulating the therapist interventions. The emergence of Artificial Intelligence techniques opens new possibilities to support this process, exploring multi-agent systems.

Intelligent agents are able to have an adaptive and independent behavior, controlling tasks, interacting with humans, simulated objects and events (Wooldridge, 2009). These agents can be software entities embedded in the system or virtual characters in virtual worlds (Yang, 2008). Jennings (2002) considers that the agents work as assistants to users, supporting them to carry out tasks. In this context, several researchers, among them, Araújo et al. (2010), Bouchard et al. (2006) and Yang et al. (2008), used intelligent agents to increase the involvement of the user with the virtual environment (VE).

Despite the three-dimensional (3-D) virtual environments being frequently used in the neuropsychiatry, the use of intelligent agents for monitoring the patients' activities is rare. An example is the Grillon & Thalmann (2008) application that uses an eye-tracking system to control interactive virtual characters that change their behavior when being observed.

While VEs are recognized as an important tool for the rehabilitation, we need to study the impact of the integration of the new intelligent strategies in these environments. In this sense we must consider human as well as methodological and technical aspects.

The purpose of our study is to model and implement intelligent agents, integrating them with virtual 3-D environments. This paper discusses the necessity to consider some Multi-agent methodologies to create VE and presents initial results of an experiment that compares two different programming languages to construct 3-D virtual environments. Finally, we discuss several possibilities to test this approach with people with various neuropsychiatric disorders.

This work is organized into 6 sections. Section 2 gives an overview of the Virtual Reality in the rehabilitation area and Section 3 describes an experiment that explored the agent technology. Section 4 presents the integration of agent modeling methodologies. In Section 5 the main characteristics of serious games for memory and attention stimulation are presented. Section 6 concludes the work and presents future research directions.

2. VIRTUAL REALITY FOR REHABILITATION

Virtual Reality includes advanced interface technologies, immersing the user in environments that can be actively interacted with and explored. The user can also accomplish navigation and interaction in a 3-D synthetic environment, using multisensorial channels. In this case, various types of stimuli can be transmitted by specific devices and perceived by one or more of the user's senses (Costa,2004) (Burdea and Coiffet, 2003).

Three-dimensional virtual environments present real-time graphics rendered by a computer, in which the user, via body position sensors or user-input devices, controls the viewpoint or the orientation of displayed objects (Pinho, 2002).

In the last few years, the potential of VE for the study and rehabilitation of human cognitive/functional abilities has been recognized. Therapy software aims to develop human capabilities and to reduce limitations, exploring different technical aids. The Virtual Reality and associated technologies provides opportunities to broaden physical and mental health applications, providing valuable scenarios with common elements for the patients, putting them in contact with daily life activities.

Recently, several experiments in this area describe case studies with positive results, which are stimulating new research that explore intelligent strategies (Dawson, 2008), (Broeren, 2008). However, in many applications there is a strong dependence on therapist's intervention to control the patient's navigation, to change the level of difficulty of a task, go back to an earlier stage, or control the order of activities in the synthetic environment. These interventions, however needed, cause distractions, reducing the level of user immersion in the simulation. As an alternative, the inclusion of intelligent agents could help to alleviate this problem, by reducing the need of therapist involvement.

3. INTELLIGENT AGENTS IN VIRTUAL ENVIRONMENTS

An agent is a software processes that are capable of independent action on behalf of its user or owner, without having to be told explicitly what to do in a given moment (Wooldridge, 2009).. An agent oriented paradigm reinforces the software flexibility and the agent's social possibilities, taking space as a solution in software engineering. Intelligent agents are used as software entities embodied in systems or as a virtual character that is part of the virtual world (Yang, 2008).

The use of an agent modeling methodology allowed the understanding of all the agents' features, as well as their communication processes. However, in general, the articles cited here do not mention the use of specific methodologies for modeling multi-agent systems. In order to fully take advantage of the opportunities offered by intelligent agents in VEs, we need to first understand how agents communicate and interact with each other.

In a perspective of educational use, there are some projects that explored intelligent agents for the support of users' learning. André et al. (1999) defined a framework for the development of presentation agents which can be utilized for multimedia help instructions. To control the agent behavior, the researchers used a decision tree and developed a declarative specification language to decide the agent behavior. The agents have a 2-D image and the objects they point to have a 3-D image. The researchers specified some pre- and post- conditions for each agent action. However, they did not mention any formal agent modeling methodology.

Johnson (2000) presented a system that has a life-like character which provides problem-solving advice in real-time, for learning environments. This work does not mention a formal methodology to model the agent behavior, but they developed a deictic model to combine the agents' speech, locomotion and gestures.

Pessin (2007) used a blackboard model to simulate the agents' conversation model. But, they do not stress any other methodology to model the agents' behavior.

Our research group developed a VE that has an Interface agent with a human-like appearance, to help patients accomplish some tasks in a kitchen (Cardoso, 2006). The system was modeled according to the precepts of the UML (Unified Modeling Language) with the definition of Use Cases, Sequence Diagrams and Description of Use Cases. The environment consists of a house with living room, two bedrooms, kitchen and

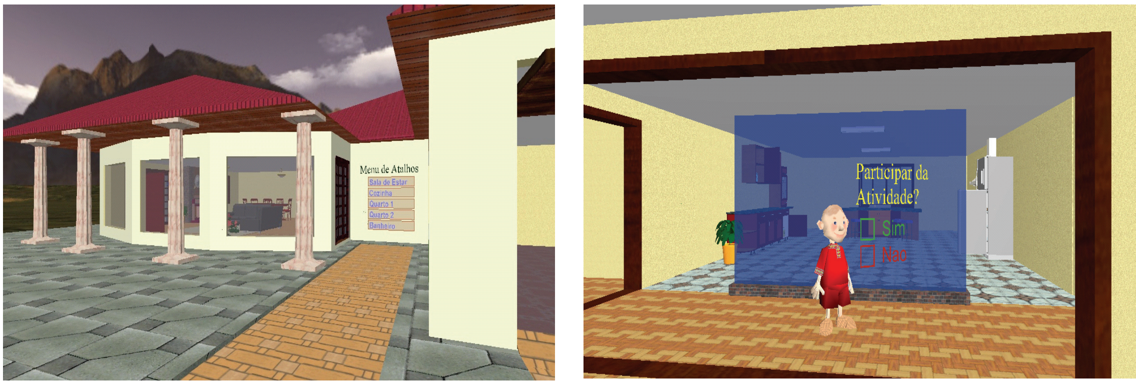
Proc. 8th Intl Conf. Disability, Virtual Reality & Associated Technologies
Viña del Mar/Valparaíso, Chile, 31 Aug. – 2 Sept. 2010
©2010 ICDVRAT; ISBN 978 07049 15022

bathroom. Each room provides cognitive stimulation activities. The interface agent (named “Bob”) is in the kitchen, where he will support the accomplishment of tasks associated with activities of daily living.

When the user approaches the kitchen, the agent asks if he wants to attend an activity. If the user answers “Yes”, the agent proposes to carry out a task: choosing the ingredients and utensils to make an omelet. The user must then open the fridge, click on the egg, open the cabinet and click on the pan, open another cabinet and click on the salt and oil. If the person misses some of these tasks, the agent speaks, giving tips and proposing an ingredient or utensil. If the user takes too long to make the decisions, the agent perceives the time delay and repeats the instructions, until the goal is reached.

The environment was developed in VRML (Virtual Reality Modeling Language) (Ames, 1997)). The functionality of the agent was implemented in JavaScript (JAVAScript,2010), which is the language that best fits VRML. The interface agent manages the tasks and decides what advice to give, from a decision tree, which considers the number of errors and the number of steps already taken on the task completion sequence.

Figure 1a presents the image of the house facade, showing a shortcut menu for users who have difficulty handling the computer mouse. Figure 1b shows the entrance to the kitchen, the agent Bob and a panel (with Portuguese text) where the patient must click to demonstrate their interest in participating in the activity.



(a) (b)

Figure 1. *Image of the facade of the house (a) and the entrance of the kitchen and the agent (b).*

After the first tests, we noticed that the behavior of the agent needed to be modified. However, since Bob’s behavior did not involve modeling, it was difficult to make specific modifications, or integration with other environments. This problem led us to adopt a formal agent modeling methodology when authoring VEs containing intelligent agents.

4. MULTI-AGENT MODELING METHODOLOGY

The main purpose of the Multi-agent System Engineering (MaSE) methodology is to support the programmer in setting initial requirements and analyzing, drawing, and implementing a multi-agent system (MAS) (Gago, 2009). This methodology is independent of any agent’s architecture, programming language, or communication framework. The MaSE treats the agents as a deeper object orientation paradigm, where the agents are object specializations. Instead of simple objects, with methods that can be invoked by other objects, the agents talk among themselves and proactively act in order to reach goals (Deloach, 2001), (O’Malley, 2001).

MaSE is a specialization of traditional software engineering methodologies with two phases (Analysis and Design) and several activities. However, the MaSE has some fragility to model the conversations between agents. For this reason we adopt some ADELFE (Atelier de Développement de Logiciels à Fonctionnalité Emergente) approaches.

According to Bernon (2003), ADELFE guarantees that the software is developed according to the Multi-agent systems theory. It follows the Rational Unified Process (RUP) (Jacobson, 1999) and uses UML (Unified Modeling Language) and AUML (Agent Unified Modeling Language) (Odell, 2000) notations. The interactions among agents and the user can be represented in the AUML Protocol Diagram, expressing how the system answers the end-user or the agents’ requests.

Next, we present some characteristics of the games that have agents control the user’s navigation.

5. GAMES FOR COGNITIVE STIMULATION

This research aims to verify the potential of intelligent agents in controlling the level of tasks offered to the user. If a user fails successively in performing a task he tends to feel discouraged. Moreover, a therapist intervention to reduce the task difficulty level can aggravate this situation, because patients can think that they are making many mistakes and feel insecure to answer. Therefore, if the software is able to perceive this weakness and automatically decrease the difficulty level of the exercises, the patient may have a decreasing level of frustration.

The environments have characteristics of serious games and aim to stimulate basic cognitive functions such as attention and concentration. They were built with JAVA3D (Java3D, 2010) and X3D languages (Brutzman, 2007) because these are free languages and offer good programming support and libraries. We constructed two similar scenes where the user must observe a geometric shape for a few seconds. It is presented in a frame on the wall. Afterwards the geometric shape disappears and three shapes appear on a table. Then user must choose the similar object among those on the table. The game has 4 difficulty levels and to change the level, the user must have 10 right answers in a row. When the level increases, the shapes can be duplicated or some colors can be redistributed to increase the degree of difficulty. However, if the user is having low success in choosing the right answers, the intelligent agents will decrease the difficulty level. To implement this control, the agents consider a combination of the number of correct/wrong answers, the time to choose an option, the task difficulty level and the results obtained in previous exercises.

In addition, to avoid common problems related to software product development, it is essential to have the support of procedures based on standards that ensure consistent results. Thus, the construction of VEs requires a process that considers the various steps involved and is appropriate to the peculiarities of Virtual Reality technology. We modeled the environments considering four phases: Requirements that identified the users, tasks and restrictions; Project, which defined software entities, the input and output hardware and modeling tools; the Implementing step made the integration of objects in the scene; and finally, the Assessment phase checks the performance, usability and value of the application.

To arrive to the third phase we made a detailed study of the JAVA3D language. This language offers various programming features and good portability that allows the visualization of virtual scenes in any browser. Figure 2 shows the main scene created with the JAVA3D language.

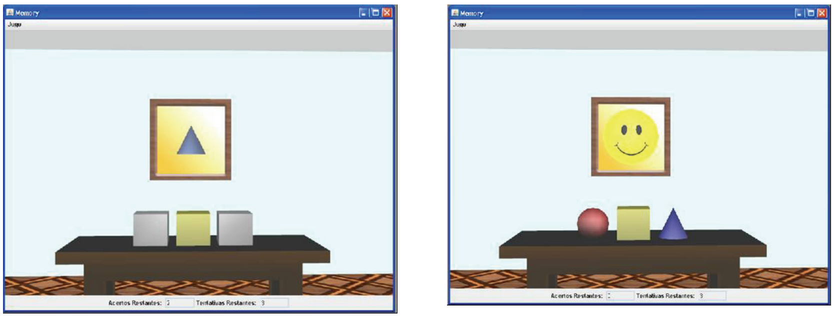


Figure 2. *Views of the game environment implemented in JAVA3D (JAVA3D, 2010).*

The X3D is an extension of VRML (Virtual Reality Modeling Language) (Ames,1997). The X3D combines geometric data and behavior descriptions in a Flash file that can have multiple transmission formats. Figure 3 shows a scene developed with X3D.

Initially, we studied different proposals for modeling intelligent agents, and authoring tools such as NetLogo tool and JADE framework (Wooldridge, 2009). However, these tools have proven inadequate for our purpose because they are not easily integrated with the JAVA3D and X3D. Then, we decided to use the structure provided by the languages, exploring the JAVA (JAVA, 2010) and JAVAScript (JAVAScript, 2010) languages, respectively.

Currently the agents are in the modeling phase. We are using the MaSE methodology (Multiagent System Engineering) (Deloach, 2001), which aims to support designers in the elicitation of the requirements set and its formal representation. For the agent communication model we are using the ADELFE AUML Protocols Diagram that express the way the agents answer an end-user's request.

The agents will control the game to decrease the therapist's interventions, making it more comfortable for the patient. According to a preliminary experiment, we perceived that the applications have good possibility to improve the users' interaction with the environment.

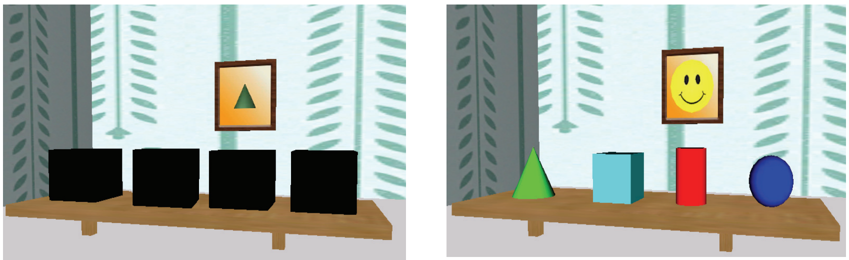


Figure 3. *View of the game environment implemented in X3D (Brutzman, 2007)).*

Until now, we realize that the JAVA3D has a small learning curve, because it uses the same structure as JAVA, such as library calls. The X3D language depends on learning a new language structure. JAVA3D is been more flexible to change the code and to integrate the multi-agent functionalities.

6. CONCLUSIONS

This paper discussed some technical questions related to the VE development process stressing the use of multi-agent approach to open new possibilities to users.

In 2000, the Brazilian Census showed that the number of people who declare themselves as having some kind of disability is around 24.5 million, approximately 14.5% of the population (Sarraf, 2000). These numbers show a high demand for treatment and social reintegration of these people. In general, the physiotherapy treatments are most commonly specified and used with the primary focus on motor rehabilitation. Commonly, the brain functions associated with cognition are neglected or receive a small amount of therapy in these treatments. To overcome this deficiency and make treatment more efficient, there is a growing movement to search procedures and practices which have as their primary purpose the reinclusion of these individuals in society and, by consequence, the minimization of their motor and cognitive disabilities.

Some types of exercises could be done at home, expanding the possibilities of rehabilitation. Thus, in this case, the virtual environment must have some kind of mechanism to control user navigation. This paper discussed some technical questions related to the VE development process, stressing the use of agent approach to open new possibilities to users.

The research described here as well as numerous studies going around the world indicate that virtual environments may be of great value in helping individuals, mainly when training them to live better in the real world. However, the use of Virtual Reality technology in cognitive rehabilitation merits further research to determine new development practices to increase access to care for a growing number of patients.

7. REFERENCES

A. Ames, D. Nadeau, J. Moreland (1997). *VRML 2.0 SourceBook*, 2nd edition, John Wiley & Sons.

E. André, T. Rist and J. Muller (1999), Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence*; **13**, 4, pp. 415-448.

A. Araújo, L. A. Carvalho, R. M. E. M. Costa, (2010), The influence of intelligent characters on users' navigation through a three-dimensional Virtual Environment. *Presence: Teleoperators and Virtual Environments,* **19**, 3, pp. 253-264.

C. Bernon, M. P. Gleizes, S, Peyruqueou, G. Picard, (2002), ADELFE, a Methodology for Adaptive Multi-Agent Systems Engineering, *LNAI 2577*, P. Petta et al. (Eds.), pp. 156-169, Springer.

S. Bouchard, S. Dumoulin, G. Chartrand-Labonté, G. Robillard, P. Renaud, (2006). Perceived realism has a significant impact on presence, *Proceedings of the 11th Annual CyberTherapy 2006 Conference: Virtual Healing–Designing Reality*, Quebec, Canadá.

J. Broeren, A-L. Bellner, M. Fogelberg, O. Göransson, D. Goude, B. Johansson, P., Larsson, Pettersson, K., Rydmark, R., (2008), Exploration of computer games in rehabilitation for brain damage, *Seventh*

International Conference on Disability, Virtual Reality and Associated Technologies, Maia, In: http://www.icdvrat.reading.ac.uk/2008/index.htm#abstracts-section-i, visited on may 2010.

D. Brutzman, L. Daly, (2007). *X3D: Extensible 3D Graphics for Web Authors,* Morgan Kauffmann.

G. Burdea, P. Coiffet, (2003). *Virtual Reality Technology*, 2nd Edition, Wiley, New Jersey.

L. Cardoso, R. M. E. M. Costa, A. Piovesana, M. Costa, M., Penna (2006). Using Virtual Environments for Stroke Rehabilitation, IEEE- 5th International Workshop on Virtual Rehabilitation, New York, pp. 1-5.

Costa, R. M. E. M., Carvalho, L., (2004). The Acceptance of Virtual Reality Devices for Cognitive Rehabilitation: a report of positive results with schizophrenia, *Computer Methods and Programs in Biomedicine*, **73**, 3, pp. 173-182.

A. Dawson, L. Buxbaum, A. Rizzo, (2008). The Virtual Reality Lateralized Attention Test: Sensitivity and validity of a new clinical tool for assessing hemispatial neglect, *Virtual Rehabilitation 2008*, Vancouver, in: http://www.virtual-rehab.org/2008/advanced_program.html, visited on march 2010.

S. A. Deloach (2001). Analysis and Design using MaSE and agentTool*. 12th Midwest Artificial Intelligence and Cognitive Science Conference*, Miami University, Ohio.

I. S. Gago, V. M. B. Werneck, Costa , R. M. E. M. Modeling an Educational Multi-Agent System in MaSE. *Lecture Notes in Computer Science,* v. 5820, p. 335-346, 2009.

H. Grillon, D. Thalmann, (2008). Eye Contact as Trigger for Modification of Virtual Character Behavior, *International Conference on Virtual Rehabilitation*, Vancouver.

I. Jacobson, G. Booch, J. Rumbaugh, (1999). *The Unified Software Development Process* – Addison-Wesley.

JAVA (2010). In: http://java.sun.com/, visited on June 2010.

JAVAScript (2010). In: http://www.javascript.com/, visited on June 2010.

JAVA3D (2010), In: http://java.sun.com/javase/technologies/desktop/java3d/, visited on June 2010.

N. Jennings, M. Wooldridge, (2002). *Agent Technology: Foundations, Applications, and Markets*, Springer.

W.L. Johnson, J. Rickel, J. Lester (2000). Animated pedagogical agents: face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education,* **11**, 1, pp. 47-78.

S. A. O'Malley, A. Scott, S. A. Deloach (2001). Determining When to Use an Agent-Oriented Software Engineering Paradigm*,* Proceedings of the Second International Workshop On Agent-Oriented Software Engineering (AOSE-2001), Montreal, Canada, May.

J. Odell, H.V. Parunak, B. Bauer (2000). Extending UML for Agents, *Proceedings of the Agent Oriented* Information Systems (AOIS) Workshop at the 17th National Conference on Artificial Intelligence (AAAI), 2000.

G. Pessin, F. Osório, S. Musse, V. Nonnemmacher, S. S. Ferreira, (2007), Simulação Virtual de Agentes Autônomos para a Identificação e Controle de Incêndios em Reservas Naturais, *IX Symposium on Virtual Reality*, Rio de Janeiro, Brazil. (in portuguese).

M. Pinho, L. Dias, C. Moreira, E. Khodjaoghlanian, G. Becker, L. Duarte (2002). A user interface model for navigation in virtual environments. *CyberPsychology & Behavior,* **5**, 5, pp.443–449.

M. Rield (2001). A computational model and Classification Framework for Social Navigation, *Proceedings of the Intelligent Users Interfaces*, Santa Fé, pp.137-144.

M. Wooldridge (2009). *An Introduction to MultiAgent Systems*, John Wiley & Sons,

H. Yang, Z. Pan, M. Zhang, C. Ju (2008). Modeling emotional action for social characters, *The Knowledge Engineering Review*, Cambridge University Press, **23**, 4, pp. 321-337.