

Towards the design of Intelligent Educational Gaming Systems

Konstantinos MARAGOS, Maria GRIGORIADOU
Informatics & Telecommunications Department, University of Athens
Panepistimioupolis, Illisia, 157 84 Greece
{kmaragos, gregor}@di.uoa.gr

Abstract. Researches on Games in Education have shown that games motivate the learners to try to develop their knowledge whilst they perform in a pleasant virtual world in order to achieve the goals proposed. However, many games are not designed in a way to keep the learners model, as this happening for Intelligent Tutoring System, so they cannot adapt the teaching strategies to users' knowledge. In addition, they lack an authoring tool used to change or alter the various components of the environment. This paper focus on these concepts, describing what Intelligent Tutoring Systems features can provide to Educational games. We also describe TALENT, a game to teach computer algorithms that incorporates features from ITS. The overall design proposed, is a step towards an Intelligent Educational Gaming System.

1. Introduction

Educational electronic games are games that encourage the growth of logic and the acquisition of skills and knowledge with a pleasant way [1]. Researches for the use of games in education [2] have proved that Games constitute a source that motivates the learners to try and to develop their knowledge, while they put it into practice. In addition, they learn things that they do not know at the same time they are engaged in an entertainment circumstance [3].

On the other hand, Intelligent Tutoring Systems (ITS) provide the beneficial of one-to-one instructions automatically and cost effective. ITS enable the participants to practice on their skills by carrying out tasks within highly interactive learning environments. Despite of other computer-based training technologies, ITS assess each learner's actions within these interactive environments and develop a model of their knowledge, skills, and expertise of the domain which is analyzed and structured by domain experts [4]. Pedagogical agents that appeared recently in ITS, have a set of normative teaching goals and plans for achieving these goals (e.g., teaching strategies), and associated resources in the learning environment [5]. Pedagogical agents can act as virtual tutors, virtual students, or virtual learning companions that can help students in the learning process. Furthermore, Pedagogical agents could lead to a more human and "social" learning environment [6] as well as providing motivation for learning [7]. Furthermore, ITS researchers have designed authoring systems in order to enable non-programmers to compose educational activities [8]. The main goal of an ITS authoring system is to make the process of building an ITS easier.

Although ITS are highly interactive, they typically lack of those components that characterize games and their potential to motivate the users. On the other hand present games lack of many of the power features of the ITS that make learning process more efficient [9]. Our proposal aims to utilizing ITS design characteristics in Educational game

design in order to achieve better learning results. By combining educational game design with some of the features of the Intelligent Tutoring Systems design, could lead us one step towards to the Intelligent Educational Gaming Systems (IEGS) design, providing better gaming and learning experience.

2. Integrating ITS Features in Educational Games Environments

The architecture of Intelligent Educational Gaming Systems, derived from the integration of ITS features in Educational Games is given in Figure1.

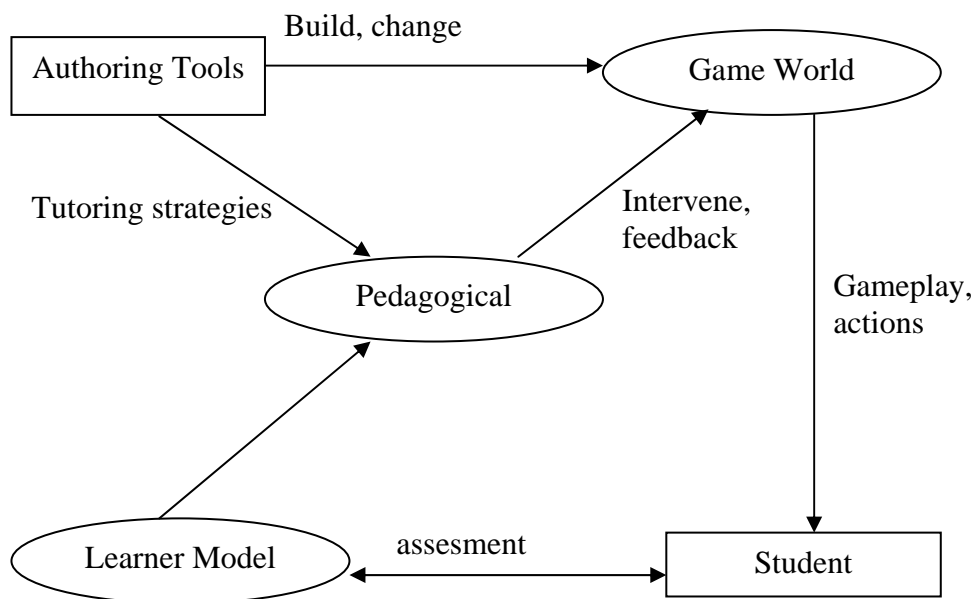


Figure 1. Architecture of Intelligent Educational Game

The architecture in Figure 1 shows that the game world (e.g. objects, stages) is constructed and categorized via an authoring tool. The categorization of the game world serves more as a classification of the educational objectives in each stage of the game. We propose a game world consisting of many different stages as opposite to a one stage game environment. This type of design provides flexibility as well as it supports learner's motivation. The end of a stage could mean the achievement of stage's educational objectives. Each stage acts as a different educational module. We could have stages that have the same educational objectives for the learner in order to strengthen the learning, as well as stages with completely different educational objectives. The sequence of stages has to be normalized according their difficulty and similarity. The teacher working with the authoring tool could be able to change or even alter the game stages as well as the entire game world consequently.

A pedagogical agent is been used by the game to support tutoring. The pedagogical agent inhabits in the educational game environment supporting the learner by giving him appropriate help based on specified teaching strategies and learner's model. The pedagogical agent is responsible to apply these strategies to learner by intervening in the environment. The intervening has to appear in a smooth and natural manner without influencing the gameplay.

3. Related Work

There are multiple commercial and academic attempts to teach specific concepts using computer applications which hide the learning process under a game. The content that is in direct relevance to a curriculum seems to be more difficult to be taught [11].

MIT along with Microsoft Corp. support in “Games to Teach” project [12], have constructed about twelve prototype games for different sciences like maths, physics, engineering, Humanities and Social Sciences. None of them seems to be supported from a pedagogical agent or authoring tools to alter the game world.

Some attempts to teach computer programming concepts include RoboCode [13], ToonTalk [14] and CeeBot-4 [15]. Robocode is a Java-based virtual robot game that is intended to teach some Java programming techniques, like the usefulness of inheritance and object-oriented programming, in general. It also provides an introduction to event-driven programming. ToonTalk is a game to teach programming concepts keeping out the writing of source code. CeeBot-4 is a game to learn programming, or to teach programming at middle school, high school and university. It uses a language close to Java and C# to program robots that will solve various tasks ranging from finding the way out of a labyrinth over car racing to playing soccer. RoboCode and CeeBot-4 lack at all the use of a pedagogical agent while ToonTalk use agents to provide hints and help but without making use of any user model. CeeBot-4 seems to provide an authoring tool as a series of files which can be easily modified.

4. TALENT Description

Currently, we are designing TALENT (Teaching ALgorithms ENvironment), an application where we aim to put into practise the previous ideas about integrating Educational Games with Intelligent Tutoring features. TALENT uses a game like environment and ideas from ITS and Pedagogical Agents to teach algorithms to students.

The student is symbolized as an avatar in the game environment that interacts with other objects and characters. The goal is the student to complete every stage of the game by providing algorithmic solutions to cope with all the obstacles in the stage. The student commands the avatar to make an action by passing chunks of code (source code) constructed by dragging avatar’s methods and framing them with the appropriate algorithmic structures. This helps students who are slow typists so we’d like to keep them off from using the keyboard [10] which in turn causes feelings of boredom.

The virtual world is also inhabited by ALEX (ALgorithms EXpert), an animated pedagogical agent who supplies hints and help, and who, eventually knows how to bypass the obstacles. The game starts assuming that the student has no previous knowledge. As the student acts into the game environment his actions are monitored and assessed in order to update his own user model. By updating the user model it provides the pedagogical agent information on how to support student’s learning, hints and suggestions to be provided to the student in order to carry out the activity.

In order to incorporate a non-isolated game learning environment we decided to support it with an authoring tool. By using this tool, one (e.g. the teacher) can create new methods and pass them as they were specific actions to the avatar (representing the learner). Moreover, he can design his own game stage with new graphics, objects, tools, obstacles and describe how the learner can deal with them.

5. Conclusions and Future Work

In this paper, we have analyzed a method to improve game like environments as learning applications using features and design characteristics from Intelligent Tutoring Systems. The main idea is to design Educational games where learner's actions in the game are assessed causing his knowledge model to be updated. Based on the learner model, a pedagogical agent tailors instructional strategies, intervene in the game providing hints or more help accordingly. The game is also supported by an authoring tool.

Putting the architecture into practice we have described our own Educational system, called TALENT. The environment is used to teach computer algorithms. It is a virtual world consisting of stages, where the learner controlling his avatar has to bypass the obstacles by ordering his own methods in algorithmic structures. Actions of the learner are monitored; learner's model is updated causing the pedagogical agent to provide help as appropriate. The application is supported by an authoring tool with which we can change or alter the game by means of graphics, stages, objects, obstacles, methods to bypass obstacles and teaching strategies applied by the pedagogical agent. One further step would be to add multi-user functionality. Students would be able to collaborate in order to design the appropriate algorithms.

References

- [1] Klawe, M. & Philips, E. (1995), A classroom Study: Electronic Games Engage Children as Researchers, *Proceedings of CSCL '95 Conference*, Bloomington, Indiana, 209-213
- [2] Gordon, A. K. (1970), Games for Growth, *Science Research Associate Inc.*, Palo Alto California
- [3] Malone, T. W. (1980), What make things fun to learn? A study of intrinsically motivating computer games, *Cognitive and Instructional Science Series, CIS-7*, Xerox Palo Alto Research Center, Palo Alto
- [4] Beal, C., Beck, J., Westbrook, D., Atkin, M., and Cohen, P. (2002), Intelligent Modeling of the User in Interactive Entertainment. In *AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*. Stanford, CA
- [5] Thalmann, D.; Noser, H.; Huang, Z. (1997), Autonomous Virtual Actors Based on Virtual Sensors. In: *Creating Personalities for Synthetic Actors: towards autonomous personality agents*. Trapp, R.; Petta, P. (Eds.). Berlin: Springer Verlag
- [6] Hermans B. "Intelligent Software Agents on the Internet", *FirstMonday*, vol. 2, 3/1997.
- [7] Morozov M., Tanakov A., Bystrov D. (2001). Pedagogical Agents in the Multimedia Natural Sciences for Children, *The 3rd IEEE International Conference on Advanced Learning Technologies*, Athens, Greece, 9-11 July 2003 pp. 62-65
- [8] Munro, A. (1995) *Authoring interactive graphic models*. In T. de Jong, D.M. Towne, and H. Spada (Eds.), *The use of computer models for explication, analysis, and experiential learning*. New York: Springer-Verlag
- [9] Ong J. & Ramachandran S. (2003). Intelligent Tutoring Systems: Using AI to Improve Training Performance and ROI, Stottler Henke Associates, Inc. online http://www.shai.com/papers/ITS_using_AI_to_improve_training_performance_and_ROI.pdf
- [10] Matthew C, et al. (2000), Alice: Lessons Learned from Building a 3D System for Novices. *ACM CHI*, pages 486-493
- [11] McFarlane A., Sparrowhawk A., and Heald Y. (2002), Report on the educational use of games. TEEM: Teachers Evaluating Educational Multimedia
- [12] Games to Teach project. <http://www.educationarcade.org/gtt/>
- [13] RoboCode Central. <http://robocode.sourceforge.net/>
- [14] ToonTalk – making programming child's play. <http://www.toontalk.com/>
- [15] CeeBot4 Teaching programming software. <http://www.ceebot.com/ceebot/4/4-e.php>