AIED 05 WORKSHOP 5

Amsterdam, 18-22 July, 2005

Educational Games as Intelligent Learning Environments

AI ED 05 WORKSHOP
Amsterdam

12th International Conference on Artificial
Intelligence in Education, Amsterdam,
the Netherlands

Table of Contents

Foreword

Over the past decade there has been an increasing interest in electronic games as educational tools. Educational games are known to be very motivating and they can naturally embody important learning design principles like exploration, immersion, feedback, increasingly difficult challenges to master. However, there are mixed results on the actual pedagogical effectiveness of educational games, indicating that this effectiveness strongly depends upon preexisting students' traits such as meta cognitive skills and learning attitudes. These results are consistent with the mixed results on the effectiveness of exploratory learning environments, not surprisingly since most educational games are exploratory learning environments with a stronger focus of entertainment.

Artificial Intelligence is already playing a increasingly integral part in both non-educational game design, and the design of more effective exploratory learning environments. This workshop aims to explore if and how AI techniques can also help improve the scope and value of educational games.

The overall goal of the workshop is to bring together people who are interested in exploring how to integrate games with intelligent educational technology, to review the state-of-the –art, and formulate directions for further exploration.

Some of the questions that the workshop aims to address include: (1) are some genres of games more effective at producing learning outcomes? (2) How do learners' individual differences (cognitive, meta-cognitive and affective) influence the genres of games they prefer/benefit from? (3) How can intelligent tutoring technologies augment gaming experience, with particular consideration to both motivational and learning outcomes? (4) How can we incorporate tutoring without interfering with game playing? (5) What role can intelligent educational games play in collaborative and social learning experiences? (6) The cost of developing games is very high, and adding AI techniques to the picture is likely to make the cost even higher. What tools exist or need to be developed to manage the development cost? (7) Should the gaming industry be involved and how?

By addressing these issues in an mixed-mode, informal set of interactions, this workshop explores the feasibility and utility of Intelligent Educational Games, identifies key problems to address, and contributes to advancing the state of the art of this emerging area of research.

Organizers

Cristina Conati
Department of Computer Science, University of British Columbia,
2366 Main Mall, Vancouver, BC, V6T1Z4, Canada
conati@cs.ubc.ca

Sowmya Ramachandran
Stottler Henke Associates, Inc,
951 Mariner's Island Blvd., Ste 360, San Mateo, CA 94404
sowmya@stottlerhenke.com

Program Committee

Amy Baylor, Florida State University, USA
Neil Heffernan, Worchester Polytechnic, USA
Lewis Johnson, ISI, USA
James Lester, North Carolina State University, USA
Frank Linton, Mitre Corp, USA
Rose Luckin, University of Sussex, UK
Heather MacLaren, University of British Columbia, Canada
Jack Mostow Carnegie Mellon University MU, USA
Ana Paiva, INESC-ID, Portugal
Amy Soller, IDA, USA
Julita Vassileva, University of Sasketchewan, Canada
Michael van Lent, Institute for Creative Technologies, USA
Beverly Park Woolf, University of Massachuset, USA
Michael Young, North Carolina State Univeristy, USA

Acknowledgments

We would like to thank the members of the program committee for agreeing to be part of this workshop, for contributing ideas and for timely reviews. We thank the workshop organizers Joe Beck and Neil Heffernan for making the workshop possible and for their support.

Web Page

http://www.cs.ubc.ca/~conati/aied-games/main.html

Book Title
Book Editors
IOS Press, 2003

Added value of task models and use of metacognitive skills on learning

Noor Christoph [1], Jacobijn Sandberg and Bob Wielinga
University of Amsterdam

Abstract. This study focuses on the effect of a task model on learning to solve problems and the use of metacognitive skills. In two conditions, students played KM Quest, a simulation-gaming environment for the domain of Knowledge Management (KM). In one condition, students had the KM model available that prescribed how to solve KM problems. The other condition provided no such task model. Forty-six students participated in the study. KMQUESTions was used to measure the acquisition of declarative and procedural knowledge. Part of the MSLQ was used in order to measure the self-reported use of metacognitive skills. A significant increase in declarative and procedural knowledge was found. Furthermore, an interaction effect was found between learning *success* and the self-reported use of metacognitive skills. Students who scored low on metacognition, achieved the biggest learning gain. No effect of condition could be reported. The explanation for these results is that KM Quest apparently has succeeded in translating the general principles of the constructivism into concrete teaching examples, and therefore, supports students that are weak in regulating their learning behaviour. Future research should indicate to what extent students in the no-model condition have developed an intuitive model for solving KM problems.

Keywords. Constructivism, Knowledge Management, Metacognition, Problem solving, Task models

1. Introduction

Constructivist learning environments generally advocate the active acquisition of knowledge and skills, collaboration and the use of authentic and realistic case material [1]. Games and simulations fit rather well in this paradigm since learners often can experiment in a highly realistic environment. However, results show that learning is suboptimal in these environments [2]. One of the problems lies in the fact that learners have difficulties in regulating their learning behaviour. In this paper the assumption is that constructivist learning environments, especially games and simulations that concern problem solving, are pre-eminently supportive of learning under the condition that they contain a task model. A task model is a model that prescribes how to solve a particular problem. Mettes and Pilot [3] for example developed a task model for problem solving in the domain of thermodynamics. This Program of Actions and Methods PAM contained all elementary executable activities stemming from the general phases of problem solving,

---

[1] Correspondence to: Noor Christoph, University of Amsterdam, Matrix I, Kruislaan 419 1098 VA Amsterdam. Tel.: +31 020 888 4671; Fax: +31 020 888 0000; E-mail: noor@science.uva.nl.

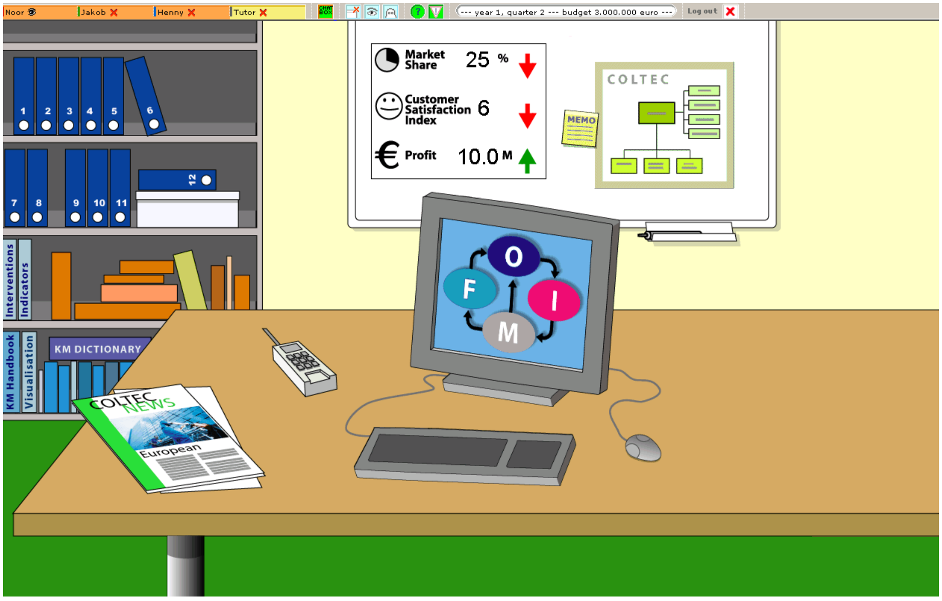


Figure 1. Screenshot of the homepage of KM Quest.

necessary for solving thermodynamics problems. It appeared that students working with this model outperformed students who did not have this model available. From the field of instructional technology and/or psychology the notion of including a task model in a learning environment is seen as a form of instructional support, especially for regulating learning behaviour. Self-regulatory behaviour concerns being able to monitor and control the learning process. It is also called metacognition, that is, the cognition of cognition [4]. In general, the use of metacognitive skills is positively related to learning success [5]. The aim of this paper is to investigate the added value of a task model in a gaming-simulation environment for the domain of Knowledge Management (KM) and the role of metacognitive skills. The learning environment suitable for this research is called KM Quest.

KM Quest[1] is a constructivist learning environment for the domain of KM. In KM Quest teams of three players have to manage the knowledge household of a fictitious company called Coltec. Coltec is a producer of adhesives and coatings and it is headquartered in Delft (The Netherlands). The behaviour of Coltec is simulated by means of a Business Model (BM). This BM contains both business indicators such as ’the number of employees in the R&D department’ or ’the number of patents pending’ and it contains knowledge related indicators such as ’the level of competence of the marketing employees’ or ’the speed of knowledge gaining in the production department’. The overall status of Coltec can be interpreted by reviewing the indicators from the BM.

Each quarter in the game, the team is confronted with a problem in the form of an event. An event is for instance, the leave of a senior marketing manager. It is up to the

---

[1] KM Quest was developed in the EC project KITS (IST-1999-13078), which consisted of the following partners: University of Twente, The Netherlands; TECNOPOLIS CSATA novus, Italy; Cibit, The Netherlands; EADS, France; ECLO, Belgium and the University of Amsterdam, The Netherlands

team to react upon the event by analysing the problem, setting goals, proposing interventions and checking effects of interventions by reviewing the status of several indicators. For this particular event knowledge retention is at stake since vital knowledge could be at loss and interventions should focus on safeguarding knowledge. The team should propose interventions that counteract this effect. They have a budget for this since interventions cost money. If the team decides to do nothing the decay function of the BM will take over and the status of Coltec will worsen. This reflects part of the competition element essential for games, teams compete against the BM model. The element of chance is represented by having the game fire events randomly. Events differ with respect to whether they represent a threat or an opportunity for Coltec or whether they are interal or external to Coltec. The aim of the team is to manage the knowledge household of Coltec as good as possible.

One of the main ingredients of KM Quest, next to the BM, is the Knowledge Management model (KM model)[6]. This is a normative model, cyclic in nature that prescribes how to solve knowledge management problems. The KM model can be seen as a set of problem solving activities that are instantiated for a specific type of task. In the KM model, the general phases of problem solving, such as for example orientation, execution and evaluation [7] are applied to a monitoring-diagnosis task. Such a task includes analysing an ongoing process and checking whether it occurs according to the expectations, identifying possible discrepancies and if needed, taking action. This process is represented in the phases FOCUS, ORGANISE, IMPLEMENT and MONITOR. The KM model prescribes how to perform this task, because each phase consists of elementary executable activities (steps) relevant for that phase. The model is therefore decomposed at the task-level. In a way, it strongly ressembles the PAM of Mettes and Pilot.

The benefit of the KM model with respect to self- regulatory skills is the following. It is argued [8,9] metacognitive skills of novices in a particular domain are general in nature. Only when they become experts, the domain independent metacognitive skills become domain-specific task schemas. The KM model supports learners in the sense that instantiating these domain independent metacognitive skills into task-related activities is already done for them. It is all laid down in the KM model. Therefore, learners that are presented with the KM model will have less problems instantiating their domain independent metacognitive skills and will learn most. The hypothesis is that the KM model in KM Quest is responsible for the learning success since it represents a compiled model of how to solve KM problems. Furthermore, when no model is present, students that have an adequate framework of metacognitive skills at their disposal will be able to use these skills in order to solve KM problems. They at least have their generic problem solving skills to tackle the new problems. Students that are weak in metacognition, will get lost more easily in the learning environment, since they have no good starting point and therefore achieve suboptimal results.

In order to study the effect of the task model on learning and the use of metacognitive skills, KM Quest will be played in two conditions: one without the task model (no-model condition) versus the standard environment (model condition). Students are assigned to conditions based on randomization. They have equal chance to be assigned to either condition. In a pre-test post-test design, measures of learning and self-reported use of metacognitive skills will be employed. The premises is that learning is taking place, since this was already established in a previous study on KM Quest [10]. Hypothesis 1 covers the main effect of condition: students in the model condition outperform students in the no-model condition with regard to the acquisition of declarative and procedural knowledge. Hypothesis 2 concerns an interaction effect of condition and metacognition. Players in the no-model condition that score high on use of metacognitive skills reach comparable scores on the knowledge tests to players in the model condition. Players in the no-model condition that score low on use of metacognitive skills, perform less on knowledge tests than students in the model condition.

2. Method

2.1. Participants

The results of 46 out of 49 participants are included in the data analysis of this study. Two students dropped out because they had participated in a previous study in this research project. They were assigned to the same team. One student fell ill during the experiment. The average age of the students is 22.7 (SD = 1.6). Thirty students are male, 16 female. No significant differences exist between conditions on the pre-test measurements of declarative and procedural knowledge. This indicates that the samples in both conditions are comparable to each other.

2.2. Learning results

The electronic test tool KMQUESTions is used in order to measure the acquisition of knowledge. It contains multiple choice items (four alternatives) specifically based on the learning goals of KM Quest. The learning goals specifically cover the acquisition of declarative and procedural knowledge. Declarative knowledge acquisition requires the acquisition of factual knowledge in the domain of KM. The learning goals focus on definitions of knowledge processes, meaning of indicators, types of events, types of interventions, effects of interventions and so on. It is tested with for instance, items such as 'What is the definition of knowledge gaining?' or 'Which knowledge domains exist in KM Quest?'. Procedural knowledge acquisition is related to acquiring knowledge about how to solve KM problems. Learning goals involve being able to reflect on the nature of the KM model, understanding the steps of the KM model, being able to apply the steps of the KM model to events and so on. Items are for example 'What do you do in the step "Where to focus on" in the KM model?' or 'Which intervention is best for this particular event?. KMQUESTions was developed in a previous study and appeared to be sufficiently reliable 10]. For the current experiment, KMQUESTions contains 96 items (38 items for declarative and 58 items for procedural knowledge). It is administered before and after the game. Sequence of the items remains the same because there is an optimal ranking in order not to reveal answers. Since there is a mere three weeks between both administrations, possible test-retest effects are considered negligible. The score reflects the proportion of items answered correctly.

2.3. Metacognition

The Motivated Strategies for Learning Questionnaire [11] is a self-report measure that focusses on motivational and learning strategies. The scale Metacognitive self-regulation (12 items) consists of three processes: planning, monitoring and regulation. Planning involves goal setting and task analysis to help activate relevant aspects of prior knowledge that make organizing and comprehending material easier. Monitoring involves tracking one's attention as one reads and self-testing and questioning. Regulation refers to the fine-tuning and continuous adjustment of one's cognitive activities). The scale consists of 12 items that are answered on a six-point scale ranging from 'Strongly disagree' to 'Strongly agree'. The reliability of this scale is sufficient [10,11]. It is administered during the post-test. Students have to keep in mind how one has just played KM Quest. This instrument represent a retrospective self-report measurement of metacognition (Veenman, in press).

2.4. Procedure

The study is carried out in a period of four weeks, prior to any other institutional instruction in KM. In week 1 students start with an introduction into the game. Students of both conditions participate in the introductory lecture. Subsequently, the training session takes place. For each condition, a specific training is developed. The main difference between the two training sessions is the explanation and demonstration of the KM model. After the training, students are administered the pre-test measurements (KMQUESTions). In week 2, students start playing the game during two game sessions that each last over 2 hours. Team members are located in different computer rooms. Communication solely takes place via the chat facilities present in the game. Students do not have access to the game outside the playing sessions. In week 3 a third and last game session takes place in order to reach quarter 7 in the game. The day after, the post-test is scheduled. The post-test consists of the MSLQ and KMQUESTions. In week 4 a debriefing lecture is organised during which students can share their experiences about the game. The instruction to students is to both perform best on the post-test and manage the knowledge household of Coltec as good as possible.

3. Results

3.1. Reliability

The reliability coefficients of different parts of KMQUESTions and the scale metacognition was moderate to sufficient. KR20 (Cronbach's alpha for dichotomous items) coefficients of KMQUESTions were 0.64 for declarative knowledge for both pre- and post-test. For procedural knowledge, this coefficient was 0.78 and 0.52 for pre- and post-test respectively. Cronbach's alpha for the scale metacognition was 0.66.

3.2. Declarative knowledge

In table 1 descriptive statistics for declarative and procedural knowledge are shown.

In order to test the hypotheses, an analysis of variance by means of a General Linear Model with repeated measures was performed. Dependent variables were the pre- and post-test measurements of declarative knowledge. Independent measures were condition and metacognition. Scores on the variable metacognition were divided in two levels based on the median, in order to discriminate between students that scored high and

	$No-model$		$Model$		$Total$	
Declarative	Pre-test	Post-test	Pre-test	Post-test	Pre-test	Post-test
$MS-$	0.51 (0.16)	0.63 (0.13)	0.49 (0.08)	0.63 (0.11)	0.51 (0.12)	0.63 (0.12)
$MS+$	0.52 (0.09)	0.64 (0.09)	0.50 (0.12)	0.59 (0.08)	0.51 (0.10)	0.61 (0.09)
$Total$	0.51 (0.13)	0.64 (0.11)	0.50 (0.10)	0.61 (0.10)	0.51 (0.11)	0.62 (0.10)
Procedural	Pre-test	Post-test	Pre-test	Post-test	Pre-test	Post-test
$MS-$	0.44 (0.14)	0.58 (0.09)	0.50 (0.12)	0.68 (0.07)	0.47 (0.14)	0.63 (0.09)
$MS+$	0.49 (0.08)	0.56 (0.08)	0.55 (0.14)	0.64 (0.07)	0.52 (0.12)	0.60 (0.08)
$Total$	0.46 (0.12)	0.57 (0.08)	0.53 (0.13)	0.66 (0.07)	0.49 (0.13)	0.62 0.09)
General	Pre-test	Post-test	Pre-test	Post-test	Pre-test	Post-test
$MS-$	0.47 (0.11)	0.62 (0.09)	0.50 (0.14)	0.64 (0.10)	0.49 (0.12)	0.63 (0.09)
$MS+$	0.51 (0.14)	0.55 (0.11)	0.56 (0.14)	0.55 (0.08)	0.54 (0.13)	0.65 (0.09)
$Total$	0.49 0.12)	0.59 (0.10)	0.53 (0.14)	0.59 (0.10)	0.51 (0.13)	0.59 (0.10)

Table 1. Mean and standard deviation of the pre- and post-test scores on declarative, procedural and general procedural knowledge measured in proportion of correct answers.

low on this variable. This analysis is repeated for the subsequent analyses. The median is a robust measure for central tendency that is not sensitive for possible outliers.

Concerning within-subject effects, a main effect for learning (comparison pre- and post-test scores) was found (F = 72.13, $p < 0.01$). Students acquired declarative knowledge as a result of playing KM Quest. No interaction effects were found. Concerning between-subject comparisons, no main effects could be reported. The hypothised interaction effect of condition and metacognition was not found (F = 0.22, p = 0.64). No other interaction effects were found. This indicates that students acquired declarative knowledge regardless of condition and metacognition. Hypothesis 1 and 2 could not be supported.

3.2.1. Procedural knowledge

Concerning within-subject effects, a main effect for learning could be reported (pre- versus post-test scores). Students acquired procedural knowledge as a result of playing KM Quest (F = 38.56, $p < 0.01$). No interaction effects existed. Concerning between-subject effects, a main effect was found for condition (F = 9.26), $p < 0.01$). Students in the model condition outperformed students in the no-model condition. This is in line with hypothesis 1. One interaction effect was found, namely between learning *success* and metacognition (F = 4.66, $p < 0.05$). Students that scored low on metacognition, showed more learning success in relation to students that scored high on metacognition. No interaction effects between condition and metacognition could be reported (F = 0.10, p = 0.75). Hypothesis 2 could not be supported.

The drawback of focussing on procedural knowledge is the fact that 33 items of this test were specific for the KM model. Students in the no-model condition did not have the KM model at their disposal, therefore, it is not fair to include these questions for them. Results indicated that the mean score of students in the no-model condition on KM model specific procedural knowledge was 0.56 (SD = 0.12) and for students in the model condition it was 0.70 (SD = 0.11). This difference was significant (T(46) = -4.14, $p < 0.01$). Therefore, mean scores on procedural knowledge were calculated again, this time without the KM model specific items. The new measure was called *general* procedural knowledge. In table 1 the mean, standard deviation and number of participants for general procedural knowledge are presented.

Concerning within-subject effect, again a main effect for learning with respect to general procedural knowledge was found (F = 15.69), $p < 0.01$). Students acquired general procedural knowledge as a result of playing KM Quest. An interaction effect of learning success and metacognition existed (F = 4.55, $p < 0.05$). Students that scored low on metacognition showed more learning gain than students that scored high on metacognition. Concerning between-subject effects, no main effects could be reported. Hypothesis 1 could not be supported. Students in the model condition, did not outperform students in the no-model condition. Additionally, no interaction between condition and metacognition existed (F = 0.09, p = 0.76. Hypothesis 2 could not be supported. It appeared that weaker students in terms of self-reported metacognitive skills after task performance, learned most, regardless of condition. Apparently, KM Quest is specifically suited to support students that are less able to monitor and control their learning process.

When conditions differ in the amount of time that is spent on playing KM Quest, it is realistic to assume that there is a relation between time-on-task and learning results. In the no-model condition, students spent on average 3 hours, 52 minutes and 32 seconds, where as students in the model conditoin spent 5 hours, 29 minutes and 22 seconds. This could confound the results. Then, time-on-task should be included as a covariate in the analysis of variance. One of the assumptions for carrying out an analysis of covariance is that a linear relation exists between the dependent variable and the covariate in each condition [12]. This was not the case for this study, therefore, time-on-task did not influence learning results.

3.3. Relation learning results and self-reported use of metacognitive skills

As for the relation between self-reported use of metacognitive skills and learning results, correlation coefficients were calculated to gain insight. The results indicated that hardly any relation was found between learning measures and self-reported use of metacognition. The retrospective measure of metacognition was only related to procedural knowledge measured in the *pre-test* (0.44, $p < 0.01$). This indicated that having sufficient common sense (or prior knowledge) about how to go about in a new task that one has not done before, related to having relevant metacognitive strategies available. This effect disappeared during the post-test measurement of procedural knowledge since then, participants had already developed an idea about the task.

4. Discussion and conclusions

In this study, the objective was to find answers on several hypotheses with the premises that learning would take place. The results reveal that students acquire declarative and procedural knowledge about the domain Knowledge Management, this replicates findings of an earlier study [10]. The first hypothesis can only partly be confirmed. Students in the model-condition outperform students in the no-model condition only with respect to procedural knowledge. Regarding the acquisition of declarative and general procedural knowledge, mean scores in the two conditions do not differ significantly. The second hypothesis, namely about the interaction effect of condition and metacognition, cannot be confirmed. Students in the no-model condition that score high on metacognition, do not exceed students in the same condition that score low on metacognition. However, an interaction effect was found between learning *success* and metacognition. Students that scored low on metacognition, obtained significantly more learning gain than students that scored high on metacognition. Finally the retrospective self-reported use of metacognitive skills did not relate with learning success.

Two conclusions can be drawn from this study. The first one concerns the predictive validity of retrospectively measured self-reported metacognition. Predictive validity is the extent to which a test is capable of predicting behaviour towards a criterion that lies in the future. The better the test can predict variances in a criterion, the higher the criterion -related validity of this test.

In general, in the literature about metacognition one often refers to the fact that it predicts learning success [5]. Students that score high on metacognition, achieve higher learning scores. In this study, the predictive validity of the retrospective self-report measure was low. No significant correlations could be reported of this measure and any post-test result of retention. The conclusion is that this self-report measure of metacognition, is not a good predictor for learning success. So although Pintrich et al. [11] initially set out to support the idea that self-regulated learning promotes learning success, their method, or at least the scale metacognition, does not underpin this relation. It appears that Veenman [13] could be right in assuming that self-report questionnaires lack predictive validity because of the individual reference point that is chosen by the respondent (e.g. comparison with best or poorest classmate or teacher etc), the social desirability of the answers, and the fact that one often does not do, what one says. The aim for the future is to employ a concurrent measure of metacognition, for instance by performing protocol analysis of the communication between team members and to score the frequency of metacognitive contributions. It will also be of interest to compare this measure of metacognition in a social, collaborative context, with individual scores of metacognition.

Secondly, in general the implication of a task model in a learning task was assumed to benefit the learning process because it supports self-regulatory behaviour. In this study, no such result was found. On the contrary, the main finding is that especially weaker students in terms of metacognitive skills appear to benefit from KM Quest, regardless wither a model is present. Their learning gain is highest compared to students that report to be stronger in using metacognitive skills. Apparently, students that are weaker in monitoring and regulating their learning behaviour, benefit most from KM Quest. It is however, not the KM model that they benefit most from, since the addition of this model to the environment does not lead to better learning results. Perhaps the fact that KM Quest is in essence a constructivist learning environment is the reason why weaker students in terms of metacognitive skills achieve more learning success. Maybe for this environment one has successfully translated the theoretical principles underpinning the constructivism into specific didactical and pedagogical teaching strategies that lead to advances self-regulatory behaviour and therefore, better learning, especially for those who need it.

References

[1] J.S. Brown, A. Collins, P. Duguid (1989): Situated cognition and the culture of learning. Educational researcher, 17, 32-41.

[2] T. de Jong, W.R. van Joolingen (1998). Scientific discovery learning with computer simulations of conceptual domains. Review of educational research, 68, 179-202.

[3] C.T.C.W. Mettes, A. Pilot (1980). Over het leren oplossen van natuurwetenschappelijke problemen. Enschede:Onderwijskundig Centrum CDO/AVC TH-Twente, The Netherlands.

[4] J.H. Flavell (1976). Metacognitive aspects of problem solving. In (ed) L.B. Resnick, The nature of intelligence. Lawrence Erlbaum Associates, Hillsdale, NJ, 231-235.

[5] M.C. Wang, G.D. Haertel, H.J. Walberg (1990). What influences learning? A content analysis of review literature. Journal of educational research, 84 30-43.

[6] De Hoog, R., Shostak, I., Purbujo, R., Anjewierden, A., Christoph, N., Kruizinga, E. (2002). Final models and visualization. Deliverable to the KITS project D12, University of Twente, 1 - 134.

[7] C. Masui and E de Corte (1999). Enhancing learning and problem solving skills: orienting and self-judging, two powerful and trainable learning tools. Learning and instruction, 9, 517-542.

[8] M.V.J. Veenman, J.J. Elshout, J.Meijer (1997). The generality vs domain-specificity of metacognitive skills in novice learning across domains. Learning and instruction, 7, 187-209.

[9] F. Prins (2002). Search & See: the roles of metacognitive skillfulsness and intellectual ability during novice inductive learning in a complex computer-simulated environment. University of Leiden, The Netherlands.

[10] N. Christoph, J.A.C. Sandberg, B.J. Wielinga (2004). Measuring learning effects of a gaming-simulation environment for the domain of knowledge management. Proceedings of IADIS CELDA conference, Lisbon, Portugal.

[11] P.R. Pintrich, D.A.F. Smith (1993). Reliability and predictive validity of the motivated strategies for learning questionnaire (MSLQ). Journal of educational psychology, 82, 33-40.

[12] J.P. Stevens. (2002). Applied multivariate statistics for the social sciences. Lawrence Erlbaum Associates, London.

[13] M.V.J. Veenman (in press). The assessment of metacognitive skills: What can be learned from multi-method designs? In B. Moschner, & C. Artelt (Eds), Lernstrategien und Metakognition: Implikationen für Forschung und Praxis. Berlin: Waxmann.

Adventure Games for Science Education: Generative Methods in Exploratory Environments

Henry M. Halff
Halff Resources
402 W. Rhapsody, Suite 110, San Antonio, TX 78216, USA

Abstract. Presented here are several techniques for design and development of computer-based adventure games for science education. Among the issues addressed are how subject matter maps to content, generative techniques for problem-solving practice, use of visualization, mechanisms for instructional support, and approaches to game development.

Introduction

In the years around 1990, I had the good fortune of creating the instructional design for a computer adventure game that taught some aspects of the US Navy's Basic Electronics and Electricity course. The game that we developed, *Electro Adventure,* was a success in many ways, and it had many shortcomings [1]. I am now involved in a project to apply what was learned from *Electro Adventure* to a new game, called *Twisted Physics,* of the same genre. This paper describes the lessons learned from *Electro Adventure* and how we intend to apply them to *Twisted Physics*. Many of these lessons should be of interest to the AI-ED community, and many are pertinent to the focal concerns of this workshop. I discuss the use of generative techniques both as an exercise in cognitive task analysis and as a method for promoting fruitful collaboration among students. I also describe a variety of mechanisms for interweaving instruction with game playing. I describe some of the ways that we use to control development costs, and I describe a nested set of game genre's that invite comparisons of instructional effectiveness.

1. Science Education

Science education, particularly these days, means many things. Hence, I first need to make clear what aspects of the enterprise are addressed here. This work concerns mainly those aspects of science education found in traditional classroom curricula. In these curricula, students are taught the content of science, Newton's laws of motion, for example, and how to solve problems that relate to that content. Recent thinking about science education has focused on other skills, namely, how to do science, and how to critically evaluate scientific claims. Both of these newer issues are important, but are not addressed in my efforts.

Even in traditional curricula, there is no dearth of challenge to science educators and no dearth of issues of interest to artificial intelligence. There are three sources of difficulty in mastery of science that are of particular interest here.

Students in many curricula are faced with a large number of concepts that are related to each other in complex ways. Physics students quickly encounter a number of topics even in the narrow domain of kinematics: velocity, acceleration, displacement, time, uniform velocity, uniform acceleration, average velocity, rotary motion, gravity, and projectiles, to name a few. Keeping the relationships among these concepts straight is a serious challenge.

In addition, many scientific concepts are abstract, invisible in the real world, and difficult to grasp. Qualitative reasoning about these concepts is known to be a serious challenge in science education. For example, students routinely fail to understand how Ohm's law applies to simple resistive circuits. Understanding how it applies even to the simplest RC circuit is an almost impossible task.

Finally, the real test of a student's mastery of science is often her ability to solve quantitative scientific problems, commonly cast as word problems. There are many challenges in the solution of these problems, including recognizing the type of problem or applicable principles, coming up with a plan for the problem's solution, and carrying out the required calculations. I take the position here that practice, especially practice that is properly structured and guided, is a powerful tool in promoting problem-solving skills.

It is these three aspects of science education—the complex structure of science, the difficulty of reasoning about abstract concepts, and the challenges that arise in quantitative problem solving—that form particularly promising targets for educational games.

2. Adventure Games

Adventure games are a genre of role-playing games in which the player assumes the role of a character in a fantasy world. The player can control his character and thereby cause the character to move about in the fantasy world, inspect that world, and interact with whatever is found in the world. The character can, for example, open doors, pick up found objects, and, in some cases, carry on dialogs with other characters in the fantasy. The fantasy world is itself rarely static; other characters and objects can move about and act on their own.

Not all role-playing games are adventure games, or at least not what I will call adventure games. A large class of role-playing games, sometimes referred to as Jump-Punch-Kick games are dominated by combat and rely mainly on eye-hand coordination. Adventure games, for our purposes, are those games in which challenges consist mainly of puzzles to be solved or discovering hidden contingencies in the fantasy world. Reaction time and coordination are rarely factors in adventure games. Most of the playing time is spent deliberating or poking around.

Adventure games can be multi-player. Indeed the granddaddy of all adventure games, *Dungeons and Dragons* [2], predates computers and is almost always a multiplayer game. However, the games that I treat here are limited to single-player versions.

One other aspect of adventure games that is important for their use in education is the strong sense of place or context that is present in many of these games. The fantasy world of a typical adventure game consists of a network of distinct physical *contexts* such as the rooms of a castle or caverns in a cave. Associated with each such context is a set of tasks that must be completed if the player is to advance in the game or even change context. The contexts are networked by doors or other mechanisms so that the player is constrained in her trajectory through the set of contexts.

3. Game Design for Science Education

The main lesson of the *Electro Adventure* project was that adventure games have considerable potential as vehicles for science education, if they are designed with that application in mind. This section describes how to design a game for purposes of science education.

3.1 *Context-Objective Mapping*

Many of the problems found in science education texts and tests can be classified into types depending on the principles that apply and the methods of solution. For example, one class of problems in electrical theory comprise those requiring computation of the equivalent resistance of a network of resistors. Another type, in the area of kinematics, requires the computation of various parameters of a projectile's trajectory. Mastery of a particular type of problem can be said to be an instructional objective, and the set of objectives, taken together, can be cast in a prerequisite structure or graph. A fragment of such a graph for kinematics is shown in Figure 1a.

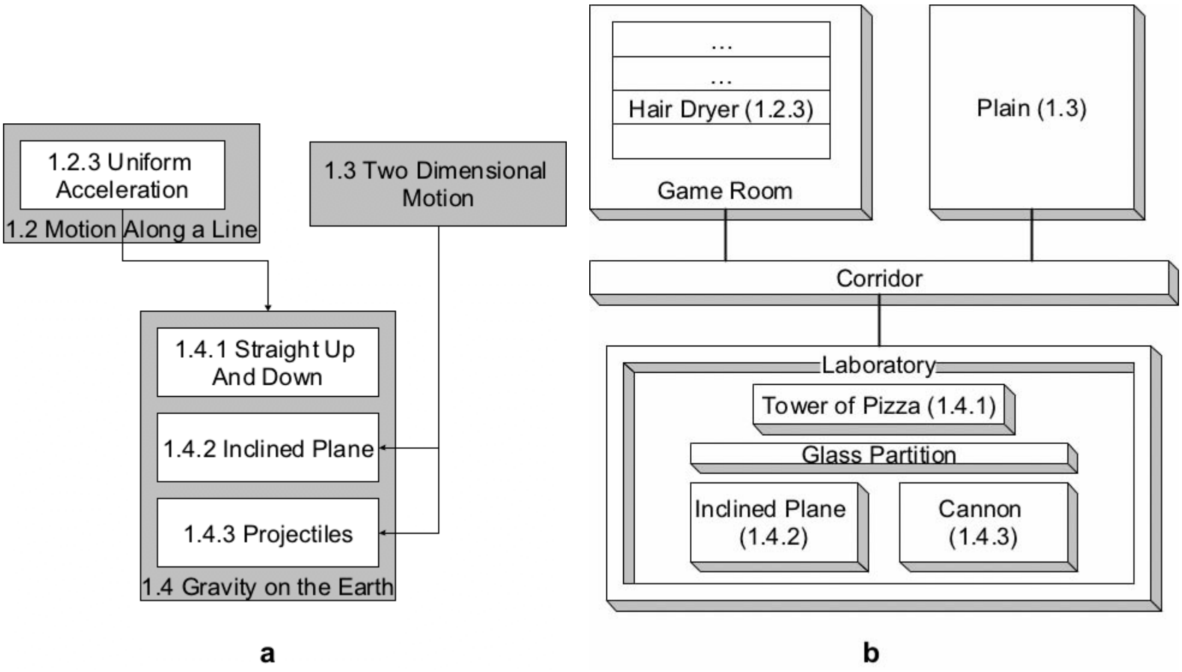


Fig. 1. Prerequisite Network Fragment for Twisted Physics (a) and Corresponding Context Chart (b).

An obvious tactic, then, for the game designer, is to take advantage of an adventure game's strong sense of place and context to associate contexts with instructional objectives and paths between contexts with the prerequisite structure of the discipline. Figure 1b shows the contexts of *Twisted Physics* that correspond to the objectives in Figure 1a. The player uses a Hair Dryer to propel a skateboard in the Game Room to learn about uniformly accelerated motion. Success in this context gives him a key to the Laboratory where he can learn about falling objects from the Tower of Pizza. On the Plain he masters the fundamentals of relative motion in two dimensions. After successfully traversing the plain he obtains a code that gets him past the Glass Partition in the Laboratory so that he can learn about motion along an inclined plane (Inclined Plane) and the trajectory of projectiles (Cannon).

3.2 *Problem-Solving Exercises*

One of the major advantages of games over other forms of instruction is their motivational property. Boring problem sets and drills in conventional instruction can become addictive puzzles in the context of a game. We take advantage of this property by embedding a problem-solving exercise in each room. This exercise naturally addresses the room's objective. Figure 2 shows how such an exercise appeared to a player in *Electro Adventure.*

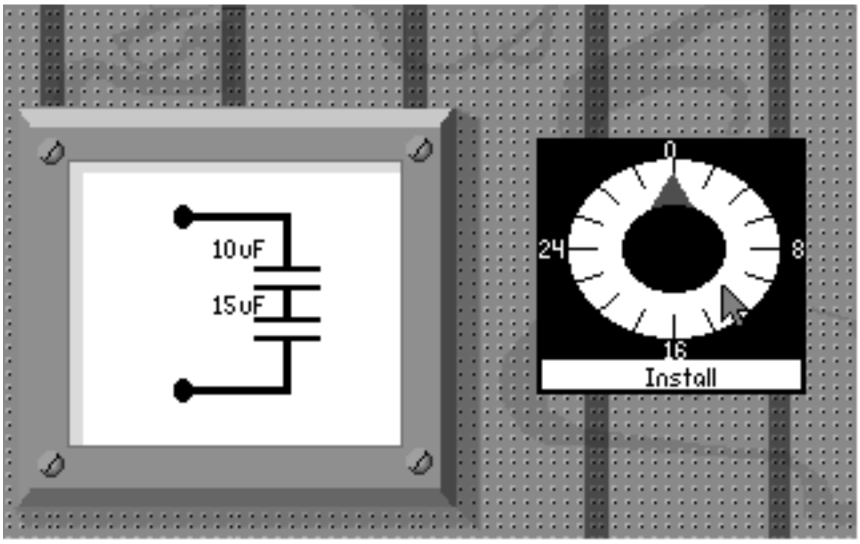


Fig. 2. *Electro Adventure* Exercise.

All of the problems in this exercise require the student to set the dial to the equivalent capacitance of the illustrated circuit. (The correct setting for Figure 2 is 6μF.). Each problem in an exercise was formed by setting the parameters of an exercise template. In the case of Figure 2, the parameters specified the circuit shown in the figure and specified the circuit topology (parallel or series), the number of capacitors, and their sizes. Many of the issues in the design of these sorts of exercises should be of interest to the AIED community.

The parameters of each problem can be generated dynamically using a semi-random process or chosen randomly from one or more pools of parameter combinations. This approach allows the player to return to an exercise and practice new problems on every replay. More importantly, it allows players to help each other without cheating. A player who has completed an exercise cannot help another player to complete the exercise by revealing the sequence of problem solutions since every player receives different problems. Hence, the helper is at least encouraged to demonstrate the solution to any player that he is helping.

Generating problem parameters, either off-line or at run time, can present some interesting problems in terms of defining the parameter space of the exercise. For example, one may want to restrict the space to problems that have integer answers or at least answers that are compatible with the answer-entry mechanism. Ranges of parameters and the units in which they are expressed can interact in complex ways. For example, the muzzle velocities in most projectile problems in physics are of the same order as the acceleration due to gravity (16m/sec^2); muzzle velocities of 2mm/hr or 5000km/sec would not be considered instructive for most purposes. At the same time, the parameter space must reflect the full range of meaningful combinations. In circuit problems, for example, component sizes (e.g., resistance and capacitance) must vary over several orders of magnitude and still confine circuit behavior (e.g. a circuit's time constant: resistance × capacitance) to a reasonable range.

Dynamic generation also supports a degree of intelligence in sequencing exercises, such as that delineated in Van Lehn's Step Theory [3]. Harder or more complex problems can be introduced only after the player has exhibited competence on simpler problems. Simpler problems can be reintroduced occasionally for review purposes, and the sequence of problems itself can be terminated based on a mastery criterion. Intelligent sequencing requires a cognitive analysis of the procedures that students use to solve typical textbook problems and the procedures that authors use to generate those problems.

As an example, consider problems involving motion in one dimension. The simplest of such problems are cases of uniform velocity, as in the following, *Problem 1.* A train traveling at a uniform velocity covers 6 km in 10 min. What is the train's velocity?

At their most complex, these problems involve objects that are uniformly accelerated over different intervals as in the following.

Problem 2. A train, starting from a dead stop, accelerates at 5 m/sec^2 for 1 min. It then travels at uniform velocity for 5 minutes. Finally, it decelerates at 2 m/sec^2 until it stops. How far does the train travel from start to stop?

These problems are typically solved by using a set of standard formulae, for example, $v = \Delta d/\Delta t$, or $v_t = v_0 + at$. One set of formulae applies to uniform velocity. Another applies to travel under uniform acceleration. In addition, the formulae must be applied recursively when velocity or acceleration varies between subintervals of an objects travel time. The skills involved in mastering these problems are therefore those of (a) determining which set of formulae apply to a situation, (b) determining which formula applies, (c) applying the formula, and (c) planning a solution path in cases where the formulas must be applied recursively.

Twisted Physics will have a context that addresses these simple problems of motion in one dimension. Exercises presented in this context will involve the player in a bizarre card game (a game within a game) that requires him to compute specified characteristics of the motion of a small creature on a skateboard. The creature can use a kick to provide an instantaneous change in velocity or a hair dryer for uniform acceleration.

Problems generated by this exercise are sequenced in a way that respects the rough skill breakdown given above. The first problems are those such as Problem 1 above that involve uniform velocity. Once those problems are mastered, the game will generate problems in which velocity is a step function of time. These problems require planning and recursive application of the formulas used to solve simpler problems. A third phase of the exercise will introduce uniformly accelerated motion, and a fourth phase will require the solution of problems such as Problem 2 above in which objects are uniformly accelerated at different rates in different intervals.

Each step in a sequence such as that described above has its own templete for generating problems and a corresponding template for help and remediation. The latter, described below in Section 3.3.4, are based on a generic solution path for a particular problem type. Explicitly representing problems in terms of solution paths also opens the door to model-tracing techniques [4].

3.3 Didactive Instruction

Whether in a game or not, students mastering an area of science, need a certain amount of didactive material. One way of providing that material is by reference to a text. One might, for example, ask students to read a particular chapter in a textbook before undertaking the instruction offered in a particular context. It is possible, however to embed such instruction in the game itself. We have used four mechanisms for doing just this.

3.3.1 Tutorials

One can provide a computer-based tutorial with each context. The tutorial can be nothing more than frame-based instruction such as that offered by Macromedia's *Authorware,* or even Microsoft *PowerPoint.* It should introduce the student to the "science" pertaining to the context and offer instruction in solving the problems provided in the context's exercise. In can contain the branching required to check student's understanding of the material.

3.3.2 Reference Library

One can also provide, in the form of a reference library support for looking up information and for browsing. We implemented this feature in an interesting way in *Electro Adventure* and hope to replicate the technique in *Twisted Physics.*

The reference library was organized around topics and entries that mapped to each other many-one. That is, each topic could have multiple entries. The topics were words or phrases that served as a table of contents to the library. Each entry presented material on its topic. The material could be in the form of text, graphics, or animation. Entries were extensively hyperlinked.

What makes the system function well with a game is the practice of keying entries to objectives (or contexts) in such a way that an entry becomes available to a player when and only when the player undertakes the associated objective. A topic becomes available only when one of its entries becomes available, and hyperlinks to an entry are invisible until the entry is available. This progressive disclosure technique allows the reference library to grow as the student progresses in the game.

3.3.3 Introductions and Reviews

Whenever the player entered a context in *Electro Adventure* he was treated to one screen's worth of text that the introduced him to the associated objective. This screen describes the scientific principles that applied to the context's objective and the kinds of problems to which they apply.

In addition, whenever a player left a context, he was treated to a screen's worth of text reviewing the material covered in the context.

3.3.4 Help

A major difference between recreational and instructional games relates to the pressure to complete the game. A recreational adventure game is often designed to keep the player engaged for as long as possible. An instructional game needs to be designed to get the player through the game in a timely fashion.

Thus, an instructional game must offer help in circumstances where the player is at an impasse. Help can, and should, be instructional in its own right. This can be done with a help system that can be invoked repeatedly in the same context, with different results on each invocation. Successive invocations can provide increasing levels of scaffolding by adding more and more definition to the context of the impasse. For example, successive requests for help in the context of a problem might

- mention the applicable principles, for example, "Any network of capacitors behaves like a single *equivalent capacitor.* You need to compute the size of the equivalent capacitor for this circuit;"
- help the student classify the problem, for example, "You first need to determine how the capacitors are connected. Capacitors are connected in series if a single pole of one is connected to a pole of the other. If each pole of one is connected to a pole of the other, they are connected in parallel;"
- describe the solution path, for example, "These capacitors are connected in series. This means that their equivalent capacitance is $\frac{1}{1/C_1 + 1/C_2}$;" and

- present the solution, for example, “In this problem, $C_1 = 10\mu F$, $C_2 = 15\mu F$, and the capacitors are in series. You need to compute $\frac{1}{1/C_1 + 1/C_2}$.”

3.3.5 *Visualization*

One of the barriers to qualitative reasoning in science is the abstract nature of the concepts involved. Charge, voltage, current, velocity, and acceleration are all invisible. Computer graphics allow us to render these abstract quantities visible so that students can inspect their behavior. This can be done by adopting a convention for displaying the concepts and then provide controllable animations that exhibit the relations among them.

In *Electro Adventure*, we used color to denote potential; red was more positive, blue was less positive, and ground was purple. We used an arrow to represent current and varies the darkness of the arrow to show the size of the current. These conventions were used in animated circuit diagrams to show how current and potential co-varied.

Visualizations can be quite effective in combating misconceptions that often plague qualitative reasoning about physics. For example, in *Twisted Physics*, the player can put on velocigoggles that show the components of an object’s velocity, and thereby show how the horizontal component of a falling object’s velocity has no influence on the vertical component.

4. Developing Instructional Adventure Games

It is a natural temptation to think about instructional games in the same way as one thinks about recreational games found on the commercial computer game market. Although we can take some lessons from the commercial game market, we can, and should, be more flexible in our thinking. Three issues have dominated my thinking about developing instructional adventure games.

4.1 *Strategy*

A prime consideration for developers of any kind of software, including instructional games, is that design is easier than development. This consideration plays out in several ways for the kinds of games described above.

Before going into detail on what this consideration implies for game development, I need to introduce a partition of the game itself in a way that makes sense for design and development.

- We can refer to the objects, events, and contingencies of the adventure itself as the *large-scale structure* of the game. This structure includes the network of contexts and their interconnections along with the objects and mechanisms for moving between them. It includes all other aspects of the fantasy world such as other characters.
- *Exercises* can, for design and development purposes, be segregated from the rest of the game. As is mentioned above, each exercise consists of a problem template and a procedure for selecting the parameter values of the template for each problem in the exercise.
- Also mentioned above are a number of ancillary *instructional mechanisms,* including tutorials, introductions and reviews, a reference library, and help.

- Finally, the design must arrange for a *wrapper* that integrates these components. A large part of this wrapper is the player interfaces to each component.

I have little to say about the last two components at this point, but the development of the large-scale structure of the game and of exercises merit some comment.

4.1.1 Large-Scale Structure

Anyone who has played a commercial adventure games knows that the range of possibilities for the large scale structure of a game is vast. One of the challenges of game design and development is that of disciplining the design of the large-scale structure so that it can be feasibly implemented and, more importantly, easily revised.

Our approach to this discipline is that of employing a language or formalism to represent the large-scale structure of the game and to limit the possibilities for this structure to those that can be expressed in the language. Currently, our inspiration for this language is a system known as the Text Adventure Development System (TADS) [5]. TADS is an object-oriented language for describing the fantasy worlds of adventures and an interpreter that can manage dialogs with a player of the sort found in early text adventure games such as *The Great Cave*. We see the same approach as applying to other sorts of player interfaces.

4.1.2 Exercises

From a design and development viewpoint, the most interesting thing about exercises is their dynamic nature. It is impossible to know how or how well an exercise will work without examining a sample of the exercises that it produces. Procedures that seem to generate perfectly reasonable problem sequences can turn out to generate completely unacceptable ones.

Hence, it is essential, before developing an exercise, to work with a functional prototype that does nothing more than generate sequences of problem parameters. If the prototype is sufficiently flexible, the designer can fine-tune the problem generation procedure before committing to development.

4.2 Style

The commercial game market is dominated by games that spare no effort on elaborate production with realistic animations and amazing special effects. These efforts are quite expensive and often involve extensive research and development projects simply to build the game development technology. Extreme efforts on the part of game manufacturers are justified because realism and effect are what sell recreational games.

The competition for instructional games, however, is not the latest edition of *Doom* [6] or some other commercial success. More often it is a textbook, a PowerPoint presentation or a “talking head.” Designers of instructional games, at least for now, can take advantage of the low-tech nature of the competition by thinking of alternate styles to the hi-fidelity, high tech game.

One approach is that of purposely lowering the sophistication of the media employed in the game to look more like that employed by the animation house, JibJab [7]; the television cartoon show, *South Park*; or the TV classic, *The Monty Python Show*. This type of indie production might result in instructional games with a very wide appeal, and it relieves one of competing in the crowded and expensive video-game market.

A second, more radical approach is that of using a text interface, such as that supported by TADS, for the large-scale structure of the space. This technique would, in effect, transfer the total burden of simulation, graphics, and animation in the large-scale space from the game program to the player's mind, on the hypothesis that the latter is well suited to the task.

Even more extreme is that notion of reducing the game to a set of exercises by reducing the large-scale space to one or menus for moving between contexts. Cliff Johnson's remarkable puzzle games (e.g., *3 in Three* [8]) are ample proof that this approach can support entertaining games in the adventure genre.

Finally, it is entirely possible to implement all of the features described above without any pretense of gaming at all. That is, one could provide, in conventional software for a network of contexts, an exercise for each context, and the associated instructional mechanisms. Such an approach might not have the appeal of a game, but it would probably be preferred to conventional instruction, computer-based or not.

4.3 Attitude

A final recommendation for design and development is easily put, but not obvious. Most instruction, computer-based or not, is designed and developed in a top-down, needs-driven, analytical, and rather dreary fashion. As the result, instructional design is an excellent way of producing really bad entertainment.

Conversely, if one hopes to produce a product with even modest entertainment value, the design and development of that product must themselves be fun. It as important to view the development of an educational adventure game as the creation of an interesting story as it is to accommodate the instructional objectives of the game.

References

[1] Halff, H. M. (November, 1994). Adventure games for technical education. *Proceedings of the 16th Interservice/Industry Training Systems and Education conference.* Orlando, FL: I/ITSEC.

[2] Gygax, G. & Arneson, D. (1974). *Dungeons & Dragons.* TSR.

[3] Van Lehn, K. (1987). Learning one subprocedure per lesson. *AI Jour*nal, *32* , 1-40.

[4] Anderson, J. R., Corbett, A.T., Fincham, J.M., Hoffman, D., & Pelletier, R. (1992). General principles for an intelligent tutoring system architecture. In J.W. Regian & V.J. Shute (Eds.), *Cognitive approaches to automated instruction* (pp. 81-106). Hillsdale, NJ: Erlbaum.

[5] Roberts, M. J. (2004). *The Text Adventure Development System*. Computer Software available at http://www.tads.org/.

[6] ID Software (2005). *Doom*^{3}. Computer software.

[7] Jibjab Media, Inc. (2004). http://www.jibjab.com/.

[8] Johnson, C. (1990). *3 in Three*. Computer software, available at http://www.fools-errand.com/.

Is support really necessary within educational games?

Henny LEEMKUIL & Robert de HOOG
University of Twente, Faculty of Behavioural Sciences,
Department of Instructional Technology
PO Box 217, 7500 AE Enschede, The Netherlands
h.h.leemkuil@utwente.nl
Phone: X - 31 - 53 - 4893651; Fax: X - 31 - 53 - 4892849

Abstract. Games can be powerful learning devices because of their interactive and multimedia capabilities, and their abilities to keep students motivated, active, deeply immersed and engaged for sustained periods of time. Yet the extent to which this translates into more effective knowledge and skill acquisition is not clear from the research reported so far. Several researchers have stressed that support tools should be added to game environments to ensure that learning takes place.

In this paper we will elaborate on this issue and will report data from experiments with a simulation game called KM Quest. In these experiments the effectiveness of several (learning) support tools was investigated. The data give indications that particular types of support (like advice) limit learning while others (feedback, just-in-time information) might enhance learning. Based on these data at the end of this paper it will be discussed whether (intelligent) support within educational games is necessary.

Keywords: simulation games, learner support, modes of learning.

Introduction

Games in potential are interesting educational tools because players (sometimes in cooperation with others) are actively solving challenging situated problems. In that sense they have elements in common with learning theories like constructivist learning, situated learning and collaborative learning. In these theories it is stressed that learning is an active social process in which meaning is given to experiences while solving situated realistic problems. Games can be powerful learning devices because of their interactive and multimedia capabilities, and their abilities to keep students motivated, active, deeply immersed and engaged for sustained periods of time. Yet the extent to which this translates into more effective knowledge and skill acquisition is not clear from the research reported so far. Klawe [1] concluded that relatively small changes in game design could strongly influence the extent of the learning effectiveness of the game. Garris, Ahlers and Driskell [2] agree that the potential for instructional games as platforms for training is appealing but they also state that "there is little consensus on game features that support learning, the process by which games engage learners, or the types of learning outcomes that can be achieved by game play. Ultimately we run the risk of designing instructional games that neither instruct nor engage the learner (p. 442)". So, little is known about the way students learn while playing a game and about game features that support learning. In this paper we will elaborate on this issue and will report data from experiments with a simulation game called KM Quest. In these experiments the effectiveness of several (learning) support tools was investigated. Based on these data at the end of this paper it will be discussed whether (intelligent) support within educational games is necessary.

1. Learning within games

We assume that players use different types of information processing while playing a game and that these different modes lead to different types of learning outcomes. This distinction is based on research by Berry and Broadbent [3], Norman [4] and Taatgen [5].

Berry and Broadbent [3] used a dynamic system, the Sugar Factory computer simulation, in which participants had to reach a certain level of sugar production by changing the number of employees. The behaviour of the simulation is based on a rule that is non-linear and contains a random component. Berry and Broadbent noticed that participants could achieve a good level of control of the system even though they remain unable to describe precisely the rules of the system in post-experimental structured questionnaires. They concluded that a task like this under certain conditions might be performed in some implicit manner. Berry and Broadbent observed a similar finding using a different task. On the basis of these results they suggested that two modes of learning could be distinguished, an implicit unselective mode (U-mode) and an explicit or selective one (that is effortful and reportable). U-mode learning is probable in situations in which there is much information in the learning environment and the key variables and their interrelationships are not salient. Especially in "rich" low transparent, interactive discovery simulations this leads to intuitive knowledge that is difficult to verbalize. Berry [6] states that these two modes of learning should be seen as the two extremes on a continuum, while Swaak and De Jong [7] assume that these two types of learning can be seen as two parallel, at least partly independent learning systems.

Taatgen [5] however, has a slightly different view. In his view implicit and explicit learning are a result of two competing modes/strategies: a search mode and a reflective mode. These are two competing strategies whose evaluations depend on the expected outcomes of the strategies. The expected outcome is determined by three elements: the estimated probability of reaching the goal using a specific strategy, the expected value of the goal, and the estimated cost of reaching the goal using the strategy. These estimates change in time due to increasing knowledge and the successes and failures of applying the strategy. In most cases people will start with a search strategy because reflective reasoning has a high cost, and it is not evident that the search strategy will be unsuccessful. Especially when one has little task relevant basic knowledge, the costs of a reflective strategy will be even higher. If the search strategy is not effective, and the estimated chance of reaching the goal will get lower, people might consider changing to a reflective strategy.

When using a search strategy implicit learning is a "by-product" of normal information processing. People are actively performing a task and while doing that unintentionally learn certain things (facts, procedures, instances, examples, sequences of actions). When using a reflective mode, learning is based on information processing based on learning strategies aimed at explicitly learning, comprehending or memorizing something.

The distinction made by Taatgen resembles the one made by Norman [4]. Norman describes two modes of cognition: an experiential mode and a reflective mode. "These two modes do not capture all of thought, nor are they completely independent (p.16)". The experiential mode is one of perceptual processing, it is a pattern or event driven activity. It requires some thought but the information processing is data driven and reactive. It leads to an accumulation of facts, it reactivates information that is already present in the memory system and it leads to a tuning and shaping of knowledge structures already available.

"The reflective mode is that of comparison and contrast, of thought, of decision making (p. 16)". It is slow and laborious. "Reflective thought requires the ability to store temporary results, to make inferences from stored knowledge and to follow chains of reasoning backward and forward, sometimes backtracking when a promising line of thought proves to be unfruitful……The use of external aids facilitates the reflective process by acting as external memory storage, allowing deeper chains of reasoning over longer periods of time than possible without the aids (p. 25)". According to Norman effective reflection requires some structure and organization and is greatly aided by systematic procedures and methods and the aid of other people.

This implies that the use of tools (inside or outside the environment) or the help of other people can facilitate the reflective process. Garris, Ahlers and Driskell [2] confirm this: "simulation games may be ineffective stand-alone training tools because people do not learn from simple exposure or experience alone to understand complex relationships …… Although our goal is to achieve self-directed, self-motivated learners, we must provide support for knowledge construction. The role of the instructor in debriefing learners is critical (if somewhat overlooked) component in the use of instructional games, as are other learner support strategies such as online help, cues/prompts, and other activities (p. 460)". A review of games research revealed that the following tools may be effective in supporting learning with games: cooperation and collaboration, debriefing and group discussions, hints and prompts, a help or advice system, additional assignments, feedback, monitoring facilities and the specification of goals.

2. KM Quest: A collaborative Internet based simulation game about knowledge management

In our research we used KM Quest: (http://www.kmquest.net) a collaborative Internet based simulation game about knowledge management. Several universities and institutions for higher education in the Netherlands like the University of Twente, the Radboud University in Nijmegen, the University of Utrecht, and Hogeschool Zuyd in Maastricht have used the KM Quest learning environment in courses on knowledge management. Furthermore, the Yeditepe University in Istanbul (Turkey) has used it. Goal of the game is to learn basic knowledge management concepts and actions and the steps of a systematic approach to solve knowledge management problems. Furthermore to learn to assess the KM situation of an organisation and to advise/implement appropriate interventions.

The simulation game can be played by three players who all play the same role of knowledge manager in a fictitious large product leadership organisation named Coltec for three years in the lifespan of the company (divided in 12 quarters). Collaboratively they have the task to improve the efficacy of the company's knowledge household. This is not an aim in itself but is related to objectives for the (management of the) company in general. The general goal of the simulation game is to optimise the level of a set of general organizational effectiveness variables: market share, profit, and the customer satisfaction index (that are in the top level of an underlying business simulation model that is used to simulate the behaviour of the company) by influencing the effectiveness and efficiency of knowledge management processes (knowledge gaining, development, retention, transfer and utilisation) that are in the lower level of the business model. These processes can be influenced by choosing and implementing interventions from a pool of 57 possible interventions.

In the game, players can use several resources while performing their task. They can inspect the status of business process indicators and knowledge process indicators (in three general domains) that are incorporated in the business simulation model, they can ask for additional information, and they have to choose knowledge management interventions to (try to) change the behaviour of the business simulation indicators. Most of these indicators are characterized by a decay factor. This means that the value of the indicators decreases over time when no interventions are implemented. The implementation of interventions involves costs, as well as several other activities that the players can perform. Players receive a limited budget that they can use to implement interventions and buy information.

Changes in the status of the business indicators will only be computed at the end of each quarter. There is no time limit to playing the simulation game. Teams set their own pace. When players think they know enough to solve the problem they indicate that they agree with the proposed interventions (by using a voting tool). After they have reached agreement the simulation game proceeds to the end of the quarter and the business simulation will calculate new values for each of the business indicators. The game ends after the players have indicated that they have implemented the last intervention(s) in the fourth quarter of the third year in the life span of the company.

To trigger activities from the players and to make sure that players are confronted with different types of knowledge management problems, at the beginning of each quarter an (unexpected) event is introduced that could affect the knowledge household of the company. Players have to decide if and how they want to react on these events. Events are generated from a pool of 50 events. Different types of events can be distinguished based on two dimensions: the locus of the event (internal or external), and the effect of the event (direct, delayed, or no effect). Effects either can be positive or negative.

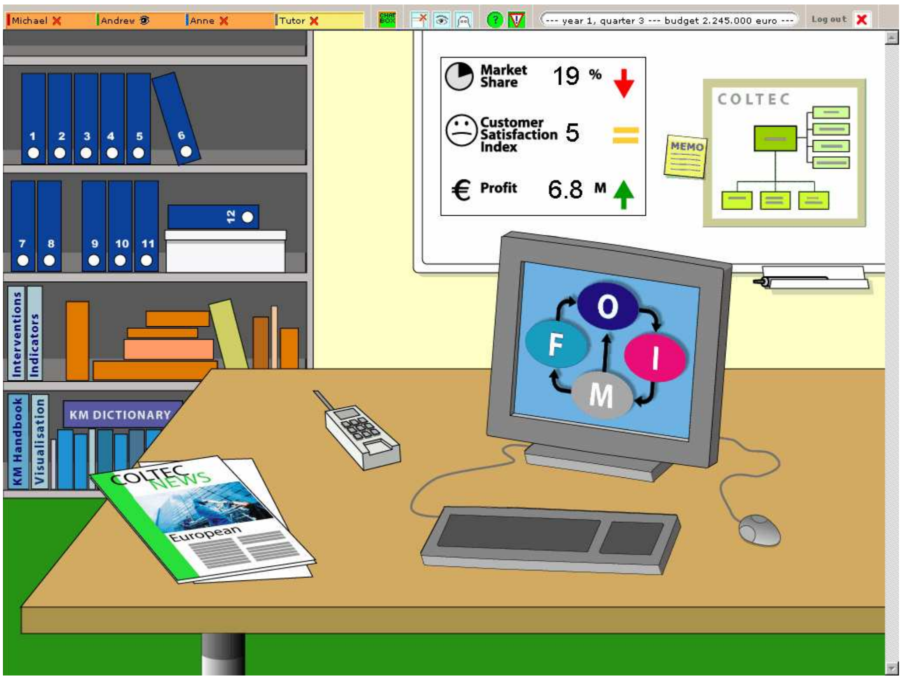


Figure 1. Virtual office interface of KM Quest.

Players can interact with the environment and with each other by using tools and resources that are presented in an Internet environment, based on a “virtual office metaphor” (see Figure 1). Clicking on a specific element in the “office” will open a window with additional resources or tools. To support the learners in performing their task and to support learning while playing the game several features have been implemented in the environment:

- A knowledge management model that describes a systematic approach to solving KM problems.

- Shared worksheets related to the steps in the knowledge management model (accessible by means of the computer).
- Just-in-time information (mainly in books on the bookshelf).
- Feedback (consisting of dynamic data from the simulation model and pre canned conceptual knowledge about knowledge management that is based on the experiences from KM experts and is coupled to certain events).
- Advice (only available when the value of certain indicators gets below a threshold value).
- Visualisation tools (different types of visualisations to display the data from the business simulation model).
- Monitoring tools (to be able to monitor their own behaviour and to reflect on it 12 quarterly reports are available on the top bookshelves).

To support collaboration and communication at a distance tools are implemented like a chat box, monitoring facilities, a voting tool, shared worksheets, and embedded forums. These tools support synchronous as well as asynchronous communication between team members.

3. Data from experiments with the learning environment

In the KM Quest simulation game several tools/resources were embedded to support learning as stated above. Logfiles from a first experiment with this game showed that from these resources the advice functionality was used most often. Advice was available when the value of certain knowledge process indicators in the business simulation model went below a threshold value.

It appeared that in 83% of the cases where advice was available it was actually consulted. Most of the time it was the first resource players consulted when they entered a new quarter in the game. Based on the available data from that experiment however it was hard to say whether the use of advice supported learning and game play.

A literature review also did not lead to many cues about the role of advice on learning in these kinds of learning environments. There is a study by Leutner [8] that also focuses on advice in a simulation game. In this research in the advice groups, during the game the players are provided with warnings if their decisions are likely to lead to problems. Results of the experiments (with 7th grade students and with university students) showed that advice increased verbal domain knowledge, but decreased game performance. Furthermore the data indicated that system-initiated adaptive advice had short-term effects (measured directly after game play), while learner requested non-adaptive background information had long-term effects (measured by a test that was administered a week after game play).

The type of advice given in the KM Quest environment is slightly different from the type of advice that was available in the Leutner study. In that study warnings were provided. In the KM Quest environment a warning is also given, but this is accompanied with general suggestions to improve the status of the specific indicators. An example of the advice is given below:

“The value of "Efficiency of knowledge gaining in research" has dropped below a value of 4.5. This may be a reason for concern. It is possible that you overlooked this decrease of this variable because you focused to much on the event or did not include it in your measurement system. If you want to improve the value of "Efficiency of knowledge gaining in research", you could consider interventions that influence this value. Most of the time several interventions are available. In this case you could take a look at interventions of the type listed below (see Interventions handbook). Note that there maybe other interventions also.

- Interventions that aim at cooperation with partners, implementation of new projects or hiring services

- Interventions that aim at the implementation of Information Communication Technology
- Interventions that aim at recruiting and hiring people”.

When a player clicks on a class of interventions the chapter of the interventions handbook will be opened in which additional information about these interventions is given.

To explore the effects of the advice on learning in another experiment two versions of the simulation game were used: one with the advice and one in which this resource was removed. A pre-test, post-test design was used with an additional transfer test. It was hypothesized that students in the advice group would gain more explicit knowledge (in line with Leutners findings) and would perform better in the game because they receive hints about possible solutions. This last hypothesis is not in line with Leutners findings, but this is because the type of advice given in this experiment is somewhat different than the advice given in the experiments performed by Leutner.

3.1 Method

In this section the subjects that participated in the research will be described as well as the instruments that were used and the experimental procedure.

The students who participated in the experiment did so because the use of the simulation game was an assignment within a third year course “Knowledge management in learning organizations” at the University of Twente. All students were from the Communication Studies department and had followed a preliminary course on Knowledge management in the second year. 29 Students participated in the course (18 women and 11 men).

The game environment that was used in both conditions was slightly different than the standard version that is described above. In this case players played individually, the order of the events was not random but fixed and the same for all students, and the starting values of the indicators were lowered (several are below 5 on a ten point scale). This is done to create a situation in which events can occur that refer to internal problems related to knowledge processes. In the original game set up in the beginning of the game a lot of events are generated that have to do with external (opportunity) events because knowledge processes within the company still were running smoothly (score of 6 or higher on a ten point scale). Observations in try-outs showed that players found these kinds of events difficult to deal with when they had limited domain knowledge.

Knowledge gain is measured by comparing the score on a pre-test and a post-test. Each consisted of 26 items that contain questions to measure explicit (18 items) as well as implicit knowledge (8 items). Explicit knowledge of knowledge management concepts and principles is tested by giving students textual multiple-choice questions that refer to declarative knowledge about concepts used in the learning environment like the knowledge processes, and steps within the knowledge management model. Furthermore, questions are included that refer to relationships between indicators and interventions. Implicit knowledge items are included to measure intuitive knowledge that is difficult to articulate. These items are based on the guidelines given by Swaak and de Jong [7] who characterize intuitive knowledge by a “quick perception of anticipated situations”. In the test a situation is given, an action is described and a set of possible post action situations is given. In each item the textual information is kept to a minimum and a picture or chart is used to present the alternatives.

A transfer task was administered after the game and the post-test were completed. The transfer task used a different type of company: a small customer relationship type of company. In the task a case description was presented with four event descriptions. For each event 4 questions were presented.

Player performance in the game was logged into a database. These data make it possible to see for each of the twelve quarters within the game, which resources people used (and in which order), which interventions they implemented and what the values of the business and knowledge process indicators were.

3.2 Procedure

In the first lecture of the course the set up of the course was described and KM Quest was introduced. In four consecutive weeks following on this lecture four two-hour sessions were planned to work with the KM Quest environment. The sessions were located in a computer room with a large set of computers. Each student had a computer. In the first session the pre-test was administered. After the game was completed the post-test was administered in the fourth session. After the students finished the post-test they received transfer task, which they had to make at home. Students were not able to get access to the game environment while they were making the transfer task.

3.3 Results

First data will be presented from the tests and about game performance. After that data will be presented about the use of advice and the other resources.

In Table 1 the test results are summarised. It shows that on the average students answered 39.6% of the questions of the pre-test correctly. For the post-test the average score was 50.3%. A paired samples t-test shows that this knowledge gain is significant (T=-4.62, df=28, p=0.00).

The data show that both groups have gained explicit knowledge. This gain is mainly in the conceptual domain. There is no significant gain in knowledge related to the knowledge management model, nor on explicit knowledge concerning relationships between indicators and interventions. The No-advice group however, has also gained implicit knowledge, while the advice group did not show any gain in this domain. The difference between both groups on the score on the implicit knowledge items of the post-test is nearly significant (Anova: F=4.18, df=1, p=0.051).

Table 1. Average number of correct answers (and standard deviations) on the explicit and implicit knowledge items of the pretest and posttest and the score on the transfer test for the whole group and for the groups with and without advice.

	Pre-expl Max=18	**Pre-impl Max=8**	**Pre-tot Max=26**	**Post-expl Max=18**	**Post-impl Max=8**	**Post-tot Max=26**	**Transfer Max=10**
Advice	6.80	3.80	10.60	8.80	3.67	12.47	7.20
N=15	(1.86)	(1.61)	(2.87)	(2.04)	(1.35)	(2.77)	(1.53)
No adv.	6.86	3.14	10.00	9.07	4.64	13.71	6.98
N=14	(2.88)	(1.56)	(3.35)	(1.97)	(1.22)	(2.30)	(1.71)
Total	6.83	3.48	10.31	8.93	4.14	13.07	7.10
N=29	(2.36)	(1.59)	(3.07)	(1.98)	(1.36)	(2.59)	(1.59)

Looking at game performance there are hardly any differences between the experimental groups (see tables 2 and 3). Both groups have managed to improve the business and knowledge management (process) indicators significantly.

Table 2. Average values for some important business (process) indicators in the business simulation model at the end of the game for the whole group and for the groups with and without advice. At the bottom row the starting values.

	MS*	CSI	Profit	JSI	PQI	ATM
Advice	23.79	8.15	64.73	9.60	8.77	1.34
	(3.76)	(0.90)	(23,26)	(0.88)	(0.99)	(0.77)
No adv.	24.43	8.33	68,58	9.50	9.04	1.14
	(3.18)	(0.61)	(19,54)	(0.76)	(0.66)	(0.55)
Total	24.10	8.23	66,59	9.55	8.90	1.24
	(3.45)	(0.77)	(21,25)	(0.81)	(0.84)	(0.67)
Start values	20.0	4.40	-	4.22	4.80	2.40

* MS = Market share, CSI = Customer satisfaction index, Profit = Total profit in 3 years in millions, JSI = Job satisfaction index, PQI = Product quality index, ATM= Average time for new product to market,

Table 3. Average level of competence in three knowledge domains for the whole group and for the groups with and without advice. At the bottom row the starting values.

	Competence in marketing	Competence in R&D	Competence in production
Advice	8.63	8.27	6.43
	(1.22)	(1.29)	(0.79)
No advice	8.72	8.60	6.40
	(1.00)	(1.02)	(0.71)
Total	8.68	8.43	6.41
	(1.10)	(1.16)	(0.74)
Start values	4.93	4.38	5.34

Advice is only available when certain indicators get below a certain threshold value. On the average advice was available in 9.33 quarters and was consulted in 7.07 quarters. This means that in 77% of the cases where advice was available it was actually consulted.

Table 4 gives an indication about the frequency of use of the other resources available in the learning environment. The Intervention handbook, the feedback and the visualization tool were the most used resources. The indicator handbook and the history books were used less frequently, and the other resources are sparsely used.

Although there are differences in the use of tools between the two experimental conditions, most of them are not significant, except for one. The no-advice group consulted the indicator handbook significantly more (Anova: F=5.22, df=1, p=0.03) than the advice-group.

Table 4. Average number of quarters in which some of the standard resources were used by the learners for the whole group and for the groups with and without advice. The resources are: feedback, the intervention handbook, the indicator handbook, history files, what and how files related to shared worksheets, and visualisation tools.

	Feedb	Intv HB	Indc HB	Hist	What	How	Help	Visual
Advice	5.73	6.13	3.13	3.67	0.80	1.00	0.47	6.73
	(4.33)	(4.36)	(1.99)	(2.19)	(0.86)	(0.76)	(0.52)	(4.93)
No adv.	5.93	4.57	5.43	3.14	1.00	1.14	0.50	5.29
	(5.08)	(4.50)	(3.30)	(1.29)	(0.55)	(0.86)	(0.52)	(3.99)
Total	5.83	5.38	4.24	3.41	0.90	1.07	0.48	6.03
	(4.62)	(4.42)	(2.90)	(1.80)	(0.72)	(0.80)	(0.51)	(4.48)

4. Discussion

The data indicate that the players appreciated the advice functionality, but there are also indications that the effectiveness of the advice given was low. Players in the no-advice condition score equally well on game performance, on the explicit knowledge items of the post-test and on the transfer test. The fact that in the no-advice condition there is a significant knowledge gain on test items that measure implicit knowledge while there was no gain in the advice condition even seems to indicate that advice (as it is given within the game) prevents the gain of implicit knowledge and the transfer of this type of knowledge into explicit knowledge. This could be due to the fact that players in the advice condition put in less mental effort in solving the knowledge management problems within the learning environment and rely too heavily on the suggestions given by the advice functionality. The players in the no-advice condition seem to be seeking more actively for data that could support problem solving. This is indicated by the fact that players in this group consulted the indicator handbook significantly more than the other group. A counter indicative finding could be that the advice group consulted the intervention handbook more often (although not significantly). This however, is probably caused by the direct links between the content of the advice and the content of certain parts of the intervention handbook.

There are some indications that players who are actively searching for information in the games resources are learning more or performing better within the game. Frequent use of feedback or the intervention handbook improved learning in a previous experiment. This finding was not replicated in this experiment. Frequent users had higher post-test scores but did not have a larger knowledge gain. This could be due to the fact that in the reported experiment the players who used these resources often already had a “high” level of prior knowledge, while this was not the case in previous experiment. The data reported in this paper however showed that frequent use of the indicator handbook lead to higher scores on the explicit knowledge items of the post-test and on game performance measured by the level of some of the business process and knowledge process indicators included in the business simulation model.

The data are in line with findings of other experiments with the same learning environment that are reported by Purbojo and de Hoog [9] and Shostak and de Hoog [10]. These data show that the availability of certain types of support resources like certain types of visualisations or certain types of indicator values, per se does not have an influence on learning results or game performance. These experiments however also show that a briefing session before game play is important as well as the fact that players have to cooperate. There are indications that a certain types of support (like the advice that is available) limit learning. When there is less directive support students are more actively searching for information in the resources available and while doing this gain more conceptual and implicit knowledge. Discussion with others during or after game play seems to support learning with games the most, as is also reported by Kirriemuir [11]. Since relatively simple support within the game seems to make players “lazy”, it can be questioned whether more advanced intelligent types of support will be effective in supporting learning within these kinds of environments. In the case of KM Quest more intelligent support could take the following form: A new version of KMQuest stores all user actions in a well-structured data base. As the program works with fixed quarters after which the values in the business model are adjusted, the moment of a quarter change also allows an analysis of the content of the data base. This analysis can lead to advice about visiting certain activities in the knowledge management model that were overlooked but are necessary for better learning. Another option is to predefine certain preferable learning trajectories and check the actual behaviour of the learners against these trajectories and give tailored advice about what to do next. A vision of the future could even be an online analysis of the content of the chat (which is also stored) to detect certain characteristic utterances that could signal learning difficulties or misconceptions that could be redressed by specific suggestions given by the system.

References

[1] Klawe, M. M. (1998). *When Does The Use Of Computer Games And Other Interactive Multimedia Software Help Students Learn Mathematics?* Electronically available at: http://www.cs.ubc.ca/nest/egems/reports/NCTM.doc.

[2] Garris R.; Ahlers R.; Driskell J.E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & Gaming, 33*, 4, 441-467

[3] Berry, D. & Broadbent, D, (1988). Interactive tasks and the implicit-explicit distinction. *British Journal of Psychology, 79*, pp. 251–272

[4] Norman, D.A. (1993). *Things that make us smart. Defending human attributes in the age of the machine.* Reading MA: Addison-Wesley Publishing Company.

[5] Taatgen, N.A. (1999*). Learning without limits. From Problem Solving towards a Unified Theory of Learning*. Rijksuniversiteit Groningen (dissertation)

[6] Berry, D. (1993). Implicit learning: reflections and prospects. In: Baddeley, A. & Weiskrantz, L. (Eds.). *Attention: Selection, awareness, and control. A tribute to Donald Broadbent*. Oxford: Oxford University Press

[7] Swaak, J., & de Jong, T. (1996). Measuring intuitive knowledge in science: The what-if test. *Studies in Educational Evaluation, 22*, 341-362.

[8] Leutner, D. (1993). Guided discovery learning with computer-based simulation games: Effects of adaptive and non-adaptive instructional support. *Learning and Instruction, 3*, 2, 113-132.

[9] Purbojo, R. & Hoog, R. de, (2004). Learning knowledge management in a collaborative game: Effects of player preparation and visualisation of variables. In: Eberle, T., Kriz, W.C., Puschert, M. & Glötzner, F. (Eds.). *Bridging the Gap: Transforming Knowledge into Action through Gaming and Simulation. Proceedings of 35th Annual Conference of the International Simulation And Gaming Association (ISAGA) Munich 2004*. Munich: Sagsaga.

[10] Shostak, I. & Hoog, R. de, (2004). The KM Quest business simulation model: supporting acquisition of decision-making skills for knowledge management. In: Eberle, T., Kriz, W.C., Puschert, M. & Glötzner, F. (Eds.). *Bridging the Gap: Transforming Knowledge into Action through Gaming and Simulation. Proceedings of 35th Annual Conference of the International Simulation And Gaming Association (ISAGA) Munich 2004*. Munich: Sagsaga

[11] Kirriemuir, J. (2005). A survey of COTS games in education. Paper presented at the Serious Games Summit on the Game Developers Conference, San Francisco, march 2005.

Building an Intelligent Pedagogical Agent with Competition Mechanism to Improve the Effectiveness of an Educational Game

Zhangbo Liu
Univ. of Electronic Sci. and Tech. of China
2-4 North Jianshe Rd.
Chengdu, Sichuan, 610054, China
zhangboliu@yahoo.com.cn

Zhi Liu
Peking University
No.5 YiHeYuan Rd. Haidian District
Beijing, 100871, China
pernula@pub.ss.pku.edu.cn

Abstract

Different kinds of pedagogical agents to interfere student's learning during the gaming procedure are proposed to improve the effectiveness of educational games in tutoring learning. However, to assess human being's emotions is a tough task and excessive input on the assessment of student's emotions during the game will degrade the level of engagement, thus degrading students' motivation. In this paper, we propose intelligent pedagogical agent with competition mechanism based on Bartle Player Model. The study results show clear increase in the effectiveness of educational games with our approach.

Key Words: Intelligent Agents, Bartle Player Model, Educational Games

1 Introduction

Electronic games nowadays are hot in educational field because they are popular among students. Researchers have been investigating whether and how electronic games can be used to assist learning [1]. Although electronic games are not regular tutoring environments because of their special traits and entertaining goals, we believe that they can be leveraged to help student's learning, if well designed.

Some pedagogical agents have already been designed to improve student learning during game playing. Some studies aimed at making pedagogical agents that can both simultaneously stimulate learning and maintain the high level of engagement, as educational games usually generate [2], and our approach just follows this way.

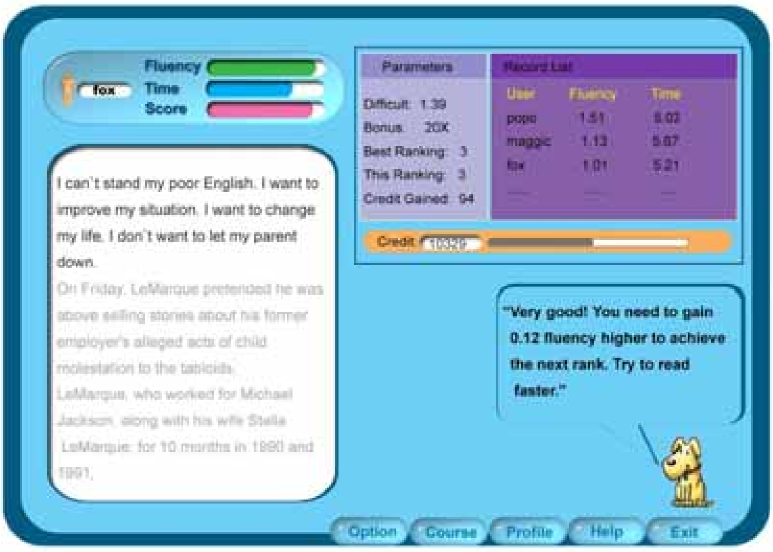
In our educational game, As Fluent As Possible, we have a multi-user environment. It is more complicated and difficult to balance the learning and entertaining within a multi-user game than a single-user game. Bartle's Player Model [3] is a well known model which helps describe player's types within a MUD (Multi-User Dungeons). We leverage Bartle Player Model to design a competition mechanism for our game so as to incorporate with the existing intelligent pedagogical agent in tutoring learning.

2 The Game and Existing Pedagogical Agent

Empirical survey shows that a large percentage of Chinese students can hardly take basic English communications with others after their graduation from colleges. One reason is that current English learning approach excessively emphasizes on reading and writing. As Fluent As Possible provides a chance for students to develop their spoken English.

However, at the beginning, student can only practice single sentences, get scores and obtain the agent—Rocky's unsolicited suggestions according to the comparison result between his speech samples and the benchmark operated by the engine.

This educational game has several drawbacks. First, student may feel tedious after the first exciting period. Second, a new comer can hardly get enough credits to surpass another student whose ability are at the same level but have got several more hours practice time. Last but not the least, information within the suggestions from the agent seems only useful to those students whose purpose of playing the game is to improve their English speaking skills.



I can't stand my poor English. I want to improve my situation. I want to change my life. I don't want to let my parent down.

On Friday, LeMarque pretended he was above selling stories about his former employer's alleged acts of child molestation to the tabloids. LeMarque, who worked for Michael Jackson, along with his wife Stella LeMarque, for 10 months in 1990 and 1991.

Difficult: 1.39
Bonus: 20X
Best Ranking: 3
This Ranking: 3
Credit Gained: 94

User	Fluency	Time
popo	1.51	5.02
maggic	1.13	5.07
fox	1.01	5.21

"Very good! You need to gain 0.12 fluency higher to achieve the next rank. Try to read faster."

Figure 1: As Fluent As Possible

3 Competition Mechanism

3.1 Killers vs. Achievers

To overcome the above disadvantages, we find that the biggest challenge is how to guarantee the student's level of engagement. First of all, we need to maintain a balanced relationship between the different types of players in this multi-player educational game.

According to Bartle Player Model, there are four types of players within a multi-player game. While in our study, we concentrate mainly on two of them: Killers and Achievers. Figure 2 clearly illustrates the relationship between them.

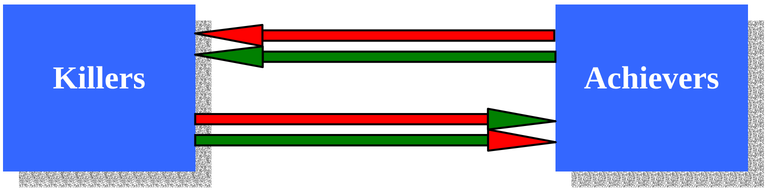


Figure 2: Killers and Achievers

In Figure 2, green color indicates increasing numbers while red color indicates decreasing numbers. A red line with a green arrowhead means that decreasing numbers of the box pointed from lead to increasing numbers of the box pointed to; a red line with a red arrowhead would mean that a decrease in one leads to a decrease in the other, and so on.

Increasing Achievers We use two approaches to achieve that:

a. *Raise the rewards for achievement*. We import the concept of Bonus, which stands for extra credits to student who breaks the records. The formula is:

$$Bonus = 10 \times C \times \lfloor n/3 \rfloor \times k \qquad (*)$$

where C represents Credits which the learner gets at a time;
n represents Record break times;
and k is a variable defined by record rank, when record rank = 1, 2, 3, k = 2, 1.5, 1.
b. *Set an extensive level system*. We propose a formula for level calculation:

$$Credit = 10 \times \left(L^3 + 4 \times L\right) \qquad (**)$$

where L represents each level;

Following this non-linear formula, it will be harder and harder for students to reach the next level. However, that's what Achievers prefer. By operating these two approaches, the interests of Achievers can be greatly enhanced.

Maintaining Killers Although decreasing the number of Killers can increase the number of Achievers, keeping a certain amount of Killers is necessary for balance. We provide Killers opportunities to demonstrate their abilities. Each sentence in the course has a record which maintains the top three students in this sentence. Every time a student breaks the top three records he will be given a considerable bonus according to (*). At the top right corner we can find the record list.

Here *Fluency* is the metric of student's performance whose formula is:

$$Fluency = \frac{Score}{Time} \times \frac{TimeofBenchmark}{ScoreofBenchmark} \qquad (***)$$

Since Killers always try to seize all the records and kick other students out of the top list, it is easy for some students who do not have much potential to compete with Killers to give up and leave. In order to avoid this situation, we endow different level records with different refreshing frequency so that Killers in a higher level are not able to disturb the order in lower levels, thus protecting new comers.

3.2 Refined Pedagogical Agent

After adding the competition mechanism into the game, Rocky can provide some more suggestions on the rank relationship between the student and others thus enhancing the interactions among them. Table 1 demonstrates possible suggestions from Rocky:

S-1	You need to improve.
S-2	Your speech is good. Try to read the word "*someword*" louder/less louder/higher/lower/longer/shorter (and louder/less louder/higher/lower/longer/shorter).
S-3	Your speech is excellent!
S-4	Very good. You need to gain "*somenumber*" fluency higher to achieve the next rank. Try

	to read faster (and/or pronounce better).
S-5	Excellent! You've got the best rank of this sentence!

Table 1: Sample Agent's Suggestions

The S-4 and S-5 are added in the current version. Take two types of students in the game into account; they will both get benefits from those suggestions. For example, speaking of S-4, Killers can get to know the gap between him and the one in front of him; while Achievers will be glad to know which he would pay more attention to improve, the pronunciation, the time, or both, in order to get a high rank.

4 Study and Evaluation

Our user study involves ten college students in the game for a period of three weeks practice. Before the study began, we did an interview in order to assess the student's English skill levels. The result showed that two out of ten students were evaluated to be *Reasonable*; others were all evaluated to be *Not So Good*. Then we divided them into two groups whom were given different versions of As Fluent As Possible: group A was given the former version and group B the new version.

Evidence showed that although both two groups gradually practiced less during three weeks, students in group B practiced more sentences and gained more credits than students in group A, mainly due to the competition mechanism. At the end of the third week, we collected their data and drew some diagrams, see Figure 3.

Then we ran a post-questionnaire with all students. Students in group A seemed to be the same in the self and mutual evaluation with one exception; while four out of five students in group B were evaluated to improve. In the comments they acknowledged that the game, especially the competition mechanism successfully kept their motivations and enhanced their self confidence, so that we can guarantee that the effectiveness of the educational game is improved by the modification.

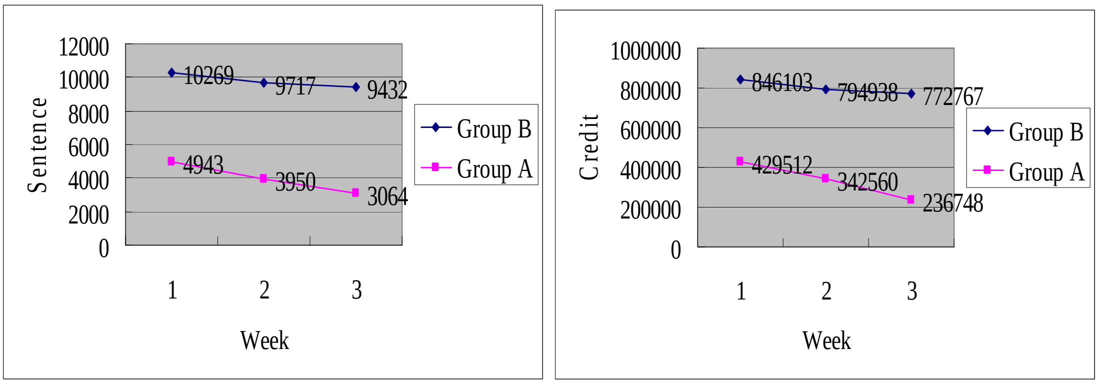


Figure 3: Study Statistics

5 Reference

[1] Randel, J.M., B.A. Morris, C.C. Wetzel, and B.V. Whitehill. *The effectiveness of games for educational purpose: A review of recent research.* Simulation & Gaming. 1992.

[2] Conati, C. and Zhao X. *Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game.* International Conference on Intelligent User Interfaces 2004.

[3] Bartle, R.A. *Hearts, Clubs, Diamonds, Spades: Players Who Suit Muds.* JVW. Volume 1, 1997.

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.

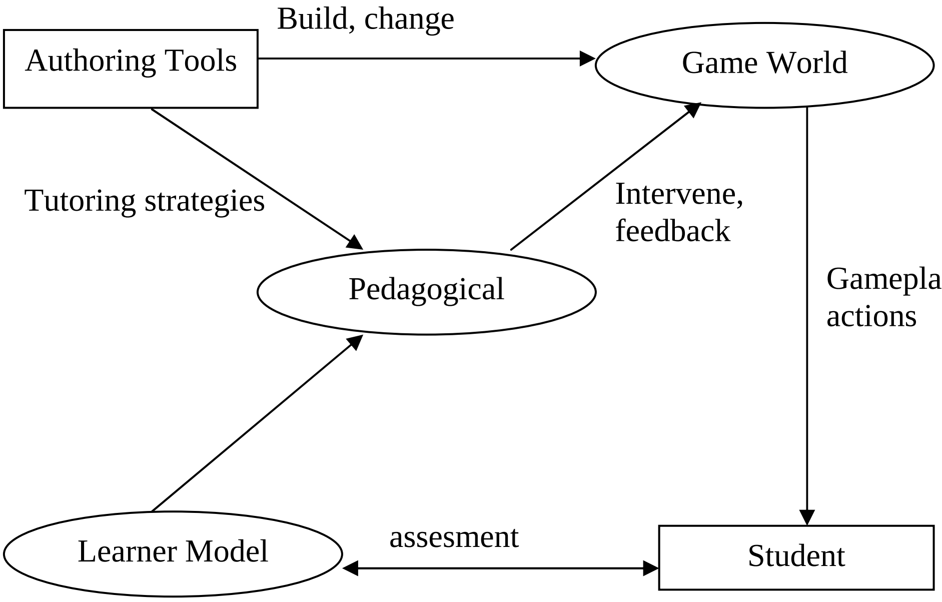


Figure 1. Architecture of Intelligent Educational Game

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. Trappl,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

MITO: an Educational Game for Learning Spanish Orthography

Eva Millan, Cristina Carmona, Roberto Sanchez
and José Luis Perex-De-La-Cruz
Departamento de Lenguajes y Ciencias de la Computación,
Universidad de Málaga
{eva,cristina, perez}@lcc.uma.es

Abstract. An educational game is a recreational activity designed to teach people, typically children, about a certain subject or help them learn a skill as they play. These games are usually successful in engagement, but sometimes fail in triggering learning. When designing an educational game for children, a way to avoid this could be to take into account the cognitive stage of the final user, as defined in Piaget's cognitive development theory. This paper briefly explains the fundamentals of this theory and discusses its implications in the design and development of an educational game to learn Spanish orthography, focused on children on the concrete operational stage.

1 Introduction

Learner motivation is one of the main objectives of any pedagogical activity. Electronic educational games are learning and recreational environments that try to increase the learner's motivation by embedding pedagogical activities in highly enjoyable interactions. In [1], a former review of literature on the effectiveness of games versus traditional classroom instruction is made. By analysing 68 studies from 1963-1991 in social science, math, language arts, logic, physic and biology, it was determined that games were more effective in the domains of language arts and mathematics. The reason is that, in these fields, greater specificity of content and more effective use of computers created a clear advantage for the exercises over traditional teaching methods. Furthermore, greater retention was shown in games, and students reported greater interest in game activities than in more conventional classroom instruction. However, there is little empirical evidence that electronic educational games can promote learning unless the interaction is led by teachers and integrated with other instructional activities [2]. A possible reason for this limitation is that learning how to play does not imply learning the domain [3].

Another important point that must be taken into account for learner motivation is that the educational game should adapt the proposed activities to the learner's cognitive stage. Probably, one of the most accurate and commonly used definitions of children cognitive stages is the defined by Piaget's theory of cognitive development [4], [5]. The cognitive abilities that Piaget studied are important when teaching young students, as they help determine how much and in what way students will understand the topic being taught.

Next section explains briefly the theory of cognitive development and the kind of games appropriate for children in the concrete operational stage (target of our system), and some considerations about designing and developing good educational games. Section 3 presents a short review of related work. Then, the electronic educational game MITO (which stands for

Multimedia Intelligent Tutor of Orthography) is presented. MITO[1] is a stand-alone application focused to help children between 8 and 12 years old learn Spanish orthography. Finally, the paper concludes by presenting some conclusions and future lines of research.

2 Psychological considerations for the design and development of educational games

This section is structured as follows: first a short description of Piagetian cognitive stages is presented. Then, the more appropriate games for each stage are discussed, and finally some considerations about the relevant features of a "good" educational game are made.

Piaget theorised that intelligence is built on in a series of stages. These stages always appear in the same order and can usually be determined by a child's age. Piaget also found there was sometimes deviations from the norm as well as possible acceleration or delay in mental age and abilities. This lead him to believe there was more than biological maturation creating the different developmental stages [5]. These four stages are: sensorimotor stage (0-2 years); preoperational stage (2-7 years); concrete operational stage (7-11 years), and formal operational stage (11-15 years). As our game has been developed for the concrete operational stage, we will only present relevant information concerning that stage:

In the concrete operational stage, children are capable of taking another person's point of view and incorporating more than one perspective simultaneously. At this stage the child begins to think logically with concrete knowledge. They understand conservation and their way of thinking is reversible (they can follow their line of reasoning back to its starting point). Their thought pattern is now logical and systematic, making it easier for them to find answers to simple problems (classification, combinations, etc.). Main limitations are the lack of abstraction capabilities and that usually thinking is limited to two characteristics at the same time. Piaget found that, though all children went through the stages of learning in the same order; the age that an individual progressed to the next stage could be quickened or stunted depending on how stimulating the learning environment is. This consideration should be taken into account when designing educational games for children.

In [6] we can find the relevant characteristics and more appropriate games for each stage. Such information for the concrete operational stage is summarized in columns 2 and 3 of Table 1, while in column 4 we show our conclusions about the implications in the design of good educational games for each period.

Stage	Relevant Characteristics	Appropriate Games	Implications For Computer Educ. Games
Concrete Operational stage	Children in this stage to represent things mentally and to manipulate such representations. They are able to tackle with different points of view, however they still lack abstraction capabilities.	Games with rules, figurines, characters, learning and informative toys, constructions, arts and crafts, remote control items, video and computer games	Almost any game (apart from those that need abstraction) can be included in an educational game and help in triggering significant learning.

Table 1. Relevant information for the design of games for the concrete operational stage

With regard to the definition of the "quality" of an educational game, we have made a literature review looking for relevant features that good games share. Next we present a list with useful quotations found in this literature review (together with some comments). From this quotations, a list of keywords that describe desirable features of a good educational game

---

[1] MITO is a tool designed with educational and non-commercial purposes. Installation files and instructions are accesible at http://www.lcc.uma.es/~eva/mito.

has been extracted. Such keywords are presented in brackets in the text, and will serve as a basis for the design of our educational game.

- "Like all instructional materials, educational games need to be developmentally appropriate. A specific game should be appealing and accessible to the target level of development" [7]. (CHALLENGING, ACCESSIBILITY, ADAPTATION).
- "Students must overcome many barriers to engagement, including fear of failure, fear of embarrassment, and aversion to losing control" [8]. (PRIVACY).
- "Game engages players through interactivity. Players take ownership of their choices, because those choices influence the outcome…. Success in a game can usually be clearly traced to effort and ability" [8]. (INTERACTIVITY, OUTCOMES, REINFORCEMENT, IMMEDIACY).
- "Good games are often distinguished by great freedom" [8]. To this respect, we would like to remark that such freedom must be supported with techniques to avoid confusion and disorientation. (NAVIGATIONAL FREEDOM, ORIENTATION).
- "A model of reinforcement and conditioning predicts that players will repeat behaviours that are rewarded and abandon behaviours that are ignored or punished" [8]. (REINFORCEMENT).
- "A good game…is one that children want to play again and again. There are several characteristics that good games share: the goal and rules are clear (SIMPLICITY, ORIENTATION); it's easy for players to keep track of their progress as they play (OUTCOMES); the game can be played with a variety of strategies; the game offers variety (for instance, because players can make different choices, or the game contains a random element such as a die) (VARIETY); and the game is so motivating that children are willing to persevere when facing challenges and to work to improve their strategies so they can become better players (MOTIVATION). Although many educational computer games available today offer attention-getting graphics, sound, and other special effects, these can become tiresome if the game itself is not well-structured and appropriately challenging" [9] (WELL-STRUCTURED, CHALLENGING). This paragraph provided with a good set of keywords (inserted in the quotation), but in our opinion a good educational game must not only fulfil the requirement "children want to play again and again", but also trigger significant learning (LEARNING EFFECTIVENESS).
- Finally we would like to add a feature to this list: in the case that a child reaches an impasse, a good game should provide immediate assistance (by means of hints/feedback) to avoid frustration (SUPPORT).

To sum up, the list (in alphabetical order) of relevant features (keywords) of a good educational game is: ACCESSIBILITY, ADAPTATION, CHALLENGING, IMMEDIACY, INTERACTIVITY, LEARNING EFFECTIVENESS, MOTIVATION, NAVIGATIONAL FREEDOM, ORIENTATION, PRIVACY, OUTCOMES, REINFORCEMENT, SIMPLICITY, SUPPORT, VARIETY, WELL-STRUCTURED.

3 Related work

Examples of educational systems designed for children and based on Piagetian stages of cognitive development are not easily found in research. An interesting work in this field is presented in [12], where Piaget's notion of cognitive development has been used in the building of pre-test that would allow improving a tutor's reasoning ability. MFD (mixed numbers, fractions and decimals) [13] is an Intelligent Tutoring System (ITS) aimed at teaching fractions, decimals and whole numbers to elementary school students. Students with different levels of cognitive development should behave differently in the tutor, and that is the reason why they need to be taught with different strategies. Before the students begin to use the tutor, they are given a computer-based pre-test that measure their level of cognitive development. The pre-test is composed by ten Piagetian tasks that determine if the students are at one of the last two stages of cognitive development (concrete operational stage and formal operational stage). The test includes exercises about number conservation, serialization, reciprocity; area conservation, class inclusion, functionality, reversibility, establishment of hypotheses, control of variables in experimental design, drawing of conclusions, proportionality; and combinatorial analysis. This measure predicts student performance at a variety of grain sizes: effectiveness of hints received, rate of failure, amount of time to solve problems, and number of problems students need to attempt to master a topic. Later on, same authors presented an independent, adaptive, and easy-to-integrate web-based component to evaluate a student's cognitive development that can be used as the pre-test of any ITS [14]. This component was constructed by including existing test's items [12] into the SIETTE [15] web-based adaptive testing system.

4 An example: the design and development of MITO

With these ideas in mind, in our group we have developed an educational game for Spanish orthography that has been developed for children in the concrete operational stage. For a definition of orthography, we have consulted some dictionaries:

- American Heritage: The art or study of correct spelling according to established usage
- *Real Academia de la Lengua*: Set of rules that determine the writing of a language

We can see that, in the case of Spanish, the word "orthography" has a different meaning. The most significant differences between English and Spanish writing systems are that in the case of spanish a) the correspondence between graphemes and phonemes is regular, with very few exceptions; and b) there is a (quite) reduced set of rules that completely determine how each word is spelled. So, though usually the best way to learn orthography is to read a lot (books, etc.) and then to use the visual memory to recall how a word must be spelled, in the case of Spanish rules are always a reliable way to know the correct spell for a word. So for example, in Spanish orthography there is a rule that states:

"before p or b m and not n should be written"

A simple rule like this determines that the correct spelling is "también" and not "tanbién", "tampoco" and not "tanpoco", and so on. In this way, learning orthography is simpler than in the case of English, as new speakers do not need rote learning of words.

In any case, either if rote learning is needed or if there is a set of rules (with their corresponding exceptions) that completely determines word spelling, orthography is an arid domain to be taught. Yet, the importance of learning orthography is clear because the ability to write correctly is fundamental in any professional environment.

However, our view of how orthography should be taught is not based in learning the rules, but shares the directions presented in [10], which we briefly explain next: to our purposes, orthography will be used as a synonymous of "correctly spelling the words". Therefore our main goal will be to teach the strategy that persons with good orthography apply. Solving exercises is not effective if the mental strategy is wrong, so the fundamental thing is to teach a mental process, which can be described as follows:

a) When in doubt, good writers mentally search for the image of the word and visualize it in their minds. Writing becomes then a copy of the word that has previously been stored in their minds.
b) This visual recall of the word can be good enough to be confident on its correctness, but in some cases doubts can be clarified by writing the word in the different ways.

Consequently, a goal when developing MITO is to help in using visual memory for orthography (by showing correctly and incorrectly written words).

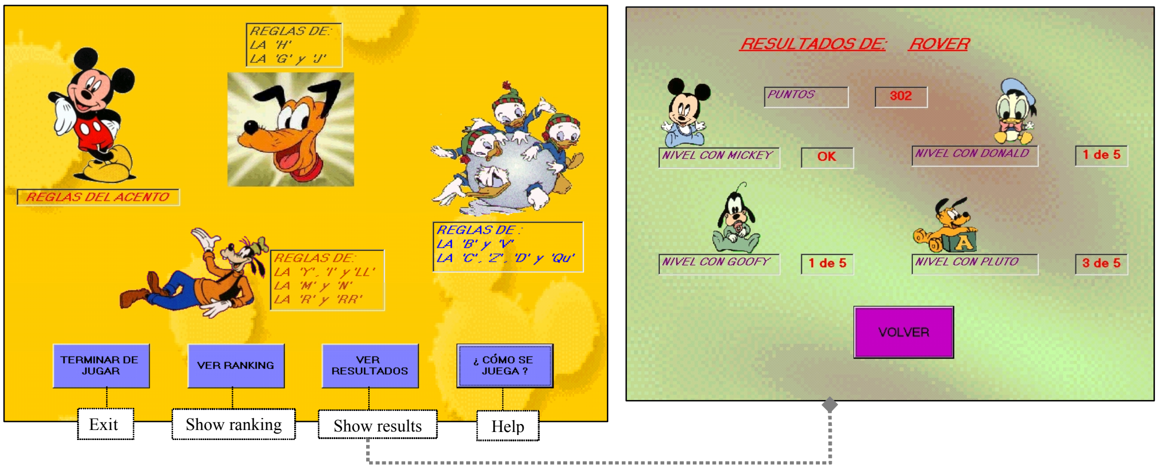
As also stated in [10], in order to learn Spanish orthography, the student must satisfy some conditions:

a) Be advanced in the concrete operational stage (about eight years),
b) Being able to write and read at a reasonable speed,
c) Being aware of the existence of rules and exceptions,
d) Being motivated to improve (though our system is designed to improve motivation).

Though MITO is a game designed for children in the concrete operational stage, it can also be used by adults. At this stage, children can understand rules and apply them to games, so the learning and use of orthography rules is an appropriate task for them.

The contents in MITO are divided in four modules that aim to teach words corresponding to different sets of rules: a) Written accent rules; b) H, G and J rules; c) B, V, C, Z, D and Q rules; and d) M, N, Y, LL, R and RR rules. As explained before, rules are used to provide support to being able to correctly spell the words but are not considered as learning goals (because for us it is more important to know the correct spelling of a word than the rule that applies). In this sense, rules are used in our system to group the exercises to be posed to the student. A Disney©[2] character is associated to each module.

For each learner, MITO creates and maintains a simple student model that keeps record of the number of correctly solved exercises of each module (represented by its associated character). To increase motivation and facilitate navigation, this simple student model is inspectable by pressing the button "*Ver resultados*" (show results). Figure 1 shows the initial screen and an example of such student model.



RESULTADOS DE: ROVER

PUNTOS 302
NIVEL CON MICKEY OK
NIVEL CON DONALD 1 de 5
NIVEL CON GOOFY 1 de 5
NIVEL CON PLUTO 3 de 5

Fig 1. Screenshots of the main menu and the inspectable student model

In order to achieve the learning goals fixed, MITO presents the following features (he associated keywords are shown in brackets):

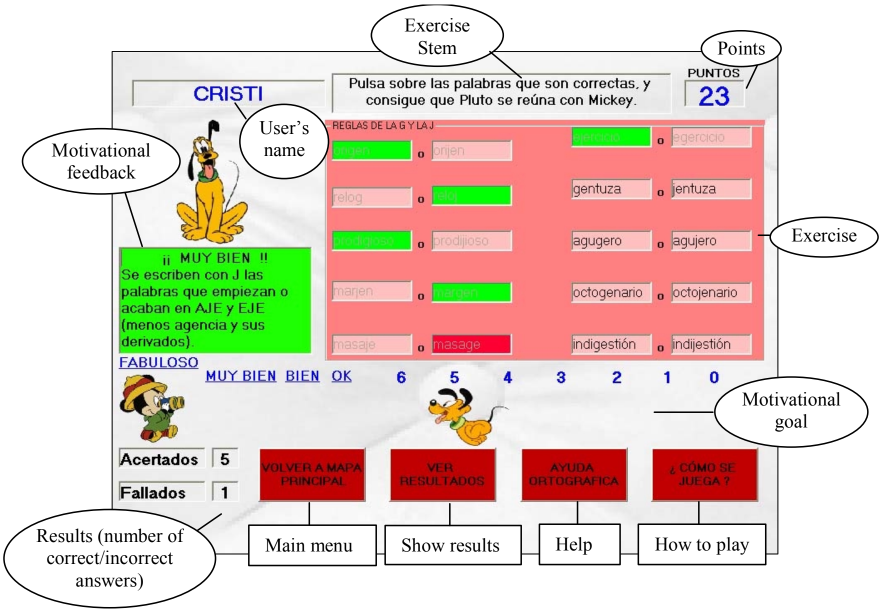
- The user can choose a different module at any moment (NAVIGATIONAL FREEDOM, INTERACTIVITY).
- Each exercise is explained briefly, but a more complete explanation can be shown by user's demand (SIMPLICITY, ACCESSIBILITY).
- The user's answers are checked immediately (IMMEDIACY).

---

[2] Disney characters were taken from http://clipart.disneysites.com and are used in our application with no commercial purposes.

- Each group is divided in several different exercises (VARIETY) that progressively get more complicated (i.e, apply more or more difficult rules) as the user correctly solves them (CHALLENGING). In order to trigger learning, the user must show knowledge about between 90% and 100% of a module's content (either by visual memory or by direct application of its rules) to be able to finish it (LEARNING EFFECTIVENESS).
- The system has a help mechanism (triggered by the user just pressing a "help" button) that explains the rules in an accurate and clear way. This help is context depending and can be shown by user's demand (SIMPLICITY, SUPPORT).
- Feedback messages are provided immediately (for both the cases of correct and incorrect answer) and are clear and brief (orthography rules are used as a basis for explanations) (SUPPORT, IMMEDIACY).
- There are many multimedia elements (childish sounds, nice pictures, etc.) designed to keep the user motivated. Also, the goal of the game is the goal is to help one Disney character reach another (see for example fig. 2 where each time the student provides the right answer, Pluto moves forward one position to reach Mickey) (MOTIVATION)
- As the user goes on with the game, his/her knowledge about the topic increases and the system adaptively selects more difficult exercises. (ADAPTATION, CHALLENGING, WELL-STRUCTURED)
- The system keeps a simple student model that is updated in every exercise. (OUTCOMES)
- At any moment, the user can see his/her student model. (OUTCOMES, ORIENTATION)
- Each time the user correctly solves an exercise, some points are granted (REINFORCEMENT).
- MITO keeps record of the score reached by its registered users, which is available at any time. The ranking of results of all registered players can be seen at any moment by pressing the button "*Ver ranking"*. In this way, competition triggers motivation and keep the children engaged. (MOTIVATION)

Figure 2 shows an exercise in MITO:



Pulsa sobre las palabras que son correctas, y consigue que Pluto se reúna con Mickey.

¡¡ MUY BIEN !!
Se escriben con J las palabras que empiezan o acaban en AJE y EJE (menos agencia y sus derivados).

Fig. 2. An exercise in MITO

The relevant elements are (from left to right and from up to down): user name, exercise stem, points reached, motivational feedback, window of the exercise (which in this case consists of selecting the correct word between the two possibilities), motivational goal (in this case, Pluto must reach Mickey and each correct answer will move him one position forward), number of correct and incorrect answers in this exercise and four buttons: main menu, show results, help, how to play (that provides an extended game description).When the student selects a word in the exercise, the background is changed to green if it is correct and to red if incorrect, and a childish voice congratulates or scolds the action while showing appropriate feedback (in this case, the corresponding rule).

MITO has a simple but effective architecture, which is shown in Figure 3. The interaction begins in the *Input Module*, where the system gets the user information. Then, the system gives access to the *Main Menu*, where he/she selects a module/game to play. The user can always access *Output Module* and leave the game (his/her student model will be saved for future sessions). The *Learner Diagnosis Module* controls the user model and provides this information when requested. The modules do not communicate with the expert module directly, but using the *Help Module* and the *Error Module*. The expert module contains the knowledge domain and structures the contents in a hierarchical way.

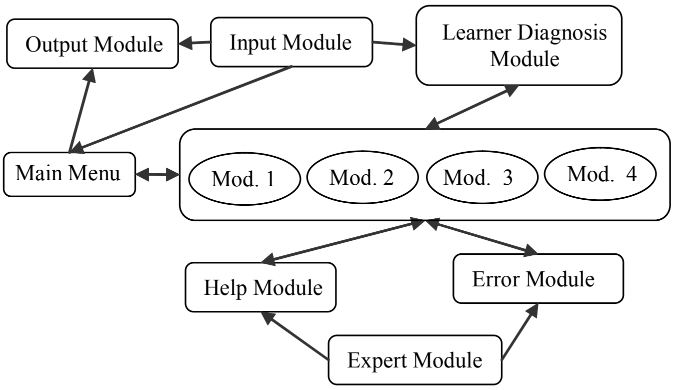


Fig. 3. MITO's architecture

5 Preliminary evaluation results

A formative evaluation of the MITO system has been performed. This evaluation had two main goals: a) to determine the degree of acceptance of the game among the users it was focused to, identifying relevant aspects that can improve learner's motivation and b) to study the effectiveness of the game in helping children to learn orthography, identifying possible ways to improve the design and behaviour of the system from an educational point of view. At this stage of our work, the evaluation was formative, so no numerical recollection and analysis of data has been performed. Instead, seven children (from 6 to nine years old and native Spanish speakers) were asked to play and a human tutor observed their behaviour and assisted them when needed (children had different computer literacy degrees, so some of them needed more support than others, though in general we found that the system was very easy to use and children only needed assistance at the very beginning).

Regarding the motivational aspects, first results have been very encouraging. Children did really become very engaged with the game and did not want to leave it until they completed all the modules. The features of the system that they liked better were the multimedia aspects, i.e., Disney characters and childish sounds (which fully fulfilled their goal to capture children's attention). The fact that they could help their favourite character to reach a goal was also very motivating, as also was the possibility to see their position in the ranking.

However, results in the use of feedback/help messages to trigger learning were not so good. More than half of the children did not want to pay any attention to feedback messages, even when pointed out that these messages could give a clue for the next exercise. Help was neither used by children, who only wanted to quickly get the correct answer from their tutor and keep playing. When reaching an impasse, most children asked their tutor for the correct answer, but very few of them tried to use help. In our opinion the reason is that, even when help messages are short and clear, children were too involved with the game and did not want to waste their time in reading. However, during the experiment we observed that children were willing to accept help from the human tutor instead of the correct answer, if it was presented as a "trick" that could provide the answer to not only the current exercise but probably to the subsequent ones. This gave us the idea that, probably, the use of a personal helping agent (as in [11]) could improve the effectiveness of the system from an educational point of view. The helping agent could be a fairy, an angel, etc. that would be assigned to the children at the beginning of the interaction and presented as a friend that can be called to provide help by means of "tricks" (simplified orthographic rules) when needed.

Though the system has been designed for children, we have also tested it with some adults (mostly illiteracy adults that were learning how to read and write). In this case, even when the images and sounds were very childish, people usually found them funny. The main difference is that help and feedback messages had the desired effect both capturing attention and triggering comprehension and learning. Ironically, the competitive aspect was the main motivating element in adult users.

6 Conclusions and future work

Our main goal while developing MITO was to build an application for children that could help them in the somewhat tedious but very important task of learning Spanish orthography. To this end, motivational aspects have had a great importance in the design and development of the system, and a first formative evaluation shows encouraging results regarding motivation. This evaluation also showed some weaknesses of the system, like the limited use of feedback and help messages. The reason possibly is that children get so involved in the game that they only want to go on playing to reach the maximum score, and therefore they do not want to lose their time in reading these messages. A possible solution that will be investigated and implemented is the introduction of a personal agent, which can provide help in a friendly way, presenting it like "tricks" that not only will help in the current exercise but will also serve to pass others. In the evaluation, the tutor played this role with satisfactory results, so probably the inclusion of such agents in the system will help in triggering significant learning.

Other improvements of the system we are planning include the development of an interface for tutors/parents that allows the inclusion of new words, to provide a greater variety of the exercises posed to the student. Finally, a formative evaluation of the whole system (the design of such evaluation has not been carried out yet) will be conducted in real settings to evaluate the effectiveness of the game in terms of the gain of significant learning.

Acknowledgements

Authors would like to express their gratitude for the many valuable and detailed comments and suggestions of improvements made by anonymous reviewers, which would serve as a very useful guide for our future work.

References

[1]. J.M. Randel, B.A. Morris, C.D. Wetzel, and B.V. Whitehill. 1992. *The Effectiveness of Games for Educational Purposes: A Review of Recent Research*. Simulation & Gaming 23. pp. 261-276.

[2]. M. Klawe. 1998. *When does the use of Computer Games and other Interactive Multimedia Software help students learn Mathematics?* NCTM Standards 2000 Technology Conference, Arlington, VA, USA. Available at http://mathforum.org/technology/papers/papers/klawe.html

[3]. C. Conati and J.F. Lehman, 1993. *EFH-Soar: Modeling Education in Highly Interactive Microworlds*. Advances in Artificial Intelligence, Proceedings of the third Congress of the Italian Association for Artificial Intelligence, AI-IA'93. pp. 47-58.

[4]. J. Piaget. 1953. *How children form mathematical concepts*. Scientific American, 189 (5), pp. 74-79

[5]. J. Piaget. 1970. *Piaget's Theory*. Carmichael's Manual of Child Psychology, third edition, volume 1. New York: Wiley.

[6]. Brown, T (2003). Age appropriate product features and characteristics. Ram Consulting. Available at http://www.tasc-toys.com/pdf/age_appropriate_product.pdf

[7]. Buchanan, K. The Heritage & Legacy of Thinking and Computer Games. Coursework for Intellectual History of Educational Psychology. At http://www.msu.edu/~buchan56/coursework/cep911/hl_thinking.htm

[8]. Buchanan, K. Theoretical Foundations for Teaching Using Computer Games. Coursework for Intellectual History of Educational Psychology. At http://www.msu.edu/~buchan56/coursework/cep911/synthesis.htm.

[9]. Kliman, M. 1999. Choosing Mathematical Software for Girls and Boys. Available at http://www.terc.edu/mathequity/gw/html/ChoosingSoftwarepaper.html

[10]. D. Gabarro y K. Puigamau. Aprender ortografía. http://zetauka.com/aprenderortografia/

[11]. C. Conati, X. Zhao, 2004. *Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game*. Intelligent User Interfaces 2004, pp. 6-13 .

[12]. Arroyo, J.E. Beck, Klaus Schultz and B.P. Woolf, 1999. Piagetian Psychology in Intelligent Tutoring Systems. Proceedings of AIED99. Le Mans, France. June 1999. pp. 600-602.

[13]. Beck, J.; Stern, M.; Woolf, B. (1997). Using the student model to control problem difficulty. The Sixth International Conference on User Modeling. pp. 277-288.

[14]. Arroyo, R. Conejo, E. Guzmán and B.P. Woolf, 2001. *An adaptive web-based component for cognitive ability estimation.* Proceedings of AIED2001. San Antonio, TX. May 2001. pp. 456-466.

[15]. R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-De-La-Cruz and A. Ríos, 2004. *SIETTE: A Web-Based Tool for Adaptive Testing*. International Journal of Artificial Intelligence in Education, 14, pp. 29-61.