Incorporating the Culture of Virtual Reality Games into Educational Software via an Authoring Tool

Maria Virvou, Constantinos Manos,

George Katsionis, Kalliopi Tourtoglou

Department of Informatics,

University of Piraeus,

Piraeus 18534, Greece

mvirvou@unipi.gr; kman@singular.gr;

gkatsion@singular.gr; ktourtog@internet.gr

***Abstract*-- Electronic games constitute one of the prevailing modes of interaction of children and adolescents with the computer. However, this popular culture has not been exploited sufficiently for educational purposes. This paper investigates the incorporation of the culture of virtual reality games into educational software. In particular, such incorporation is meant to be achieved through an authoring tool, which is called Ed-Game Author and provides facilities to human instructors for the creation of their own educational game applications. The authoring tool has the advantage of the generality and domain-independence while ensuring the active participation of human instructors in the development of the final educational application. Such participation guarantees the acceptance of the tutoring game by the human instructors who play a significant role in the educating process. On the other hand, the game environments generated by Ed-Game Author maximize the students' engagement and preoccupation with the educational application. Indeed, the system has been evaluated and the results show that Ed-Game Author has a lot of advantages that may be expanded further.**

Keywords: **Authoring tools, educational software, student modeling, virtual reality, computer games.**

I. INTRODUCTION

Over the past decades, electronic games have become an important part of young people's entertainment culture. Simple observations of current every-day life as well as formal studies show that electronic games have gained the affection of many children and adolescents who spend much of their leisure time and possibly even some of their supposed working time on them. For example, Griffiths and Hunt [1] conducted a study involving 387 adolescents and found among other things that approximately 30% of the adolescents play every day and the same proportion plays once a month. However, as pointed out by researchers (e.g. [2]) very few play electronic games at school, since these games are not welcomed in class.

Indeed, a lot of educators are alarmed by the possible addiction of adolescents and children to computer games. However, there are also quite a lot of educators and researchers who believe that the attractiveness of computer games should be exploited for the benefits of education. Papert [3] claims that computer games teach children that some forms of learning are fast-paced, immensely compelling and rewarding. Boyle [4] notes that games promote imaginative engagement and thus provide a powerful format for educational environments. In this sense computer-based education may profit from the popularity of computer games to achieve better learning effects among students.

However, many researchers [e.g. 5, 6] point out that technology is effective when developers thoughtfully consider the merit and limitations of a particular application while employing effective pedagogical practices to achieve a specific objective. This raises the issue of the design of the educational software application so that it may be educationally beneficial to students. This is also a major issue in the case of educational computer games. If educators are to include electronic games as part of the curriculum then there is a need to do much more than invite the popular electronic games culture of children inside the classroom walls [2].

In the case of educational software games there are three important parts of each application that need to be addressed. First, the design of the game environment has to be suitable for learning purposes. Second, the design of the educational content has to be suitable for the needs of students and their human instructors. Third, pedagogy strategies have to be incorporated in the educational game context.

The above issues constitute a complex problem that has to be addressed in the design of educational computer game software. However, if each game is designed to teach a specific domain and has been developed in a domain-dependent way then there will be few possibilities of reusability. At the same time, the construction of the application is probably going to take long if all issues involved are to be addressed.

As Murray [7] points out, inspired by goals of elegance, parsimony and/or cost-effectiveness, software designers are driven to write software that is general and reusable; in the context of educational applications, authoring tools are general and reusable. Authoring tools are meant to be used by instructors who wish to author their own educational applications on a certain domain. Therefore, the methods incorporated in the authoring tools have to be domain-independent.

Indeed a solution to the problem of reusability and cost effectiveness is the development of authoring tools that may be used for the creation of many computer game software applications. In this paper we describe an authoring tool that may be used by instructors to create virtual reality games for education. This tool is called Ed-Game Author (Educational Game Author). In addition to the objective of cost-effectiveness in the design of multiple applications, Ed-Game Author assigns an important role to the human instructor who has to author the application and customize several parameters, such as the exact way of grading the students. In this way, the authoring tool may ensure its acceptability from the human instructors who constitute an important part of the school teaching process.

II. Design Issues of ED-GAME Author

The tool offers multiple virtual reality game environments and the basic story of these games. It also incorporates a learner modelling mechanism that builds the individual profile of each player who is also a learner. Then, instructors may insert the material that they wish to teach to students. They may also insert domain facts, which will be used by the authoring tool for the automatic construction of questions that are going to be asked to students in the process of the game. The instructor has also the possibility of inserting frequent misconceptions of students relating to certain correct facts. In this sense, the instructor may also construct a "bug-list" which may be used by the system in the context of the game.

One important issue that has been addressed in the authoring tool is the design of the computer game environments. Characteristics such as music, lights, colours and noise play an important role in the attractiveness of a game [8, 9]. Moreover, the familiarity of children and adolescents with many computer games renders them quite demanding and thus imposes a high standard in the quality of the game environment and the plot of the story. Otherwise, these games run the risk of being considered as dull by the students.

In order to design game environments that would be acceptable to students we conducted an empirical study among school children and adolescents so that we could find out what their preferences were in computer games. As a result of this study, most students of the sample seemed to favour virtual reality games of the type of "DOOM" (ID-software 1993), which has many virtual theme worlds and castles with dragons that the player has to navigate through and achieve the goal of reaching the exit. The authoring tool may also generate many virtual worlds where the student has to navigate through. There are mediaeval castles in foreign lands, castles under the water, corridors and passages through the fire, temples hiding secrets, dungeons and dragons.

The interaction of the student with the resulting educational applications takes place through animated agents or through windows showing text to students. Questions to students are always asked by animated agents. Then, students may type their answer in a dialog box.

III. Creating the Educational Content

Human instructors who act as authors are responsible for inserting their own teaching material, which consists of lessons and tests accompanying these lessons. Instructors may also provide a list of frequent errors for each question or they may type explanations of errors in the multiple choice tests.

Tests may consist of questions of the following types:

1. Multiple choice questions
2. Fill-in the blank space
3. True/False questions
4. Questions where the student has to type in the answer

Each type of question is associated with certain facilities that Ed-Game Author may provide to instructors for the creation of a sophisticated educational application. In multiple choice and true/false questions the instructor has the ability to associate erroneous answers to particular causes and explanations of errors so that these may be used by the system to give more detailed and informative feedback to students. Moreover, these explanations are used to create each student's profile, which is recorded permanently and is updated after each interaction of the student with the educational application. For example, the same explanation of error may hold for more than one faulty answers of the student. In this case the long term student model counts the number of occurrences of the same type of explanation. Then it compares the numbers of occurrences of different explanations and finds the student's weaknesses and proneness to errors. These numbers are also used to find out whether the student has made any progress since the last time s/he interacted with the educational application or whether s/he has forgotten parts of the syllabus that s/he seemed to have known in previous interactions.

For example, in an educational application about geography, a student may have made 10 errors in questions concerning Greece and none in questions concerning other countries. In this case, the system will record the fact that the student has a serious lack of knowledge about the particular country and will compare this finding with the findings of previous interactions and future interactions to determine how the student is progressing.

In questions where the student has to type in the answer and fill-in the blank space questions, the student is allowed more freedom in the answer s/he may give. Error diagnosis in these categories of questions is more difficult than other categories of questions where the possible students' answers are more limited.

The explanation of a mistake may be difficult for the system to spot. Hollnagel [10, 11] makes an important distinction between the underlying cause or genotype of an error and the observable manifestation or phenotype of the error. In addition, ambiguity may be a problem, since there may be different explanations of observed incorrect users' actions [12]. For example, a student may give an erroneous answer due to a typing or spelling error and may appear that the

student does not know the answer in the domain being taught.

Ed-Game Author offers the facility of spotting spelling and typing errors. For example if the student types an answer, which contains an extra letter in comparison with the correct one then it has probably been a typing error. If the student types an erroneous answer that is pronounced in a similar way as the correct one then s/he has probably made a spelling error. If the student has typed a word, which is completely different from the correct one then s/he has made a domain error. For example, Fig. 1 illustrates the percentages of several causes of error in the answers of Student 1. These statistics may also be used for ambiguity resolution in case an error may be attributed to more than one categories of explanation. If a student is prone to typing errors then this cause may be favoured in cases of ambiguity.

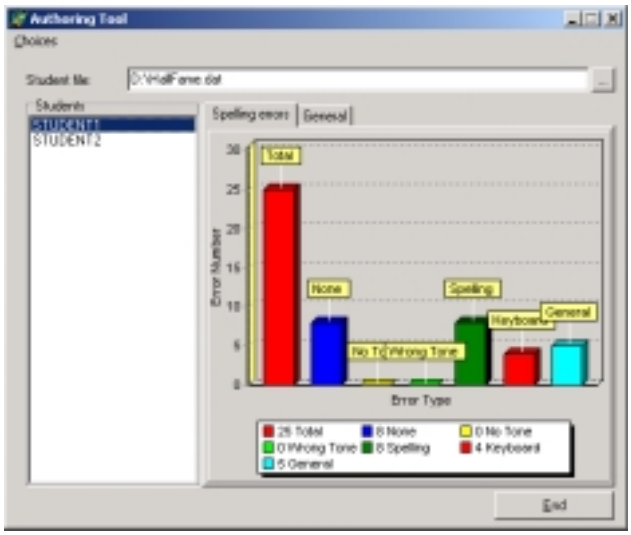

Fig. 1: Example of percentages of different types of students' errors

Domain errors may be examined further for the identification of a deeper cause of error. For example, the instructor may have provided a list of frequent errors and each of them may have been associated with an underlying cause of error. In this way the instructors may create a bug-list, which is based on their experience of students' making errors. Such lists may be used for further classification of domain errors and the student model is updated.

Thus each game created by Ed-Game Author may contribute both domain-dependent and domain-independent information about particular students to their long-term individual student models. For example, a student may be consistently making a lot of spelling mistakes when s/he is typing answers to questions posed by the game. This is a domain-independent feature of the student concerning the student's carefulness or carelessness when s/he types answers. This kind of feature is recorded in the student model and is updated constantly.

All questions that belong to a test are connected to a game map so that each question is going to turn up at a certain location of the virtual world and the student will have to answer it correctly to proceed further into the world. The instructor may decide whether the test will consist of questions of one type only or a combination of types.

The interaction of instructors with the authoring tool is performed through a user-friendly interface. An example of part of such interaction is illustrated in Fig. 2, where the instructor inserts multiple choice questions for tests in geography. At first the instructor has to type each question in the question list. For each question the instructor is also expected to type the choices that students will have for answering these questions. To insert the choices, which are attached to a question, the instructor has to select this question and then add the choices in the choice list next to the question list (Fig.2).

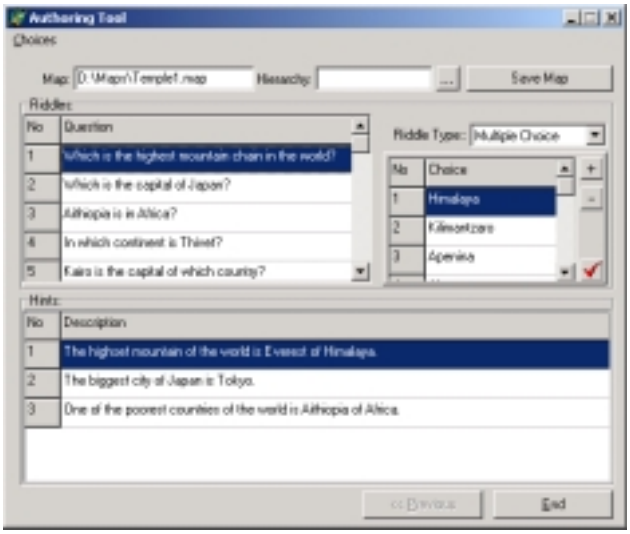

Fig. 2: Example of instructor interaction with Ed-Game Author

Instructors may optionally give hints concerning some questions of the test. If they decide they want hints to be given to their students concerning certain questions, then they have to select the questions from the list and give the description of the hints as can be seen at the bottom of the example screen in Fig. 2. Instructors are responsible for deciding which questions the hints will refer to and what these hints will be. These hints are going to be used in the educational game applications. In particular, as part of the adventure of the game the player may come across certain objects where s/he may click on. These objects appear at random and give hints to students. However, these hints are not immediately usable by the students since they refer to questions that the students will have to answer at a location of the virtual world other than the one they are currently at. Hence, the students will have to remember these hints so that they may use them when the time comes.

IV. Resulting Educational Game Application

The story of the educational games that result from Ed-Game Author incorporates a lot of elements from adventure games. However, each of these elements is connected to ideas and approaches from educational software technology.

Fig. 3: Virtual water world

The ultimate goal of a player is to navigate through a virtual world and find the book of wisdom, which is hidden. While the player is navigating through the virtual world, s/he finds keys, which are guarded by dragons. A guard dragon poses a question to the player from the domain of the particular educational application. If the player gives a correct answer then the dragon allows him/her to take the key. Each of these keys opens a door, which leads the player closer to the "book of wisdom".

Fig. 4: Virtual woods world

There are many virtual worlds that may be generated through Ed-Game Author. Instructors may have explicitly connected a test to a specific world. In any other case, Ed-Game Author selects the least frequently used world to connect it to a test. Examples of a virtual water world and a virtual world of woods are illustrated in Fig. 3 and Fig. 4 respectively.

In the resulting educational applications, the system communicates with the student via two types of animated agent, the virtual enemy and the virtual companion. The virtual enemy is usually a dragon who threatens the student by asking questions. In Fig.3 and Fig. 4 the dragons may be seen on the right of the screen. The virtual enemy is destroyed by the student if the student answers correctly.

The virtual companion appears in cases where the student has given an answer, which is close to the correct one but is not the correct one. In this case, the virtual companion tries to help the student give the correct answer. The existence of the virtual companion has been considered quite important by many researchers for the purpose of improving the educational benefit of tutoring systems. For example, Van Lehn and his colleagues [13] argue that students can improve their learning in collaboration with a simulated student; because the simulated student can be simultaneously an expert and a co-learner, it can scaffold and guide the human's learning in subtle ways.

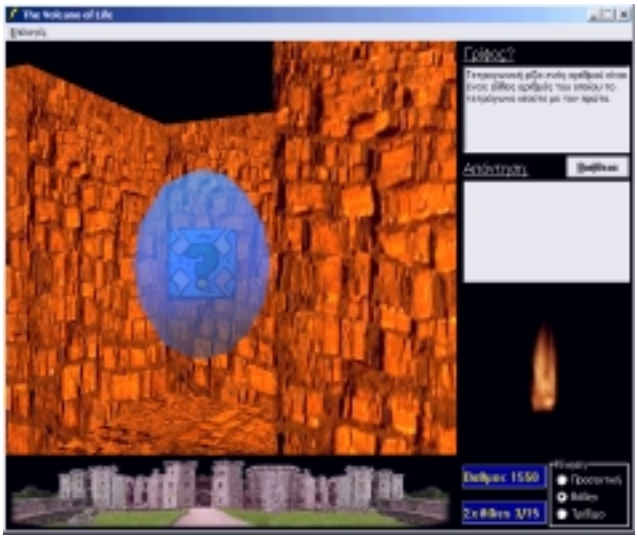

Fig. 5: A hint in the form of a blue ball in a virtual volcano world

Finally there are cases where certain objects appear at random to give hints to the player concerning questions that s/he will be asked in the future. At the time when the hint is given, the player does not know which question this hint refers to. Therefore s/he has to remember the hints for future use. An example of a hint in the form of a blue ball that has a question mark in the middle is illustrated in Fig. 5 in a virtual volcano world.

V. Evaluation

Educational applications may be considered successful if they are educationally beneficial to students. Therefore

evaluation of this kind of software is very important. In particular, formative evaluation is one of the most critical steps in the development of learning materials because it helps the designer improve the cost-effectiveness of the software and this increases the likelihood that the final product will achieve its stated goals [14].

The fact that educational software has many special features which differ from other applications has led many researchers to the creation of models dedicated to the evaluation of educational software [e.g. 15, 16, 17]. One such framework outlines three dimensions to evaluate: (i) context; (ii) interactions; and (iii) attitudes and outcomes [15]. The context determines the reason why the educational software is adopted in the first place, i.e. the underlying rationale for its development and use; different rationales require different evaluation approaches. Students' interactions with the software reveal information about the students' learning processes. The "outcomes" stage examines information from a variety of sources, such as pre and post-achievement tests, interviews and questionnaires with students and tutors. The focus of this framework is on students. However, in the case of an authoring tool, there is also one very important category of users, that of instructors acting as authors. Therefore an evaluation of an authoring tool has to involve instructors as well.

In view of the above, the evaluation of Ed-Game Author involved both instructors and students and was conducted in two different phases. At the first phase, the authoring procedure was evaluated by instructors. The second phase concerned the evaluation of the resulting educational applications and involved mainly students.

At the first phase, 6 instructors were involved. Half of them were school teachers in primary schools and were asked to prepare lessons and tests in geography using Ed-Game Author; the resulting educational applications were going to be used by students who were 8-9 years old. The other half of instructors were history high school teachers and their educational applications would be used by students who were 13-14 years old. All of the instructors who participated in the experiment were familiar with the use of computers. In addition, they had been trained for the use of Ed-Game Author before the experiment.

In general, instructors did not have many problems while authoring their educational applications. However, half of them did not make use of all the facilities that Ed-Game Author gave to them. For example, they did not attach explanations to errors or they did not any insert any bug list. The rest of the instructors made use of all the facilities of Ed-Game Author. All 6 instructors made use of the hint facilities.

When interviewed, the authors confirmed that Ed-Game Author had a user-friendly interface and stated that they were quite satisfied with the facilities that Ed-Game Author could provide for adding content. Among the instructors who did not make use of all the facilities of Ed-Game Author, two of them said that they did not consider it necessary to include these facilities in their educational applications and one of them said that he was not quite sure about how to make use of these facilities. The instructors who made use of these facilities were very pleased with the outcome. Finally, 5 of the instructors said that they had tried the resulting educational games and they had liked them a lot.

The educational applications that resulted from the first phase were used in the second evaluation phase. The second phase involved 10 students from the respective classes of the 6 instructors who participated in the first phase. The underlying rationale of the educational games lies on the hypothesis that these applications are more attractive and engaging and thus they may increase the students' motivation while retaining the educational quality. At a first glance, the validity of this hypothesis might look obvious. However, there may be students who are not familiar with virtual reality games and thus might not like the particular applications. On the other hand, there may be students, who play games very often and thus may have very high demands from computer games. Hence, one important aspect of the evaluation is to find out whether students were indeed pleased with the game environment. Another very important aspect was to find out whether students had gained educational benefits from the games.

Students were asked to use the games as part of their duties in class. Their instructors were present during the experiment. Moreover there were computer assistants who could help students with their interaction with the game, in case they needed help. After the interaction with the game the students were interviewed.

There were two categories of student in terms of their familiarity with computer games, the experienced computer game players and the inexperienced ones. In general, the experienced users found the game very interesting. However, they also pointed out that they would like it to be more adventurous. Some of them were very fascinated with the idea of the game being incorporated into their duties.

The inexperienced players did not have many problems interacting with the game. Some problems they did have could be easily addressed in a subsequent version of the game. For example, some students had problems with disorientation in the navigation through the worlds or with their movement around the world (e.g. they might have got stuck into some corridors). The problem of the disorientation could be addressed by the addition of some maps where the student could see where s/he was at any time. The problem of the movement through the corridors could be addressed by expanding the width of the corridors and by allowing students to move around at a lower speed if they wished so. However, in general the inexperienced users were also very pleased with the idea of the game as part of their duties.

All students were asked questions from the tests they had taken while they played the game and they seemed to remember most of the issues they had learned. Moreover they all seemed to remember the hints they were given while playing the game.

VI. CONCLUSIONS

This paper has shown how virtual reality games may be incorporated into educational software by providing an authoring tool that can turn ordinary tests into educational games. The authoring tool is called Ed-Game Author and is addressed to instructors who wish to author their own educational game applications. The resulting games offer a variety of virtual reality worlds that the student has to navigate through to win the prize of the game.

The authoring tool has been evaluated by instructors and students. The results of the evaluation were quite encouraging. In general, instructors have found the authoring tool quite easy to use and helpful. Students have found the resulting educational applications more interesting and appealing than other forms of educational means but they noted that they would like the games to have a yet richer virtual reality environment. The learning effects of the application were quite high and there was almost no percentage of drop out from the educational software application.

REFERENCES

[1] Griffiths, M. D. and Hunt, N. (1995) Computer game playing in adlescence: Prevalence and demographic indicators. Journal of Community and Applied Social Psychology, 5, pp. 189-193.

[2] Inkpen, K., Upitis, R., Klawe, M., Lawry, J., Anderson, A., Mutindi, N., Sedighian, K., Leroux, S. & Hsu, D. (1994). "We Have Never-Forgetful Flowers In Our Garden: Girl's Responses to Electronic Games". Journal of Computers in Math and Science Teaching, 13(4), 383-403.

[3] Papert, S. (1993). "The Children's Machine: Rethinking School in the Age of the Computers". Basic Books, New York, 1993.

[4] Boyle T. (1997). Design for Multimedia Learning. Prentice Hall, 1997.

[5] Salomon G. (1990) Studying the flute and the orchestra: Controlled vs. classroom research on computers. International Journal of Educational Research, 14, pp. 521-532.

[6] Welch M. and Brownell K. (2000) The development and evaluation of a multimedia course on educational collaboration. Journal of Educational Multimedia and Hypermedia 9(3), pp. 169-194.

[7] Murray, T. (1999). Authoring intelligent tutoring systems: an analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10, pp. 98-129.

[8] Griffiths, M.D, (1995). "Technological addictions". Clinical Psychology Forum, 76, 14-19.

[9] Wolfson, S. and Case, G. (2000). The effects of sound and colour on responses to a computer game". Interacting with Computers, 13, pp. 183-192.

[10] Hollnagel, E. (1991). The Phenotype of Erroneous Actions: Implications for HCI Design. Human-ComputerInteraction and Complex Systems, G. R. S. Weir and J.L. Alty (eds.), London Academic Press Ltd.

[11] Hollnagel, E. (1993). The Phenotype of Erroneous Actions. International Journal of Man-Machine Studies, 39, 1-32.

[12] Mitrovic, A., Djordjevic-Kajan, S., Stoimenov, L. (1996). INSTRUCT: Modeling students by asking questions. User Modeling and User Adapted Interaction, 6 (4), 273-302

[13] VanLehn K., Ohlsson S. and Nason R. (1994) Applications of Simulated Students: An Exploration, Journal of Artificial Intelligence in Education 8, pp. 262-283.

[14] Chou, C. (1999). Developing CLUE: A Formative Evaluation System for Computer Network Learning Courseware, Journal of Interactive Learning Research, 10(2).

[15] Jones, A., Scanlon, E., Tosunoglu, C., Morris, E., Ross, S., Butcher, P. & Greenberg, J. (1999). Contexts for Evaluating Educational Software. *Interacting with Computers*, 11(5), 499-516.

[16] Squires, D. & Preece, J. (1996). Usability and Learning: Evaluating the Potential of Educational Software, Computers and Education, 27(1), 15-22.

[17] Squires, D. & Preece, J. (1999). Predicting Quality in Educational Software: Evaluating for learning, usability and the synergy between them, Interacting with Computers, 11(5), 467-483.