A virtual reality user interface for learning in 3D environments

George Katsionis
Department of informatics
University of Piraeus
80 Karaoli & Dimitriou Str. Piraeus
18534, Greece
+3-210-5813659
gkatsion@kman.gr

Maria Virvou
Department of informatics
University of Piraeus
80 Karaoli & Dimitriou Str. Piraeus
18534, Greece
+3-210-4142269
mvirvou@unipi.gr

ABSTRACT

Computer assisted learning is a common issue among the research communities globally. The benefits of such technologies are widely accepted, so more and more educational applications are being developed. However, educational software aims to be used by students for learning. Therefore educational applications have to be as attractive as possible to increase the engagement of students. To serve this purpose many researchers employ several aspects of multimedia technology to improve the aesthetics and the appeal of educational software. One very attractive and popular application area, that involves and demands advanced multimedia technologies and could be used for educational purposes, is computer games.

In this paper a virtual reality game for educational purposes is described. This tutoring system incorporates a virtual reality user interface similar to that of common commercial games. The game is enriched with student modeling mechanisms that ensure the individualisation of the interaction. In order to create the virtual environment of the game, Virtual Reality Modeling Language (VRML) was used. Implementation issues of the virtual reality environment, and associated matters are being discussed.

Categories and Subject Descriptors

K.3.1 [**Computers and Education**]: Computer Uses in Education – *Computer-assisted instruction (CAI), Computer-managed instruction (CMI), Distance learning.*

General Terms

Design, Experimentation, Languages.

Keywords

Educational software, Virtual reality, Student modeling, Computer games, Intelligent Tutoring Systems.

1. INTRODUCTION

Children and adolescents are often fascinated by electronic games. Indeed, it has been widely acknowledged that electronic games are part of the popular culture of many children [18]. For example, Papert [16] acknowledges the fact that the greatest amount of children's time with the computer is devoted to playing games and Griffiths and Hunt [10] who conducted a study among adolescents, found that approximately one third of the subjects of their sample played computer games every day and the same amount played once a month.

Although children's fixation with these games initially alarmed parents and educators, educational researchers soon questioned whether the motivation to play could be tapped and harnessed for educational purposes [15]. Hence a lot of researchers have recently highlighted the advantages of computer games relating to education. For example, Boyle [5] notes that games can produce engagement and delight in learning. In addition there are many researchers who encourage this entertaining aspect of education and have developed games for educational purposes ([3], [14], [19]).

Additionally, the scientific community in general has acknowledged the need for a high degree of adaptivity and dynamic individualisation to each student that educational applications of any kind should provide. To this end, Intelligent Tutoring Systems (ITSs) have made significant contributions. Indeed, as Self [20] points out, ITSs are computer-based learning systems, which attempt to adapt to the needs of learners and are therefore the only such systems, which "care" about learners in that sense. It is simple logic that response individualised to a particular student must be based on some information about that student; in ITSs this realisation led to student modelling, which became a core or even defining issue for the field [8].

For the purposes of this kind of research, in this paper, we present an Intelligent Tutoring System (ITS) that operates as a virtual reality game. This game is an educational application for teaching English orthography and grammatical rules, inside three-dimensional virtual worlds. The environment of the game aims at increasing students' motivation and engagement and is quite similar to that of common commercial games.

2. VIRTUAL REALITY GAME

The environment of a game plays a very important role for its popularity. Griffiths [11] after conducting a questionnaire and interview study, found that the machine's "aura" typified by characteristics such as music, lights, colors and noise was perceived as one of the machine's most exciting features for a large part of the population questioned.

Our educational application invites the culture of computer games for creating a language Intelligent Tutoring System that can be very engaging, motivating. In the case of language tutoring systems the use of computer games may additionally provide a cultural internationalisation and wide acceptance of these systems.

The environment of the virtual game is similar to that of most popular virtual reality games, which has many virtual theme worlds with castles, corridors and dragons that the player has to navigate through and achieve the goal of reaching the exit. The main similarity of this tutoring system with computer games lies in their use of a 3D-engine. However, this game unlike commercial computer games of this kind is not violent at all and is connected to an educational application. One must fight one's way through by using one's knowledge. However, to achieve this, the player has to obtain a good score, which is accumulated while the player navigates through the virtual world and answers questions concerning English spelling. These virtual worlds look like the one in Figures 1,2.

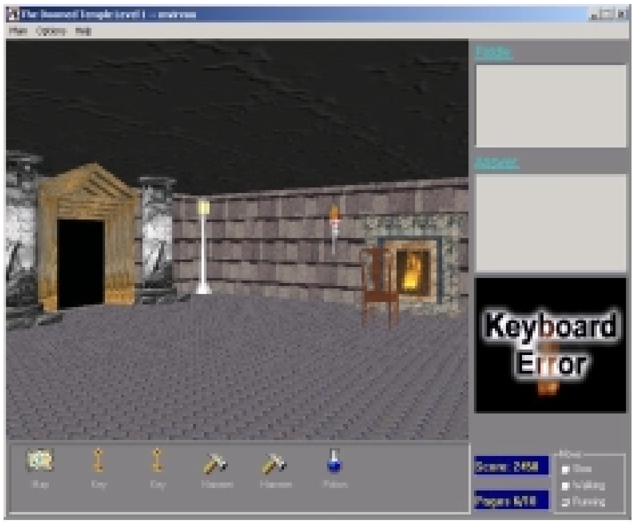


Figure 1. Virtual worlds of learning

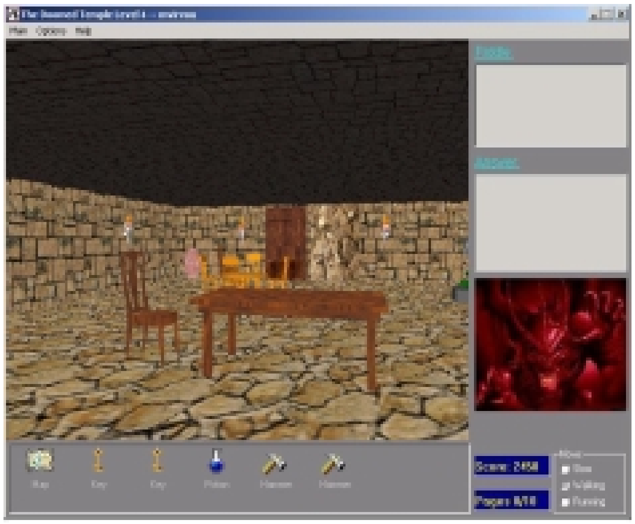


Figure 2. Virtual worlds of learning

As it can be seen in figures 1,2 part of the adventure of the game is the variety of inventory objects that the student can accumulate during the game and make the game even more attractive. There is map that every player has access to, and is an essential part of the game. The map shows an overview of the structure of the world, and is very useful for the navigation of the players in the world. The keys can be used to open doors that the students do not know the answer to the questions asked. Additionally there are other objects with useful functions including potions, hammers, teleports and books.

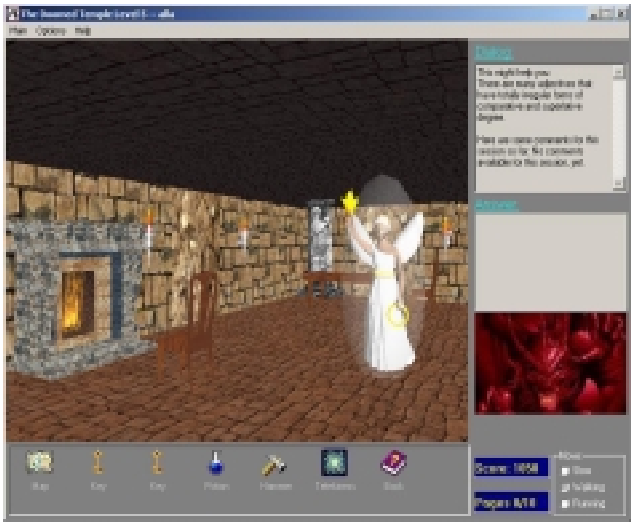
As we said the student-players of the virtual game have to answer certain questions in order to pass through the doors of the virtual worlds. When a student is asked a question s/he may type the answer in a dialog box. The ITS takes into account the history of answers of students and constructs a student-model for each one of them. The student characteristics that are being modelled concern the knowledge level of students (answers' results - errors) as well as some general actions. Additionally it provides individualised assistance, guidance, error diagnosis and navigation support based on these student models.

The game aims at helping students learn simple grammatical rules and check their spelling. For example, one set of grammatical rules concerns the plural form of nouns. Another one concerns the comparative and superlative form of adjectives consisting of one or two syllables. For both of the above sets, there are grammatical rules that categorise the nouns and adjectives into different categories. Thus, in the domain representation of the game, there are different categories of rules accordingly. While students interact with the game, it examines their answers in questions that relate to grammatical rules and categorises the errors that they may make. In this way, it may provide help according to the category of error.

The educational application communicates its messages to students through animated agents or through windows that display text. The user interface employs three types of animated agent: The dragon that is the virtual enemy of the player and is responsible for asking questions outside every door, the angel, which is the virtual advisor of the player and is responsible for providing hints and advice, and the virtual companion of the player who has the form of an elf.

The virtual advisor agent appears in some special situations. One of them is to inform the student to read new parts of the theory in order to continue playing the game, and be able to answer the questions that will follow. Another is to advice the student to repeat parts of the theory that s/he appears not to know well. Such information is obtained by making inferences about the mistakes of the student that are being tracked. In figure 3 below is illustrated an example of the advisor agent interacting with the student.

The virtual companion, who has the form of an elf, appears in cases where the student has given an answer, either correct or wrong, and the student has declined much from his usual actions (made a mistake in a category that s/he was always correct) or has made a repeated mistake. Then the virtual companion appears, asks the student about this situation and makes some notes. It simulates the behaviour of a friend for the student. However, it is essential not to become disturbing for the student by interrupting him/her continuously. 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 and promoting the student's sense of collaboration.



This might help you:
There are many adjectives that have totally irregular forms of comparative and superlative degree.

Here are some comments for this session so far. No comments available for this session, yet.

Score: 1058

Pages: 8/18

Figure 3. Virtual advisor agent

3. VIRTUAL REALITY ENVIRONMENT IMPLEMENTATION

Due to the advantages of computer games relating to education our main goal was to make an educational application that would have a user interface quite similar to the environment of common commercial games. The game category that we selected for our educational application was virtual reality games.

Virtual reality describes an environment that is simulated by a computer. Most virtual reality environments, including virtual reality games, are primarily visual experiences, displayed on a computer screen, and some include additional sensory information, such as sound through speakers. Because of using such advanced multimedia technologies, virtual reality gaming applications create rich and engaging environments. Such applications, which typically are very attractive and provoke a wealth of emotions to users, can become an advanced test bed for educational purposes.

There was no 3D commercial engine that we used to create our virtual application. We have created the 3D environment of the game from scratch. To create the virtual environment for our educational application we have used the Virtual Reality Modeling Language (VRML) [2]. VRML was originally designed to allow 3D worlds to be delivered over the World Wide Web. Using a VRML browser the user can explore this world, zooming in and out, moving around and interacting with the virtual environment. This allows fairly complex 3D graphics to be transmitted across networks without the very high bandwidth that would be necessary if the files were transmitted as standard graphic files. VRML can also include multimedia elements, such as texture images, video and sounds. The file extension for VRML files is ".wrl".

VRML was used to create the virtual worlds of the user interface of our educational application. However, due to luck of the VRML's language capabilities for creating a window application, all the rest parts of the functionality, the utilities, and the information kept for the students, were constructed with a visual programming language.

So the educational game's windows, forms, buttons, menus, etc, have all been created using DELPHI [12]. We then used a highly interactive 3D viewer to integrate our virtual worlds, which were created with VRML, into our DELPHI application. This 3D viewer is called Cortona VRML client [13] and it provides visual programming languages, DELPHI included, with an active-x component named Cortona in which virtual worlds created in VRML can be viewed by loading the according files. It should be mentioned here that the window application has the capability to use VRML as to make changes in the virtual worlds, create new .wrl files, and load them again. This implementation architecture is illustrated in figure 4 below.

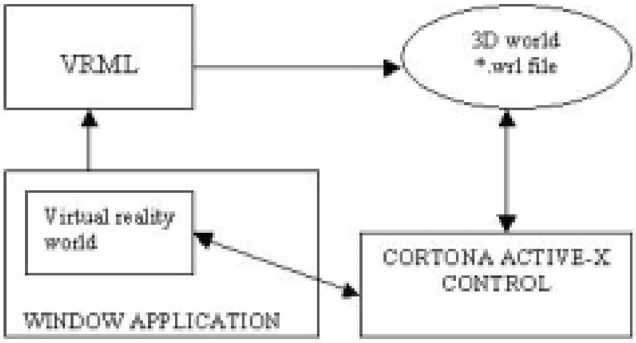


Figure 4. Virtual game's implementation architecture

The combination of a common and easy to use visual programming language to create the user interface of the educational application, and VRML for creating the virtual worlds to be embedded in it, provided the result presented in the above first three figures.

4. WEB3D PROBLEMS AND POTENTIAL SOLUTIONS

Educational applications express the need for a high degree of adaptivity and dynamic individualisation. Student modeling can cover this need. The shift from standalone to networked PC computing offers the capability of modelling a large population of individuals. Information about the learner is no longer stored locally on each learner's computer but in a central repository that can be accessed by any client of the application that requests it. This is usually implemented by using client-server architectures. A step forward for these architectures is educational applications that operate trough the Internet. Some researchers have used VRML language for creating simple educational applications that work through the Internet ([6], [22]).

At the beginning of our research our aim was to create a virtual game for educational purposes. An educational system that was a standalone, local application that worked as a virtual game was created. However, in order to exploit the benefits of computer assisted learning in multi-user environments and distance learning, we wanted to make that educational standalone-local application accessible to all students through the Internet. That meant that there should be a client-server program that would allow the students to interact with the application. Each student would play through the Internet, on the client installed on his or her system, and all of the information results would be stored in the server, located at the teacher's computer. In this way the teacher would always have access to all the information about the students' answers and their user-models.

In the past years there were some approaches to make client-server applications work through the Internet. For example, a method used for the deployment of an ITS over the web [1] was to take all the parts of the client program and provide them through the Internet. In that way, the client-server program worked by having the client as web pages on the Internet and the server located at the Web server where the web pages existed. The server was receiving information from the web pages.

A VRML world is made up of lots of simple shapes, such as cones and spheres, grouped together to form objects. The more shapes in the file, the more detailed the world, but at a cost of increasing the file size and the time taken by the browser to display the world. In our application there exist complex graphical objects that constitute of hundreds of simple shapes. Thus, it is made clear that a virtual reality game like our educational application, which has numerous objects inside its virtual worlds that are highly detailed, would take too much time to load in a web browser.

As a result, to take all the parts of the client program, including the VRML virtual worlds, and provide them through the Internet was out of the question. So what was best was to have the client installed at each student's computer, and by some way to communicate with the server at the teacher's computer through the Internet. For this goal we used the new trend in web communications, the Web services.

Web services can be defined as modular programs, generally independent and self-describing, that can be discovered and invoked across the Internet or an institution intranet [17]. Web services are interfaces that describe a collection of operations that are network-accessible through standardised XML messaging. Additionally Web services perform a specific task or a set of tasks. A Web service is described using a standard, formal XML notation, called its *service descriptio*n, which provides all of the details necessary to interact with the service, including message formats (that detail the operations), transport protocols, and location [9].

From a pure technology standpoint, Web services represent a shift toward the broad-based adoption of standard interfaces, like SOAP (Simple Object Access Protocol) [4], UDDI (Universal Description, Discovery and Integration) [21], and WSDL (Web Service Description Language) [7], which are used to access application functionality over the Internet.

More precisely, Web services are basically designed to allow loose coupling between client and server, and they do not require clients to use a specific platform or language. In other words Web services are language neutral. Mainly for these reasons among others, these services are becoming very popular. One important application area that may benefit from the advantages of this new technology is the field of education, where there may exist very demanding applications such as Intelligent Tutoring Systems (ITSs) that have to be transferred over the Web.

What is needed is to have some services and their interfaces mentioned on a Web server, for example the Internet Information Server (IIS) of the main computer of our computer lab. That makes these services known for every client program that can connect to the IIS of the main computer through the Internet. The services are implemented on the main computer in a server program that can be another executable application (EXE) or even a DLL (Dynamic Link Library). So a client calls a Web service form the IIS (Web server) by knowing its interface, and the IIS automatically sends the message for that service to the server program. The server program also aware of the service's interface executes it.

In a similar way we have developed our virtual game and installed it in different computers, which calls Web services from our IIS server of our main computer for performing specific tasks. The interface language exploited to publish services and exchange messages is WSDL. DELPHI provides special components responsible to make this kind of connections by using interface languages like WSDL and SOAP. Our client (virtual game) is aware of the services that are implemented on the server program. So the VR-Game calls a Web service form the IIS, and the IIS automatically sends the message for that service to the server program (DLL) that executes it. One such task is the addition of a student's answer to the database, which is located at the main computer. Along with his answer there are stored its characteristics (was it correct, what kind of mistake). Other tasks are any questions regarding the information that exists in the database that keeps the students model. For example the companion agent frequently asks about the category of knowledge being taught, or the specific rule of this category, that the student makes most mistakes, as to provide help for the student. The resulting architecture of the system is illustrated in figure 5.

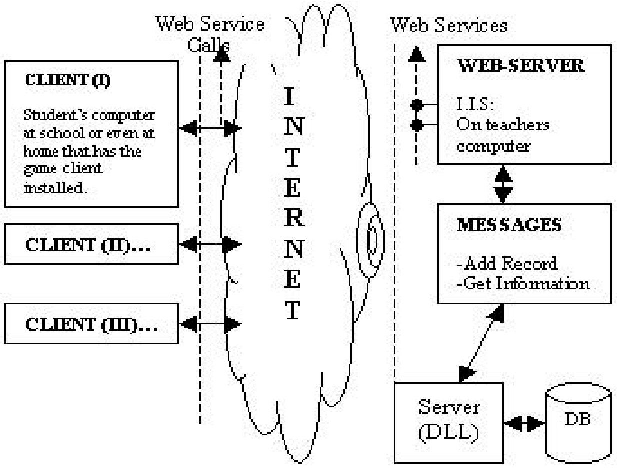


Figure 5**: The architecture.**

An example of the operation of the architecture of the system is the following. Every client, located in each student's computer can add an answer of that student to the database located at the main computer. So the client calls a Web service using an interface, through the Internet, from the Web server (Internet

Information Server in our example (IIS)) located at the main computer. Every call provides the Web service with a declaration and some variables, which contain the information about the student's answer. The Web server (IIS) then sends the message to the server program (.DLL in our example) that implements this interface. Then the server program executes the service and stores the answer to the database. In this way, from a user's point of view the game works in the same way as the standalone version.

Until now the results we are getting from our experiments show that the transfer to a Web-based ITS will help us a lot. We have not converted the whole program to work by using Web-services. It is certain that this is going to take us some time. Most of its' operations are executed directly from the client. We have implemented though some parts of it that are general for all the students and should be located at the teacher's computer. For example the database that keeps the user models and it is updated instantly from every client. The main usability difference that we have gained is that through Web services we are managing to make this demanding application to work through the Internet.

5. CONCLUSIONS

Educational applications may benefit from the technology of virtual reality games, which can increase the students' engagement and motivation. Individualisation of such tutoring systems is achieved by monitoring users while they play and by creating and maintaining detailed student models. These models are used for adapting instruction and advice accordingly.

However, one major problem of this kind of educational application is the construction and use of the game itself. The virtual reality user interface of such an application is extremely demanding. In this paper we have examined an implementation technique of a virtual reality educational game, and have discussed the difficulties faced to make it work through the web.

6. REFERENCES

[1] Alpert S. R., Singley M.K. & Fairweather P.G. (1999) "Deploying Intelligent Tutors on the Web: An Architecture and an Example", *International Journal of Artificial Intelligence in Education*, 10, 183-197.

[2] Ames A., Nadeau D., Moreland J. "VRML 2.0 Sourcebook", *2nd Edition*.

[3] Amory, A., Naicker, K., Vincent, J. & Claudia, A. (1998). Computer Games as a Learning Resource. *Proceedings of ED-MEDIA, ED-TELECOM 98, World Conference on Education Multimedia and Educational Telecommunications,* Vol. 1, pp. 50-55.

[4] Box D., D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, D. Winer, "Simple Object Access Protocol (SOAP) 1.1", W3C Note, May 2000, http://www.w3.org/TR/SOAP.

[5] Boyle T. "Design for Multimedia Learning". *Prentice Hall*, 1997.

[6] Chover M., Ó. Belmonte, I. Remolar, R. Quirós, J. Ribelles. "Web-based Virtual Environments for Teaching", Proceedings of the Eurographics/ACM SIGGRAPH *Workshop on Computer Graphics Education*, 2002.

[7] Christensen E., F. Curbera, G. Meredith, S. Weerawarana (eds.) "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, "http://www.w3.org/2001/NOTE-wsdl-20010315".

[8] Cumming G. & McDougall A.: "Mainstreaming AIED into Education?", *International Journal of Artificial Intelligence in Education*, Vol. 11, (2000), 197-207.

[9] Gottschalk K., Graham S., Kreger H. & Snell J. "Introduction to Web services architecture". *IBM SYSTEMS JOURNAL*, VOL 41, NO 2, 2002.

[10] 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.

[11] Griffiths, M.D. "Technological addictions". *Clinical Psychology Forum*, 76, 1995, pp. 14-19.

[12] http://www.borland.com/

[13] http://www.parallelgraphics.com

[14] 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), pp. 383-403.

[15] Mc Grenere, J.L. "Design: Educational Electronic Multi-Player Games. A Literature Review". *Technical Report* 96-12, the University of British Columbia, Canada, 1996.

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

[17] Pires P.F., Benevides M. & Mattoso M. (2002) "Building Reliable Web Services Compositions", Net.Object Days - WS-RSD'02, 551-562.

[18] Provenzo, E.F "Video Kids: Making sense of Nintendo". *MA: Harvard University Press*, Cambridge, 1991.

[19] Sedighian, K. (1997). "Challenge-Driven Learning: A Model for Children's Multimedia Mathematics Learning Environments". *Conference of Educational Multimedia & Hypermedia & Educational Telecommunications,* 1997, Calgary, Canada.

[20] Self J. "The Defining Characteristics of Intelligent Tutoring Systems Research: ITSs Care, Precisely", *International Journal of Artificial Intelligence in Education* ,10 , 1999, pp. 350-364.

[21] Universal Description, Discovery and Integration (UDDI) Version 2.0 Specification, June 2001, "http:/uddi.org/specification.html".

[22] Warrick, P.A. Funnell, W.R.J.:" A VRML-based anatomical visualisation tool for medical education", *Information Technology in Biomedicine*, Vol. 2, (1998), 55-61.