ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ενσωμάτωση γνωστικών θεωριών αιτιολόγησης και απομνημόνευσης σε λογισμικό ευφυών εκπαιδευτικών παιχνιδιών εικονικής πραγματικότητας

Incorporating cognitive theories on reasoning and memorization in intelligent virtual reality educational games

Διδακτορική Διατριβή

Κωνσταντίνος Σ. Μανός

Πειραιάς Ιούλιος 2011

Dedicated to my wife,

PHD THESIS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τριμελής Συμβουλευτική Επιτροπή

Επιβλέπουσα:
Μαρία Βίρβου
Καθηγήτρια
Πανεπιστημίου Πειραιώς

Μέλη:
Γεώργιος Τσιχριντζής
Καθηγητής
Πανεπιστημίου Πειραιώς

Αλεξανδρής Νικόλαος
Καθηγητής
Πανεπιστημίου Πειραιώς

Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Διατριβή
Για την απόκτηση Διδακτορικού
Διπλώματος του Τμήματος Πληροφορικής

Κωνσταντίνος Σ. Μανός

«Ενσωμάτωση γνωστικών θεωριών αιτιολόγησης και απομνημόνευσης σε λογισμικό ευφυών εκπαιδευτικών παιχνιδιών εικονικής πραγματικότητας»

Επταμελής Εξεταστική Επιτροπή

Μέλη:
Αχιλλέας Καμέας
Επίκουρος Καθηγητής Ελληνικό Ανοικτό Πανεπιστήμιο

Αλεξανδρής Νικόλαος
Καθηγητής Πανεπιστημίου Πειραιώς

Βίρβου Μαρία
Καθηγήτρια Πανεπιστημίου Πειραιώς

Τσιχριντζής Γεώργιος
Καθηγητής Πανεπιστημίου Πειραιώς

Μαρία Μουντρίδου
Επίκουρος Καθηγήτρια ΑΣΠΑΙΤΕ

Αθανασία Αλωνιστιώτη
Λέκτορας ΕΚΠΑ

Αικατερίνη Καμπάση
Καθηγήτρια Εφαρμογών ΤΕΙ Ιονίων Νήσων

Table of Contents

1. Introduction 7
1.1 VR-ENGAGE 7
1.2 VR-INTEGATE 8
1.3 Student Modeling 9
1.4 Memory Retention Capabilities 10
1.5 Using Virtual Agents to enhance the teaching experience 11
1.6 Initializing Cognitive Model and evaluation 12
2. Field Overview 14
2.1 Computer Games 14
2.1.1 Adventure Games 16
2.1.2 Action / Arcade games 16
2.1.3 Role Playing Games (RPG) 18
2.1.4 Simulation Games 19
2.1.5 Strategy Games 20
2.2 Educational Software 20
2.2.1 Target: Knowledge 21
2.2.2 Target: Physical / Reflexive Skills 21
2.2.3 Authoring Tools 22
2.3 Virtual Reality (VR) 23
2.3.1 Audio and Visual Enhancements 23
2.3.2 Virtual Environments 25
2.3.3 Artificial Intelligence & Dynamic Environments 26
2.4 Cognitive Psychology – Memory Retention 26
2.4.1 Ebbinghaus' Mathematical Model 27
2.5 Student Modeling 27
2.5.1 General Techniques 28
2.5.2 The "Overlay" technique 29
2.6 Conclusions 29
3. VR-ENGAGE: A Virtual Reality – Educational Negotiation Game. 31
3.1 The Game's Environment and Game Play 31
3.2 Rewards, Prizes, Threats, Negotiation and Virtual Companions 32
3.3 Student Modeling 34
3.4 Evaluation 36
3.4.1 Classroom experiment 37

3.4.2 Lab Experiment..38
3.5 Conclusion 39
4. Combining Intelligent Tutoring Systems and Virtual Reality Games 40
4.1 Evaluation aims and experiment 41
4.2 Evaluation results 46
4.2.1 First part of the evaluation 46
4.2.2 Second, third and fourth part of the evaluation 47
4.3 Interviews of students and teachers 49
4.4 Discussion and conclusions...50
5. VR-INTEGATE: A knowledge-based authoring tool for Intelligent Tutoring Systems integrated in a virtual reality game. 52
5.1 Operation of the authoring mode 53
5.2 Design Issues of VR-INTEGATE...53
5.3 Creating the Educational Content 54
5.4 Resulting Educational Game Application 57
5.5. Evaluation...59
5.6 Conclusions..61
6. Applying Cognitive Psychology to Student Modeling 63
6.1 The simulated students as evaluation agents in the software engineering process of an ITS 64
6.2 Learning and Forgetting...66
6.2.1 Base Retention Percentage..68
6.2.2 Individual Memorization Ability 70
6.2.3 Individual Response Quality...71
6.3 Conclusions..73
7. Using Intelligent Agents to evaluate the educational software 74
7.1 Reasoning of the evaluation agent 75
7.2 Temperamental model..76
7.3 Cognitive model...78
7.3.1 Retention and memorization capabilities 78
7.3.2 Diagnostics 79
7.4 The overall architecture 79
7.5 Iterations of the authoring process...81
7.6 Conclusions...82
8. Initializing a student model via the use of an online web game.............83
8.1 The game as an initialization application 84

8.2 "Two-of-a-kind" 84
8.3 Implementation Architecture 85
8.4 Cognitive Model 86
8.4.1 Retention Capabilities 86
8.4.2 Model Usage 88
8.5 Conclusions 88
9. Re-usability of the cognitive model via the use of Web Services 90
9.1 Web services 90
9.1.1 function CreateProfile (BRF:integer; MA:integer): string 90
9.1.2 ModifyProfile (aProfileId, BRF, MA): boolean 90
9.1.3 EncounterFact (aProfileId:string, aFactId:string) 91
9.1.4 QueryFact (aProfileId, aFactId, aResponseQuality): integer 91
9.1.5 ResetFactDatabase (aProfileId) 91
9.2 Interoperability 91
10. Evaluating the Memory Retention Model 92
10.1 The game "Two-of-a-kind" 93
10.2 Implementation 94
10.3 Evaluation Environment’s Effectiveness Results 100
10.4 Mathematical Model Evaluation Results 102
11. Conclusion 104
11.1 Contribution to Educational Games development 104
11.2 Contribution to Student Modeling 105
11.3 Contribution to the evaluation of educational software 105
11.4 Contribution to Applied Cognitive Psychology 106
11.5 Future work 106
REFERENCES 107

1. Introduction

Over the past few 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 playing them. For example, Griffiths and Hunt (Griffiths, M.D. & Hunt, N., 1995) conducted a study involving 387 adolescents and found among other things that approximately 30% of the adolescents play electronic games every day and the same proportion play once a month. However, as pointed out by researchers (Inkpen et al, 1994) very few play electronic games at school, since these games are not welcomed in class.

Indeed, many of educators are concerned 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 benefit of education. Papert (1993) found that computer games are fast-paced, immensely compelling and rewarding. Boyle (1997) noted 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. Salomon G., 1990; Welch M. and Brownell K., 2000) point out that technology is effective when developers fully 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 (Inkpen et al. 1994).

1.1 VR-ENGAGE

This dissertation will start by creating a set of necessary tools and platforms to generate a base for the research to be conducted. These tools constitute of a Virtual Reality game platform called VR-ENGAGE, which stands for Virtual Reality - Educational Negotiation Game. The environment of the game aims to increase students' motivation and engagement. However, the game also incorporates intelligence. It has the main components of an Intelligent Tutoring System (ITS), namely the domain knowledge, the student modeling component, and the tutoring component. In particular, the student modeling component models the student's knowledge and his/her ability to reason plausibly about domain knowledge acquired. In this way, while playing, students may practice both their factual knowledge (e.g. on geography) and their reasoning ability and thus they are led to "enjoyable" consolidation of knowledge.

The environment of a game plays an important role in its popularity. Griffiths (1995), 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. As such, VR-ENGAGE consists of four virtual reality games that share the same story and presentation principles such as music, lights, colors, and noise. However, each game is designed for a specific application domain and has a different virtual reality world associated with it. The domains are biology, history, spelling and mathematics. Each domain is taught in a different virtual world. History is taught in a virtual world of lands with castles and warriors, biology is taught in a virtual water world, spelling is taught in a virtual world of woods, and the domain of mathematics is taught in a virtual world of planets of the outer space.

The story of VR-ENGAGE incorporates a lot of elements from adventure games. Such elements include (but not limited to) dungeons, dragons, castles, keys, etc. However, each of these elements is connected to ideas and pedagogic approaches from educational software technology.

The ultimate goal of the player is to navigate through a virtual world and find the book of wisdom, which is hidden. To achieve the ultimate goal, the player has to be able to go through all the passages of the virtual worlds that are guarded by dragons and to obtain a score of points, which is higher than a predefined threshold. The total score is the sum of the points that the player has obtained by answering questions. Before being able to answer a question, the student needs to first find an agent inside the game's environment who will teach him/her the necessary “theory”. After that when the student faces the question s/he needs to recall the theory and answer correctly to be allowed to proceed deeper in the dungeon.

1.2 VR-INTEGATE

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 re-usability. At the same time, the construction of the application will probably take long if all issues involved are to be addressed.

As Murray (1999) 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.

Given the fact the ultimate goal is to make the VR-ENGAGE platform as flexible and adjustable to the various needs of a classroom environment and its students as possible, the use of an authoring tool was mandatory. For that the VR-INTEGATE tool was created. VR-INTEGATE stands for Virtual Reality- INTElligent Game Authoring Tuition Environment. This tool initially was designed to provide to the teachers an easy to use application through which they could design and manipulate the domain-specific content of the game. Using this tool, a teacher could define all the parts of the "theory" being taught in each of the virtual worlds, along with a complete set of questions that the students have to answer correctly to win.

As soon as student modeling started to be applied in the VR-ENGAGE environment, the authoring tool had to be extended to include a wider set of functions and services. Mainly a set of reporting graphs and tables that displayed the performance of each student based on their respected profile.

1.3 Student Modeling

Many researchers aim to make their multimedia systems more "intelligent" and adaptive to the learner's demands, abilities, and knowledge (Hasebrook & Gremm, 1999). Adaptivity and intelligence may be added to educational software if a student modeling component is incorporated into it. The student modeling component involves the construction of a qualitative representation that accounts for student behavior in terms of existing background knowledge about the domain and about students learning the domain (Sison & Shimura 1998). The VR-ENGAGE incorporates a common student modeling component for all four domains involved in the game. For each domain there are certain categories of error that have been encoded into the system. In this way, the system may analyze possible erroneous answers of students and perform error diagnosis. Following the diagnosis, the student is given a mark, which is translated to points for his/her total score, depending on the severity of his/her error. For example, if the system diagnoses that the student has only made a typographic error then s/he is given almost full points for his/her answer. However, if s/he gives a totally irrelevant answer then s/he receives almost no points for this answer.

VR-ENGAGE holds long-term information about each individual student. Such information is provided by all four domains involved. In particular, each domain contributes both domain-dependent and domain-independent information about each student. For example, a student may be consistently making a lot of spelling mistakes. This is a domain-independent error proneness of the student. This kind of feature is recorded in the student model and is updated constantly in all four domains. Conversely, there may be domain-dependent errors, which may only be made in the corresponding domain game.

The student modeling component of VR-ENGAGE examines the correctness of the students' answers in terms of the students' factual knowledge and reasoning that they have used. The long-term student model (Rich E. 1983) keeps a history record of the student and is used to adapt the presentation of lessons to the particular student's knowledge and possible weaknesses. The student modeling capabilities needed for the negotiation mode of the game are based on Human Plausible Reasoning theory. This theory formalizes the plausible inferences based on similarities, dissimilarities, generalizations and specializations that people often use to make plausible guesses about matters that they know partially. Important inference patterns in the theory are the statement transforms. These inferences may lead to either correct or incorrect guesses. In any case, these guesses are plausible.

1.4 Memory Retention Capabilities

In this dissertation, the goal is to extend the aforementioned student modeling by including measurements of the student's individual memory retention capabilities. Given the fact that the main goal of the educational software is to actually teach a set of facts to the students (using a somehow more appealing environment), a way to track the effectiveness of the whole process is needed.

Student modeling in VR-ENGAGE is based on the overlay technique. The overlay model was invented by Stansfield, Carr, and Goldstein (1976) and has been used in many early user-modeling systems (Goldstein, 1982) and more recent systems (Matthews et al. 2000). The main assumption underlying the overlay model is that a user may have incomplete knowledge of the domain. Therefore, the user model may be constructed as a subset of the domain knowledge. This subset represents the user's partial knowledge of a domain, enabling the system to know which parts of the theory the user knows and which s/he does not know. However, as Rivers (1989) points out, overlay models are inadequate for sophisticated modeling because they do not take into account the way users make inferences, how they integrate new knowledge with knowledge they already have or how their own representational structures change with learning. One additional problem with the overlay technique is that it assumes for the student an "all or nothing" knowledge of each part of the domain (either a student does or does not know something).

The overlay technique has to be used in conjunction with inference mechanisms about the students' knowledge. This research takes into account what parts of the theory the student has been shown, how often this has happened, and what the student is likely to remember. For this purpose, the overlay technique has been extended to include degrees of knowledge for each fact. Each degree represents the possibility of a student knowing and remembering something given the time at which it was learnt. For this purpose, a forgetting model was used.

There are two popular views on forgetting (Anderson, 2000). One of them, the decay theory, supports the view that memory traces simply fade with time if they are not "called up" now and then. The second view states that once some material is learnt, it remains forever in one's mental library, but for various reasons it may be difficult to retrieve. These may seem to be conflicting theories, but when someone has forgotten something, there is really no way for us to tell whether it has been completely removed from his or her mental library or is simply very (almost impossibly) difficult for him or her to retrieve it. For this study, both theories have practically the same meaning: If a student finds it hard to remember a fact that he or she has learnt (either due to memory fading or difficulty of retrieval) then the learning process was not good enough and should be modified.

A classic approach on how people forget is based on research conducted by Herman Ebbinghaus and appears in a reprinted form in (Ebbinghaus, 1998). Ebbinghaus worked for a period of one month and showed that memory loss was rapid soon after initial learning and then tapered off. In particular, Ebbinghaus' empirical research led him to create a mathematical formula which calculates an approximation of how much may be remembered by an individual in relation to how much time has passed since the end of learning.

In this dissertation, that formula was extended by the addition of two more factors, the Base Retention Factor and the Memorization Ability, so that it could be personalized and better tailored to the individual student's performance. Then the effectiveness of the formula was evaluated with great success using a classroom of students and conducting an educational session. During the session, everything the students did (choices, answers, performance, etc) was recorded. Afterward, the data gathered were analyzed and compared using the extended formula.

1.5 Using Virtual Agents to enhance the teaching experience

In view of the high demands on the reasoning abilities of educational software and the apparent need for the improvement of the software engineering process, simulated students have been created that can be used to improve the performance of educational applications dynamically (on the fly) and can be used as evaluation agents in an iterative software engineering process with these applications.

The simulated student-player incorporates the previously described cognitive model that keeps track of the students' memory of facts that have been taught to them. As a result, the educational application takes into account the time that has passed since the learning of a fact and combines this information with evidence from each individual student's actions. Such evidence includes how easily a student can memorize new facts and how well s/he can answer questions concerning the material being taught. In this way, the system knows when each individual student needs to revise each part of the theory being taught.

As soon as the system realizes that a student needs to revise a specific fact, then it can readjust the virtual world so that the student will be faced again with the same piece of "theory" to read and a relevant question to be tested with.

Moreover, the virtual agents can have an even better application. Whenever a teacher prepares a virtual world using the authoring tool (VR-INTEGATE), s/he has no other way to test the effectiveness of the lesson, but to actually let the students play the game and then check their scores/results. By using virtual agents, the teacher may prepare the lesson and then assign a number of virtual agents, initialized with different student's profiles, to "play" the game. With the use of the cognitive model the agents can evaluate the effectiveness of the session, providing enough information to the teacher so as to be able to make any necessary corrections prior to releasing the lesson to the students.

As it can be understood, the use of virtual agents can be extremely important both to make the game session more effective in teaching the specific domain knowledge that the teacher decides, but also to give the opportunity to the teacher to enhance the effectiveness of the teaching process before giving the game to the students.

1.6 Initializing Cognitive Model and evaluation

As soon as the effectiveness of the use of the cognitive model in the evaluation of the educational game was proven, a search for a better way to create the initial student profiles began.

To accomplish that, a new process was created through which the Base Retention Factor (BRF) and the Memorization Ability (MA) of a student could be calculated. By having those two factors, one could use Ebbinghaus' mathematical formula to effectively keep track of a student's performance. This process was first evaluated by conducting an experiment inside a classroom. The results were positive, but the target group was relatively small (just one class). We wanted to stress test the effectiveness of the mathematical model by testing it on a larger population. The goal was to have the model tested by people of different ages and different places in the world.

To accomplish this, an online web memory game was created. The game was based on the popular “two-of-a-kind” memory game where the player is asked to find and match pairs of the same cards. The game was created using DHTML, PHP, and MySQL. It was deployed on a web server and numerous invitations to players in existing online games were sent.

The project was a huge success. It managed to attract 115 players that played 1585 games. The data gathered were then applied to the aforementioned process to evaluate its effectiveness. Once again, it was found that both the cognitive model's formula and the evaluation process were successful in tracking the players’ memory retention capabilities and providing insights on how much information they can retain.

In the next chapters we will go through all the techniques and theories that were used in the creation of the final student model and it’s application to a virtual reality educational software.

2. Field Overview

As already stated, the ultimate aim of this thesis is the enhancement of a student's profile in such a way that it can be used by an educational software to provide information about the student's memory retention capabilities. Before detailing the approach and implementation techniques of this project, it is necessary to give an overview of the various sectors involved.

First, various computer game genres and their characteristics will be examined. As the aim is the enhancement of an educational game, it is important to review the current game industry. The basic description of each genre, along with their special characteristics, will be discussed.

Next, the use and functionality of educational software will be described. It should be noted that not all educational software are computer games and not all games strive to develop the same skills. In the following pages, these differences will be revealed and an explanation of this project's objective will be given.

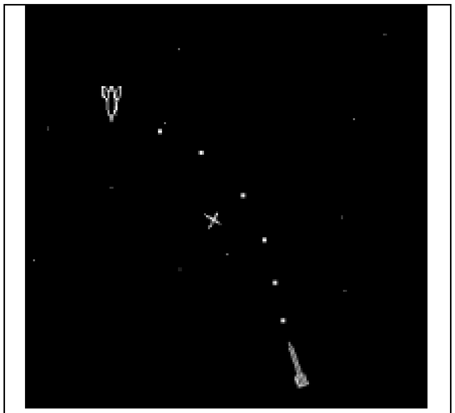
Computer games with no visual and audio effects do not exist (at least not nowadays). In order to have success, a computer game must have an internal engine that, with the use of graphical and audio effects, will be able to produce a stunning virtual environment. Thus, this chapter will go through the basic ingredients for the creation of such an environment.

Finally, an overview of the artificial intelligence that is used inside computer games will be discussed. The final product of this thesis will be an artificial intelligence module that, when used by the computer game software, will give it the ability to measure the student's memory retention capabilities. Therefore, this chapter will also identify the artificial intelligence techniques already used by the gaming industry and determine whether this project’s module will be useful.

2.1 Computer Games

Although personal computers only became popular with the development of the microprocessor, mainframe and minicomputers, computer gaming has existed since at least the 1960s. One of the first computer games was developed in 1961, when MIT students Martin Graetz and Alan Kotok, with MIT employee Steve Russell, developed *Spacewar!* on a PDP-1 computer used for statistical calculations.

The first generation of PC games was often text adventures or interactive fiction, in which the player communicated with the computer by entering commands through a keyboard. The first text-adventure, *Adventure*, was developed for the PDP-11 by Will Crowther in 1976, and expanded by Don Woods in 1977. By the 1980s, personal computers had become powerful enough to run games like *Adventure*, but by this time, graphics were beginning to become an important factor in games. Later games combined textual commands with basic graphics, as seen in the SSI Gold Box games such as *Pool of Radiance*, or *Bard's Tale*.



Spacewar!, developed for the PDP-1 in 1961, is often credited as being the first ever computer game. The game consisted of two player-controlled spaceships maneuvering around a central star, each attempting to destroy the other.

During the last 15 years, electronic games have evolved significantly while at the same time gained a special place in our everyday life. The technological advancements helped the gaming industry to create games with great characteristics. Virtual environments, 3D graphics, audio effects and broadband Internet connections have made the creation of extremely realistic games possible.

Special 3D engines that simulate real life's laws of physics have been developed and have made the virtual environments seem real. Along with digital audio and scenarios that are equivalent to movies, the budget of the gaming industry was boosted to great heights.

The investments made by gaming corporations, along with widespread broadband Internet connections, opened up a multimillion dollar market. This increased competition, which in turn lead to more technological advancements. The have games evolved from mere action coin-up games with 16 colors to full blown online interactive virtual environments that can be played at a home computer or gaming console (like Nintendo, Xbox, etc). New game types have been created trying to address all gamers' needs.

Depending on their characteristics, digital games can be categorized in many different ways. For example, you can categorize by game type, requirements, gaming style, or scenario type.

There are five major 'abstract' game categories that depend on the style of the game. These categories are: Action/Arcade games, Adventure Games, RPG Games, Simulations, and Strategy Games. Although modern games tend to have characteristics from more than one of these categories, every game can be put into at least one.

2.1.1 Adventure Games

Myst used high-quality 3D rendered graphics to deliver images that were unparalleled at the time of its release.

Adventure games present the player with various situations that require correct judgment and analytical skills to be successfully overcome. Generally, the player is asked to solve a series of riddles to unravel the mystery of the game. Knowledge skills are put to a test and a very good command of the game's language is required (arcade games can be played without knowledge of the game's language as long as someone explains the rules).

For much of the 1980s, adventure games were one of the most popular types of computer games produced. However, their market share drastically declined in the mid-1990s. Action games took a greater share of the market, particularly first person shooter games such as *Doom* and *Half-Life*, which progressively began featuring strong, story-structured solo games. This slump in popularity led many publishers and developers to see adventure games as financially unfeasible in comparison. Text adventures met the same fate much earlier, but their simplicity has allowed them to thrive as non-commercially developed interactive fiction.

Few recent commercial adventure games have been successful in the United States, but they maintain popularity in Europe (95% of all adventures released in the United States are in fact translated European products). It has been suggested that this is because today's average American gamer was introduced early on to console video games and first person shooters rather than "traditional" computer games. Others believe that MMORPGs (Massive Multiplayer Online Role Playing Games), which offer a persistent multiplayer world, have at least partially supplanted the genre.

Examples of such games are: *Myst*, *Space Quest Series*, *Legend of Kyrandia*, *Quest for Glory*, *Under a killing moon*, *Loom*, *The dig*, *Little big adventure* and many more. Unfortunately, games that clearly fall into this category tend to disappear. Several years ago there were companies dedicated to this type of game (i.e. Sierra), but now games tend to have more arcade and RPG features.

2.1.2 Action / Arcade games

Action games require players to use quick reflexes and timing to overcome obstacles. They often include tactical conflict, exploration challenges, and puzzle-solving, but these are not defining elements. Action games are the broadest and most inclusive genre in gaming, encompassing many diverse sub-genres such as fighting games, first-person shooters, beat 'em ups, and platform games.

Action games typically feature violent physical force, especially shooting or melee combat as their main interactive feature.

Final Fight, a scrolling beat 'em up

While the earliest action games appeared on computers, most action game genres were popularized in the video arcades that became popular in the 1970s and '80s. At that point, the vast majority of games focused on tests of dexterity that lent themselves to the short, addictive play that the arcade format thrived on. Sports and driving themes were common, but other games with more varied (and usually violent) themes began to form the action genre.

While the objective of an action game varies from game to game, it generally involves advancing through stages (or levels), eliminating hordes of enemies, and solving puzzles. Many games include one or more "Bosses", often preceded by "Mini-Bosses". A Mini-Boss is usually the climax to a level or series of levels, with a Boss encountered either at the end of the game or periodically throughout the game, leading up to an "End-game Boss", whose defeat is the objective of the game.

Bosses are typically defeated through use of pattern recognition skills and physical reaction speed. In most older action games and even many modern ones, the bosses are programmed with a simple pattern of attacks or moves that players learn through trial and error. These simple patterns would often include combo moves that require a player to jump, dodge, or block an attack, then strike at certain points to deal damage, perhaps even waiting out or timing the patterns to get in attacks.

Many sub-genres, such as platform games and action-adventure games, add gymnastic-style puzzles, such as timing jumps to and from moving platforms. Platform games, whether 2D or 3D, are usually similar in concept to the original *Mario Bros.* game series. Some action games feature third-person shooter gameplay, enabling the player to acquire and upgrade various weapons, each sporting its own special abilities.

Another common sub-genre is the shoot 'em up, which usually involves the player controlling a character or vehicle brandishing many weapons and shooting anything that moves on-screen. This genre is well known for its side and vertical scrolling shooter games.

Studies have shown that people can improve their eyesight by playing action video games. Tests by scientists at the University of Rochester on college students showed that over a period of a month, performance in eye examinations improved by approximately 20% in those playing *Unreal Tournament* compared to those playing *Tetris*. (New Scientist Tech) . It is believed that this is due to the action game improving the spatial resolution of the players' vision.

Some strategy, or tactics, is often required to accomplish various tasks, but with practice you can overcome every obstacle. Games that fall into this category include: *Pacman*, *Cabal*, *Kickoff*, *Tetris*, *Doom*, *Sands of Time*, *Unreal* and many more.

2.1.3 Role Playing Games (RPG)

RPGs are the games where (as per the title) you have control of a character with a very specific role in the game. Participants determine the actions of their characters based on their characterization (Kim, John) and the actions succeed or fail according to a formal system of rules and guidelines. Within the rules, players have the freedom to improvise. Their choices shape the direction and outcome of the game.

OFFICIAL
Advanced Dungeons & Dragons
COMPUTER PRODUCT
EYE OF THE BEHOLDER
IBM PC
A LEGEND SERIES
Fantasy Role-Playing Saga, Vol. I
SSI
STRATEGIC SIMULATIONS, INC.

Eye of the Beholder, one of the first and very famous RPG

The main difference of this category is the fact that the game also includes an advancement procedure for the characteristics and abilities of the game's main character. For example, the player can have his/her character train the “strength” so s/he can do more damage or his/her “stamina” so that s/he can run faster and so on. Whenever the character accomplishes a specific task in the game, s/he is awarded XPs (eXperience Points) which can be spent to build up a character's abilities.

These games were invented when people tried to move the older tabletop DnD (Dungeon & Dragons) games to the computer. DnD games are conducted like radio drama, only the spoken component is acted. One player, the game master (GM), creates a setting in which the other players play the role of a single character. (Kim, John) The GM describes the game world and its inhabitants. The other players describe the intended actions of their characters and the GM describes the outcomes. Some outcomes are determined by the game system and some are chosen by the GM.

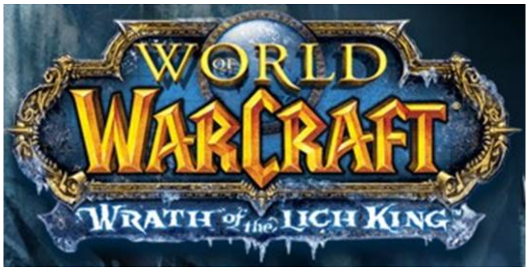
Examples of such games are: *Eye of the Beholder*, *Lands of Lore*, *Wizards and Warriors*, *Diablo*, *Menzoberanzan*, *Baldur's Gates*, *Neverwinter Nights,* etc.

With the introduction of fast Internet connections, the above category evolved to **MMORPGs**, which stands for Massive Multiplayer Online Role Playing Games. As per the title, the game can be played online using Internet and the players have the option of combining forces in parties to accomplish difficult tasks inside a virtual world.

MMORPGs are distinguished from single-player or small multi-player CRPGs by the number of players and by the game's persistent world, usually hosted by the game's publisher, which continues to exist and evolve while the player is away from the game.

MMORPGs are very popular throughout the world. Worldwide revenues for MMORPGs exceeded half a billion dollars in 2005 and Western revenues exceeded US$1 billion in 2006 (Park Associates).

Although modern MMORPGs sometimes differ dramatically from their antecedents, many of them share some basic characteristics. These include common themes, some form of progression, social interaction within the game, in-game culture, system architecture, and a large degree of character customization. The characters can often be customized quite extensively, both in the technical and visual aspects, with new choices being added constantly. Players might also "mod" in order to allow for even greater flexibility of choice.



World of Warcraft, the most played Massive Multiplayer Online Role Playing Game (MMORPG)

Character abilities are often very specific. Depending on the particular game, the specialties might be as basic as simply having a greater affinity in one statistic, gaining certain bonuses of in-game resources or related in-game race, job, etc.

The most successful game of that category is *World of Warcraft* with approximately 10 million player accounts. Other games in this category include: *Lineage*, *GuildWars*, *Lord of the Rings Online*, *Conan*, *Unreal Tournament*.

2.1.4 Simulation Games

Simulation Games' category is a bit confusing. At the beginning, it included games that simulated specific technological aspects, like flight simulators, driving simulators, space simulators, and they were normally used for

SimCity Societies (@2007)

educational purposes. Then the game evolved to real life simulation where players took charge of a big company (*ThemePark*, *Railroad Tycoon*), the position of a mayor (*SimCity*), or even God *(Black & White*). These games simulated the reactions of the environment played in (city, company, hospital, the world, etc) to the actions of the player. These games have very strong elements of strategy, but they are clearly described as simulators. Some of the most known games of this category include: *The Sims Series*, *Railroad Tycoon, SimCity, Theme Hospital, Theme Park, Black & White, Microsoft Flight Simulator, Test Drive Series* and many more.

2.1.5 Strategy Games

Command & Conquer Red Alert

A **strategy game** is a game (e.g. computer, video or board game) in which the players' decision-making skills have a high significance in determining the outcome. Many games include this element to a greater or lesser degree, making demarcation difficult. It is therefore more accurate to describe a particular game as having a certain degree of strategic elements, as in being mainly based around strategic principles.

The crucial factor that separates this type of game from all others is that there is relatively little chance involved. All players have equal degree of knowledge of the elements of the game. There is no physical skill required other than that necessary to interact with the game pieces.

The most classical game of this category is chess. Games in this category include: *Warcraft*, *Starcraft, UFO Enemy Unknown, Command & Conquer, Dune2,* and *Supreme Commander*.

2.2 Educational Software

It is commonly known and accepted that games (whatever their nature, be them dolls, tabletops, athletic, gambling, or computer based) have a great impact on a child's education. Children can acquire a great number of skills through playing, and sometimes with even better results than through simple, lecture-based teaching. Based on this idea, the computer games industry started to produce educational software. With computers entering almost every home in the world, a new area of educational skills and techniques laid ready to be discovered.

Before delving deeper, it is important to note that there are a large number of educational software for enhancing one's knowledge skills in the market (i.e. learn a foreign language, school lesson helpers, etc). This dissertation will only be concerned with software that is actually a computer game. This includes educational computer games and not educational software in general.

For the purposes of this dissertation, educational games will be divided into two categories based on their educational target: games that intend to enhance the player's knowledge skills, and games that try to enhance the player's physical or reflexive skills. There are of course games that include features from both of these categories, but due to specific hardware requirements, these are counted in the latter category.

2.2.1 Target: Knowledge

The majority of educational software games aim to teach the player a specific content. Be it mathematics, physics or national history, the game uses a varied number of techniques to present the content to be taught, lets the player study it, and then tests him/her on the subject.

The whole process is wrapped up in a game's environment, which makes it more appealing to the player who often does not feel like s/he is being tested (thus psychologically speaking s/he is more relaxed, more focused and more productive [ref]) or being taught.

In general, children tend to resist traditional education systems. If asked to choose between going to school to learn or going outside to play, children are more likely to choose play over and over again. By incorporating the educational process inside a game, children can play and learn at the same time.

2.2.2 Target: Physical / Reflexive Skills

Recently, a new category of games hit the market, which aims to enhance the players physical skills. Games that need good reflexives have existed from the beginning of the software gaming industry, but games that make you workout or exercise while playing have appeared during the last five years.

The main reason for the 'late' appearance of this category is hardware support. These games came along with various hardware accessories that only lately can be produced at reasonable prices for the general public. One of the most popular games that helped launched this game category is *Guitar Hero*. Nintendo's Wii Platform and, more recently, Xbox 360's Kinect, have become leaders in this category.

Guitar Hero comes with a digital guitar that players plug into the gaming console and are asked to play a series of famous songs in increasing difficulty. With this game, player's enhance their reflexive, their understanding of musical rhythm, and learn to synchronize their hands and fingers. These skills are essential when playing a real musical instrument. The latest version of the game comes with the addition of drums, a bass, and a microphone for vocals.

Nintendo's Wii platform is not specialized, but comes with a variety of games that require body movements to be completed. One of the latest games of the Wii series is called *Wii-fit*. This game comes with an extra hardware device that can be used to perform various physical exercises and it is being promoted more as a workout device than a game.

2.2.3 Authoring Tools

From the beginning of the existence of educational software, it became clear that there is no way to create an educational process where human intervention (i.e. the teacher) is not required. With this in mind, the idea of a 'helping' software, which gives the ability to modify the content of the educational software to better match the player's (i.e. student's) needs, arose. These software programs are called 'Authoring Tools'.

Knowledge-based authoring tools are meant to be used by instructors who wish to author their own Intelligent Tutoring Systems (ITSs) on a certain domain. Murray [8] highlights the potential of ITS authoring tools in giving the instructional designer a combination of facilities to produce visually appealing, interactive screens and a deep representation of content and pedagogy. Authoring tools have to be used for multiple domains. Therefore, the methods incorporated in the authoring tools have to be domain-independent. In addition to domain-independence, the resulting ITSs should have a learner modeling capability that may diagnose the learners' weaknesses and support a teaching method adaptive to the learners' needs. The following chapters include further description of such models.

Although there is a great need for authoring tools, the gaming industry does not provide them with the games it produces. This is mainly due to small market penetration and corporate marketing decisions. On the contrary, computer game **mods** are very popular. Mod, or modification, is a term generally applied to computer games, especially first-person shooters, RPGs and real-time strategy games. Mods are made by the general public or a developer, and can be entirely new games in themselves. However, mods are not standalone software and require the user to have the original release in order to run. They can include new items, weapons, characters, enemies, models, textures, levels, story lines, music, and game modes.

Thus, although the need of “modding” in educational software is great, the “modding” functionality is not implemented in educational software. Educational software is released as a stand-alone fixed bundle, which can not be changed by either the user or a teacher.

2.3 Virtual Reality (VR)

Virtual reality (VR) is a technology which allows a user to interact with a computer-simulated environment, be it a real or imagined one. Most current VR environments are primarily visual experiences, displayed either on a computer screen or through special or stereoscopic displays, but some simulations include additional sensory information, such as sound through speakers or headphones. Some advanced, haptic systems now include tactile information, generally known as force feedback, in medical and gaming applications. Users can interact with a virtual environment or a virtual artifact (VA) either through the use of standard input devices such as a keyboard and mouse, or through multimodal devices such as a wired glove, the Polhemus boom arm, or an omnidirectional treadmill. The simulated environment can be similar to the real world, for example, simulations for pilot or combat training, or it can differ significantly from reality, as in VR games. In practice, it is currently very difficult to create a high-fidelity virtual reality experience, due largely to technical limitations on processing power, image resolution and communication bandwidth. However, those limitations are expected to eventually be overcome as processor, imaging, and data communication technologies become more powerful and cost-effective over time.

VR is often used to describe a wide variety of applications, commonly associated with its immersive, highly visual, 3D environments. The development of CAD software, graphics hardware acceleration, head mounted displays, database gloves, and miniaturization have helped popularize the notion. In the book *The Metaphysics of Virtual Reality*, Michael Heim identifies seven different concepts of VR: simulation, interaction, artificiality, immersion, tele-presence, full-body immersion, and network communication. The definition still has a certain futuristic romanticism attached. People often identify VR with head mounted displays and data suits.

2.3.1 Audio and Visual Enhancements

There are two major factors that play an important role in the representation of a game. These include visual and audio capabilities. For each of these factors, a very specific and specialized piece of hardware is needed in home computers: sound cards and video cards. These cards are responsible for the reproduction of audio or visual effects respectively.

Sound cards compatible with the IBM PC were very uncommon until 1988, which left the single internal PC speaker as the only way early PC software could produce sound and music. The speaker hardware was typically limited to square waves, which fit the common nickname of "beeper".

One of the first manufacturers of sound cards for the IBM PC was AdLib, who produced a card based on the Yamaha YM3812 sound chip, aka the OPL2. Creative Labs also marketed a sound card at about the same time called the Creative Music System (C/MS). Although the C/MS had twelve voices compared with AdLib's nine and was a stereo card while the AdLib was mono, the basic technology behind it was based on the Philips SAA 1099 chip, which was essentially a square-wave generator.

A large change in the IBM PC compatible sound card market occurred when Creative Labs introduced the Sound Blaster card. The Sound Blaster added a sound coprocessor for recording and play back of digital audio, a game port for adding a joystick, and capability to interface with MIDI equipment (using the game port and a special cable). The Sound Blaster line of cards, together with the first inexpensive CD-ROM drives and evolving video technology, ushered in a new era of multimedia computer applications that could play back CD audio, add recorded dialogue to computer games, or even reproduce motion video.

The first IBM PC video card, which was released with the first IBM PC, was developed by IBM in 1981. The MDA (Monochrome Display Adapter) could only work in text mode representing 25x80 lines on the screen. It had a 4KB video memory and just one color. [5]

Following the MDA, several other video cards were released. One of them was VGA, which was widely accepted. This led some corporations such as ATI, Cirrus Logic and S3 to work with that video card, improving its resolution and the number of colours it used. As a result, the SVGA (Super VGA) standard was created, which reached 2 MB of video memory and a resolution of 1024x768 at 256 color mode.

In 1995, the first consumer 2D/3D cards were released, developed by Matrox, Creative, S3, ATI, and others. Those video cards followed the SVGA standard, but incorporated 3D functions. In 1997, 3dfx released the Voodoo graphics chip, which was very powerful compared to other consumer graphics cards, introducing 3D effects such mip mapping, Z-buffering, and anti-aliasing into the consumer market. From this point, a series of 3D video cards were released, like Voodoo2 from 3dfx, and TNT and TNT2 from NVIDIA. The bandwidth required by these cards was approaching the limits of the PCI bus capacity. Intel developed the AGP (Accelerated Graphics Port), which solved the bottleneck between the microprocessor and the video card.

Having created a solid base in both video and audio hardware, a race in both areas between the manufacturers began. This led to a dramatic advancement over the last 10 years in the technology (hardware) and the mathematical formula and functions that support 3D graphics and 3D audio.

3D graphic engines have been invented that are extremely fast and can create millions of polygons in real time (instead of pre-rendering techniques that were used at the beginning). Backed by very powerful graphic cards (which are cheap and available to the majority of the common computer users), the possibility of creating visually an almost perfect game environment has appeared. The following chapter will discuss the techniques and the workings of such environments in greater detail.

From single channel audio, the industry quickly moved to stereo. From there, with the introduction of Dolby Surround technique, audio was given a 3D feeling. Furthering that technique, the 4.1 surround system was invented. With the use of a subwoofer and 4 speakers (left, right, back and front), listeners were given the sensation that they were actually in the "middle" of a scene. For example, during a war movie or game, the sound of enemy fire could come from a back speaker and a comrade’s voice from the left, making the listener feel s/he is truly experiencing the event. The final step was the addition of a fifth speaker that produced only bass audio. With that, the 5.1 surround system was born, which is something that every gamer can have in his/her house.

2.3.2 Virtual Environments

With a very advanced audio and visual technique background and support, one can create a virtual environment that "feels" almost like the real thing. What needs to be created are mathematical formulas that simulate the laws of physics and how items inside the virtual world interact with them.

Currently, one of the most known engines with extremely good results is the 3D engine of Crysis Corporation. The techniques used inside this engine can reproduce incredible details, from lighting effects and shadows on moving objects to exploding particles that interact with other objects of the environment that come into their path.

The techniques and the laws are not new. They have been used in films for over a decade. However, in films, everything is pre-rendered. It might take two days of rendering to see three seconds of great graphics in a movie. What Crysis has accomplished is the creation of such graphical representations in real time.

2.3.3 Artificial Intelligence & Dynamic Environments

Educational software is a special kind of software that aims to facilitate the difficult cognitive process of students' learning. In this respect, educational software has to combine many qualities to make the most of the interactive means provided by computers and to be educationally beneficial. Such qualities include attractive multimedia presentations, the individualization of tutoring, reasoning abilities, user-friendly interfaces, and etc. To achieve all these qualities there has to be a combination of educational software technologies and ideas, such as the combination of multimedia educational software technology with the underlying reasoning mechanisms of Intelligent Tutoring Systems (ITSs).

Thus, taking into consideration the games' adapting capabilities we can create another abstract categorization. In it we find two entries: Sandbox/Static games and Dynamic Games.

The majority of the games depend on a static and linear scenario. The player has to overcome a series of tests (mental or reflexive) in a specific order so that s/he can start from the 'beginning' and reach the 'finale' of the game. Every player that finishes a game gets more or less the same experience out of it. These games are static and because the player can not change the environment they are called 'sandboxes'.

On the other hand, there are games that evolve along with the player. Their environment and their scenario are not pre-defined, they start from a specific point, but from then on everything changes depending on the player's choices. The changes can be from very small to very complicated and can end up giving each player a very unique experience, tailored to his/her specific needs.

This dissertation will examine and present how to enhance such a game by providing a framework and services that can create a profile of the player's memory retention capabilities. Although memory is a skill that needs to be practiced in all games, its use is far stronger in educational software; therefore, we will focus on them for the rest of the work.

2.4 Cognitive Psychology – Memory Retention

There are two popular views of forgetting (Anderson, 2000). First, the decay theory holds that memory traces simply fade with time if they are not “called up” now and then. The second view states that once a material is learned, it remains forever in one’s mental library, but for various reasons it may be difficult to retrieve. These theories may seem to be conflicting, but when someone has forgotten something, there is really no way for us to tell whether it has been completely removed from his/her mental library or if it is very difficult for him/her to retrieve it. For our study, both theories have practically the same meaning; if a student finds it hard to remember a fact that s/he has learnt (either due to memory fading or difficulty of retrieval) then the learning process was not that good and should be modified.

2.4.1 Ebbinghaus' Mathematical Model

Based on research conducted by Herman Ebbinghaus (1998), a classical model on how people forget was developed. Ebbinghaus worked for a period of one month and showed that memory loss was rapid soon after initial learning and then tapered off. In particular, Ebbinghaus' empirical research led him to the creation of a mathematical formula that calculates an approximation of how much can be remembered by an individual in relation to the time it was learned (Equation 1).

$$b=\frac{100*k}{(\log t)^{c}+k} \qquad (1)$$

In Equation 1:

- t: is the time in minutes counting from one minute before the end of the learning
- b: is the equivalent of the amount remembered from the first learning. As it is evident from the logarithmic nature of the formula, b lowers greatly at the beginning and starts to stabilize after time passes on.
- c and k: are two constants with the following calculated values: *k* = 1.84 and *c* = 1.25

Linton (1979) also conducted research on retention of knowledge and worked for a period of six years. Linton's results were similar to Ebbinghaus' results. Finally, Klatzky (1980) also reports the results of a study that consisted of experiments on retention. These experiments involved repetitions of a memorized list of words after a pre-specified break length, typically up to few days. This study showed that memory decay is a power function of the break length. For example, subjects forget 55% of the words within a six hour break time and 80% percent within 72 hours. However, these results are very close to Ebbinghaus results. Indeed, if Ebbinghaus' formula was used, one would find that subjects forget 60% of the words within a six hour break and 75% within 72 hours. Such differences in the results have little significance for the purposes of the incorporation of a forgetting model in the educational application. Therefore, Ebbinghaus' mathematical formula has been used in this thesis to give the system insight about the students' learning and forgetfulness.

2.5 Student Modeling

Since Information Technology has been widely spread and offered its services to many domains and disciplines, it has also been widely acknowledged that it can be very useful for assisting education. To this end, many educational software researchers employ several means of computer technology to improve the presentation, the structure, and reasoning abilities of educational software. One important goal has been the improvement of attractiveness and aesthetics of educational applications based on multimedia, virtual reality, and etc. However, on the other hand, multimedia educational products are often criticized because they do not support the learner well nor exploit the capability of the medium (Laurillard 1995, Montgomery 1997, Moore 2000). From this point of view, the incorporation of a student modeling component into educational software may be quite important for rendering the system more adaptive to the student's learning needs, abilities, weaknesses, and knowledge.

The educational software application aims at teaching students in a motivating way. Therefore, teaching and testing takes place in the environment of a virtual reality game. Indeed, recently researchers in educational software point out the virtues of computer games relating to children and adolescents' education. For example, Muntaz (2001) notes that a range of cognitive skills are practiced in computer game playing given the sheer number of decisions children make as they weave their way through various games.

2.5.1 General Techniques

Student modeling is an important process for ITSs since it may provide detailed reasoning concerning the students' needs and progress and thus make the applications highly individualized. Indeed, student modeling has become a core or even defining issue for ITSs (Cumming & McDougall 2000). In the research described in this thesis, the student models give input for the creation of simulated students. Then the simulated students are used during the execution of the educational applications to give insight to the system as to how much a student has learnt from the material that has been taught to him/her and what needs to be reviewed. If something needs to be reviewed the ITS reschedules the teaching material and presents the topic to the student to be reviewed. Moreover, the simulated students may be used by instructors and ITS designers to evaluate the courses that they have created before they are delivered to real students. Thus, designers are given the opportunity to fine-tune the ITSs to achieve better results with the real students.

Simulated students have been created and used in past ITSs, mainly to assist the learning process of students. For example, the mode of the simulated co-learner has been considered quite important by many researchers for the purpose of improving the educational benefit of tutoring systems. One reason for this is the fact that the simulated student can simultaneously be an expert and a co-learner and can thus scaffold and guide the human's learning in subtle ways (VanLehn et al. 1994). However, simulated students have not been used as evaluation components in the software engineering process of

ITSs. Such evaluation components may be very useful because they allow and encourage multiple iterations of the design process, which in turn may ensure that the resulting educational software applications are of better quality.

2.5.2 The “Overlay” technique

Student modeling in this dissertation is based on the overlay technique. The overlay model was invented by Stansfield, Carr and Goldstein (1976) and has been used in many early user modeling systems (Goldstein, 1982) and more recent systems (Matthews et al. 2000). The main assumption underlying the overlay model is that a user may have incomplete knowledge of the domain. Therefore, the user model may be constructed as a subset of the domain knowledge. This subset represents the user's partial knowledge of a domain and thus the system may know which parts of the theory the user knows and which s/he does not know. However, as Rivers (1989) points out, overlay models are inadequate for sophisticated modeling because they do not take into account the way users make inferences, how they integrate new knowledge with knowledge they already have or how their own representational structures change with learning. An additional problem with the overlay technique is that it assumes for the student an "all or nothing" knowledge of each part of the domain (either a student knows something or not).

The overlay technique has to be used in conjunction with inference mechanisms about the students' knowledge. The inference mechanisms that have been employed so far have been mainly based on students' actions in assessment tests that show evidence of the students' knowing or not knowing something. However, even in cases where the student shows evidence of knowing something at a particular time, s/he may forget it after a while. Therefore in our research we take into account what parts of the theory the student has been shown, how often this has happened and what s/he is likely to remember. Therefore, the overlay technique is extended to include degrees of knowledge for each fact. Each degree represents the possibility of a student knowing and remembering something given the time it was learnt. For this purpose, Ebbinghaus’ mathematical forgetting model will be used.

2.6 Conclusions

The gaming industry has gone to great lengths to attract their target audience. Using state of the art graphical engines and cards, along with audio software and hardware, it managed to create very realistic virtual environments that can react to the basic laws of physics.

With the use of artificial intelligence in these environments, NPCs (Non Player Characters) can acquire behavior models and react like normal people to the player's actions, thus increasing the game reality level.

The above techniques enable the creation of educational games that are appealing to students. Gaining and maintaining the attention of students is essential to the learning process.

It should be noted that the teaching process is not static. While teaching, a teacher monitors the students’ reactions, answers, emotions, and mood, trying to figure out how s/he can better adjust the process so that students will be able to learn as much as they can. In order to make educational games effective, it is important to use techniques that will give the gaming engine the ability to modify the game accordingly. They should emulate the way teachers work, thus making the player (i.e. student) practice the aimed skill as much as possible.

Artificial intelligence modules and techniques that can monitor and report on various psychological emotions of the user already exist (e.g. whether s/he is having fun, feels frustrated etc.). Also, the techniques used to dynamically change the games’ virtual environment and content already exists. There is, however, still a need to find a way to monitor and report on the student's performance.

The objective of this dissertation is to explore the creation of an artificial intelligence module that will monitor a student's progress in a game. With the use of a modified student profile and algorithms that can simulate a student's retention capabilities. This module will report to the game's engine facts on whether the student has actually learnt the targeted piece of information or not. The game should then decide if should dynamically modify its content and virtual environment to allow the player to revise that piece of information. This will effectively enhance the teaching process of the educational software.

3. VR-ENGAGE: A Virtual Reality – Educational Negotiation Game.

In this chapter, a virtual reality educational game will be presented. The game is called *VR-ENGAGE* which stands for *Virtual Reality - Educational Negotiation Game*. The environment of the game aims to increase students' motivation and engagement. However, the game also incorporates intelligence. It has the main components of an ITS, namely the domain knowledge, the student modeling component and the tutoring component. In particular, the student modeling component sculpts the student's knowledge and his/her ability to reason plausibly about the domain knowledge acquired. In this way, while playing, students may practice both their factual knowledge on geography and their reasoning ability and thus they are led to an "enjoyable" consolidation of knowledge.

3.1 The Game's Environment and Game Play

VR-ENGAGE consists of four virtual reality games that share the same story and presentation principles, such as music, lights, colors, and noise. Moreover, they share the same underlying reasoning mechanisms such as the student modeling component. However, each game is designed for a specific application domain and has a different virtual reality world associated with it. The domains are biology, history, spelling, and mathematics. Each domain is taught in a different virtual world. History is taught in a virtual world of lands with castles and warriors. Biology is taught in a virtual water world. Spelling is taught in a virtual world of woods, and the domain of mathematics is taught in a virtual world of planets in outer space.

The game has features that are quite common in virtual reality adventure games. Such features include dungeons, dragons, castles, keys, etc. In *VR-ENGAGE* the player tries to reach the "land of knowledge" and find the treasure, which has been hidden there. However, to achieve this, the player has to obtain a good score, which is accumulated by all four domains. The idea behind this is to motivate students to have a good standard in all lessons, which are different from each other.

The environment of a game plays a very important role in its popularity. Griffiths (1995), 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.

The environment of *VR-ENGAGE* is similar to that of the popular game called *DOOM (ID-Software 1993)*, which has many virtual theme worlds with castles and dragons that a player has to navigate through to achieve the goal of reaching the exit. *VR-ENGAGE* has also many virtual worlds where a student has to navigate through. There are medieval castles in foreign lands, castles under water, corridors and passages through fire, temples hiding secrets, dungeons and dragons. The main similarity of *VR-ENGAGE* with computer games like *DOOM* lies in their use of a 3D-engine. However, *VR-ENGAGE,* unlike *DOOM* and other computer games of this kind, is not violent and is connected to an educational application.

In all four worlds there are animated agents that communicate with the players. There are two types of animated agents, the advisor and the guard of a passage. Advisors lead students to lessons that they have to read. Guards of passages ask the players questions and allow them to continue into the passage and receive points. If a student does not know how to answer a question, s/he may ask for help. In such cases, the advisor helps the student give the correct answer and thus the student may continue on his/her way into the passage, but s/he does not receive any points for the total score.

VR-ENGAGE communicates its messages to students through animated agents or through windows that display text. When a student is asked a question s/he may type the answer into a dialog box. The user interface employs two types of animated agents, the dragon which is the virtual enemy of the player and the virtual companion of the player. Both types of animated agents use synthesized voice as well as written messages. However, their voices are different so that the player is able to distinguish between them. The reason why the animated agents use voice is that there are studies that show that voice messages may be more effective than written ones in the way that students react to the educational applications (e.g. Walker et. Al 1994). In addition, it was considered important for the "aura" of the game.

Players are also allowed to select whether they want background music or not. If they do, they are allowed to select the background music that they prefer from a menu. The reason there is a high degree of choice for the status of the background music in *VR-ENGAGE* is because there is controversy as to what affects background sounds may have on performance. For example, a study conducted by Smith (1997) has shown that background sounds may be stimulating, but they may also have negative effects on performance. On the contrary, another experiment, which involved five computer games (Wolfson & Case, 2000) has shown, among other things, that sound level had little influence on performance scores and errors. Therefore, in *VR-ENGAGE*, which is primarily aiming at educating players and stimulating them to think, a player may turn off the sound if s/he feels that s/he is disrupted.

3.2 Rewards, Prizes, Threats, Negotiation and Virtual Companions

The story of *VR-ENGAGE* incorporates many elements from adventure games. However, each element is connected to ideas and pedagogic approaches from educational software technology.

The ultimate goal of a player is to navigate through a virtual world and find the book of wisdom, which is hidden. To achieve the ultimate goal, the player has to be able to go through all the passages of the virtual worlds that are guarded by dragons and obtain a score of points, which is higher than a predefined threshold. The total score is the sum of the points that the player has obtained by answering questions.

In particular, while the player is navigating through the virtual world, s/he finds closed doors, which are guarded by dragons as illustrated in the example of Figure 1. A guard dragon poses a question to the player from the domain of geography. If players give a correct answer then they receive full points for this question and the dragon allows them to pass through the door. This will lead them closer to the "book of wisdom".

However, if a player is not certain about the correct answer s/he is allowed to ask the dragon for a "negotiation". In this case, the student is allowed to make a guess for which s/he has to provide a justification. The amount of points that the student is going to receive in the negotiation mode, depends on how close the student's answer is to the correct answer and/or how plausible the reasoning that s/he has used is. If the answer that the student gives is absolutely correct then the dragon allows him/her to proceed through the door. However, if the answer is not completely correct then the system performs error diagnosis. The results of the diagnosis are communicated to the student through the virtual companion agent that appears to help the student.

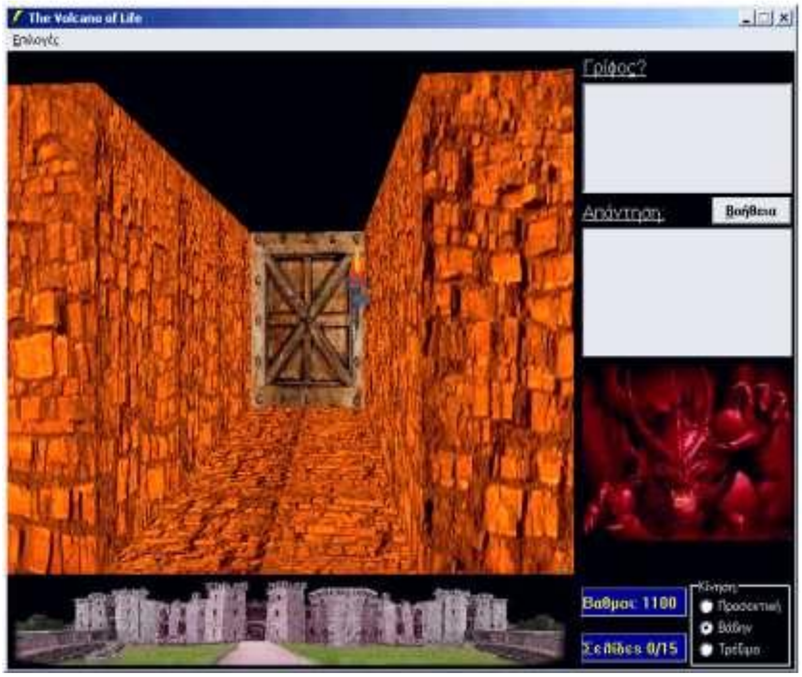


Figure 1. A door guarded by a dragon which asks questions

In the negotiation mode, the student modeling component performs error diagnosis based on a cognitive theory of Human Plausible Reasoning (Collins & Michalski, 1989). At the end of this interaction, possible errors of the student and/or evidence of the student's lack of knowledge on a topic are recorded in the long term student model. For example, the student may have been asked the following question:"What is the capital town/city of the geographical compartment called Achaia (in Greece)?" While being in the

negotiation mode, the student may give an answer such as: "My guess is that Rio is the capital of Achaia. I know that Rio belongs to Achaia; Rio is an important town in Achaia. Therefore, it is likely that Rio is the capital of Achaia." The student's guess may be correct or incorrect; in the case of the example, it is incorrect because Patras is the correct answer. However, the reasoning that s/he has used may reveal whether the student has a good knowledge of geography and whether s/he is able to use it plausibly.

In this sense the game provides an environment where there is opportunity for a negotiating teaching-learning dialogue between the ITS and the students. Collaborative discourse is an issue that has attracted a lot of research energy in recent years (e.g. Moore 2000, Baker 1994). The process of becoming an expert in a certain domain should no longer be solely viewed as the acquisition of a representation of correct knowledge; the knowledge to be acquired should flexibly manage open problems (Andriessen & Sanberg 1999).

If a player does not know the answer at all or has given an incorrect answer without having asked for negotiation, then s/he does not receive any points and may only continue on his/her way if s/he asks for help. In such cases the virtual companion appears and lets the student know what the correct answer is, so that the door may be opened. In addition, the virtual companion suggests to the student to read a particular section of the lesson, which is mostly relevant to the question that s/he did not know how to answer correctly. The appropriate section is selected based on the error diagnosis performed by the student modeling process.

As in other recent educational systems, the existence of the virtual companion in the game has been considered quite important for the promotion of the student's sense of collaboration. For example, Kay (2001) notes that there is a growing acknowledgment of the importance of the learner's social context; therefore, systems are increasingly being designed for learners working in groups of real or simulated peers.

3.3 Student Modeling

Many researchers aim to make their multimedia systems more "intelligent" and adaptive to the learner's demands, abilities and knowledge (Hasebrook & Gremm, 1999). Adaptation and intelligence may be added to educational software if a student modeling component is incorporated. The student modeling component involves the construction of a qualitative representation that accounts for student behaviour in terms of existing background knowledge about the domain and about students learning the domain (Sison & Shimura 1998). *VR-ENGAGE* incorporates a common student modeling component for all four domains involved in the game. For each domain, there are certain categories of error that have been encoded into the system. The system analyzes possible erroneous answers and performs an error diagnosis. Following the diagnosis, the student is given a mark, which is translated to points for his/her total score. The number of points depends on the severity of his/her error. For example, if the system diagnoses that the student has only made a typographic error then s/he is given almost full points for the answer. However, if s/he gives a totally irrelevant answer then s/he receives almost no points for the answer.

VR-ENGAGE holds long-term information about each individual student. Such information is provided by all four domains involved. In particular, each domain contributes both domain-dependent and domain-independent information about each student. For example, a student may consistently make spelling mistakes when s/he is typing answers to questions posed by the game. This is a domain-independent error proneness of the student, which is associated with the degree of the student's carefulness when s/he types answers. This kind of feature is recorded in the student model and is updated constantly in all four domains. Another student feature, which is updated in all four domains, concerns the student's spelling skills. If a student makes spelling mistakes in the game worlds of history, biology, and mathematics, this affects his/her student model regarding his/her spelling skills and reduces his/her score in the spelling game world. However, there are other domain-dependent errors, which may only be made in the corresponding domain game.

Furthermore, the student modeling component of *VR-ENGAGE* examines the correctness of the students' answers in terms of the students' factual knowledge and reasoning used. Information about each student concerning his/her knowledge and reasoning ability is recorded in his/her long-term student model. The long-term student model (Rich 1983) keeps a history record of the student and is updated every time the student answers a question. The long-term student model is used to adapt the presentation of lessons to the particular student's knowledge and possible weaknesses.

The student modeling capabilities needed for the negotiation mode of the game are based on the Human Plausible Reasoning (HPR) theory. This theory formalizes the plausible inferences based on similarities, dissimilarities, generalizations, and specializations that people often use to make plausible guesses about matters that they are not entirely familiar with. Important inference patterns in the theory are called statement transforms. These inferences may lead to either correct or incorrect guesses. In any case, these guesses are plausible.

HPR has been adapted and used previously in intelligent environments for novice users of UNIX (Virvou & Boulay 1999) and for novice users of a Graphical User Interface (Virvou & Kabassi 2001). Moreover, it has been applied in an ITS authoring tool (Virvou 2000). The previous adaptations of HPR in a variety of domains, which were very different from one another and from the present one, shows that HPR could be promising as an underlying reasoning mechanism in educational applications. Therefore, it has been adapted for the particular circumstances of an educational computer game that aims to teach students both the domain of geography and the way to plausibly reason facts.

In the context of the game, HPR has been used to add human-like reasoning abilities to the animated agents that interact with the students. In particular, when a student is asked a question from the domain of geography, HPR is used to perform error diagnosis and, in the case of an error, it will determine how close the erroneous answer was to the correct one. The outcome of the negotiation process is recorded in the long-term student model and is used to adapt the presentation of the teaching material to the individual student. Moreover, if a student asks for negotiation when s/he is expected to give an answer to a question in geography, the system employs the inference mechanism of HPR to evaluate the plausibility of the student's answer in terms of the model of human reasoning that HPR represents. In the negotiation mode, the student is asked to give explicitly the reasoning for the answer that s/he gives and is not certain about.

For example, the question "What is the capital town/city of Achaia?" that was mentioned previously, corresponds to the statement: *capital(Achaia)=Patras*, where *capital* is a descriptor, *Achaia* is an argument and *Patras* is a referent. Based on HPR, the erroneous answer that the student has given in the example: *capital(Achaia)=Rio* corresponds to a similarity referent transform because the two towns belong to Achaia and they are similar in terms of the importance of their harbours. Therefore, the student's answer is considered close to the correct one and the student receives some marks for his/her answer (although not full marks). However, if the student gives a totally irrelevant answer then s/he does not receive any marks at all.

3.4 Evaluation

An evaluation of VR-ENGAGE was conducted so that the educational value of the virtual reality game could be assessed. Evaluation is a crucial part of the design process of educational software, which has to be iterative to a large extent. The educational material was tested on students and refined, then tested again. The cycle continued for as long as necessary (Jones et al. 1993).

One important aspect of the evaluation is to justify the reason why educational software is adopted in the first place, i.e. what the underlying rationale is (Jones et al. 1999). In the case of VR-ENGAGE, one important reason for the integration of educational software with a virtual reality game was the objective of making educational software more engaging and motivating than other forms of software while retaining and even improving the underlying reasoning mechanisms. Therefore, one major part of the evaluation consisted of a comparison between VR-ENGAGE and educational software with a conventional user interface, but with the same underlying reasoning mechanisms as VR-ENGAGE. This part of the evaluation was conducted as an experiment, which involved school children and took place in classrooms while human tutors were present, but was not actively involved in the evaluation.

Another part of the evaluation aimed at finding out the extent to which VR-ENGAGE could be used by children during their leisure time. The underlying rationale of this part of the evaluation was to find out whether VR-ENGAGE could replace computer games with no educational value that children prefer for their entertainment. In this way, the children's game culture could be enriched with educationally beneficial games. This part of the evaluation consisted of an experiment that was conducted in a computer lab where no human teachers were present.

3.4.1 Classroom experiment

School children usually have a preconception of educational means as being totally different from entertainment. In this respect, the first experiment aimed at finding out how school children would react to an educational game in the settings of a real classroom where an entertaining aspect of education would be rather unexpected. Therefore, the first experiment took place in a school-classroom.

The experiment involved a class of 16 school children of 11-12 years old and four human teachers that taught history, biology, spelling, and mathematics to this class. The class was divided into two groups of 8 children. The division of children into two groups was based on the human teachers' selection of children in such a way that the two groups had the same distribution of students having good, mediocre, and bad grades on average in the four domains involved.

The first group was given VR-ENGAGE to work with. The second group was given educational software, which consisted of the underlying reasoning mechanisms of VR-ENGAGE, but had a simple user interface with no game. Both groups were told that they could use the software for as long as they wished. Moreover, both groups were asked by their human teachers to complete a test using the software. This test was given to them as an informal assignment. In the environment of VR-ENGAGE this meant that they had to open all doors in a virtual world and complete their total score. In the environment of the conventional educational software they had to answer a set of questions, which were displayed to them in plain text and context. The rules for the students' receiving their marks through the software were the same for both groups. Finally, both groups were supervised by two computer assistants who helped them with their interaction with the computer.

While children of both groups were using their respective software, their actions and scores were recorded in protocols. After the children of both groups had finished using the programs, the scores they had obtained and the errors they had made were collected in their user protocols, since all their actions were given to their school teachers. Then the school teachers were asked to repeat the questions where students of both groups had originally given erroneous answers while they used the software. This test would reveal the degree to which students had learnt from their mistakes while they used the software. The players of VR-ENGAGE remembered the correct answers to a slightly higher extent than the other group of students. This showed that VR-ENGAGE had achieved its aim of being at least as effective as conventional educational software in the learning outcomes and was in fact slightly better in this respect.

Another aspect that was tested in this experiment was the amount of time that children had spent using the educational software they were given. On average, the students who had used VR-ENGAGE had spent more time with the system than the students who had interacted with the conventional educational game. This was partly due to the fact that there was more to explore in the game, therefore, students needed more time to complete the game. However, it was also partly due to the fact that the players of VR-ENGAGE had spent more time reading the lessons that were shown to them than the other group of students. This showed that VR-ENGAGE was indeed more engaging.

Finally, both groups of students were interviewed concerning the software they had used. These interviews revealed that the players of VR-ENGAGE were fascinated by the idea of a game in the classroom and they were certainly more enthusiastic about the software that they had used than the other group of students.

3.4.2 Lab Experiment

The second experiment took place in a computer lab. The main aim of this experiment was the assessment of the entertaining aspect of VR-ENGAGE. The idea behind this was to find out whether VR-ENGAGE could be competitive to non-educational computer games in terms of entertainment. If this was the case then VR-ENGAGE would have the advantage of being favoured by children in their leisure time. In this way, the educational game could be used both at work time and leisure time and thus would have a greater educational impact on children.

This experiment involved 20 children of 11-12 years old who were all from the same school and class. These children were asked to try this new game and their reactions were observed and analyzed. In addition, the children were interviewed after they had completed their interaction with the game. Unlike the first experiment, there were no human teachers present in this experiment and the game was not associated with school assignments. Therefore, the experiment took place in a building away from the children's school. However, as in the previous experiment, there were two computer assistants to help children with the use of the game.

The results from this experiment were quite different from the first one. Since children were not given the game to work with it as an assignment, they considered it merely as a game similar to the commercial games they were familiar with. Therefore, their judgment of it focused on the game environment. Most of them (73%) pointed out that it would be a better game if it had more virtual objects, more background sounds, and more adventure. This was due to the fact that most of them were familiar with commercial virtual reality games; therefore, they compared VR-ENGAGE with them and had higher expectations in this respect.

Quite a lot of the children (46%) commented on the educational aspect of the game and they said that they found the game quite informative with interesting subjects. Another 35% did not make any comment about the educational content of the game. Finally, 19% of them said that they were annoyed by the fact that the game reminded them of the school syllabus. However, most of the children (even those who did not like the school syllabus) remembered to a large degree what they had learnt from the game in the domain of the lessons.

3.5 Conclusion

Educational applications may benefit from the technology of virtual reality games, which can increase the students' engagement and motivation. However, one major problem of this kind of educational application is the construction of the game itself and the connection of pedagogy and student adaptivity with the story of the game. The approach taken in VR-ENGAGE that was described in this chapter offers a solution to this problem. VR-ENGAGE employs animated agents who take part in the story of the game by asking questions, and by providing adaptive advice and collaboration to the student. The tutoring adaptivity to the student's needs is provided by a domain-independent reasoning mechanism that performs error diagnosis and records the student's progress in the student model. The system has been evaluated and the results have shown that school children would be quite happy to work with a computer game, which represents a more amusing teaching fashion than that of conventional educational software. Moreover, the educational benefits of the game are at least as good as those of conventional educational software. However, the experiments also revealed that children are quite familiar with commercial games and therefore have high expectations for the game environment.

4. Combining Intelligent Tutoring Systems and Virtual Reality Games

The process of learning is a very complex cognitive task that can be very imposing on students since it requires a lot of effort from them. Consequently, they need a lot of motivation to cope with it. In view of this, it is within the benefit of education to create educational software that is interesting and stimulating for students. On the other hand, there is a fast growing area of computer technology, that of computer games, that is extremely appealing to children and adolescents. Indeed, anyone who interacts with children and adolescents in every-day life can easily observe that they like computer games. This is also a view that has been supported by many researchers who have conducted empirical studies (e.g. Mumtaz 2001). Thus the computer games technology could be used to render educational software more motivating and engaging. In this respect, the difficult process of learning could become more amusing.

Indeed, there are many researchers and educators that advocate the use of software games for the purposes of education. Papert (1993) notes that software games teach children that some forms of learning are fast-paced, immensely compelling and rewarding whereas by comparison school strikes many young people as slow and boring. Boyle (1997) points out, that games can produce engagement and delight in learning; they thus offer a powerful format for educational environments. Moreover, there are studies that have shown that the use of carefully selected computer games may improve thinking (Aliya 2002). As a result, many researchers have developed games for educational purposes (e.g. Amory et al. 1998; Conati & Zhou 2002). At the same time, a story can be viewed as a set of concepts linked together by a narrative (Gerdt, Kommers, Looi & Sutinen 2001).

However, the attempts to create educational games have not reached schools yet. There are several reasons for this. First, not all educators and parents are convinced that educational games can be beneficial to students. Second there are criticisms about the quality of the existing educational games. For example, Brody (1993) points out that the marriage of education and game-like entertainment has produced some not-very-educational games and some not very-entertaining learning activities.

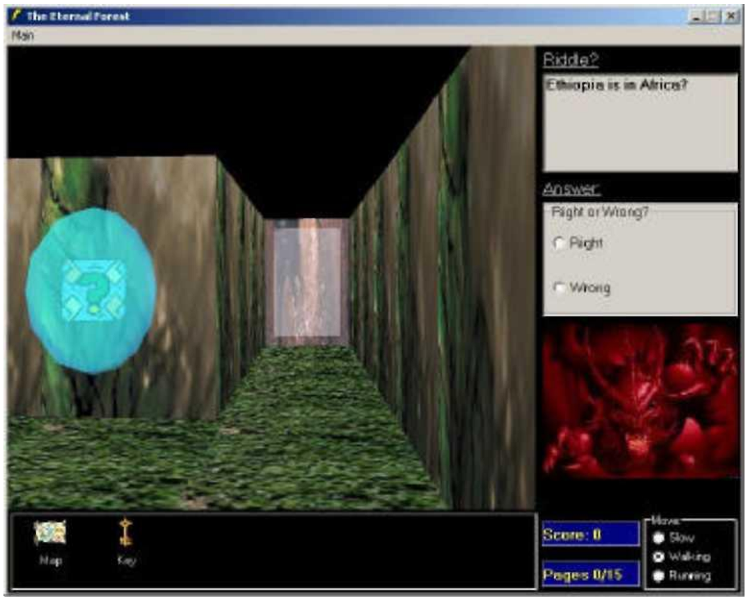
Given the motivational advantages of software games as well as the criticisms that have been made on educational games, there has to be further investigation on the advantages and limitations of software games for education. Such investigation may lead to useful guidelines for the design of effective educational software games. Indeed, educational software games should be designed in such a way that they are educationally beneficial for all students, even those that are not familiar with computer games.

In view of the above, an evaluation study on the virtual reality educational game VR-ENGAGE was conducted. VR-ENGAGE teaches students geography. VR-ENGAGE aims to increase students' engagement by providing a popular and motivating virtual reality environment. In this way, it aims at being more effective in teaching students than other educational software and traditional media of education. The main focus of the research described in this chapter is to measure the educational effectiveness of an educational VR-game as compared to educational software that does not incorporate the gaming aspect.

The main objective of this comparison is to find out whether the gaming environment may improve education.

4.1 Evaluation aims and experiment

The evaluation that was conducted on VR-ENGAGE focused primarily on evaluating the educational effectiveness of the gaming aspect of the educational software. One could argue that the greatest advantage of games is the motivation provided to students by the game environment whereas one possible disadvantage for the learning process could be the students' distraction by this game environment. However, even the motivational advantage of educational games may be questioned since in a classroom there may be students who do not like games or students who find it difficult to navigate through the virtual world and thus may not be able to benefit to the full from the educational content of the software.



The Eternal Forest

Riddle?

Ethiopia is in Africa?

Answer

Right or Wrong?

○ Right

○ Wrong

Score: 0

Pages 0/15

Move

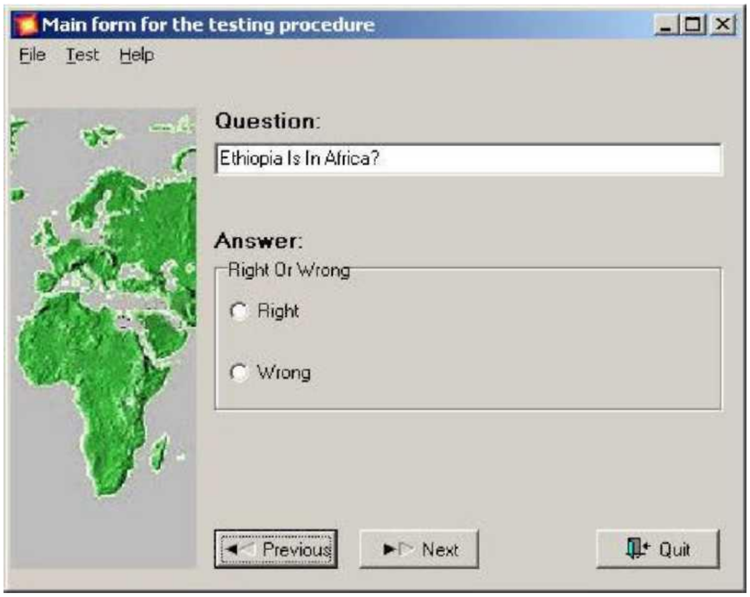
● Slow

● Walking

● Running

Fig. 2: A question posed

A common theme found in the literature for educational games, both electronic and non-electronic, is that these games and software are considered successful only if they are as effective as traditional classroom education (Mc Grenere 1996). However, this kind of comparison implies that games are not meant to be included in traditional classroom education but rather they are meant to replace it. In our view, games should be used to supplement traditional classroom education. Human teachers still have more abilities in explaining domain issues and diagnosing students' problems than any kind of software irrespective of its sophistication. This view is reinforced by empirical studies that show that no matter how successful an ITS may be, students still prefer the human teacher (e.g. Tsiriga & Virvou 2004).



Question:
Ethiopia Is In Africa?

Fig.3: A question posed in Simple UI application

Therefore, in the present evaluation, conducting a comparison between human teaching and tutoring through the game was not considered. Thus, to find out whether the game environment is in fact motivating and educationally beneficial to students and not distractive, we conducted an experiment where the game-ITS could be compared to an ITS that had a conventional user interface without any virtual reality components. Both educational software applications had the same underlying reasoning mechanisms with respect to student modeling as well as the same help and theory functionalities. The main difference between the two educational software applications (game-ITS and ITS with a simple user interface) was that one had a gaming approach whereas the other one did not. In fact, the software with the simple user interface had a hypertext display of domain theory and exercises that were communicated to students through forms, dialogue boxes, buttons, drop-down menus, etc. However, these exercises were not part of any story as in the gaming approach. Moreover, there was no virtual reality environment and no animated-speaking agents. For example, the way that the exam question "Is Ethiopia in Africa?" is presented to the user of VR-ENGAGE is illustrated in Figure 2, and the way the same question is presented to the user of the software with a simple user interface is illustrated in Figure 3.

The evaluation experiment was connected to the underlying rationale of the educational game, which was to engage students in learning the domain concepts that were taught to them. Thus, the aim of the experiment was to find out whether the educational game was in fact more motivating while it was at least as effective with respect to students' learning as the educational software with the conventional interface. Moreover, one of the primary aims of the experiment was to reveal the degree of educational effectiveness (if any) for students whose performance was considered good, mediocre, or poor respectively from their human teachers.

This experiment took place in classrooms. School children usually have a preconception of educational means as being totally different from entertainment. In this respect, the experiment aimed at finding out how school children would react to an educational game in the settings of a real classroom where an entertaining aspect of education would be rather unexpected. This was the main reason why the experiment took place in school-classrooms. Human tutors were present and they were asked to observe their students while they interacted with the computer but were not actively involved in the evaluation. There were, however, lab assistants that helped students with the interaction with the game if the students needed help.

The experiment consisted of four parts. All four parts were similarly set up and involved a comparison between VR-ENGAGE and the ITS with the simple User Interface (UI) in terms of the educational effectiveness and motivation. All four parts were conducted in parallel. All of the children who participated in all four parts of the experiment were of 9-10 years old and attended the fourth grade of elementary schools in Greece. They had been taught the same syllabus on geography and they had a similar background on the use of computers. More specifically, all of them were computer-literate and had been trained in their respective schools in the use of Windows, the Internet and other popular software packages such as word-processors, etc.

Each part of the experiment was different from the other parts in the type of the school-children that participated in it. Specifically, the first part of the experiment involved all the students of 5 classes of school children of the fourth grade of an elementary school, 90 children altogether, and their respective geography teachers.

For the second, third, and fourth parts of the experiment, the students who participated, were also of 9-10 years old (fourth grade of the elementary school), but the selection of them was based on the mark that these students had received from their respective human teachers in geography in the previous term. The term-marks that the students of the fourth grade of elementary schools receive usually range from A to C. "A" is given to students with good performance, "B" is given to students with mediocre performance and "C" is given to students with poor performance. The participants were selected from the total students of 7 classes (127 students) of the fourth grade of an elementary school, which was different from the one that had been used for the first part of the experiment. From the total of the 127 students, 30 students were selected for the second part of the experiment based on the criterion of their having received the mark "A" in the previous term, 30 students were selected for the third part of the experiment based on the criterion of their having received the mark "B" in the previous term and finally 30 students were selected for the third part of the experiment based on the criterion of their having received the mark "C" in the previous term. The number 30 was selected so that we could have equal numbers of students participating in each of the remaining three parts of the experiment.

Variable	**VR-ENGAGE Sub-Group (n=45)**		**Simple ITS Sub-Group (n=45)**		**Tv; Cv**
	Mean Value	*Standard Deviation*	*Mean Value*	*Standard Deviation*	
Pre-test errors of students of previously **poor academic performance** (Between 0 and 100)	61.40	7.70	60.40	5.84	Tv = 0.53; Cv = 2.05
Pre-test errors of students of previously **mediocre academic performance** (Between 0 and 100)	35.67	9.43	32.87	8.62	Tv = 1.27; Cv = 2.05
Pre-test errors of students of previously **good academic performance** (Between 0 and 100)	14.20	4.36	13.40	3.40	Tv = 0.56; Cv = 2.05
Post-test errors of students of previously **poor academic performance** (Between 0 and 100)	31.33		41.33		
Post-test errors of students of previously **mediocre academic performance** (Between 0 and 100)	21.93		22.47		
Post-test errors of students of previously **good academic performance** (Between 0 and 100)	9.40		9.00		
Improvement percentage on mistakes between pre-test and post-test for students of previously **poor academic performance**	48.97%	10.94	31.57%	7.72	Tv = 4.86; Cv = 2.05
Improvement percentage on mistakes between pre-test and post-test for students of previously **mediocre academic performance**	38.50%	10.06	31.64%	5.08	Tv = 2.28; Cv = 2.05
Improvement percentage on mistakes between pre-test and post-test for students of previously **good academic performance**	33.80%	9.66	32.84%	9.67	Tv = 0.27; Cv = 2.05

Each group of children that were selected to participate in each part of the experiment was randomly divided into two independent sub-groups of the same number of children. Thus, there were two independent sub-groups of 45 students for the first part of the experiment, two independent sub-groups of 15 students for the second part of the experiment, two independent sub-groups of 15 students for the third part of the experiment and two independent sub-groups of 15 students for the fourth part of the experiment. The first sub-group of each group would use VR-ENGAGE and the second sub-group of each group would use the ITS with the simple UI (User Interface).

Before using their respective version of educational software, the students of both sub-groups of all the groups, were asked to work on a pre-test using paper and pencil. This pre-test was an ordinary classroom test in which every student had to answer 100 questions by filling in a test paper. The students' performance in the pre-test was compared to the students' performance in a post-test that was given to the students after the use of their respective software. The post-test was of a similar level of difficulty as the pre-test and consisted of the same number of questions (100). The comparison of students' results in the pre-test and the post-test was used to draw conclusions about the educational effectiveness of VR-ENGAGE as compared to the simple ITS. In particular, the school teachers were asked to count the number of erroneous answers of each student in the pre-test and the post-test.

The students' pre-test and post-test performance was compared using t-test statistics. In particular, the educational effect of VR-ENGAGE was compared to that of the simple ITS by comparing the number of mistakes of the students of the VR-ENGAGE sub-groups with the number of mistakes of the students of the respective sub-groups that had used the software with the simple UI. It was expected that the number of mistakes that the students would make after the use of either of the software versions would be reduced in comparison with the pre-test because both applications provided quite sophisticated tutoring from the adaptive presentation of the theory and the reasoning and student modeling of both applications. However, the post-test would reveal the degree to which students that had used VR-ENGAGE exhibited greater or less improvement than those who had used the ITS with the standardized user interface.

After the post-test, all the students who had participated in the experiment were also interviewed about their experiences using their respective educational software. Moreover, the teachers of the school classes who had participated in the experiment were also interviewed concerning their students' performance and behavior during the experiment. Teachers were also asked to give their comments on their students' performance on the pre-tests and post-tests.

4.2 Evaluation results

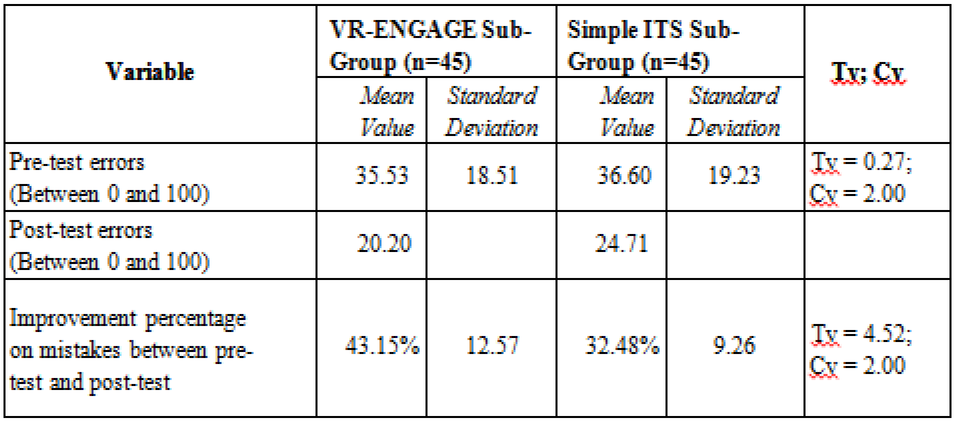
4.2.1 First part of the evaluation

As mentioned earlier, the first part of the evaluation involved 90 students of the fourth grade of an elementary school who were separated into two sub-groups of 45 children that would use VR-ENGAGE and the simple ITS respectively. The results showed a greater improvement of the VR-ENGAGE users over the users of the other software. In particular, in the post-test, the players of VR-ENGAGE made 43.15% less mistakes than in the pre-test. The other sub-group of students that had used the simple ITS resulted in 32.48% less mistakes of their answers in total, as compared to the pre-test. Thus the players of VR-ENGAGE resulted in a higher improvement of 10.67% in terms of their mistakes than the users of the simple ITS. This showed that VR-ENGAGE had achieved its aim of being at least as effective as conventional educational software in the learning outcomes and was in fact better in this respect.

In more detail, the total questions that were asked to the total number of students of each sub-group were 4500: *45 students x 100 questions = 4500 questions.* In total, the students who had worked with VR-ENGAGE failed during the pre-test in *1599* questions. The mean value of errors per student was 35.53 and the standard deviation 18.51. In total, the students who had worked with the simple ITS failed during the pre-test in *1647* questions. The mean value of errors per student was 36.6 and the standard deviation 19.23. An initial analysis concerning the comparison of the number of mistakes of each sub-group in the pre-test was not statistically significant showing that the two sub-groups had similar background knowledge on geography. Indeed, there was a t-test performed for the pre-test of the students of the two sub-groups. The null hypothesis (H0) was that there was no difference between the mistakes of the two sub-groups and the research hypothesis, (H1) was that there was a difference between the mistakes of the two sub-groups. The t-value result of 0.27 was smaller than its critical value 2.00. This showed that the students of the two sub-groups had similar prior knowledge of the domain of geography.

Then, after the students had completed their interactions with the two applications, they were given the post-test. The players of VR-ENGAGE made 909 mistakes in the post-test. This number of mistakes as compared to the 1599 of the pre-test constituted an improvement of 43.15%. The students who had worked with the other educational software made 1112 mistakes, which constituted an improvement of 32.48% in the number of erroneous answers. The second statistical analysis concerned the improvement on the number of mistakes for each sub-group between the pre-test and the post-test. There was a t-test performed for the improvement on the number of mistakes for each sub-group. The null hypothesis (H0) that there was no difference in the improvement on the number of mistakes for the two sub-groups and the research hypothesis, (H1) was that there was difference in the improvement on the number of mistakes for the two sub-groups. The t-value result of 4.52 was significantly greater than its critical value 2.00. This showed that the difference was statistically significant for the first sub-group in comparison with the difference of the second sub-group, leading to the result that students who had used VR-ENGAGE had a higher educational benefit than the students who had used the simple ITS.

Table 1 illustrates the mean value of the errors made during the pre-test, the mean value of the errors made during the post-test and the mean value of the percentage improvement on mistakes between the two tests, for both of the sub-groups, the first who had used VR-ENGAGE and the other who had used the simple ITS. Additionally, it includes the respective results of the t-tests. These are the results of the first t-test, after the pre-test, which show that there was no significant difference on the background knowledge on geography for the two sub-groups, and the results of the second t-test, after the post-test, which show that there was a greater improvement on the number of mistakes for the sub-group of VR-ENGAGE users over the other group. These results involve the standard deviations, the T values and the Critical values of the t-tests.



Variable	VR-ENGAGE Sub-Group (n=45)		Simple ITS Sub-Group (n=45)		Tx; Cx
	Mean Value	*Standard Deviation*	*Mean Value*	*Standard Deviation*	
Pre-test errors (Between 0 and 100)	35.53	18.51	36.60	19.23	Tx = 0.27; Cx = 2.00
Post-test errors (Between 0 and 100)	20.20		24.71		
Improvement percentage on mistakes between pre-test and post-test	43.15%	12.57	32.48%	9.26	Tx = 4.52; Cx = 2.00

Table 1: Results of the analysis of the students' mistakes.

In the above t-tests, the t-value of each t-test is calculated by performing a t-test for independent samples for each of the null and research hypotheses (Voelker, 2001). The critical value for each t-test is the value taken from Table T for a two-tailed research hypothesis depending on the sample number. The t-test results show that there is a statistically significant difference for the two samples in favor of the educational benefits of VR-ENGAGE for the two sub-groups.

4.2.2 Second, third and fourth part of the evaluation

The second, third and fourth part of the evaluation involved 90 students of the fourth grade of an elementary school, which was different from the first part of the evaluation, separated into three groups of 30 children having poor, mediocre, and good performance in geography. Every group of 30 children was then separated into two sub-groups of 15 children that would use VR-ENGAGE and the simple ITS respectively.

In the post-test, the VR-ENGAGE students who used to be poor and mediocre performers made 48.97% and 38.5% less mistakes respectively than in the pre-test. The students of poor and mediocre academic performance that had used the simple ITS resulted in 31.57% and 31.64% less mistakes respectively, as compared to the pre-test. Thus, the students of previous poor and mediocre academic performance that had used VR-ENGAGE resulted in a higher improvement of 17.4% and 6.86% respectively in terms of their mistakes than the students of the respective groups that had used the other application. Moreover, the good students who had used VR-ENGAGE resulted in a 33.8% improvement while the good students who had used the other application resulted in a 32.84% improvement. This showed, that for the two sub-groups of the good students there was also a small difference in making fewer mistakes when using VR-ENGAGE, but this difference was not statistically significant. The overall results showed that VR-ENGAGE had achieved its aim of being at least as effective as a non-game ITS in the learning outcomes and was in fact better in this respect for the categories of students of poor and mediocre academic performance.

To be more specific, the total questions that were asked to the total number of students of each sub-group were 1500: *15 students x 100 questions = 1500 questions.* In total, the students of previous poor, mediocre, and good academic performance who had worked with VR-ENGAGE failed during the pre-test in *921, 535, and 213* questions respectively. In total, the students of previous poor, mediocre, and good academic performance who worked with the conventional educational software failed during the pre-test in *906, 493, and 201* questions respectively. An initial analysis concerning the number of mistakes of each sub-group in the pre-test involved 3 t-tests for the students of poor, mediocre, and good academic performance respectively. The null hypothesis (H0) was that there was no difference between the mistakes of the VR-ENGAGE group and the simple ITS group and the research hypothesis (H_1) was that there was a difference between the mistakes of the two groups. The t-value results of 0.53 for the students of poor previous performance, 1.27 for the students of mediocre previous performance, and 0.56 for the good students were smaller than their critical values of 2.05, 2.05 and 2.05 respectively. This led to the acceptance of the null hypothesis (H_0), which showed that the two groups (VR-ENGAGE and non-game students) of each of the three categories of students had similar prior knowledge of the domain of geography.

Then, after the students had completed their interactions with the two applications and answered the questions of the post-test, the following results were found. The players of VR-ENGAGE (poor, mediocre and good previous performance) made 470, 329, and 141 mistakes respectively. These mistakes compared to the *921, 535, and 213* in the pre-test constituted an improvement of 48.97%, 38.5%, and 33.8% respectively. The students who had worked with the non-game ITS made 620, 337, and 135 mistakes, which constituted an improvement of 31.57%, 31.64%, and 32.84% respectively in the number of answers failed.

The statistical analysis which took place after the post-test, concerned the improvement on the number of mistakes for each sub-group between the pre-test and the post-test. There were 3 t-tests performed for the improvement on the number of mistakes for each sub-group. The null hypothesis (H0) that there was no difference in the improvement on the number of mistakes for the two sub-groups and the research hypothesis, (H_1) was that there was difference in the improvement on the number of mistakes for the two sub-groups. The t-value results of 4.86 for the poor students and 2.28 for the average students were significantly greater than their critical values 2.05 and 2.05 respectively. This showed that the difference was statistically significant for the first sub-group in comparison with the difference of the second group, leading to the result that students of previous poor and average performance who had used VR-ENGAGE benefited more than the non-game users since they made less mistakes. The t-value result of 0.27 for the good students was significantly smaller than its critical values 2.05. This showed that the difference was not statistically significant for the first sub-group in comparison with the difference of the second group, leading to the result that good students who had used VR-ENGAGE benefited in a similar way with the good students that had used the non-game ITS.

The mean values of the pre-tests and the post-tests of the students and the results of the above t-tests are summarized in Table 2. In particular, Table 2 illustrates the mean values of the errors made during the pre-tests, the mean value of the errors made during the post-tests, and the mean value of the percentage improvement on mistakes between the test pairs, for all of the sub-groups pairs, the first who had used VR-ENGAGE and the other who had used the simple UI application. Additionally, it includes the respective results of the t-tests. These are the results of the three t-tests, after the pre-tests, to find out any difference on the background knowledge on geography for the sub-group pairs, and the results of the three t-tests, after the post-tests, to find out any difference in the improvement on the number of mistakes for the sub-groups pairs. These results involve the standard deviations, the T values and the Critical values of the t-tests.

In the above t-tests the t-value of each t-test is calculated by performing a t-test for independent samples for each of the null and research hypotheses. The critical value for each t-test is the value taken from Table T for a two-tailed research hypothesis depending on the sample number. The t-test results show that there is a statistically significant difference for the two samples in favor of the educational benefits of VR-ENGAGE for the sub-group pairs of poor and mediocre students.

4.3 Interviews of students and teachers

All of the students who had participated in the experiment were interviewed concerning the software they had used. These interviews revealed that the players of VR-ENGAGE were fascinated by the idea of a game in the classroom and they were certainly more enthusiastic about the software that they had used than the other group of students. However, despite the fact that all students had liked the game in the context of their classroom work, a large part of them criticized the game in comparison with other commercial games and said that they would like VR-ENGAGE to have more virtual objects, a more sophisticated environment, more adventure, and more action. Students who were experienced game-players mainly made these comments. Such students had high expectations from VR-games.

As for the teachers, most were particularly impressed by the effect that the game had on students who were previously poor performers on geography. This category of students included quite a lot of those students who the teachers thought were not easily disciplined in class. The teachers reported that these students seemed absolutely absorbed by the game environment and kept working in peace and quiet without talking to anyone and without disturbing anyone. To some extent, this comment was also made for the same category of students who were given the other educational application to work with. In general, the teachers thought that the use of computers was very good for the students who they used to consider as non-disciplined in class. However, they thought that those who had used the game seemed so immersed that their behavior in class had changed completely and they had appeared to be very satisfied and interested in the educational content. The teachers were very happy with their students' performance on the post-test and most of them said that they would certainly wish to include educational games of this kind in the classroom. Some of them suggested that they might even use the game on their own laptop in classroom and show the action of the game through a projector to their whole class so that the whole class could participate in a single game play.

4.4 Discussion and conclusions

The results from the evaluation showed that students would benefit from educational games in classrooms and would be quite happy to work with a computer game, which represents a more amusing teaching fashion than that of conventional educational software. Moreover, one important finding that should be noted from the t-tests of the second, third, and fourth part of the evaluation is that when the subgroups of students who previously had good, average, and poor performance respectively and were compared separately, it was revealed that the subgroup of students who used to be poor performers had benefited the most from the game environment whereas the subgroup of good students had benefited the least from the game environment. This finding may be explained by the fact that good students usually perform well under any circumstances, whereas the rest of the students and particularly those who perform poorly may do so because of lack of interest in their lessons and tests. Thus, students with little interest in their courses may benefit from extra motivating environments such as those of VR-educational games. This finding was also confirmed by the teachers' impression about the students who they thought they were not easily disciplined in class. These students were reported to have been absorbed by the game and they did not seem willing to take time out to talk to other students or to try to cheat on the test and so on. This is probably due to the fact that games are able to attract the attention of students who do not easily concentrate on their assignments due to boredom or other distractions.

The students who used to have good academic performance did not have any significant difference in their improvement through the use of the game or the use of the other software. However, one important finding is that the performance of previously good students has not deteriorated by the use of the educational game due to possible usability problems in the VR-environment or their possible distraction through the game. It seems that good academic performers can keep their good academic record despite the fact that some of them were not experienced virtual reality game players. From the interviews it was evident that they too had enjoyed the learning experience through the game to a large extent. However, it must be noted that during the experiment all students had as much help as they needed from lab instructors concerning their interaction with the VR environment of the game. If the students had used the software on their own at home, then perhaps they might have had more usability problems, especially those who were not sufficiently experienced in virtual reality game playing. Such problems might have resulted in less good educational results. Therefore, in future versions of VR-ENGAGE, the goal will be to improve the usability of the game environment and incorporate more on-line help. Finally, the game environment of the educational game has to be competitive with commercial games to attract a high degree of interest from students. This is so because children are quite familiar with commercial games and therefore they have high expectations of game environments.

5. VR-INTEGATE: A knowledge-based authoring tool for Intelligent Tutoring Systems integrated in a virtual reality game.

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 re-usability. 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 (1999) 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 re-usability and cost effectiveness is the development of authoring tools that may be used for the creation of many computer game software applications. In this chapter, an authoring tool that may be used by instructors to create virtual reality games for education is described. The authoring tool is called VR-INTEGATE, which stands for Virtual Reality- INTelligent Game Authoring Tuition Environment. VR-INTEGATE provides an authoring environment to instructors who wish to create ITSs that operate through a virtual reality game. The concept of the game is used so that the ITSs may become more motivating and engaging. Moreover, the ITSs are able to provide diagnostic reasoning concerning the students' answers to questions about the domain being taught. In addition to the objective of cost-effectiveness in the design of multiple applications, VR-INTEGATE 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.

5.1 Operation of the authoring mode

The initial input to the authoring tool is given by a human tutor who is acting as an author. The initial input consists of a description of domain knowledge in terms of hierarchies. Therefore, the author has to decide what the main concepts of the lesson are, that may be represented in hierarchies. Then s/he may create hierarchies by giving data to a dialogue box of the system. First the author has to declare a description of the nodes of a hierarchy and then s/he has to declare the nodes' attributes. Finally s/he has to enter the actual data, both for the nodes and their attributes. For example, in the creation of a lesson in geography an author may decide to declare the nodes "Continent", "Country", "County". Then s/he may insert attributes for the nodes, for example the capital city of a country. Finally s/he may give data, such as Continent: Europe, Country: Greece, Capital of Greece: Athens etc. After this input has been given, the tool constructs a knowledge base concerning the specific domain in the form of hierarchies. Finally, the authoring tool may automatically construct tests that consist of questions relating to the factual knowledge of the domain. All tests are part of the story of the virtual reality game.

The student modeling component examines the correctness of the students' answers in terms of the students' factual knowledge and reasoning that they have used. The diagnostic process is based on previous research (Virvou 1999; Virvou & Du Boulay 1999) in error diagnosis which explored the utility of a formal theory of Human Plausible Reasoning (Collins & Michalski 1989) in the context of an Intelligent Help System for novice users of operating systems. This research showed that Human Plausible Reasoning could be a helpful tool when employed for error diagnosis. The Human Plausible Reasoning Theory (henceforth referred to as HPR) was originally constructed to provide a formal model of the reasoning that people use to reach some conclusions about questions for which they do not know the immediate answer. Starting from a question asked to a person, the theory tries to model the inferences made, based on similarities, dissimilarities, generalizations, and specializations that people often use to make plausible guesses. These guesses may be correct as well as incorrect. For the purposes of error diagnosis, we exploit the fact that the human plausible reasoning that a student may have used may have led him/her to make an error. The diagnostic process makes use of domain knowledge represented in "is a" and "is part" hierarchies.

5.2 Design Issues of VR-INTEGATE

The tool offers multiple virtual reality game environments and the basic story of these games. It also incorporates a learner modeling 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, colors, and noise play an important role in the attractiveness of a game (Griffiths 1995, Wolfson & Case 2000). 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 favor 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 medieval 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.

5.3 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:

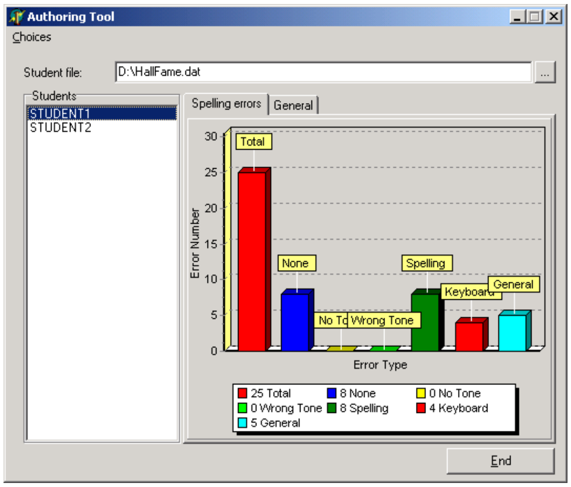
1. Multiple choice questions
2. Fill-in the blank space
3. True/False questions
4. Questions where the student has to type in the answer

Each type of question is associated with certain facilities that VR-INTEGATE 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 answer 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.

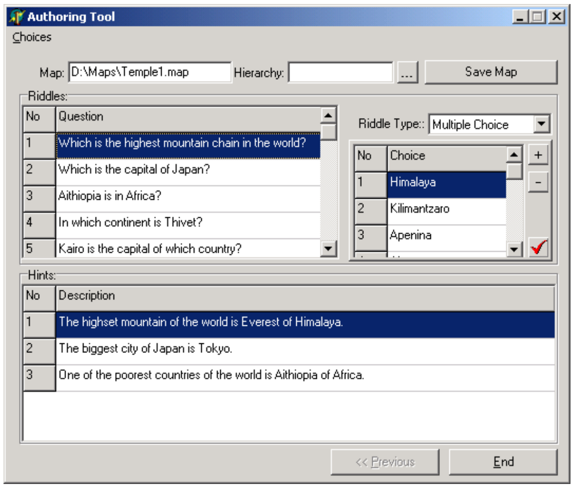
The explanation of a mistake may be difficult for the system to spot. Hollnagel (1991 & 1993) makes an important distinction between the underlying cause or genotype of an error and the observable manifestation or phenotype of the error. In addition, ambiguity may be a problem, since there may be different explanations of observed incorrect users' actions (Mitrovic et al 1996). 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.



Student file: D:\HallFame.dat

Students
STUDENT1
STUDENT2

Figure 1. Example of percentages of different types of students' errors



Map: D:\Maps\Temple1.map Hierarchy:

No	Question
1	Which is the highest mountain chain in the world?
2	Which is the capital of Japan?
3	Aithiopia is in Africa?
4	In which continent is Thivet?
5	Kairo is the capital of which country?

Riddle Type:: Multiple Choice

No	Choice
1	Himalaya
2	Kilimantzaro
3	Apenina

No	Description
1	The highset mountain of the world is Everest of Himalaya.
2	The biggest city of Japan is Tokyo.
3	One of the poorest countries of the world is Aithiopia of Africa.

Figure 2. Example of instruction interaction with VR-INTEGATE.

VR-INTEGATE 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 category of explanation. If a student is prone to typing errors then this cause may be favored in cases of ambiguity.

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 VR-INTEGATE 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).

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.

5.4 Resulting Educational Game Application

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

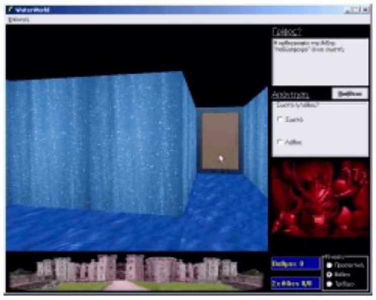


Figure 3. Virtual Water World

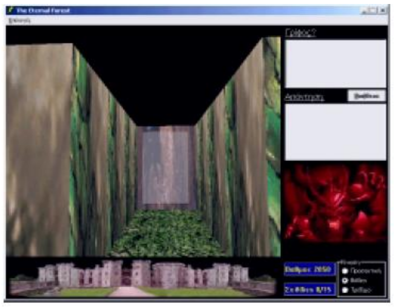


Figure 4. Virtual woods world

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".

There are many virtual worlds that may be generated through VR-INTEGATE. Instructors may have explicitly connected a test to a specific world. In any other case, VR-INTEGATE selects the least frequently used world to connect 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.

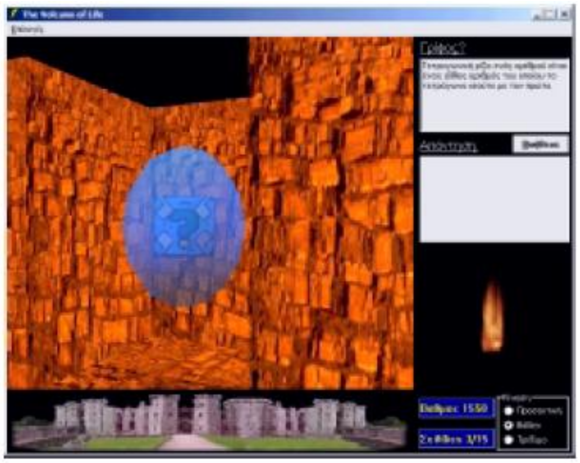


Figure 5. A theory hint in the form of a blue ball in a virtual volcano world

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 (Vanlehn et al 1994) argue that students can improve their learning in collaboration with a simulated student. As the simulated student can be simultaneously an expert and a co-learner, it can scaffold and guide the human's learning in subtle ways.

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.

5.5. 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 (Chou 1999).

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. Jones et al 1999, Squires & Preece 1996 & 1997). One such framework outlines three dimensions to evaluate: (i) context; (ii) interactions; and (iii) attitudes and outcomes (Jones et al 1999). 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 VR-INTEGATE involved both instructors and students and was conducted in two different phases. During 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.

During the first phase, six instructors were involved. Half of them were school teachers in primary schools and were asked to prepare lessons and tests in geography using VR-INTEGATE; 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 VR-INTEGATE 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 VR-INTEGATE gave to them. For example, they did not attach explanations to errors or they did not any insert a bug list. The rest of the instructors made use of all the facilities of VR-INTEGATE. All six instructors made use of the hint facilities.

When interviewed, the authors confirmed that VR-INTEGATE had a user-friendly interface and stated that they were quite satisfied with the facilities that VR-INTEGATE could provide for adding content. Among the instructors who did not make use of all the facilities of VR-INTEGATE, 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, five 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 six 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 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 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 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 in 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.

5.6 Conclusions

This chapter 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 VR-INTEGATE 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.

Virtual reality games may well be used for educational purposes since they guarantee the students' engagement into the educational application. One problem of such applications is the construction of the game itself and the connection of pedagogy with the story of the game. The authoring tool that was described in this chapter offers a solution to this problem. Instructors may author their own educational application, which will result in a knowledge-based educational game. In this way, game environments may be re-usable and enhanced with the domain-independent reasoning of an ITS that performs error diagnosis and records the student's progress in the student model. Then the system may adapt advice to the particular student's needs. In addition the game employs animated agents who take part in the story of the game by asking questions, providing advice, instruction, and collaboration to the student.

6. Applying Cognitive Psychology to Student Modeling

Educational software is a special kind of software that aims at facilitating the difficult cognitive process of students' learning. In this respect, educational software has to combine many qualities to make the most of the interactive means provided by computers and be educationally beneficial. Such qualities include attractive multimedia presentations, the individualization of tutoring, reasoning abilities, user-friendly interfaces, etc. To achieve all these qualities there has to be a combination of educational software technologies and ideas, such as the combination of multimedia educational software technology with the underlying reasoning mechanisms of Intelligent Tutoring Systems (ITSs). Moreover, the coexistence of these qualities demands a lot of effort during the software engineering process, which has to be iterative so that it may allow many evaluations of the educational application life cycle.

Indeed, an iterative software engineering process that allows many evaluations of the software is recommended for any kind of software. For this reason there have been older software life cycle models, such as the spiral model (Sommerville, 1992), and newer, very successful ones, such as the Rational Unified Process (Kruchten, 99; Quatrani, 98), which advocate multiple iterations of the developmental process. Multiple iterations of the software engineering process can be very beneficial for educational software as well. This means that there may be a first prototype which has to be evaluated, and subsequently there can be an improved executable release based on the results of the evaluation and so on. As Dix et al. (1993) point out, evaluation is an integral part of the design process and should take place throughout the design phase of the life cycle. However, evaluations have often been neglected in educational software development. For example, Gilbert (1999) analyses the Teaching and Learning Technology Programme (TLTP) evaluation report by the Higher Education Funding Council for England. Among other things, he notes that the programme seriously underestimated the complexity of designing materials that could be considered, in any sense of the word, "intelligent", and that there has been a serious lack of evaluation.

In view of the high demands on the reasoning abilities of educational software and the apparent need for the improvement of the software engineering process, simulated students were created that can be used to improve the performance of educational applications dynamically (on the fly) and can be used as evaluation agents in an iterative software engineering process with these applications. In particular, we are going to describe how a student modeling process has been incorporated into simulated students. This particular student modeling process focuses on keeping track of what a student is being taught and will actually remember after the end of the lesson (Virvou & Manos 2003a). This is achieved by the adaptation and application of models of cognitive psychology to the particular circumstances of the educational software application.

Student modeling is an important process for ITSs since it may provide detailed reasoning concerning the students' needs and progress and thus make the applications highly individualized. Indeed, student modeling has become a core or even defining issue for ITSs (Cumming & McDougall 2000). In the research described in this chapter, the student models give input for the creation of simulated students. Then the simulated students are used during the execution of the educational applications to give insight to the system as to how much a student has learnt from the material that has been taught to him or her and what needs to be reviewed. If something needs to be reviewed the ITS reschedules the teaching material and presents the topic to the student to be reviewed. Moreover, the simulated students may be used by instructors and ITS designers to evaluate the courses that they have created before these are delivered to real students. Thus, designers are given the opportunity to fine-tune the ITSs so as to achieve better results with the real students.

Simulated students have been created and used in past ITSs, mainly to assist the learning process of students. For example, the mode of the simulated co-learner has been considered quite important by many researchers for the purpose of improving the educational benefit of tutoring systems. One reason for this is the fact that the simulated student can simultaneously be an expert and a co-learner and can thus scaffold and guide the human's learning in subtle ways (VanLehn et al. 1994). However, simulated students have not been used as evaluation components in the software engineering process of ITSs. Such evaluation components may be very useful because they allow and encourage multiple iterations of the design process, which in turn may ensure that the resulting educational software applications are of better quality.

6.1 The simulated students as evaluation agents in the software engineering process of an ITS

As a test-bed for the research contacted, VR-ENGAGE (Virvou et al. 2002) was used. VR-ENGAGE has been enhanced by the addition of a module that measures/simulates the way students learn and possibly forget throughout the process of a game/lesson.

Student modeling in VR-ENGAGE is based on the overlay technique. The overlay model was invented by Stansfield, Carr, and Goldstein (1976) and has been used in many early user-modeling systems (Goldstein, 1982) and more recent systems (e.g. Matthews et al. 2000). The main assumption underlying the overlay model is that a user may have incomplete knowledge of the domain. Therefore, the user model may be constructed as a subset of the domain knowledge. This subset represents the user's partial knowledge of a domain, enabling the system to know which parts of the theory the user knows and which s/he does not know. However, as Rivers (1989) points out, overlay models are inadequate for sophisticated modeling because they do not take into account the way users make inferences, how they integrate new knowledge with knowledge they already have, or how their own representational structures change with learning. One additional problem with the overlay technique is that it assumes for the student an “all or nothing” knowledge of each part of the domain (either a student does or does not know something).

The overlay technique has to be used in conjunction with inference mechanisms about the students’ knowledge. The inference mechanisms that have been employed so far in the literature have been mainly based on actions students make in assessment tests that show evidence of their knowing or not knowing something. However, even in cases when the student shows evidence of knowing something at a particular time, s/he may forget it after a while. Therefore, in our research we take into account what parts of the theory the student has been shown, how often this has happened and what the student is likely to remember. For this purpose, the overlay technique has been extended to include degrees of knowledge for each fact.

Each degree represents the possibility of a student knowing and remembering something, given the time at which it was learnt. For this purpose, we use a forgetting model.

There are two popular views on forgetting (Anderson, 2000). One of them, the decay theory, supports the view that memory traces simply fade with time if they are not “called up” now and then. The second view states that once some material is learned, it remains forever in one’s mental library, but for various reasons it may be difficult to retrieve. These may seem to be conflicting theories, but when someone has forgotten something, there is really no way for us to tell whether it has been completely removed from his or her mental library or is simply very (almost impossibly) difficult for him or her to retrieve it. For our study, both theories have practically the same meaning: If a student finds it hard to remember a fact that s/he has learnt (either due to memory fading or difficulty of retrieval) then the learning process was not good enough and should be modified.

A classic approach on how people forget is based on research conducted by Herman Ebbinghaus and appears in a reprinted form in (Ebbinghaus, 1998). Ebbinghaus worked for a period of one month and showed that memory loss was rapid soon after initial learning and then tapered off. In particular, Ebbinghaus’ empirical research led him to create a mathematical formula which calculates an approximation of how much may be remembered by an individual in relation to how much time has passed since the end of learning (Equation 1).

$$b = \frac{100 * k}{(\log t)^c + k} \qquad (1)$$

In Equation 1:

- t: is the time in minutes, starting one minute before the end of learning
- b: is the equivalent of the amount remembered from the first learning. As it is evident from the logarithmic nature of the formula, b decreases greatly at the beginning and starts to stabilize as time passes.
- c and k: are two constants with the following calculated values: k =1.84 and c = 1.25

Linton (1979) also conducted research on the retention of knowledge and worked for a period of six years. Linton's results were similar to Ebbinghaus' results. Finally, Klatzky (1980) also reports the results of a study that consisted of experiments on retention. These experiments involved repetitions of a memorized list of words after a pre-specified break length, typically up to a few days. This study showed that memory decay is a power function of the break length. For example, subjects forget 55% of the words within a six hour break time and 80% percent within 72 hours. However, these results are very close to Ebbinghaus' results. Indeed, if Ebbinghaus' formula was used, one would find that subjects forget 60% of the words within a six hour break and 75% within 72 hours. Such differences in the results have little importance for the purpose of incorporating a forgetting model into an educational application. Therefore, Ebbinghaus' mathematical formula has been used in VR-ENGAGE to give the system insight on the students' learning and forgetfulness.

In our model, there is a database that simulates the mental library of the student. Each fact a student encounters during the game/lesson is stored in this database as a record. In addition, the database also stores the date and the time the fact was last used, along with a numerical factor describing the likelihood of the student's recalling the given fact. The smaller the factor the less likely it is that the pupil will remember the fact after the end of the game/lesson.

6.2 Learning and Forgetting

Our research goal is to make the educational game more effective in teaching the student. This will happen if the student ends up with many facts, with high factors in his or her mental library, after the course. To model this, it is assumed that the student has a blank mental library on the subject being taught. This means that during the first lesson there is nothing in the mental library of the student to be retrieved.

While the student plays the educational game, s/he encounters a tutor that provides him or her with a piece of information to be taught. This is the first encounter with this information, and it is thus added to the memory database. The data saved in the database are:

- **ID**: a string ID of the fact being taught
- **TeachDate**: the date and time of the first occurrence of the fact
- **RetentionFactor(RF)**: a number showing how likely it is that the student will remember the given fact after the end of a "game lesson"

When a fact is inserted into the database, the TeachDate is set to the current date and time, while the RF is set to a base number. The RF stored in the "mental" database for each fact is the one representing the student's memory state at the time shown by the TeachDate field.

Having saved the above data, one may calculate the percentage of retention of a given fact that a particular student is likely to have at a particular time. We call this Retention Percentage (RP). Whenever we need to know the current RP of the fact, equation 2 is used.

$$X\% = \frac{b}{100} * RF \quad (2)$$

Where:

- b: is Ebbinghaus' power function result (Equation 1), setting t=Now-TeachDate

- RF: is the Retention Factor stored in our database.

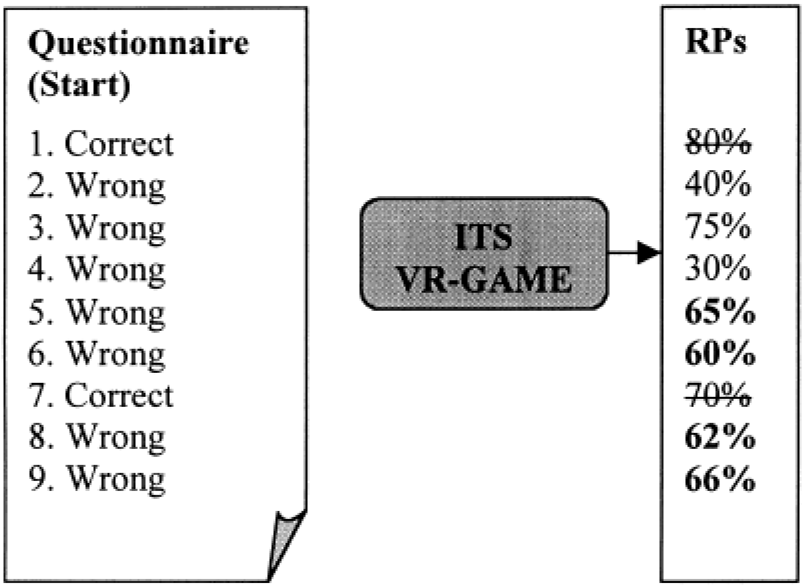
The Retention Factor is used to individualize this equation for the particular circumstances of each student by taking into account evidence from his or her own actions. If the system does not take into account this evidence from the individual students' actions, then the Retention Factor may be set to 100, in which case the result is identical to Ebbinghaus' generic calculations concerning human memory in general. However, if the system has collected sufficient evidence for a particular student, the Retention Factor is set to 95 when a fact is first encountered by this student and then modified accordingly. This will be described in detail in the following sections.

The mathematical formula (1) by Ebbinghaus gives an estimation of how students learn and forget, which applies to all kinds of students and does not take into account individual characteristics. However, the information stored in each individual student model provides more information about each student's ability to learn and memorize new facts. This kind of information has been used to individualize the results provided by the Ebbinghaus formula (Virvou & Manos 2003b). In particular, we have used what we call the Personal Base Retention Percentage, the Memorization Ability factor and the Response Quality factor, which will be explained in detail in the following subsections.

6.2.1 Base Retention Percentage

To estimate whether a student has learnt a fact that has been taught to him or her during a lesson, the student’s RP for that particular fact has to be calculated at the end of the lesson. Then the RP has to be compared with a number that represents a threshold of learning for students. We call this number the Base Retention Percentage (BRP). This number is set by default to 70. This is because any number below 70 corresponds to a forgotten fact according to the Ebbinghaus formula. If the calculated RP of a particular student for a particular fact is greater than the BRP, the student is assumed to have learnt the fact. Otherwise, he or she needs to review it.

A BRP of 70 may give more or less accurate results for a wide variety of students. However, each individual student is a unique entity and has his or her own personal BRP. For example, students with strong memorization abilities tend to have a lower BRP while, on the other hand, weak students tend to have a higher BRP. For example, there may be cases where a student is believed to remember 60% of a fact. This percentage is below 70%, and thus the student is by default believed to have forgotten the fact.



Calculating the personal Base Retention Percentage

However, the student may answer a question correctly that concerns this fact. This shows that the student knows the fact although he or she is believed to remember only 60% of it. If this happens for many facts for that particular student, then the student has stronger memorization abilities than the average student modeled by the Ebbinghaus formula. In such cases, the BRP for this kind of student should be lower than 70, which is the average

BRP. The unique BRP for each student is what we call the student's Personal BRP. Unfortunately, there is no automatic or mathematical way of calculating the personal BRP for every student. One way to calculate it is by giving the student a sequence of tests. First, we need to have the student answer all the questions he or she will encounter in the virtual world. These answers should be given to the student before he or she plays the game and without his or her having read the theory. In this way, the system may find out what the student already knows before he or she reads the theory to be taught. Then, the student is left to play the game. When he or she finishes playing the game, the ITS calculates an RP for each and every part of the theory. Last, we give the student the same questionnaire as at the beginning and mark the results (Figure 4). The answers that the student provides to the questions at the start are compared with those that s/he provides at the end. Moreover, it is examined whether the RP corresponding to every fact associated with each question is less or greater than 70, which is the default BRP.

First, during the comparison of the student's answers at the start with those at the end, we cross out all the facts that correspond to a correctly answered question in the first questionnaire. This is because we assume that the student already knew these facts and that they thus might tamper with our calculations. Indeed, these facts in the example in Figure 3 have been crossed out. Next, we find all the RPs that correspond to correctly answered questions which were not answered correctly in the first questionnaire and for which the calculated RP is below 70 (for our example these are facts 5, 6, 8, and 9, with RPs 65, 60, 62, and 66 respectively). These questions correspond to facts that were not known by the student before the game/lesson and seem to have been learnt by the student during the game/lesson since he or she answered the questions correctly after the end of it. Moreover, the facts are found to have been learnt by the student although the RPs that correspond to them are lower than 70. If such RPs are more than or equal to 4 (to limit the possibility of such a case being a "one-time" event), we take the highest of them (for our case that is 66), and that is one metric (RP1).

After this, we find all the RPs that correspond to wrong answers for which the calculated RP is above 70. Again, if such RPs are more than or equal to 4, we take the lowest of them and use it as the final metric (RP2). In the example we do not have an RP2 metric.

To define the student's personal BRP, we need to consider the following issues:

1. If no RP1 exists we examine RP2. If RP2 exists then this is the student's personal BRP; otherwise, we keep the value of 70.
2. If RP1 exists but no RP2 was found, then this is the student's personal BRP; otherwise, we keep the value of 70. For the example above, the student's personal BRP is set to 66%.

3. Experiments have shown that in rare cases, both metrics can exist. However, in these cases, it is probable that the student has been answering the questions (during the phases of the test) by chance.

For the rest of this study, whenever referring to BRP, it should be assumed that that its value is 70. Given the fact that any RP below 70 corresponds to a forgotten fact (using Ebbinghaus' power function), the "lifespan" of any given fact can be calculated.

6.2.2 Individual Memorization Ability

One important individual student characteristic that is taken into account is the ability of each student to memorize new facts. Some students have to repeat a fact many times to learn it while others may remember it from the first occurrence with no repetition. To take into account these differences, we have introduced the student's Memorization Ability factor (MA). The values of this factor range from 0 and 4. The value 0 corresponds to "very weak memory," 1 to "weak memory," 2 to "moderate memory," 3 to "strong memory" and 4 to "very strong memory."

During the course of a virtual game there are many different clues that can give insight on the student's MA. One important hint can be found in the interval of time between a student having read about a fact and his or her answering a question concerning that fact. For example, if the student has given a wrong answer about a fact that he or she has just read about then s/he is considered to have a weak memory. On the other hand, if s/he gives a correct answer concerning something s/he read about a long time ago then s/he is considered to have a strong memory.

Taking into consideration such evidence, one may calculate the student's MA value. Using MA, the Retention Factor is modified according to the MA value of the student in the manner illustrated in Table 1. As mentioned earlier, every fact inserted in the database has an initial RF of 95.

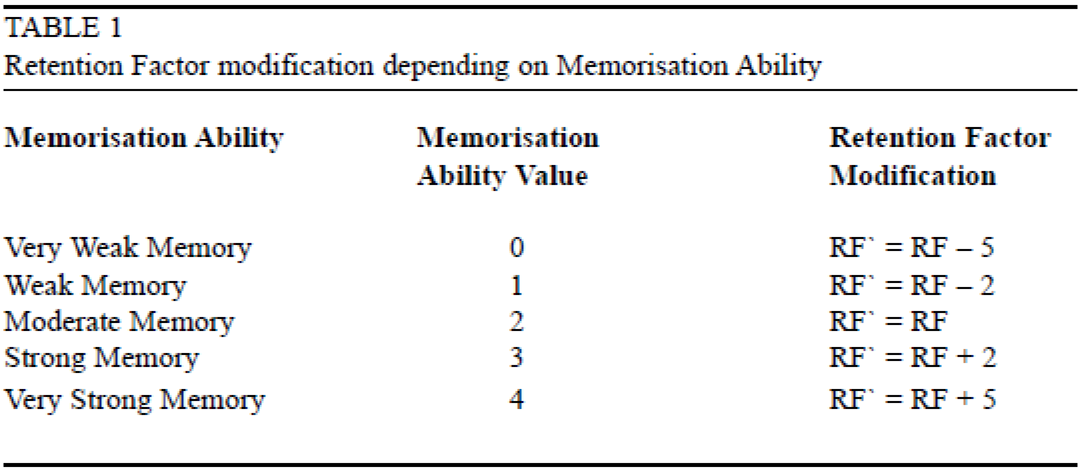


TABLE 1
Retention Factor modification depending on Memorisation Ability

Memorisation Ability	Memorisation Ability Value	Retention Factor Modification
Very Weak Memory	0	RF` = RF – 5
Weak Memory	1	RF` = RF – 2
Moderate Memory	2	RF` = RF
Strong Memory	3	RF` = RF + 2
Very Strong Memory	4	RF` = RF + 5

After the modifications, which are based on the MA factor, the student's personal RF ranges from 90 (very weak memory) to 100 (very strong memory), depending on his or her profile. Taking as a fact that any RP below 70 corresponds to a forgotten fact, one may calculate the lifespan of any given fact for the MA mentioned above using Equation 2. Thus, a student with a very weak memory would remember a fact for 3 minutes while a student with a very strong memory would remember it for 6.

6.2.3 Individual Response Quality

During the game, the student also faces question-riddles (which require the recalling of some facts in order to be answered correctly). In that case, the RP of the fact is updated according to the student's answer. An additional factor, the Response Quality (RQ) factor, is used for this modification. This factor ranges from 0 to 3 and reflects the quality of the student's answer. In particular, 0 represents "no memory of the fact", 1 represents an "incorrect response: but the student was close to the answer", 2 represents "correct response: but the student hesitated", and 3 represents a "perfect response".

The formulae for the calculation of the new RF depending on the Response Quality Factor are illustrated in Table 2.

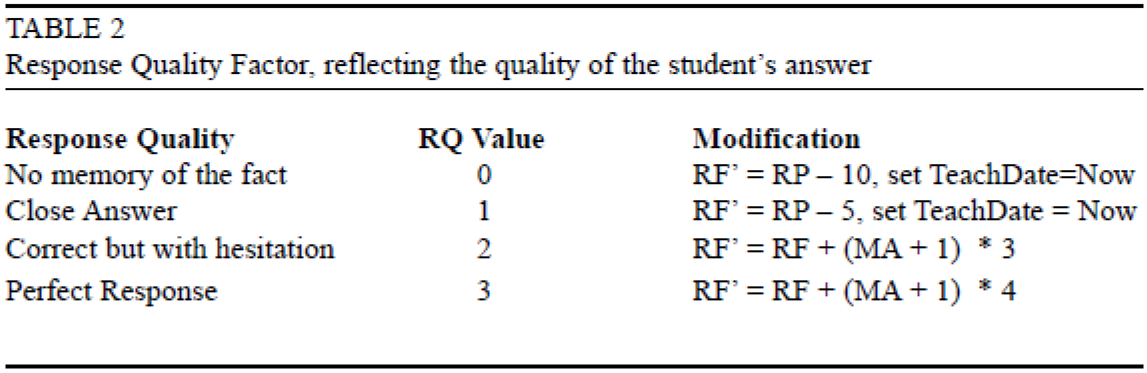


TABLE 2
Response Quality Factor, reflecting the quality of the student’s answer

Response Quality	RQ Value	Modification
No memory of the fact	0	RF’ = RP – 10, set TeachDate=Now
Close Answer	1	RF’ = RP – 5, set TeachDate = Now
Correct but with hesitation	2	RF’ = RF + (MA + 1) * 3
Perfect Response	3	RF’ = RF + (MA + 1) * 4

When a student gives an incorrect answer, the TeachDate is reset, so that Ebbinghaus’ power function is restarted. This is the case both when the student gives a completely incorrect answer (RQ value = 0) and when the student gives an incorrect answer which is close to the correct one (RQ value = 1). When a student gives a correct answer, the increase of his or her Retention Factor depends on his or her profile and more specifically on his or her Memorization Ability factor. In particular, if the student’s RQ is 2 and s/he has a very weak memory then the RF will be increased by 3 points (extending the lifespan of the memory of a fact by about a minute), while if s/he has a very strong memory the RF will be increased by 15 (extending the lifespan by over 6 minutes). These formulae for the calculation of the RF give the cognitive model a more personal aspect since they are not generic but based on the student’s profile.

The previously mentioned individualizations were made and refined based on empirical research data. In the case of an RQ of 0 or 1 (wrong answer), there is strong evidence that the student has forgotten the fact, and thus we calculate the RP as if the student had first seen the fact at the time s/he gave the answer, then we lower it and assign the new value as the RF. Finally, we also reset the time.

Indeed, in the case of an RQ of 0 or 1 we achieve a rapid loss of retention by resetting the time in the formulae for the calculation of RF. This is due to the logarithmic nature of the Ebbinghaus Power Function. Moreover, the RP is also decreased by 10 and 5 respectively. As a consequence of these modifications, the resulting RP will almost definitely correspond to a forgotten fact (one with a final RP lower than 70). After these modifications, the only time the system may come up with a fact of an RP value greater than 70 is if the student has an MA of 4 (very strong memory) and the answer is among the latest ones given (thus the time difference that is applied to the Ebbinghaus function is small and as a consequence the retention decline smaller too). Our experiments have shown that such cases are rare and when they do occur the student has accidentally answered incorrectly.

If a student has an RQ of 2 and 3 then we know that s/he has answered correctly and we can thus raise the RF value in accordance to the student’s personal MA value. The stronger the student’s memorization abilities are the higher we can raise the RF. There are cases when a correct answer may be given as the result of a lucky choice, but the system has adequate information about the student’s profile to track such cases down and remove them from the retention process.

6.3 Conclusions

In this chapter, it has been shown how simulated students can be created and used for the enhancement of the performance and the software engineering process of an ITS. For this purpose, the student modeling component of the ITS was used to allow a simulated student to be created with the characteristics from each individual student model as input. In particular, it takes into account what the student has been able to remember from the material taught as this has been recorded in his or her performance on tests. This information is combined with principles of cognitive psychology, giving the ITS insight on what students may remember from the material being taught to them.

This memory information is used by the system to adapt the teaching process accordingly. Depending on what a student does or does not remember, the system proceeds by presenting new course material or repeating certain parts of the course material that have already been taught. In this way, the educational software application becomes more personalized and adaptive by responding appropriately to each individual student’s needs regarding the way the course material is being taught to him or her. Moreover, and most importantly, the simulated students are used instead of real students for the evaluation of the ITS. In this way, the developers of the ITS (teachers and ITS designers) may find out what needs to be corrected in a subsequent version of the ITS without cost to the educational process.

Indeed, evaluating a course on real students is not fair for them since they would have to suffer the consequences of all the possible mistakes that the developers may have made and would have corrected if they had discovered them earlier. As a result, the simulated students make a major contribution to the software engineering process of educational applications by encouraging and facilitating many iterations of the software life-cycle and many evaluations. This process can guarantee that the end result will be of higher quality than it would be without the use of simulated students.

7. Using Intelligent Agents to evaluate the educational software

Educational software may serve the aims of education very effectively since it can assist students to learn and practice new skills without necessarily the presence of a human instructor. However, to benefit from educational software to its full extent, this software has to be included in the educational process and has to be designed very carefully. Indeed, a major issue is how to design an educational system that is beneficial to students. Towards this end, there is a need for the incorporation of reasoning aspects into educational software technology, so that the interactivity and individualization abilities of the tutoring software may be maximized. Such reasoning abilities may be provided by Intelligent Tutoring Systems (ITSs).

ITSs have been quite good at providing dynamic aspects to the reasoning ability of educational applications. They have been shown to be effective at increasing students' motivation and performance in comparison with traditional learning methods and thus ITSs may significantly improve the learning outcomes (Self 1999, Shute et al 1989). It has been widely agreed that an ITS should consist of four components, namely the domain knowledge, the student modeling component, the tutoring component, and the user interface [10, 11]. In particular, the student modeling component contributes significantly to the individualization of the electronic tutoring to each student's needs. Indeed, the student modeling component aims at gaining an understanding of what individual students know, how they learn, and what their problems are while they learn. ITSs are mainly based on Artificial Intelligence (AI) techniques. AI in education (AIED) can offer ways to develop and test precise theories and important concepts relevant to individualized learning that have largely been overlooked by Education, such as that of learner modeling. However, AIED can scarcely claim to be in Education (Cumming & McDougall 2000). Indeed a common criticism on ITSs is that they may miss the mark in terms of task reality, feasibility, and effectiveness (McGraw, 1994).

In this chapter, the problem of task reality, feasibility, and effectiveness of ITSs will be addressed by introducing a novel approach, which is based on the reasoning capabilities of ITSs themselves. It will be argued that the development and evaluation process of ITSs can be assisted by extending the techniques used for student modeling in ITSs. To this end, an evaluation component was developed that can be used in ITSs. The evaluation component is an agent that acts as a simulated student and is meant to be used by instructors-authors to evaluate the ITSs that they have authored before they are delivered to real students. Thus, authors are given the opportunity to fine-tune the courses of the ITSs that they have created so as to have better results for the real students.

As a test-bed for our agent, we have used VR-INTEGATE (Virvou et al, 2002), an authoring tool for ITSs that operate as virtual reality games. An ITS authoring tool is a generalized framework for building ITSs along with a user interface that allows non-programmers (prospective authors) to formalize and visualize their knowledge (Murray, 1999). VR-INTEGATE is meant to be used primarily by authors-instructors, who may author their courses, and then by students who are going to use the resulting courses. It offers multiple virtual reality game environments and the basic story of these games. Indeed, recently a lot of researchers are convinced that education may benefit a lot from the incorporation of computer games (e.g. Amory et al 1998, Boyle 1997, Inkpen et al 1994, Jayakanthan 2002). VR-INTEGATE also incorporates a learner modeling 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.

The present research work has led to considerable enhancements of the learner modeling component of VR-INTEGATE so that it can constantly make observations about the students' behavior. These observations mainly concern the way students respond to assessment questions in terms of the quality and correctness of their answers. For example, whether a student's answer was correct, whether the answer was given with certainty or with hesitation, and if the answer was wrong whether this kind of mistake is a frequent one for the particular student, etc.

The evaluation agent that has been incorporated in *Ed-Game Author* is an application that, given a learner model, starts playing the virtual lesson inside the ITS, simulating a real user's reactions.

7.1 Reasoning of the evaluation agent

The evaluation agent is constructed for a sole purpose: To be able to simulate a user in any aspect inside the ITS. To accomplish this, the agent should have adequate information so as to be able to mimic the student's actions inside the system. This information is stored in the learner's individual model within the ITS. Obviously, to have a solid learner profile, that is stable and has adequate information for the agent, a student must have interacted and used the system for at least a couple of sessions, during which time the digital image of the student is being composed (and stored in the learner model).

The kind of information that is needed for the creation of the digital image of a student is determined by the way a student's image is perceived by the ITS. This means that a student model is determined by the way the system actually understands the term "student". From the system's point of view, the "student-user" is nothing more than digital-binary input, either from the mouse (clicking on the screen) or from the keyboard. To be more exact, this is a very low-level image of a "student-user". If we see it from the ITS's layer, then the "student-user" represents input of very specific type, for example: students' answers to questions, movement inside the virtual environment, responses to the system's interaction, etc. For the composition of the "student-user's" digital image, the ITS needs to keep data for each category of input that it can understand and use. Such data may be classified into two major categories: temperamental data and cognitive data.

Temperamental data include all the information that is needed to mimic the student's reaction inside the virtual environment of an ITS game. Cognitive data have to do with the student's mental capabilities and his or her knowledge level of the domain.

The reason, why both categories of information are needed, is the fact that the evaluation agent needs to simulate both the knowledge level of a student and the way s/he learns. From the way that a student learns, depending on the type of person that s/he is and his or her cognitive abilities, the instructor may understand how motivating and educationally effective a course is.

7.2 Temperamental model

Temperamental data is connected to the way that a student behaves and responds to the system. In the case of VR-INTEGATE the students' behavior is related to the way that the student plays the educational game in the virtual world.

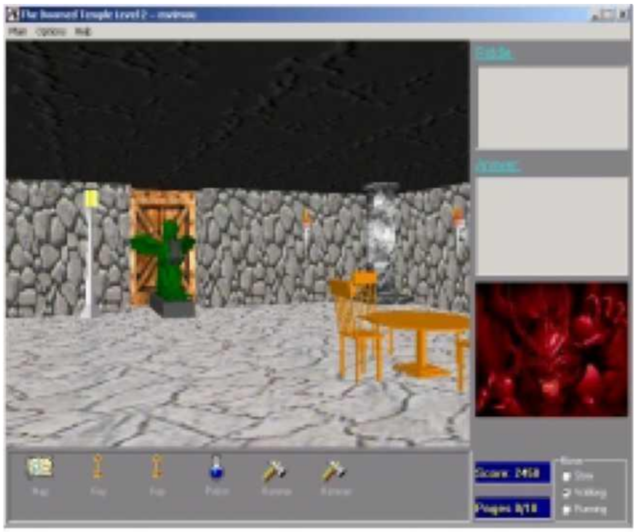


Figure 1. A guardian

The features of virtual reality games include dungeons, dragons, castles, keys, etc. In these games the student-player tries to reach the Land of Knowledge and find the hidden treasure. The difference between educational games and commercial games is that in the first ones, players must fight through by using their own knowledge. To win, the player has to obtain a good score, which is accumulated while the player navigates through the virtual world and answers questions concerning the domain being taught.

In the game worlds there are animated agents that communicate with the players. There are three types of animated agent: the advisor, the guard of a passage, and the student's companion. Animated agents, who act as advisors, lead the student to lessons that s/he has to read. Animated agents, who act as companions are responsible for showing empathy to the students and help them in managing their emotions while playing and answering questions. On the other hand, animated agents who act as guards of passages ask questions to players. These questions have to be answered correctly by the students so that they are allowed to continue their way into the passage and receive points for their total score. An example of an animated agent who acts as a guard is the dragon, which is illustrated in Figure 1.

Temperamental data include:

- The way that the user walks around the virtual world. Is s/he lost easily inside the labyrinth? Does s/he keep walking around the same places (maybe looking for something or just checking around)? Does s/he take a lot of time at specific locations (maybe staring the surroundings and thus being distracted from the system)?

- How familiar s/he is with the use of a computer. While moving around the virtual environment, does s/he keep bumping into physical obstacles? Does s/he take a lot of time to navigate through the user interface?

Time plays an important role in temperamental measurements. There are many inferences that can be drawn from the students' feelings and reactions depending on the time they spend before and after they make certain actions. Some examples of inferences based on observations of time spent for various activities are the following:

- The time that a student takes to answer a question. This measures the degree of speed of the student.

- Pausing time after a system's response. The time the computer is left idle after a response to the student is used to measure the degree of surprise that the response may have caused to the student. At the first five times that the student answers questions the system measures this pausing time and calculates an average. After that, the system has a base measure about whether a student's pausing time is out of the ordinary and may have been caused by surprise.

In addition, certain patterns of actions are used to show aspects of the students' cognitive and emotional state.

Some examples of students' actions that are used as evidence are the following:

- The number of times that a student presses the "backspace" and "delete" button while forming an answer. This evidence is used to measure the degree of certainty of the student concerning a particular answer. The more times the student presses "backspace" and "delete" the less certain s/he is about the answer. If the student consistently hesitates and does not seem certain about his or her answers irrespective of their correctness, then this may reflect a personality attribute of lack of self-confidence. On the other hand, if lack of certainty is occasional then this probably means that the student does not likely know the particular piece of the domain that is related to the exam question that the student has answered.

- Mouse movements without any obvious intent in the virtual reality space of the game. This kind of evidence is mainly connected to the degree of concentration or frustration or intimidation of the student. The more mouse movements without any obvious intent, the less concentrated or the more frustrated or intimidated the student is.

In some cases, inferences are drawn from the combination of two different categories of evidence. For example, the degree of determination is calculated as the means of the degree of speed and the degree of certainty of a student.

7.3 Cognitive model

Cognitive data include:

- The level of the student's knowledge. This is measured by keeping track of the right and wrong answers s/he gives and the time that s/he needs to give a correct one. Moreover, since the ITS provides help through the use of virtual tutors, a metric is also provided from the number of times the student uses a tutor to help him or her answer a question.

- The student's retention capabilities. The agent incorporates a cognitive model, which is based on cognitive psychology. This model calculates and simulates the retention and memorization capabilities of a student and gives the teacher an insight on the proportion of the information that is actually learnt by a student-player during the virtual lesson.

- The ITS also keeps track of specific statistics that have to do with the causes underlying an error. For example, whether the student made a typographic, a syntactical, or a spelling error. These metrics are also stored in the student model.

7.3.1 Retention and memorization capabilities

The simulated student-player incorporates a cognitive model that keeps track of the students' memory of facts that have been taught to them. For this reason, principles of cognitive psychology have been adapted and incorporated into the system. As a result, the educational application takes into account the time that has passed since the learning of a fact has occurred and combines this information with evidence from each individual student's actions. Such evidence includes how easily a student can memorize new facts and how well s/he can answer questions concerning the material being taught. In this way, the system may know when each individual student needs to revise each part of the theory being taught.

The cognitive model is based on a classical approach about how people forget, which was introduced by Ebbinghaus (1998). Ebbinghaus' empirical research led him to the creation of a mathematical formula which calculates an approximation of how much may be remembered by an individual in relation to the time from the end of learning. The mathematical model and its function have been extensively described in Chapter 6 of this dissertation.

In the cognitive model of the simulated student-player, the Ebbinghaus calculations have been the basis for finding out how much is remembered by an average student. In particular, there is a database that simulates the mental library of the student. Each fact a student encounters during the game-lesson is stored in this database as a record and a Retention Factor (RF) is calculated for the fact. The RF represents the student's memory state at that time.

A further enhancement of the student model is the addition of a new factor, the Student's Retention Factor (SRF). The initial Retention Factor (RF) is used to measure the portion of a fact that is actually remembered by the student after a specific time interval. Although the actual calculation of the RF has been modified to take into account a portion of the existing student model, the factor continues to be based mostly on Ebbinghaus' mathematical formula, which is very general and does not take account the particular circumstances of an individual student model. Experiments have shown that this is not enough and that each student tends to have a personal way of reacting inside the system. More specifically, the level of retention in previous studies was considered to be static and equal to 75%. If a fact had an RF value equal or higher than this level of retention, it was considered to have been successfully memorized by a student. The result of the research conducted for this dissertation was that each student's base level of retention varies. Therefore, with the use of questionnaires at the end of the lesson, the facts that were actually learnt were recorded and compared with results provided by the cognitive model. From this procedure the SRF was created, which represented a personalized level of retention for each student.

7.3.2 Diagnostics

The educational games perform error diagnosis and record all errors that the student may have made. Thus, the system records a detailed report for every student in the student model. While the educational game examines the knowledge of students, it can distinguish between spelling mistakes, typing/keyboard mistakes, and errors that are due to lack of domain knowledge. 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 answer that contains a letter substituting the correct one, which is near the correct one on the keyboard, then it has probably been a keyboard 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. The results of these answers are then kept in the statistical part of the system. In particular, the students' errors are kept and classified in different categories depending on their underlying cause.

7.4 The overall architecture

In the approach and implementation, functionality provided by the new technology of the Web Services was used. From the perspective of a system's architecture, Web Services are a collection of procedures and/or functions that have the ability to be called remotely by any external system.

What the system may gain from this new technology is greater scalability and flexibility.

By using Web Services, the system can be cut down to relatively small independent pieces and then distributed along the network. For the simulated student-player, a web service was created from all the input procedures and functions of an ITS, making the system's core independent of the source of input.

The cognitive and temperamental models have also been implemented as two Web Services. Using the flexibility that this new technology provides, the models are expandable while at the same time they can be used with almost no effort by various other modules. Web Services also provide an important advantage, the fact that they can run over the Internet. In that way, the models and the ITS can actually be in different machines across the web. This is extremely useful for the authors who may try their courses on simulated student-players using real students' profiles which may reside in different PCs.

The implementation of the simulated student-player requires computer-computer interaction rather than human-computer interaction, which is needed for the user-interface of a standard ITS. Thus, the input from the mouse or keyboard, which is suitable for real students, is not suitable for the simulated student-agent. For this reason, another interface for the ITSs was created which was called IITSController (Interface Intelligent Tutoring System Controller). IITSController is a Web Service provided by the ITS, that gives the caller full control over the virtual lesson's interface. This means that one can implement a program, which can reproduce any input for the system that would otherwise be generated from the mouse or the keyboard. Moreover, since IITSController is a Web Service, it may be called by any computer in the web, so an application (in our case the agent) may invoke remotely the ITS system.

By using this architecture the system's core no longer "knows" who/what is providing the input, because this is not of any interest to it. As a result, it still works, calculating user profiles and statistic results, but irrespective of who/what is interacting with it in the local computer. Both the simulated student-player and a real student-player are regarded as valid users of the ITS and are treated in the same way.

The simulated student-player consists of a core module and two proxy classes, one that communicates with the ITS and one that accesses the cognitive model. At this point, we should point out that the agent itself is implemented as a web service, thus it can be invoked and used through the Internet.

The agent takes as input a web service for the cognitive model, an IITSController, and an ASCII file containing the student's profile as it has been compiled through the last couple of virtual sessions. The agent uses the information stored inside the student's profile to simulate the student's actions.

7.5 Iterations of the authoring process

During the authoring process, after the teacher finishes describing a new virtual world s/he may ask the simulated student-player, which acts as an evaluation agent, to "play" the virtual game using different student profiles. These student profiles contain long-term characteristics of real student-players that have played in other parts of the game-course, which had been previously authored by the instructor and used by students. These profiles may have been stored in the students' PCs, which may be different from the author's PC. However, they can be collected through the Internet.

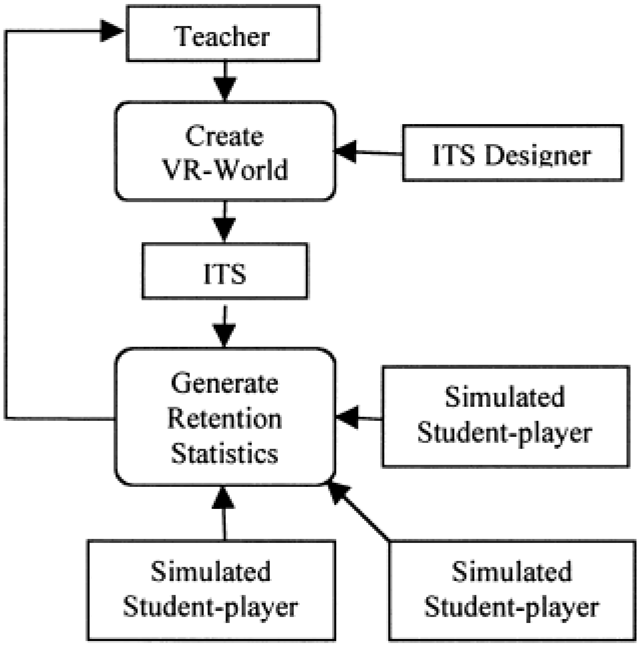


Figure 2: Iterations of the authoring process

In the end, the teacher views the results and may choose to modify the virtual world's content so as to emphasize some parts of the theory more than others or s/he may find a mistake in the flow of the lesson which s/he may wish to correct. With that tool, the teacher has a measure of the virtual lesson's efficiency before taking it to class. This allows an iteration of the authoring process of the ITS and thus ensures better quality of the resulting educational application. Thus, the life-cycle of the educational games that are created by instructors may contain several iterations as illustrated in Figure 2. Each iteration, improves the previous version of the system and leads to better quality at no cost to the educational process.

7.6 Conclusions

This chapter has presented and discussed an evaluation agent that can be used in the context of an ITS authoring tool. This component can be particularly useful to instructors-authors who can evaluate the courses that they have created using the evaluation agent rather than real students. In this way, instructors may easily identify possible deficiencies of the courses that they have constructed and thus they may make the necessary amendments before the courses are delivered to real students. This process may allow instructors to produce courses of very high quality to the benefit of education.

The evaluation agent uses data from real students in order to imitate their behavior while they learn. Moreover, it uses theories from cognitive psychology to gain an understanding of how much new knowledge a student may learn and remember after each lesson. For these purposes, the evaluation agent described uses both temperamental data and cognitive data concerning students in order to create a simulation of them.

8. Initializing a student model via the use of an online web game

Personalization and adaptivity are desired characteristics for software applications that are addressed to a wide range of users of various backgrounds capabilities and needs. These desirable features in user interfaces may be achieved if they have user models inside the software applications. Applications that may benefit greatly from user models are Intelligent Tutoring Systems (ITSs). In such cases user modeling is learner modeling. Learner modeling involves the construction of a qualitative representation that accounts for student behavior in terms of existing background knowledge about a domain and about students learning the domain (Sison & Simura, 1998). Such a representation, called a learner model, can assist an ITS, an Intelligent Learning Environment (ILE), or an intelligent collaborative learner in adapting to specific aspects of student behavior (McCalla, 1992).

User modeling requires collecting information about each user so that several aspects of his/her state may be modeled. This means that a system can only achieve an accurate user model after the user has provided explicitly and/or implicitly information about him/her. When a new user starts interacting with an application then the system must initialize the user model which is then refined when more information about the user is known. The initialization of the user model is extremely important because if the user model is initialized incorrectly then the application may lose its credibility from the first interactions with the user. Although a lot of research has focused on the identification of efficient methods for updating the user model, the process of the initialization has often been neglected or it has been dealt with by using trivial techniques. One common way for the initialization of user models involves the user answering questionnaires. However, questionnaires cannot be too long because users may find it tiresome to answer them. Moreover, users may not be able to describe themselves as accurately as it would be needed.

In view of the above, the aim of the research described in this chapter is the initialization of user models concerning user's memory retention capabilities through a pleasant and user-friendly memory game application. The game provides the asset of not being tiresome and, as it can be played in any place, it does not have to take productive time away from the user. Rather, it may be played at leisure time. Thus the game is implemented in such a way so that it is small and portable, being able to run everywhere even on mobile phones. This game incorporates a cognitive psychology model which measures the user's memory retention capabilities. The cognitive psychology model is based on a classical approach on how people forget, which appears in a reprinted form in Ebbinghaus (1998).

8.1 The game as an initialization application

It is widely accepted that one of the most popular ways of human-computer interaction is a computer game. Thus, humans tend to react more naturally to the interaction with a game than with any other kind of application. The same stands for the output that may be collected from experimenting with someone’s abilities. The data that the computer may acquire when testing a subject inside a natural and friendly environment, such as a game, is by far better than the data collected using for example a simple questionnaire.

The game serves as an auxiliary application that aims at producing output about the user's retention capabilities. Thus, the game serves as an initialization application for the user model. The output produced by this game may be imported to another application, the target application, which needs a model of the user's retention capabilities. Such application can be an ITS. The initialization procedure is illustrated in Figure 1.

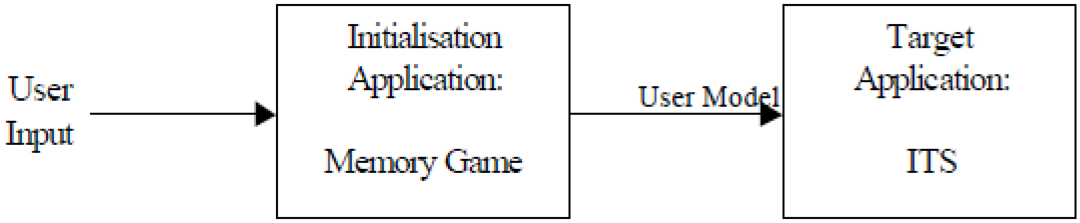


Figure.1 Initialization procedure through the memory-game.

8.2 "Two-of-a-kind"

The application that was selected to use the cognitive model is a simple memory game known as "Two-of-a-kind". This game is a popular mind-game and very simple to play. The player starts with a number of cards placed in front of him/her upside-down. These cards are grouped into pairs which have the same image. Taking turns, the player is asked to flip two cards at a time. If both cards of the pair that he/she has chosen bear the same picture then the cards remain flipped while the player proceeds to the selection of the next pair of cards. If, on the other hand, the cards of the pair do not bear the same image, the cards are flipped back to the upside-down position and the player is asked to select another pair. In this game the user's goal is to find all the images that match, revealing the whole board. Screenshots of this game can be viewed in Figures 2 and 3.

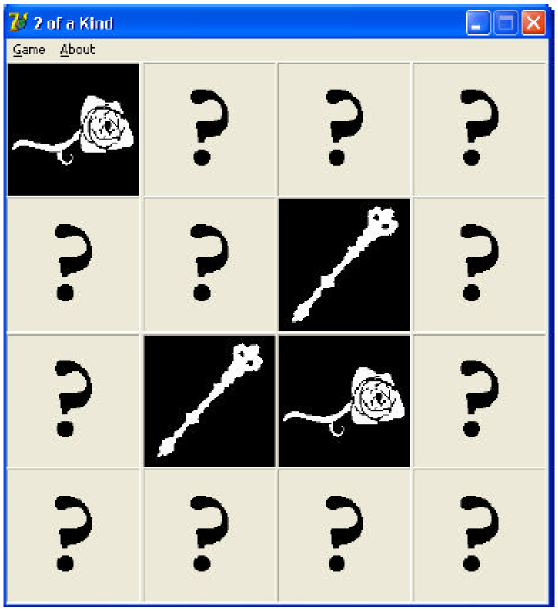


Figure 2. Client Screenshot of the memory game

This game is quite well-known and has been packaged by gaming companies in various forms. It is solely based on the players' pattern matching and short memory retention capabilities. Other variations of the game include timed challenges where the player is required to complete the board not only successfully but also as fast as he/she can by making the fewest possible mistakes. It is clear that if we assume that the player does not find any pair by luck then s/he will need to at least flip all tiles once (thus making N mistakes, where N is the amount of pairs inside the board), and then match all pairs unerringly. This gives us the best result of N mistakes, likely by someone with a photographic memory.

8.3 Implementation Architecture

Technically the application "Two-of-a-kind" has been developed using a Microsoft SQL Server as the back-end and Microsoft .NET technology for the framework. We have used 2 layers of application development. The first one is an application server that handles all the functional logic of our application. It is the one holding all the necessary modules that simulate the Ebbinghaus' cognitive model. The advantage of using an application server is the fact that one does not need to rewrite the actual functional processes of the application, but only the front end. For example, we have created an application in Delphi which acts as a client to the specified application server (Figure 2). Furthermore we have created a collection of ASPX web pages which can provide a front-end either in normal HTML pages (thus being able to be viewed by usual Web browsers like Netscape, Internet Explorer, Mozilla, etc.) or in HTML pages specifically designed for mobile devices (like cell-phones or palmtop) that have the ability to surf the Internet. Figure 3 illustrates the system's architectural design.

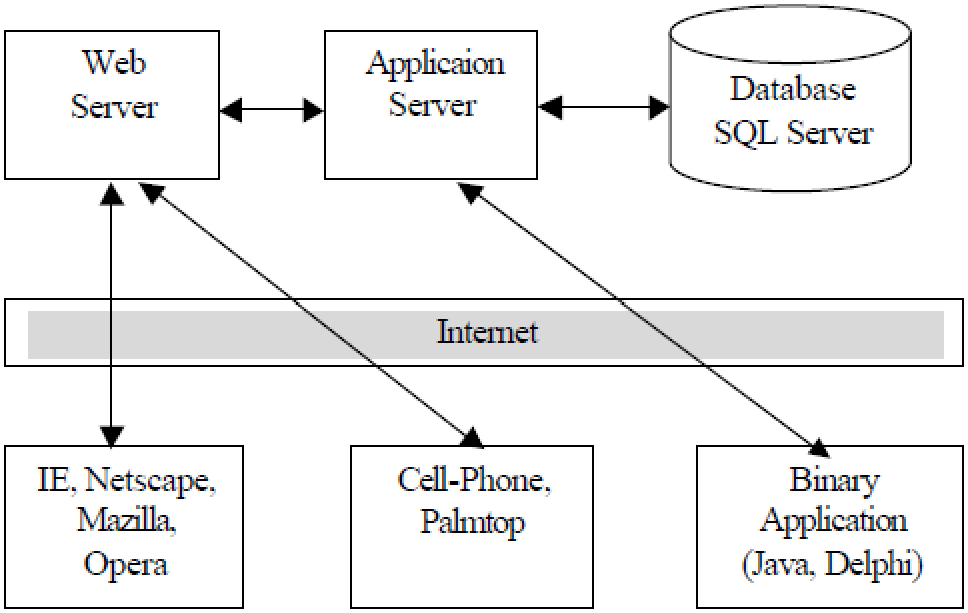


Figure 3. System's Architecture

8.4 Cognitive Model

Behind this simple game we have incorporated a powerful cognitive model that can measure the retention capabilities of the player. Based on the research studies of Ebbinghaus (1998), who gave a mathematical formula that can simulate the learning curves of the human brain, the model was extended so that it could be implemented and used in a user model.

8.4.1 Retention Capabilities

While the user plays the game, the cognitive model measures his/her retention capabilities and creates a user retention profile. If the user is playing the game for the first time, then the first statistics are gathered and an initial user model is calculated. During each subsequent play of the game, along with the player, in the background the application tries to simulate the user's choices using the previously generated user model. If these choices are very different from the user's actual ones then the user model is modified accordingly. This process is performed in silent while the user plays the game unobtrusively. This process keeps repeating itself until the user model is stabilized, at which point the system has collected enough data to be able to model the user's retention capabilities accurately.

During the process of acquiring data, extensive effort has to be put on the separation of the "luck" factor from the actual data. When dealing with such mind games, apart from the retention factors, one must always keep track of the data that are being compromised by the fact that the player was just "lucky". In this application there are many different ways to find out which of the data are true and which are the "lucky" ones. For example, if a player flips an image for the first time (which means that s/he was in no position of knowing that s/he would find this image underneath), then whatever the outcome, it is discarded since it does not contribute any information about the user's memory capabilities. The same stands for the last image for which the player actually has no other choice. Whether the user had already seen the last card or not, s/he has to flip it because there are no more images to be flipped. As a result this move too does not contribute any information about the user's memory capabilities and therefore it is discarded.

Through experimentation it was concluded that the value of an outcome depends directly on the amount of pairs that have not yet been discovered. At that point, if this problem is to be addressed, the number of pairs should be increased and the profiling should stop when there are 4 or 5 left. However, that approach would again lead to false data. Given the nature of the game and the fact that the player does not actually gain anything by remembering the positions of the images (in contrast with educational software where the information is related to knowledge being taught); by increasing the playing field, the results may not match the users accurately. This is because users may get lost in a high number of choices. Consequently, they may get frustrated, irritated, and lose their interest to the game. If this happens, then the existence of the game as a "pleasant" initialization way becomes pointless. Thus, an optimum number of pairs must be selected, which will be able to give the data needed and at the same time it will ensure that the game is pleasant for the users.

For this purpose, 15 pairs were selected, meaning 30 tiles in a 6 by 5 matrix. When the game starts, the matrix is filled randomly with the various images. Each time the positioning is completely different from the previous one. Monitoring is started from the beginning up to and including the 11the pair. From then on, the results of the 12th and 13th are stored for comparison and experimentation reasons and the rest are discarded due to the very high "being lucky" probability.

The information that the application gathers during the play of the game is timestamps that have to do with the exact time each tile is flipped and whether the flip resulted to a successful pair match or not. Ebbinghaus' mathematical model depends greatly on the time passed between seeing an item and using an item. Thus these data are acquired from the user interface and then fed into the model. The model then produces a retention percentage. This percentage shows the chance that the user has to actually remember the tiles he flips. The closer to 100% the percentage is, the higher the possibility of the gamer to remember the tile he chooses.

The mathematical model also depends on various factors that have to do with the gamer, for example the Base Retention Factor and Memorization Ability (Manos & Virvou 2003). The aim of the application is to calculate these factors so that the model will be able to simulate the gamer's responses. Whenever the model predicts falsely (e.g. the model says that the gamer knows where the tile is while the gamer chooses incorrectly and vice versa), these factors are being recalculated to fit the new information. After the game has been played a number of times, the model can predict correctly the choices about 80% of the time.

8.4.2 Model Usage

The retention user model that is generated by this application is not used in the same application for any other reason but for being updated and synchronized with the user's actual retention capabilities (since these are changed through the course of time). The principal aim of the creation of the memory user models is for them to be used as initial memory user models by other applications that need to know the user's retention capabilities. One such application is VR-INTEGATE (Virvou et al. 2002; Manos & Virvou 2003). VR-INTEGATE is an authoring tool that generates educational games. VR-INTEGATE imports and uses the initial user models generated by the mobile game "Two-of-a-kind" to test the effectiveness of the educational game that it produces. It supplies the user models to artificial agents which walk through the educational software (that it generates) measuring which things the actual user learns (remembers) and which s/he forgets after the end of the virtual lesson. An accurate user model concerning the user's retention capabilities is very useful for the dynamic planning of the lessons to be taught by the educational game to each individual student so that s/he can learn and consolidate the knowledge at his/her individual pace. Thus, the memory game "Two-of-a-kind" serves as a way to initialize the individual user models concerning the memory abilities of each user. Specifically, users may be asked to play the memory game at their leisure, at any time and any place where they have some spare time. As a result, their memory user model is initialized.

8.5 Conclusions

In this chapter, it was shown how a pleasant memory-game application may be used as a way of initializing individual user models concerning memory retention capabilities of users. The advantages of this method is that users do not have to fill in cumbersome questionnaires and the user modeling procedure does not have to depend on the users' beliefs about themselves, which may be inaccurate. Moreover, users do not have to devote time from their work, which can be counterproductive, but rather they can use the memory game during their leisure time, anywhere they wish since the application is portable and can run both on desktop and mobile devices. The initialized user models may then be imported to another adaptive application that needs to have a model of users' retention capabilities.

Such applications can be Intelligent Tutoring Systems that need to have a representation of the students' way of learning, how fast they memorize a new syllabus, etc.

9. Re-usability of the cognitive model via the use of Web Services

In order to be able to have the formula used by anyone who might want to exploit its advantages, a set of Web Services was created. Based on SOAP (SOAP ver 1.2) and implemented using PHP on an Apache web server, these services strive to expose an easy-to-use interface that brings all the research inside any application development environment.

The use of SOAP for this implementation was decided based on the fact that this is the easiest way to expose any kind of software service, in such a way that others might be able to use them and integrate them in their own applications. The language that each application is written is not a barrier since SOAP implementations exist for all RAD (Rapid Application Development) packages (Microsoft .NET Visual Studio, Embarcadero, Delphi, PHP, Python, Java, C++ etc).

9.1 Web services

In order for someone to use the cognitive model, s/he needs to integrate and use five functions exposed by the Web Service definition. These functions are used to create a new profile, initialize its factors, and then feed it with facts that need to be memorized. At any time, you may ask the system to provide you with the relevant retention factor for a specific fact. The following is the description of each function and the way it works.

9.1.1 function CreateProfile (BRF:integer; MA:integer): string

This function creates a new student profile in the systems database. It takes two arguments corresponding to the BRF (Base Retention Factor) of the student and the MA (Memorization Ability). The BRF is expected to be a number from 0 to 100, while the MA from 0 to 4. If these values are not known at the creation of the profile then you should pass -1 as a value to both of them. This will result in the system initializing these values with the default ones (75 for BRF and 2 for MA).

When successfully called, this function will return the unique profile ID that corresponds to the record created. This profile ID is actually a string representation of a GUI and is required by all subsequent calls to this Web Service. You will likely need to store this ID in your own database.

9.1.2 ModifyProfile (aProfileId, BRF, MA): boolean

With this function you can change the BRF and the MA values of a specific profile at any time, provided that you have the relevant profile ID. The Web Service returns true if everything was updated successfully or false in the case of an error.

9.1.3 EncounterFact (aProfileId:string, aFactId:string)

The first time a student is faced with a new fact/theory, you should call this service to notify the system of the event. The system stores the time stamp of the event to be used for the future calculations. The fact ID that you pass as a parameter should be unique for this event and you should use the same one when querying about its retention Factor (RF) .

9.1.4 QueryFact (aProfileId, aFactId, aResponseQuality): integer

This function can be called under two different cases:

a) If you just want to know the RF of a specific fact then you call it passing the relevant profile ID and the fact ID that you want to check. You set the value of the Response Quality Parameter to -1. This function will then return the RF for that fact ID as it is calculated by applying the cognitive model to the profile's data.

b) If you also know the Response Quality of the student then, by supplying it to the function, you will get the calculated RF, but you will also allow the system to update the information on this fact by applying the RQ to the cognitive model. As it can be understood, in cases where the RQ can not be measured, one can always use -1 as a value and use the rest of the model. If your case has the ability to track RQs, then you can easily use them too.

9.1.5 ResetFactDatabase (aProfileId)

With this function you can request from the system to clear all cognitive data (facts, timestamps, Rqs, etc.) for a specific profile. This function should be used with caution because it can not be undone.

9.2 Interoperability

By exposing the whole system and the research done behind it via web services, we hope that other researchers will be inclined to use it “as is” and gain in development time, while at the same time the system will be tested in different environments and under different sets of rules and restrictions (domain-depended or not). The more the system is used, the better it can become. Gathering more statistical data will help future upgrades and enhancements in both the Ebbinghaus' cognitive model and the system's architectural design and implementation.

10. Evaluating the Memory Retention Model

After expanding Ebbinghaus' mathematical formula (chapter 6) and implementing a way to initialize the personalized data for any student profile created (chapter 8), the time came to put everything inside an evaluation platform and test the effectiveness of the project.

The attractiveness of software games has often been considered a promising means for the creation of attractive educational software. Indeed, there are many researchers and educators that advocate the use of software games for the purposes of education. Papert (1993) notes that software games teach children that some forms of learning are fast-paced, immensely compelling, and rewarding whereas by comparison, school strikes many young people as slow and boring. As a result, many researchers have developed games for educational purposes (e.g. Inkpen et al 1994, Amory et al 1998, Conati & Zhou 2002). For this reason, the "two-of-a-kind" memory game was chosen to provide the base for the evaluation platform and experiment.

One more problem every evaluator faces is how s/he can acquire a sufficient amount of data that will be enough to solidify his/her work. Since a computer game has been chosen to hide the evaluation environment, the specified problem was converted to a different one: How can someone find as many players as possible and motivate them to play this game? This problem consists of two layers: One is how to make the game easily available to a great number of players, while the second is to find a good motive for them to play!

The first layer of the problem is easily addressed and solved if the correct implementation of the game is chosen. To be more exact, the game must have the ability to be easily distributed to a great number of players. This can be achieved if the game runs over Internet. This way, every computer game player, that has Internet access, automatically has access to the evaluation environment. Given the fact that nowadays a game must have the ability to be playable over Internet, or it has no future at all, there are already a great number of players who are accustomed to using Internet for their gaming hours. By implementing the game so as to be able to run online (i.e. Internet), this great number of already existing and trained "test subjects" can be tapped.

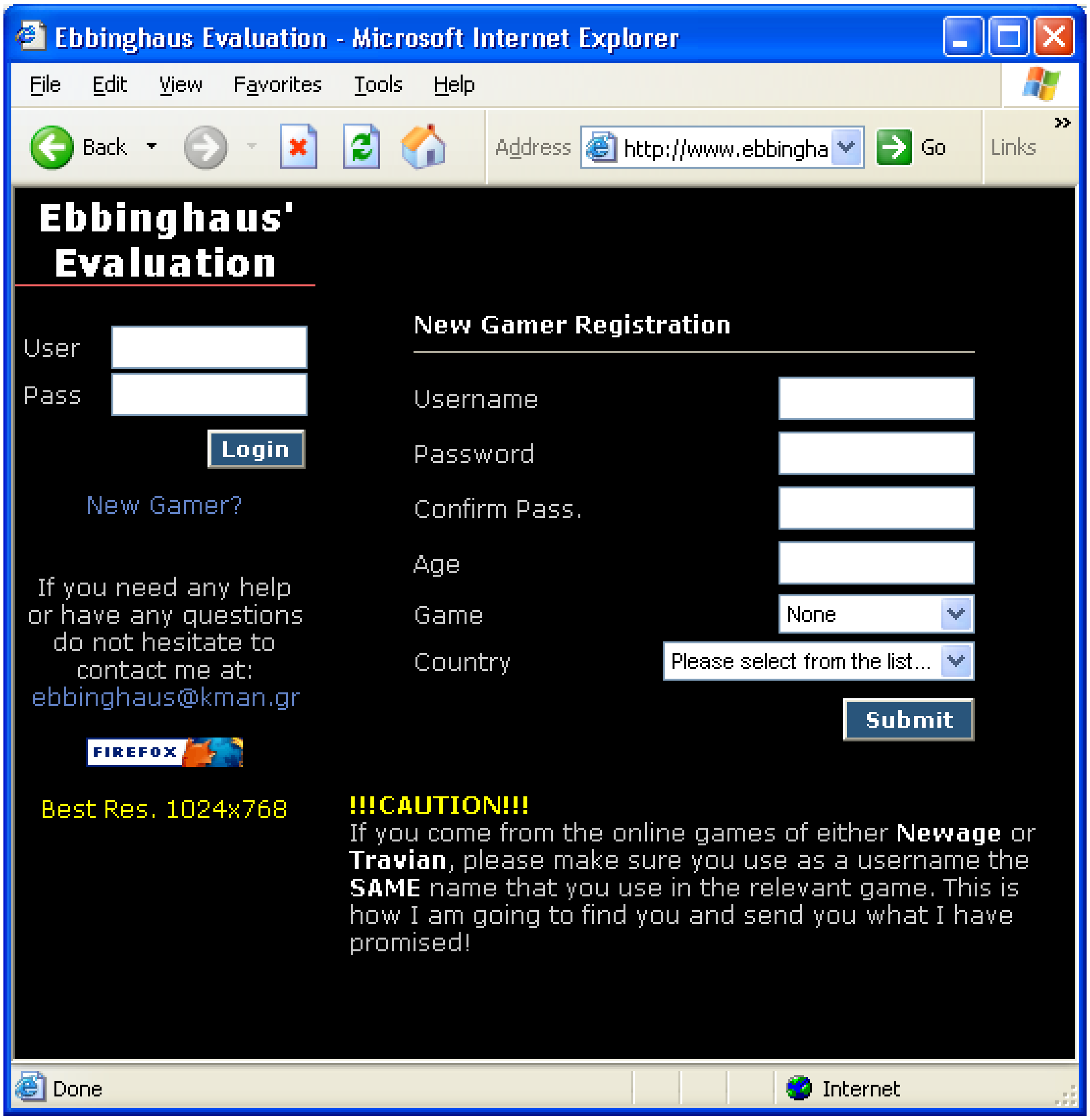
To gain access to a large pool of potential players, two player accounts were created in two different and very famous games (Newage3 Online and Travian). Both games were real time resource planning strategy games. This means that their core function is the gathering and handling of specific resources (oil, wood, iron, etc). In the course of the 5 months, a very good amount of these resources were gathered in each of these two accounts. Alongside the gathering of resources, communication channels with various other players in these games were established and maintained.

As it can be easily understood, the second layer of the problem was solved by deciding to "pay" the players for their gaming time. Therefore, a contest was held in both games. The players were invited to play the memory game by informing them that it was for a student's university project and also that they would be rewarded. The rewards would be "resources" for each game played. This meant that the corresponding resources would be transferred (in game) form the relevant accounts to the players playing the memory game. The results of this approach were far better than expected and are discussed in greater detail later in this chapter.

10.1 The game "Two-of-a-kind"

"Two-of-a-kind" is a classical memory-related puzzle game. The game consists of a number of pairs of cards. Depending on the difficulty level you may have more or less pairs of cards. The bigger the number of pairs, the greater the difficulty of the game will be.

When setting up the board of the game, all the cards are gathered and shuffled together, then they are placed face-down forming a square layout. After the board is set up, the player is asked to flip two cards. If the cards represent the same image, then a matching pair is found and the player proceeds to the next one. If the cards do not match then they are flipped back to the face-down position and the player continues by selecting the next pair. This procedure continues until all pairs have been found.


Ebbinghaus Evaluation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://www.ebbingha
Go
Links
FIREFOX
Done
Internet

Ebbinghaus' Evaluation

User
Pass
Login

New Gamer?

If you need any help or have any questions do not hesitate to contact me at: ebbinghaus@kman.gr

Best Res. 1024x768

New Gamer Registration

Username	
Password	
Confirm Pass.	
Age	
Game	None
Country	Please select from the list...

Submit

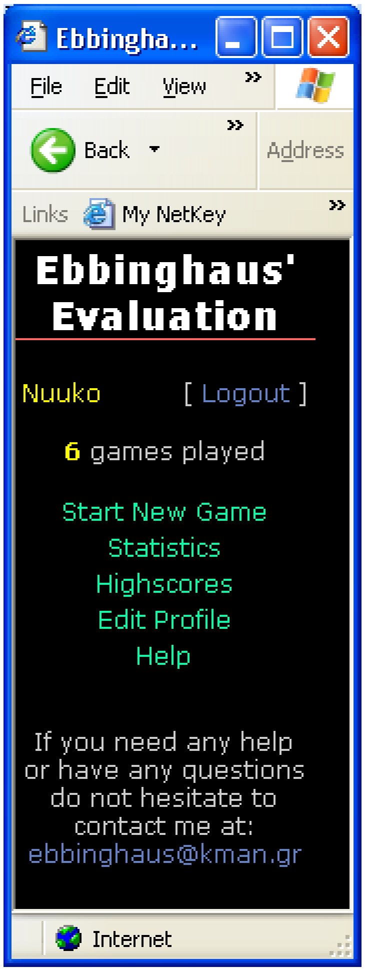
!!!CAUTION!!!
If you come from the online games of either **Newage** or **Travian**, please make sure you use as a username the **SAME** name that you use in the relevant game. This is how I am going to find you and send you what I have promised!

Figure 1. Players' profile Information

10.2 Implementation

Considering all the aforementioned requirements, the game was implemented using DHTML, PHP and MySQL. The player is required to create an account by filling out a form with some personal data and by selecting the game where the "resources" should be sent. This information is stored in an online database instance of MySQL and helps the tracking of our formula's effectiveness across ages and countries. A screen shot of the players' profile can be seen in Figure 1.

After a player has successfully created a new profile s/he can login to the game. From then on, s/he has the following options (as seen in Figure 2):



Ebbinghaus' Evaluation

Nuuko [Logout]

6 games played

Start New Game
Statistics
Highscores
Edit Profile
Help

If you need any help or have any questions do not hesitate to contact me at: ebbinghaus@kman.gr

Figure 2. Menu

- **Start New Game.** With this option the player may begin a new game session. All the choices s/he makes during the session are recorded in the game's engine. Since the exact timing of the various moves is of great importance, measures were taken to ensure that nothing would distort them. An example of such a distortion would appear if we chose to save each event at the moment of execution. Given the fact that the game runs on the player's computer while the database is on a remote server, the communication between these two computers would create a delay in the game process. Although this delay is very small when a high speed Internet connection is used, it would still restrict the evaluation platform and would create a set of cases which could generate an amount of data distortion.

The solution to this problem was to implement the game using HTML and JavaScript in such a way that during the game no communication was needed with the server. The game runs completely inside the Internet Browser of the player, storing locally all the necessary data. When the game ends, all the data are sent to the server for storage and analysis.

In that way, the timestamps generated during the game session are genuine and clean of any kind of distortion from the environment (Internet Connection Speed, CPU Speed, hardware, etc). A screenshot of the game environment can be seen in Figure 3.

As seen below, a deck of cards was selected for this implementation of the "two-of-a-kind" game. The cards used are from number 4 to 10. It was decided to exclude all kings, queens, and jacks because these cards are easier to memorize due to their optical difference with the other cards. The same stands for aces, twos, and threes (they all have too few marks on them). Furthermore, only hearts and spades were used so that the difficulty of the game wouldn't be increased.

As it can be understood with the given layout, the player is required to remember two different variants, color and number. If families of the same color were used, then the player should also remember the family, which would make the game harder and would add a variant that would tamper with the final data. As it has been stated previously, the goal of the evaluation platform was to create as "sterile" an environment as possible, which also included removing information that the player is required to remember.

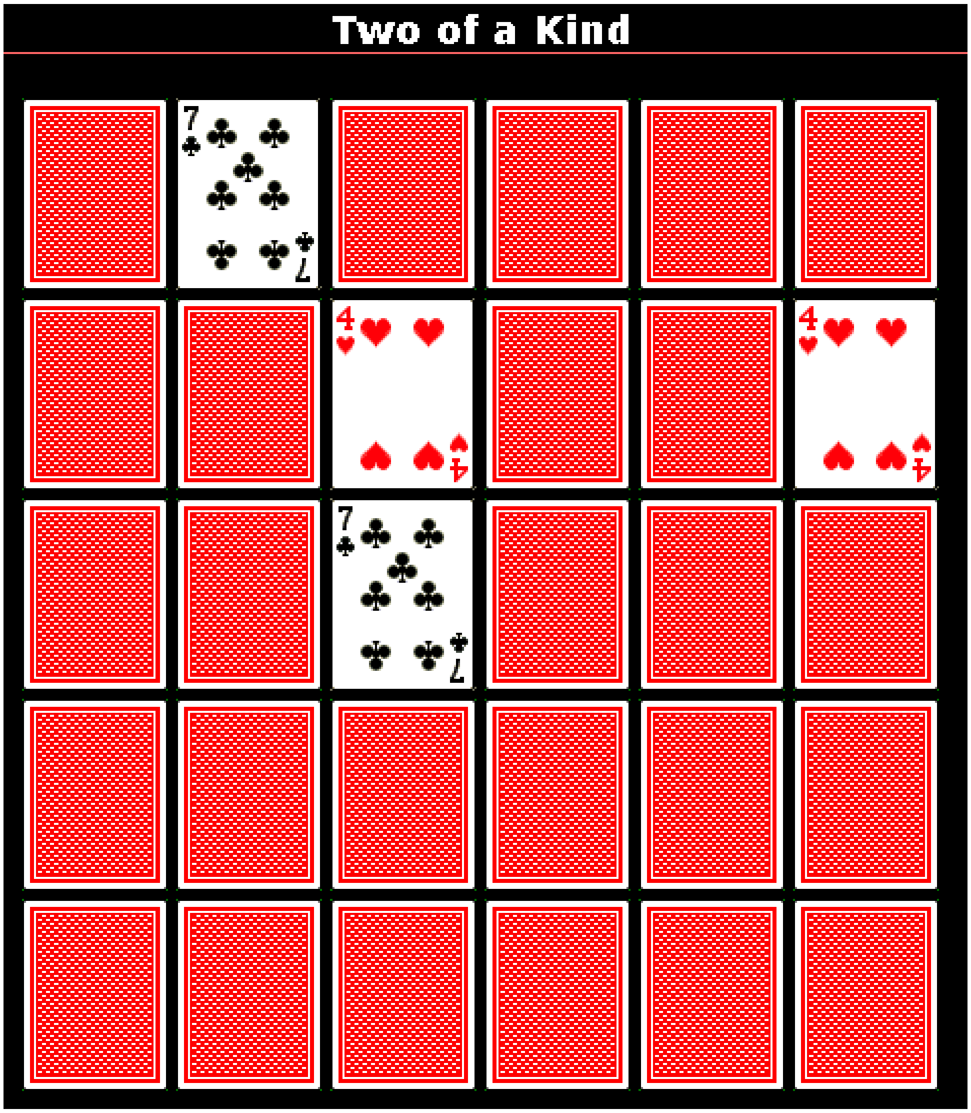
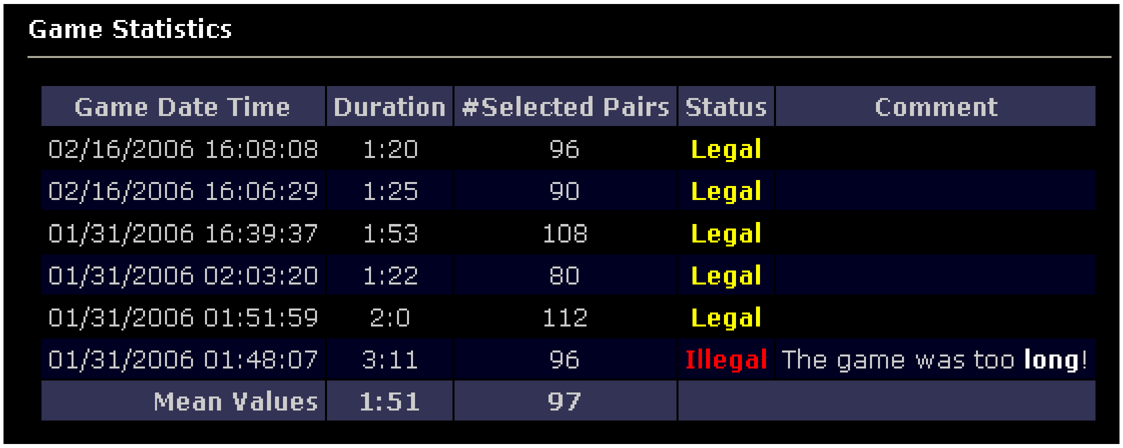


Figure 3. Game Board

- **Statistics.** With this option the player may see some statistics of the sessions s/he has already played. The information displayed includes the timestamp of the beginning of the game, the duration (in minutes), the number of different pairs that s/he selected until s/he managed to clear the board, and a status. The status can be "legal" or "illegal". In the second case, the "Comment" column is also filled indicating the reason why the session has been rejected. These reasons can be twofold. First, the session might have taken too long. An upper bound of three minutes to complete the session has been established. Three minutes are more than enough for a player that is dedicated to the game to complete the board. If the session takes longer then it can be safely assumed that the player was not focused on the task so the statistics should be discarded. Second, a player may be very "lucky" and clear the board without him testing his memory skills. The engine can track whether a hit is made by luck or the user knew where the card was placed, so if the number of "lucky shots" is more than three then the session is discarded. A screenshot of the statistics can be seen in Figure 4.

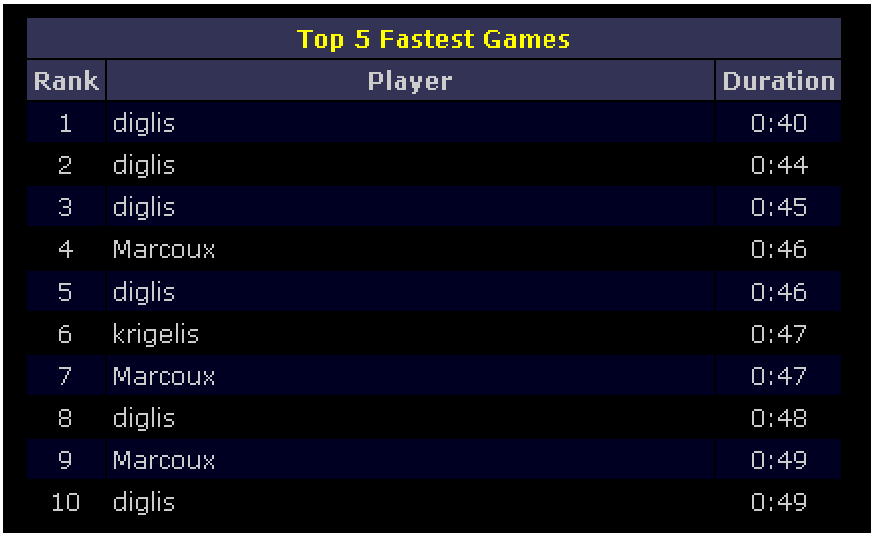


Game Statistics

Game Date Time	Duration	#Selected Pairs	Status	Comment
02/16/2006 16:08:08	1:20	96	Legal	
02/16/2006 16:06:29	1:25	90	Legal	
01/31/2006 16:39:37	1:53	108	Legal	
01/31/2006 02:03:20	1:22	80	Legal	
01/31/2006 01:51:59	2:0	112	Legal	
01/31/2006 01:48:07	3:11	96	Illegal	The game was too **long**!
Mean Values	1:51	97		

Figure 4. Player's Sessions' Statistics

- **Highscores.** In this screen, the player sees which players have achieved "top scores" in different sectors of the game. These sectors are speed and memory retention capabilities. The first sector is easy to understand. It consists of the players with the fastest legal game sessions. You can see a screen shot of that board in Figure 5.

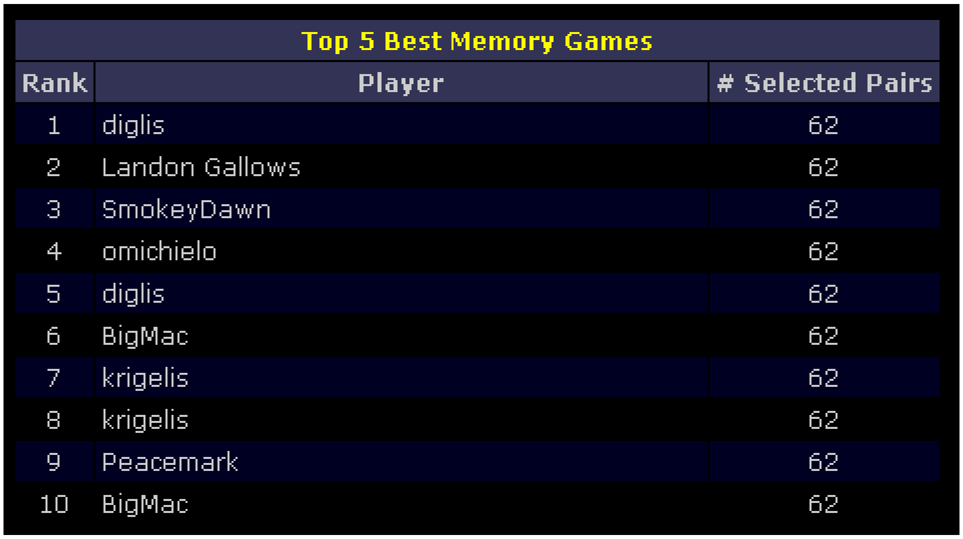


Top 5 Fastest Games

Rank	Player	Duration
1	diglis	0:40
2	diglis	0:44
3	diglis	0:45
4	Marcoux	0:46
5	diglis	0:46
6	krigelis	0:47
7	Marcoux	0:47
8	diglis	0:48
9	Marcoux	0:49
10	diglis	0:49

Figure 5. Fastest Games' Score Board

For the second sector, it should be noted that the way the memory retention capabilities are measured is by counting the number of pairs selected to complete a session without it being characterized as "lucky". The fewer the number of these pairs, the better the memorization capabilities of the player. A screenshot of this board can be seen in Figure 6.

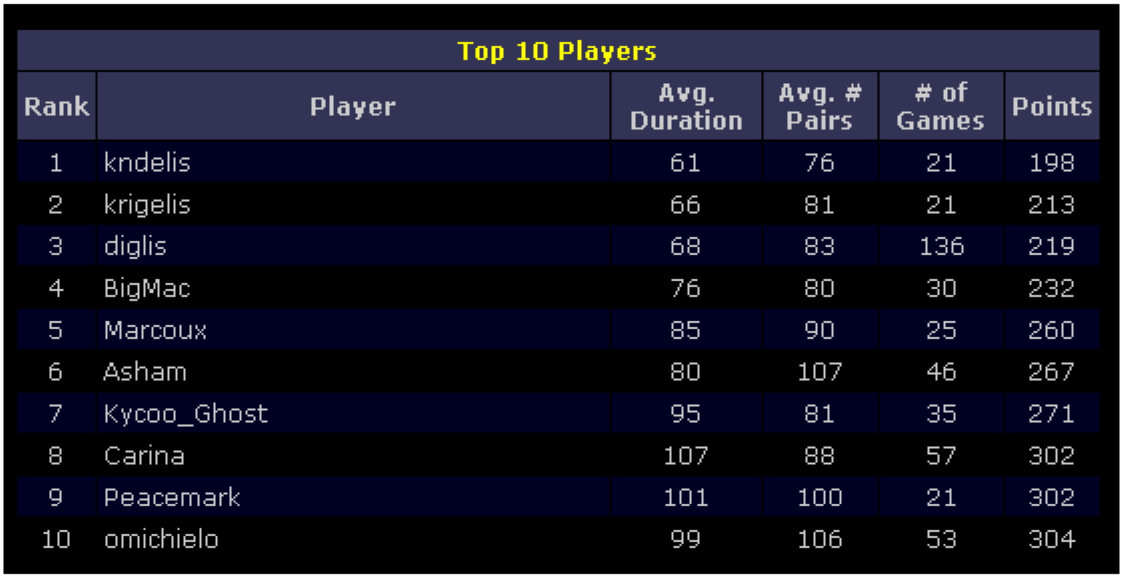


Top 5 Best Memory Games

Rank	Player	# Selected Pairs
1	diglis	62
2	Landon Gallows	62
3	SmokeyDawn	62
4	omichielo	62
5	diglis	62
6	BigMac	62
7	krigelis	62
8	krigelis	62
9	Peacemark	62
10	BigMac	62

Figure 6. Memory Retention Score Board

Apart from these two sectors, a "catch-all" high score board was also implemented. There, the two sectors are combined giving out the best player in both. For this reason, a point system was created in which a player earns one point for each pair s/he selects during a game session, while s/he gets two points for each second passed. The session with the least number of points is the best. Thus, the player with the least number of points is the best. The Top 10 Players' Scoreboard can be seen in the Figure 7.

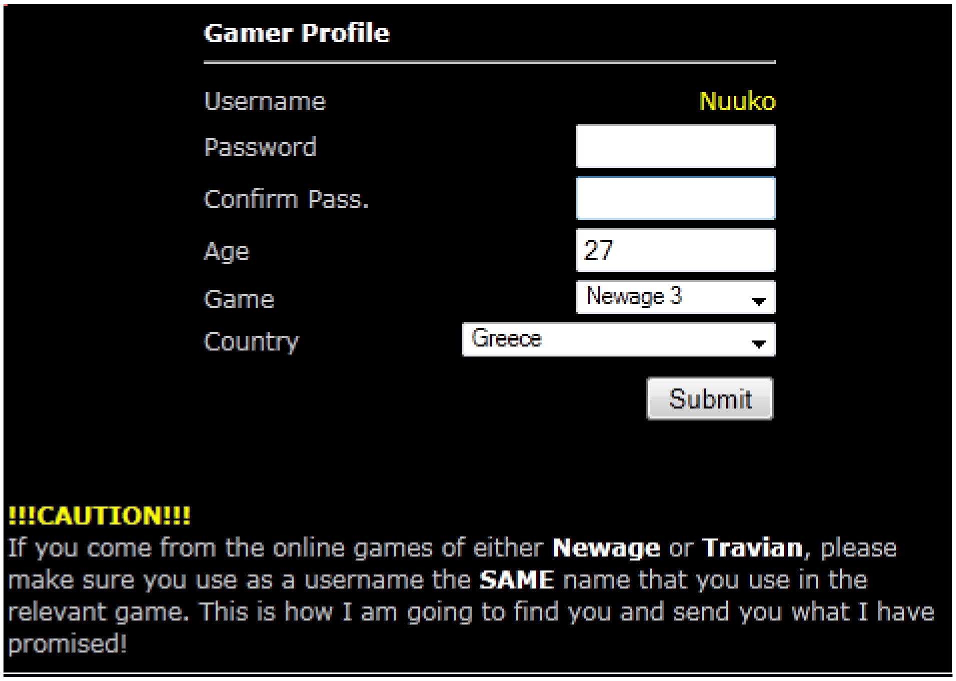


Top 10 Players					
Rank	Player	Avg. Duration	Avg. # Pairs	# of Games	Points
1	kndelis	61	76	21	198
2	krigelis	66	81	21	213
3	diglis	68	83	136	219
4	BigMac	76	80	30	232
5	Marcoux	85	90	25	260
6	Asham	80	107	46	267
7	Kycoo_Ghost	95	81	35	271
8	Carina	107	88	57	302
9	Peacemark	101	100	21	302
10	omichielo	99	106	53	304

Figure 7. Top 10 Players' Scoreboard

Since the player named *diglis* was one of the fastest gamers, he managed to have his name multiple times on the relevant scoreboard. Unfortunately for him, his average performance was not that good, so his position in the Top 10 Players' list was not as good.

- **Edit Profile.** With this option, a player can edit some of his/her profile settings. S/he can change everything except the username.

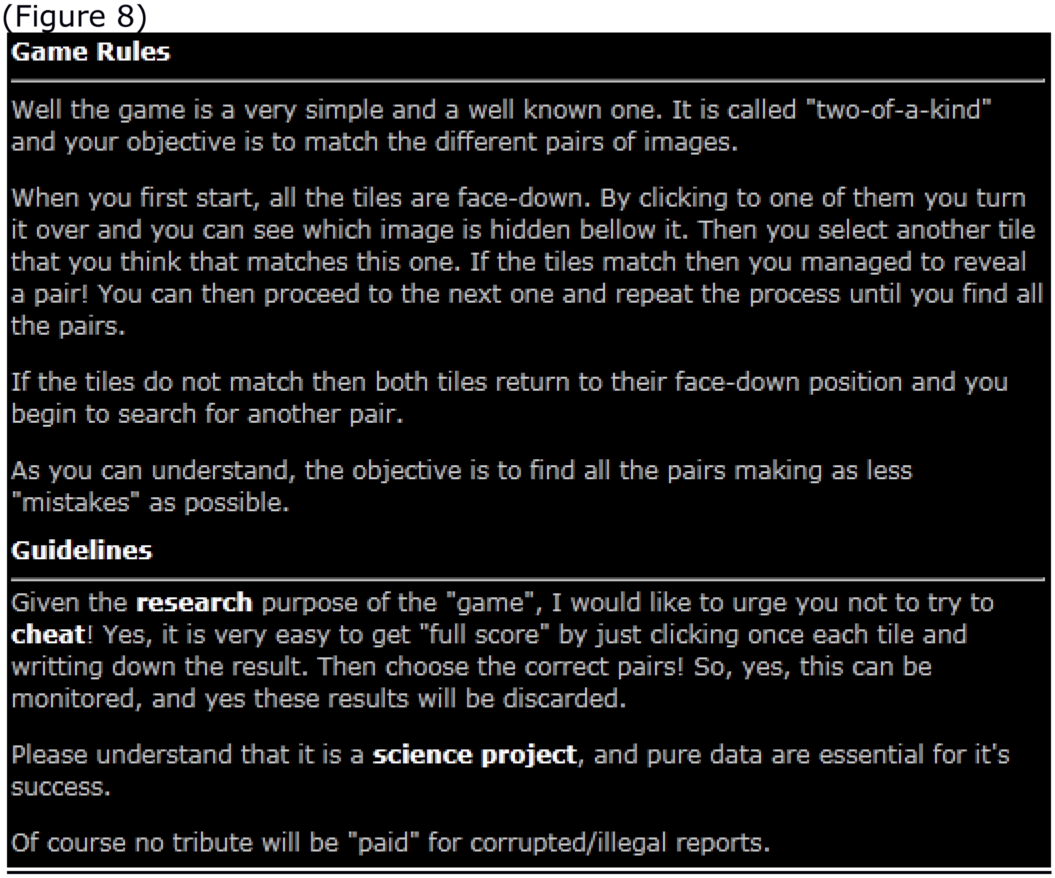


Gamer Profile

Username	Nuuko
Password	
Confirm Pass.	
Age	27
Game	Newage 3
Country	Greece

!!!CAUTION!!!
If you come from the online games of either **Newage** or **Travian**, please make sure you use as a username the **SAME** name that you use in the relevant game. This is how I am going to find you and send you what I have promised!

- **Help.** This option displays a small help document which gives the basic guidelines of how the game is played along with the "house-rules". (Figure 8)



Game Rules

Well the game is a very simple and a well known one. It is called "two-of-a-kind" and your objective is to match the different pairs of images.

When you first start, all the tiles are face-down. By clicking to one of them you turn it over and you can see which image is hidden bellow it. Then you select another tile that you think that matches this one. If the tiles match then you managed to reveal a pair! You can then proceed to the next one and repeat the process until you find all the pairs.

If the tiles do not match then both tiles return to their face-down position and you begin to search for another pair.

As you can understand, the objective is to find all the pairs making as less "mistakes" as possible.

Guidelines

Given the **research** purpose of the "game", I would like to urge you not to try to **cheat**! Yes, it is very easy to get "full score" by just clicking once each tile and writting down the result. Then choose the correct pairs! So, yes, this can be monitored, and yes these results will be discarded.

Please understand that it is a **science project**, and pure data are essential for it's success.

Of course no tribute will be "paid" for corrupted/illegal reports.

Figure 8. Help notes

10.3 Evaluation Environment's Effectiveness Results

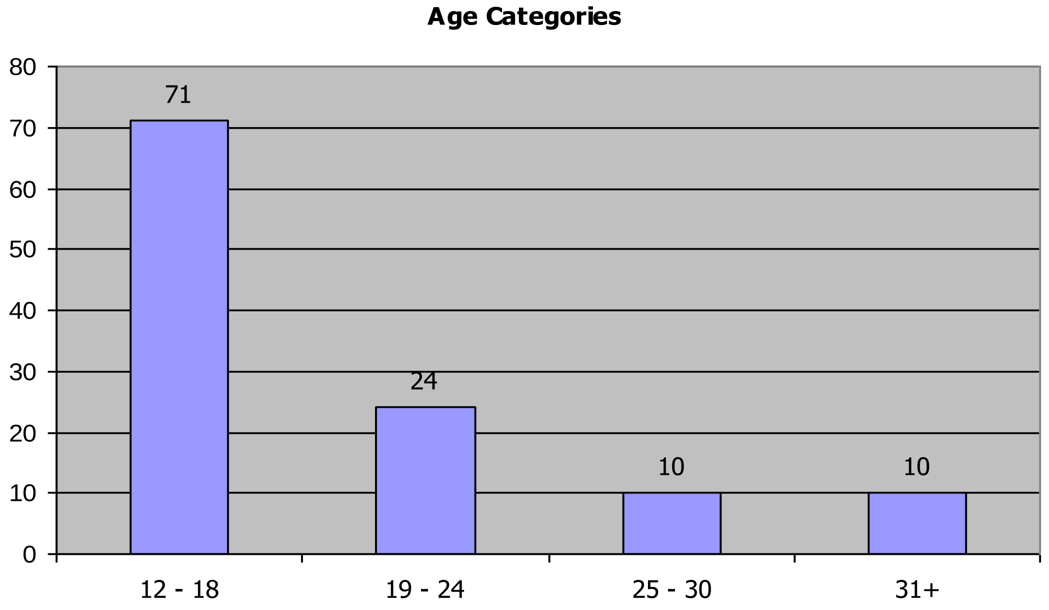
The project as a whole had far better results than expected. At first, the game managed to attract and hold the attention of the players far more than what was initially hoped. This can be proved by the following facts:

1. The initial agreement with the players was that they would be paid 1000 gold pieces (or resources – depending on the game) for each game they played to a maximum of 20 games. If someone wanted to play more games, s/he could, but no rewards would be paid. It was expected for everyone to play only 20 games, but this was not the outcome. After the first games were played, the high-score board was implemented for the players to see their status. This was enough to create a feeling of competition between the players, who started trying to win the first position on the board. The evaluation platform managed to provide them with a far better motive than a mere reward.

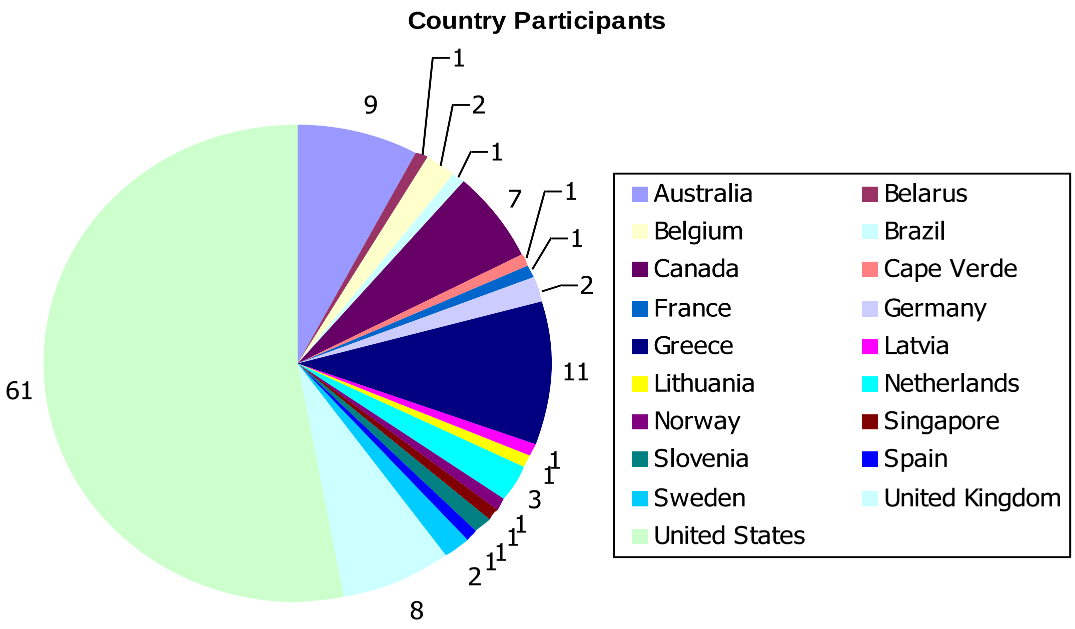
2. Many players were students and they sympathized with the project's efforts. To help, they even contributed to our cause by sending gold

pieces/resources to keep the competition going. The number of active players was far more than it was initially expected and the resources gathered in the two accounts were not enough to cover all the rewards. However, with the students' extra help, the competition ended up being a great success.

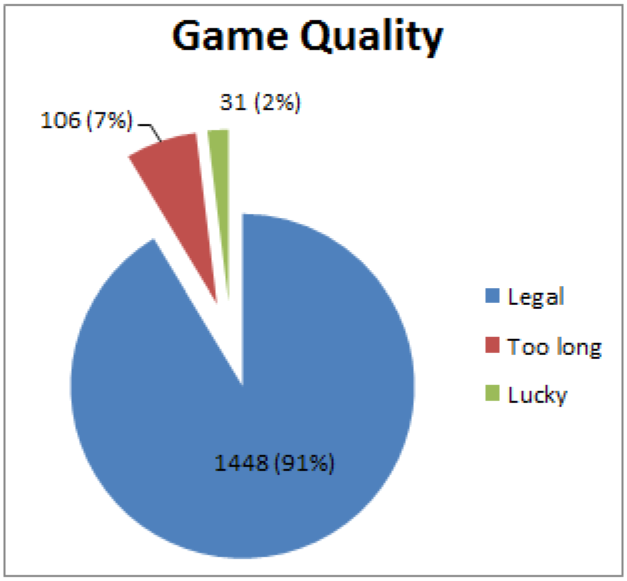
3. 115 players from different age categories and countries played 1585 memory games. In the charts bellow, the distribution of the players across different age categories and countries is displayed. As it was expected, most of the players were of ages 12 to 18. After all, the final goal was the creation of a model that can be helpful in the production of educational software; therefore, this was exactly the target group. However, it was fortunate that the formula had the opportunity to be tested on older and more mature players as well.



Also, although the players were mainly from United States, there were many players from different countries around the world. By examining these charts, the value of evaluation over the Internet becomes clear. Using the Internet is the only way to run such an evaluation, with so many players from different countries, while keeping the resources (costs) at a minimum.



4. Finally, the algorithms used to detect and discard illegal games were also proven to be extremely helpful in keeping the final set of data as clean as possible. In the sum of 1585 games that were played, 106 "long" games and 31 "lucky" ones were rejected. The relevant data can be seen in the pie chart below:



10.4 Mathematical Model Evaluation Results

At the end of the contest, there were 1585 games in the evaluation's database. For each of the games, all user actions were recorded. These actions include:

a) The first time a player saw each card

b) The time when the player successfully matched a pair

c) The time when a player falsely chose a card

These data need to be filtered so that only the events that actually give information on the player’s memory retention capabilities are kept. For example, if at the time of choosing the second card a successful match was found, but the second card was encountered for the first time, then this is a lucky match and all events associated with those two cards need to be discarded.

This information is then applied to the Ebbinghaus' memory retention mathematical formula, so that the Basic Retention Factor (BRF) of each player can be calculated.

For that to be accomplished the following assumptions were made:

a) The Response Quality Factor (RQ) takes only two values, 0 if the match was wrong and 3 if the match was correct.

b) Given the fact that the time each game lasts is less than a minute, the actual effect of the Memorization Ability Factor in the specific case cannot be measured effectively. As it was mentioned in Chapter 6 (paragraph 6.2.2), the Memorization Ability factor is used to adjust the time an event is retained in the memory of the test-subject, but the modification applied is in minutes (from 3 up to 6). For that reason, when applying the calculations to the formula, a Memorization Ability Factor of 2 is used for all cases.

The process that was implemented was the following: For each matching pair, the time difference between two events was calculated. Applying this difference to the main formula, the Retention Factor (RF) was calculated. Furthermore, the result of the match (whether it was a correct one or not) was noted. Therefore, a complete set of RFs plus actual results was created. By applying the technique described in Chapter 6 (paragraph 6.2.1), the Base Retention Percentage (BRP) of the player was calculated.

It is clear that the more results one feeds into the technique, the better the BRP calculation. This experiment has proven the fact that after 12 pairs (given the fact that the pairs are already filtered as described previously to include only valid ones) the modification to the BRP calculated never exceeded +/-1%. This was true even for players that played 40+ games.

This proves that the mathematical model and the personalized extension that was proposed in this dissertation, is accurate when applied to retention capabilities of non-domain specific information (like numbers and colors).

11. Conclusion

The main goal of this thesis was to find a way to enhance the student modeling techniques used by educational software. To accomplish this task a set of necessary tools was created which would provide the base for a test environment and an evaluation platform.

These tools included an Intelligent Tutoring System (ITS) masked as a Virtual Reality Game called VR-Engage and an authoring tool for that system called VR-Integate.

11.1 Contribution to Educational Games development

In VR-Engage the student was asked to explore 4 different virtual worlds in a quest to find the lost pages from the Book of Wisdom (Virvou et al, 2002). Each virtual world is linked to a different knowledge domain (Mathematics, Geography, Grammar and Spelling). In order to accomplish this task the student had to find his/her way out of 4 labyrinths by opening doors guarded by virtual dragons. To open each door the student had to answer a domain-specific question.

In the process of creating VR-Engage it became apparent that an easy way was needed to be able to change the domain specific information/data of the game. The final goal was for teachers to be able to "feed" the virtual worlds with domain theory and questions, and to be able to customize the student's experience, tailoring it to their respected needs. At that point VR-Integate was created (Virvou et all, 2002). VR-Integate started as plain authoring tool for the VR-Engage platform. The teacher could use it and easily generate/update/delete all the domain content for each of the different virtual worlds.

The next step was to upgrade the system to be able to track the student's performance in various different sectors. For that, different modules were created which helped generate the student's educational model inside the virtual environment. These modules were able to track the quality of the student's answers in various questions s/he faced, the difficulty s/he might had to navigate through the virtual environment (Virvou et all, 2002), and even measure how focused on the game a student was.

By applying these techniques, a robust educational platform has been created. The final platform via the use of the authoring tools was flexible, dynamic and easily adaptable to fit the needs of any teacher. The resulting tools were extremely easy to learn how to use, requiring only minimum effort from the teacher.

11.2 Contribution to Student Modeling

During the evaluation process of the VR-Engage platform, a serious problem appeared. Whenever a teacher was creating a new domain-specific world, s/he had to test its effectiveness by letting the students play it. There was no way to have any kind of evaluation information in advance and before releasing the lesson to the students. The solution to this problem came from the Cognitive Psychology sector.

Based on the initial research conducted by Ebbinghaus (1998), the student model was enhanced by extending the system to be able to track the memory retention capabilities of the students.

11.3 Contribution to the evaluation of educational software

Using this new module, the creation of Virtual Agents that could actually "play" the game, simulating different students (depending on the profile they were using) was made possible. By feeding the agents with different student profiles, the virtual world created by the teacher could be evaluated before it was released to the classroom. By using the memory retention module, the virtual agents could report the possible amount of information each student could retain after the end of the game, giving the teacher the opportunity to readjust the domain-specific questions and theory so that s/he could achieve the desired outcome.

Furthermore, the use of the virtual agents was extended even more. Instead of using the agents only as a "pre"-evaluation process, they were integrated inside the VR-Engage platform for real time tracking and reporting. While the student was interacting with the environment, all information was fed to the virtual agent. The agent then, using the specific student's profile, could generate reports on what the student is supposed to have assimilated and what s/he has forgotten. Using this information the system could then on-the-fly readjust the virtual world so that the student would be forced to repeat parts of the lesson that the agent reported as "forgotten".

With those two different applications of the extended memory retention formula, it was made possible to create, for the first time, such a development process for an educational game that not only it was pre-evaluated before being released to the classroom, but the final ITS had also the ability to track in real-time the effectiveness of the lesson and the student's performance.

11.4 Contribution to Applied Cognitive Psychology

The initial mathematical models proposed by Klatzky, Anderson and Ebbinghaus were very abstract and general. Although they could successfully simulate the memory retention of facts most of the times, in real life not all human minds work in the same way. Others memorize much faster and for longer periods while others need to repeat the study over and over again until they can retain the same amount of information.

Thus, the mathematical formula proposed was extended with the use of several factors that made it possible to personalize it. The Base Retention Factor (BRF) and Memorization Ability (MA) have been introduced which helped in the creation of a retention model much more tailored to each student.

The new formula has been extensively evaluated using an online memory game implemented for the said cause. The online memory game was played by hundreds of players providing thousands of games and as a result a very extensive database of events that were used to prove the effectiveness of the formula.

11.5 Future work

By integrating cognitive psychology to the student modeling of intelligent tutoring systems, a new research area was revealed. The initial mathematical model to simulate the memory retention capabilities (Ebbinghaus, 1998) was successfully extended to become more personalized. The next steps in the field could be the definition of different memory retention factors for each domain being taught. For example, not all students are as good in all domains. This could be tracked by keeping different BRFs and Mas for each domain and maybe using all these factors to calculate the overall memory retention capabilities of a subject.

For that reason it was decided (after having numerous inquires) to open the platform to all researchers wanting to follow this road. The mathematical model and it's application was exposed via the use of Internet Web Services (SOAP) so that anyone wishing to use it would not have to implement it from scratch.

Truth be told, the applications of this extensive model, are too many and very important. Especially for the educational software development, it can radically improve the effectiveness of both the lesson designed by the teacher, and the final performance of the student.

REFERENCES

A Brief History of Computing © Copyright 1996-2005, Stephen White

A Task Oriented Natural Language Understanding Model, T.Tan, Y.Shi, and W.Gao (Eds): ICMI 2000, LNCS 1948, pp 526-533, 2000.

"Action computer games can sharpen eyesight" retrieved from New Scientist Tech http://technology.newscientist.com/channel/tech/dn11123-action-computer-games-can-sharpen-eyesight.html

Aliya, S., K. (2002) "The role of computer games in the development of theoretical analysis, flexibility and reflective thinking in children: A longitudinal study." In International Journal of Psychophysiology, 45, p 149.

Amory, A., Naicker, K., Vincent, J. & Claudia, A. "Computer Games as a Learning Resource". Proceedings of ED-MEDIA, ED-TELECOM 1998. World Conference on Educational Multimedia and Educational Telecommunications, Vol.1, 1998, pp. 50-55.

An Agent model for NL dialog interfaces, F. Giunchiglia (Ed.): AIMSA '98, LNAI 1480, pp. 14-27, 1998

Anderson J.R. (2000). "Learning and Memory: An Integrated Approach" (2nd ed.). John Wiley & Sons, Inc.

Andreal L.Ames & David R.Nadeau & John L.Moreland, "VRML 2.0 Sourcebook", Wiley, ISBN 0-471-16507-7

Andriessen J. and Sanberg. "Where is Education Heading and How about AI?". International Journal of Artificial Intelligence in Education, 10, 1999, pp. 130-150.

Baker, M. "A Model for Negotiation in Teaching-Learning Dialogues". International Journal of Artificial Intelligence in Education, 5(2), 1994, pp. 199-254.

Boyle T. "Design for Multimedia Learning". Prentice Hall, 1997.

Bulletin of the British Psychological Society 32 (Feb): 80-80 1979

Brody, H. (1993) "Video Games that Teach?". In Technology Review, November/December 1993, pp. 51-57.

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

Collins A. and Michalski R. "The Logic of Plausible Reasoning: A core Theory", Cognitive Science 13, 1989, pp. 1-49.

Conati, C. & Zhou, X. (2002) " Modeling students' emotions from cognitive appraisal in educational games". In S. A. Cerri, G. Gouarderes and F. Paraguacu (Eds.)

Intelligent Tutoring Systems 2002, LNCS, 2363, pp. 944-954, Springer-Verlag Berlin Heidelberg 2002.

Connectionist Analysis and Creation of Context for Natural Language Understanding and Knowledge Management, P. Bouquet et al. (Eds): CONTEXT'99, LNAI 1688, pp. 479-482, 1999.

Cortona VRML Client, Parallel Graphics, http://www.parallelgraphics.com

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

Cunningham, H.: Gender and computer games. Media Education Journal, 17, (1994) 13-14.

Danny Thorpe, "Delphi Component Design", Addison Wesley Developer's Press, 1997, ISBN 0-201-46136-6

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1993) Human-Computer Interaction. New York: Prentice-Hall.

Downes, T.: Playing with computer technologies in the home. Education and Information Technologies, 4, (1999) 65-79.

Dudani, S. (1976), The distance-weighted k-nearest-neighbor rule, IEEE Transactions on Systems,Man and Cybernetics, 6(4), 325-327.

Ebbinghaus, H. (1998) "Classics in Psychology, 1885: Vol. 20, Memory", R.H. Wozniak (Ed.), Thoemmes Press, 1998

El-Sheikh E. & Sticklen, J.: Generating Intelligent Tutoring Systems from Reusable Components and Knowledge-Based Systems. In S.A. Cerri, G. Gouarderes and F. Paraguacu (Eds.): ITS 2002, LNCS 2363, (2002), 199-207.

Gerdt, P., Kommers, P. Looi, C-K & Sutinen, E. (2001) "Woven stories as a cognitive tool". In M. Benyon, C.L. Nehaniv and K. Dautenhahn (Eds.): Cognitive Technology (CT) 2001, LNAI 2117, pp. 233-247, Springer-Verlag Berlin Heidelberg 2001.

Gilbert, L. (1999) "Some Valuable Lessons from the Teaching and Learning Technology Programme in the U.K", Journal of Interactive Learning Research, 10, 1, 67-85.

Goldstein, I. (1982). "The Genetic Graph: A Representation for the Evolution of Procedural Knowledge". In D. Sleeman & L. Brown (Eds.), Intelligent Tutoring Systems. London: Academic Press.

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

Griffiths, M.D. & Hunt, N. (1995) "Computer game playing in adolescence: Prevalence and demographic indicators". In Journal of Community and Applied Social Psychology, 5, pp. 189-193.

Hasebrook, J.P. & Gremm, M. (1999) "Multimedia for Vocational Guidance: Effects of Individualized Testing, Videos and Photography on Acceptance and Recall". Journal of Educational Multimedia and Hypermedia, 8(4), pp. 377-400.

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

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

ID-Software (1993), Doom - Virtual Reality Computer Game, Id Software Company Texas.

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.

Intelligent Virtual Environments, T. Panayiotopoulos, S. Vosinakis, N.Avradinis, Seminar/Workshop on Intelligent Agents and Virtual Reality, Artificial Intelligence Human Networks, Athens University of Business and Economics, pp.3-8, June 29, 2001.

Interacting with Computers, Vol. 11, No 5, pp. 499-516.

Jacob - An Animated Instruction Agent in Virtual Reality, T.Tan, Y.Shi, and W.Gao (Eds): ICMI 2000, LNCS 1948, pp 526-533, 2000.

James Allen, "Natural Language Understanding", Benjamin/Communings Publishing Company, 1995, ISBN 0-8053-0334-0

Jayakanthan R. (2002): "Application of computer games in the field of education" Electronic Library vol. 20(2), pp. 98-102.

Jones, A., Kirkup, G. & Kirkwood, A. (1993) Personal Computers for Distance Learning, Paul Chapman, 1993.

Jones, A., Scanlon E., Tosunoglu C., Morris, E., Ross, S., Butcher, P. & Greenberg, J. "Contexts for Evaluating Educational Software". Interacting with Computers, Vol. 11, 1999, No 5, pp. 499-516.

Kabassi, K. & Virvou, M. (2003), Using Web Services for Personalised Web-based Learning.Educational Technology & Society, Journal of International Forum of Educational Technology & Society and IEEE Learning Technology Task Force, 6(3), 61-71.

Kabassi, K. & Virvou, M. (2004), Personalised Adult e-Training on Computer Use based on Multiple Attribute Decision Making. Interacting with Computers, 16(1), 115-132.

Kay, J. "Learner Control", User Modeling and User Adapted Interaction, 11, 2001, pp. 111-127.

Kerawalla, L. & Crook, C. (2002) "Children's computer use at home and at school: context and continuity". British Educational Research Journal, 28(6), pp. 751-771.

Kim, John. "What is a Role-Playing Game?", http://www.darkshire.net/~jhkim/rpg/whatis/

Kim, John. "Narrative" or "Tabletop" RPGs, http://www.darkshire.net/~jhkim/rpg/whatis/tabletop.html

Kirkman, C.: Computer experience and attitudes of 12-year-old students: implications for the U.K. national curriculum. Journal of Computer Assisted Learning, 9, (1993) 51-62.

Klatzky, R.L. (1980) "Human memory - Structure and Processes", W.H. Freedman and Co., San Francisco.

Kruchten, P.: "Rational Unified Process-An Introduction", Addison-Wesley, 1999.

Laurillard D. (1995) 'Multimedia and the Changing Experience of the Learner', British Journal of Educational Technology 26(3), pp. 179-189.

Learning to Generate CGs from Domain Specific Sentences, H. Delugach and G. Stumme (Eds): ICCS 2001, LNAI 2120, pp. 44-57, 2001.

Lester, J.C., Converse, S.A., Kahler, S.E., Barlow, S.T., Stone, B.A., Bhogal, R.S.: The persona effect: affective impact of animated pedagogical agents. In: CHI'97, Proceedings (1997) pp. 359-366

Linton, M. (1978) "Temporal codes for memories - Issues and problems - Underwood BJ", American Journal of Psychology, 91 (2): 364-367 1978.

Linton, M. (1979) "Real-world memory after 6 years- Invivo study of very long-term-memory" Bulletin of the British Psychological Society 32 (Feb): 80-80 1979

Manos, K. & Virvou, M. (2003) "Memory Features in Simulated Students to Improve the Software Engineering Process and the Performance of Intelligent Tutoring Systems" In Technology, Instruction, Cognition & Learning, Vol. 1, No 4, pp.303-322, OCP Science Journals.

Mark, M.A. & Greer, J. E. (1991). "The VCR Tutor: Evaluating instructional effectiveness" In Hammond K.J. & Gentner D. Q. (Eds.) Proceedings of the 13th Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 564-569.

Matthews, M., Pharr, W., Biswas G. & Neelakandan, (2000). "USCSH: An Active Intelligent Assistance System," Artificial Intelligence Review 14, pp. 121-141.

Mc Grenere, J. L. (1996) "Design: Educational Electronic Multi-Player Games. A Literature Review" Technical Report 96-12, the University of British Columbia, Vancouver, BC, V6T 1Z4, Canada.

McCalla, G. (1992) "The central importance of student modelling to intelligent tutoring". In E. Costa (Ed.) New Directions for Intelligent Tutoring Systems, Berlin: Springer Verlag.

McGraw, K.L. (1994) "Performance Support Systems: Integrating AI, Hypermedia and CBT to enhance user performance" International Journal of Artificial Intelligence in Education, vol 5(1), pp. 3-26.

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

Montgomery M. (1997) Developing a Laurillardian CAL Design Method, Proceedings of ED-MEDIA/ED-TELECOM World Conferences On Educational Multimedia, Hypermedia and Educational Telecommunications, Vol. 2, pp. 1322-1323.

Moore D. "A framework for using multimedia within argumentation systems". Journal of Educational Multimedia and Hypermedia 9(2), 2000, pp. 83-98.

Moundridou, M. & Virvou, M. (2001), Authoring and Delivering Adaptive Web-Based Textbooks using WEAR, In Okamoto, T., Hartley, R., Kinshuk & Klus, J.P. (eds.): IEEE International Conference on Advanced Learning Technologies ICALT 2001, 185-188.

Multi Modal Cooperation with the DenK System, Harry Bunt, Rene Ahn, Robbert-Jan Beun, Tijn Borghuis, and Kees van Overveld

Multi-agent Systems as Intelligent Virtual Environments, G. Anastassakis, T. Ritchings, T. Panayiotopoulos, KI 2001 : Advances in Artificial Intelligence, F. Baader, G. Brewka, T. Eiter (Eds.), Lecture Notes in AI, (LNAI 2174), Springer-Verlag, pp.381-395, 2001.

Muntaz, S. (2001) " Children's enjoyment and perception of computer use in the home and the school" In Computers & Education, 36 (2001), pp. 347-362.

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.

NAUTILUS (Navy AUTomated Intelligent Language Understanding System) - a general purpose natural language processing system, http://www.aic.nrl.navy.mil/~severett/vr.html

Ossa - A Conceptual Modelling System for Virtual Realities, H. Delugach and G. Stumme (Eds): ICCS 2001, LNAI 2120, pp. 333-345, 2001.

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

Papous: The Virtual Storyteller, A. Silva, M. Vala, A. Paiva, Intelligent Virtual Agents, A. de Antonio, R. Aylett, D. Ballin (Eds.), Lecture Notes in AI, (LNAI 2190), Springer-Verlag, pp.171-180, 2001.

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

Quatrani, T. (1998) "Visual Modeling with Rational Rose and UML", Addison-Wesley, 1998.

Research & Analysis for digital living technologies, Parks Associates http://www.parksassociates.com/press/press_releases/2005/gaming-1.html

Rich E. "Users are Individuals: Individualizing User Models", International Journal of Man-Machine Studies, 18, 1983, pp. 199-214.

Rivers, R. (1989) "Embedded User Models - Where Next?" Interacting with Computers 1, pp. 14-30.

Robert Sedgewick, "Algorithms", Addison Wesley, Second Edition 1998, ISBN 0-201-06673-4

Rokeby, D. (1998) "The construction of experience: Interface as content" In Dodsworth Jr., C. (Ed.) Digital Illusion: Entertaining the future with high technology, pp. 22-47, Addison-Wesley Publishing Co. 1998.

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

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.

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.

Shute, V., Glaser, R. & Raghaven, K. (1989). " Inference and Discovery in an Exploratory Laboratory" In Ackerman, P.L., Sternberg, R. J. & Glaser, R. (Eds.) Learning and Individual Differences, San Francisco: Freeman, pp. 279-326.

SimHuman : A Platform for Real-Time Virtual Agents with Planning Capabilities, S. Vosinakis, T. Panayiotopoulos, Intelligent Virtual Agents, A. de Antonio, R. Aylett, D. Ballin (Eds.), Lecture Notes in AI, (LNAI 2190), Springer-Verlag, pp.210-223, 2001.

Sison, R. & Shimura, M. (1998) "Student Modeling and Machine Learning". International Journal of Artificial Intelligence in Education, 9, 128-158.

Sleeman, D., Hirsh, H., Ellery, I. & Kim, I. (1990). " Extending domain theories: Two case studies in student modeling", Machine Learning, 5, pp. 11-37.

Smith, A., Whitney, H., Thomas, M., Perry, K., Brockman, P. "Effect of caffeine and noise on mood, performance and cardiovascular functioning." Human Psychopharmacology-Clinical and Experimental, 12, 1997, pp. 27-33.

"SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)". W3C. April 27, 2007. Retrieved 2011-02-01.

Sommerville, I. (1992) "Software Engineering, Reading," Mass.: Addison-Wesley Pub. Co.

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

Stansfield, J.C., Carr, B., & Goldstein, I.P. (1976) "Wumpus Advisor I: A First Implementation of a Program that Tutors Logical and Probabilistic Reasoning Skills". At Lab Memo 381. Massachusetts Institute of Technology, Cambridge, Massachusetts.

SynEdit, Open Source Editor, http://synedit.sourceforge.net

Tsiriga, V. & Virvou, M. (2004) "Evaluating the intelligent features of a web-based intelligent computer assisted language learning", International journal of AI tools, to appear.

VanLehn, K., Ohlsson, S. & Nason, R. (1994) "Applications of simulated students: An exploration", Journal of Artificial Intelligence in Education, 8, pp. 262-283.

Virtual Agent Societies with the mVITAL Intelligent Agent System, G. Anastassakis, T. Panayiotopoulos, T. Ritchings, Intelligent Virtual Agents, A. de Antonio, R. Aylett, D. Ballin (Eds.), Lecture Notes in AI, (LNAI 2190), Springer-Verlag, pp.112-125, 2001.

Virvou, M. (1999) Automatic Reasoning and Help about Human Errors in Using an Operating System, Interacting With Computers, 11(5), pp. 545-573.

Virvou M. "A Domain-Independent Reasoning Mechanism for an ITS Authoring Tool". Proceedings of ED-MEDIA 2000 World Conferences on Educational Multimedia, Hypermedia and Telecommunications, 2000, pp.1727-1729.

Virvou M. and Du Boulay B. "Human Plausible Reasoning for Intelligent Help". User Modeling and User Adapted Interaction, 9, 1999, pp. 321-375.

Virvou M. and Kabassi K. "Evaluation of the Advice Generator of an Intelligent Learning Environment". Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001), pp. 339-342.

Virvou, M. & Alepis E. (2004b), Mobile authoring and management of educational software applications: Usefulness and usability for teachers, In Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA) 2004, to appear.

Virvou, M. & Alepis, E. (2004a), Mobile educational features in authoring tools for personalised tutoring. Computers & Education, to appear.

Virvou, M. & Kabassi, K. (2003), Experimental studies within the software engineering process for intelligent assistance in a GUI, Journal of Universal Computer Science, 9(1), 51-85.

Virvou, M. & Manos, K. (2003), Individualising a cognitive model of students' memory in Intelligent Tutoring Systems, Lecture Notes in Artificial Intelligence, 2773, 893-897.

Virvou, M. & Manos, K. (2003a) "A Simulated Student-player in Support of the Authoring Process in a Knowledge-based Authoring Tool for Educational Games" Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies 2003, Athens, Greece, July 9-11, 2002

Virvou, M. & Manos, K. (2003b) "Individualising a Cognitive Model of Students' Memory in Intelligent Tutoring Systems" To appear in Lecture Notes in Artificial Intelligence, Editors V. Palade, R.J. Howlett, L.C. Jain: "Knowledge-based Intelligent Information and Engineering Systems", 2003, Springer-Verlag.

Virvou, M. & Tsiriga, V. (2002): An object-oriented software life cycle of an intelligent tutoring system, Journal of Computer Assisted Learning, 17(2), 200-205.

Virvou, M. (2003), Modelling the Knowledge and Reasoning of Users in a Knowledge-Based Authoring Tool, International Journal of Continuing Engineering Education and Lifelong Learning, 13(3/4), 399-412.

Virvou, M., & Katsionis, G. (2003), Relating Error Diagnosis and Performance Characteristics for Affect Perception and Empathy in an Educational Software Application, Proceedings of the 10th International Conference on Human Computer Interaction HCI 2003.

Virvou, M., Katsionis, G. & Manos, K. (2004), On the motivation and attractiveness scope of the virtual reality user interface of an educational game, Proceedings of the International Conference on Computational Science, Lecture Notes in Computer Science, to appear.

Virvou, M., Manos, C., Katsionis, G. & Tourtoglou, K.: VR-ENGAGE: A Virtual Reality Educational Game that Incorporates Intelligence In IEEE International Conference on Advanced Learning Technologies (2002), Kazan, Russia, September 16-19, 2002

Virvou, M., Manos, K., Katsionis, G. & Tourtoglou, K (2002): Incorporating the Culture of Virtual Reality Games into Educational Software via an Authoring Tool, Proceedings of the IEEE International Conference on Systems Man and Cybernetics 2002 (SMC 02), 2, 326-331.

Virvou, M., Manos, K., Katsionis, G. & Tourtoglou, K. (2002): "VR-INTEGATE: A Knowledge-Based Authoring Tool for Virtual Reality Educational Games" In Proceedings of ED-MEDIA 2002 World Conferences on Educational Multimedia, Hypermedia and Telecommunications.

Voelker, D.: Statistics, Wiley Publishing, Inc. New York 2001.

Walker, J.H., Sproull, L. & Subramani, R. "Using a Human Face in an Interface". In B. Adelson, S. Dumais & J. Olson (eds.) Proceedings of ACM Conference on Human Factors in Computing Systems CHI , 1994 , pp. 85-91.

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.

Wenger E. (1987) 'Artificial Intelligence and Tutoring Systems', Morgan Kaufmann.

Wolfson, S. & Case, G. "The effects of sound and colour on responses to a computer game". Interacting with computers, 13, 2000, pp. 183-192.