

Intelligent Games for Education - An Intention Monitoring Approach based on Dynamic Bayesian Network

I. Cheng¹ F. Chen¹ S. Rodrigues¹ O.G. Pañella² L. Vicent² and A. Basu²

¹Department of Computing Science, University of Alberta, Canada

²GTM - Grup de Recerca en Tecnologies Mèdia Enginyeria i Arquitectura La Salle Universitat Ramon Llull, Spain

Abstract

Computer games have become one of the preferred choices for entertainment in our society primarily because they are interactive, have appealing multimedia content, and provide an immersive and rewarding environment for players. These qualities constitute an essential psychophysical factor that inspires learning abilities and new knowledge. Despite all these promising elements, studies have shown that current educational games are not as effective as they could be. A lack of adaptive tutoring and feedback tools, lack of proper knowledge assessment, and weakly designed gameplay are the major factors for their inefficiency. We address these problems by proposing an Intelligent Tutoring System (ITS) for computer games. An important contribution of this ITS is its capability to track player intentions and award partial marks, which provides more accurate assessment than simply giving full mark to the correct result and none to an incorrect answer. Two strategies adopted in this system are Bayesian Networks based student modeling and individualized tutoring. The system can incorporate one or more games and can address one or more educational topic. The information collected from student interaction with computer games is used to update a student module that reports a student's current level of knowledge, making adaptive tutoring and assessment with computer games more effective. In order to provide an engaging and interactive environment, each game in the system has a local student module constructed based on a Dynamic Bayesian Network. We describe the design and evaluation of our ITS using a prototype implementation with several game examples. Positive evaluation results support the feasibility of the proposed system.

Categories and Subject Descriptors (according to ACM CCS): K.3.1 [Computing Mulieux]: Computers and Education—Computer Uses in Education

1. Introduction

Historically, computer games had been overlooked as a serious research topic. The first fields of science that had interest in game-related research were psychology and sociology. Although the focus of their research was to find the negative influence that games had on children, they discovered that games have many positive attributes that contribute to learning new concepts [Pre01] [Gee03] [Squ05]. Thus, research into educational computer games was initiated. Many scholars discuss the potential learning principles that computer games provide. Computer games are learning machines, a player can learn complicated rules, plan strategies, realize complex manoeuvres, all without reading a manual and just in the first few minutes of playing a game. Engagement is a critical attribute that games have. This attribute leads to a

number of positive elements that provide an optimal environment for learning. Trying to take advantage of the potential learning environment that computer games provide, numerous educational computer games have been developed; although when compared to commercial “non-educational games,” they fail to demonstrate a motivating and exciting experience. One of the reasons behind this is that the production values tend to be sub-par, resulting in games with simple graphics and audio, and over-simplified interfaces. The gameplay (*i.e.*, the way a game is played, its rules and feeling) of these games rely on simple replications of exercises that a student does with pen and paper. These games present the educational content and the game section independently. Therefore, students perceive the educational game as two separate entities, the educational “boring” section (s)he has to pass to get to the “fun” section of the game. Several

scholars have concluded that this approach results in a game that is not educational, as a student tends to only remember the game section and forgets about the subject knowledge. Thus, designing an effective educational game can be quite a challenge. How can educational games be improved? In this paper an Intelligent Tutoring System (ITS) for computer games is proposed, following the research direction of serious games [KTK05]. A prototype of the ITS based on topics in physics is designed, developed and evaluated as well. The rest of this paper is organized as follows: Section 2 presents related work in educational games and Intelligent Tutoring Systems. System design of the proposed ITS and game examples are described in Section 3 and Section 4 respectively. Discussion of experimental results and evaluation are presented in Section 5. Finally, concluding remarks are given in Section 6.

2. Related Work

There has been extensive research in the graphics education area during recent years [OSZ00] [KS04] [SSR07] [FH98] [CAC*05] [KP95] [KB04]. Among these, most of the work related to educational games is based on analyzing a game's potential for education. A few other articles deal with design choices that increment the effectiveness of educational games [MN05]. A large number of publications propose new games to educate in different fields [VDBNR98] [Won93] [LH93] [KV04] [Stu03] [EH05] [KK05] [Yan05]. However, only a few of the articles provide an empirical evaluation of the effectiveness of their games [YHT05]. Related work in the field of Intelligent Tutoring Systems includes ANDES [GCV98] (an ITS for Newtonian physics). ANDES has the typical components of an ITS. Although not game-based, ANDES is a representative ITS that uses Bayesian networks at its core. Evaluations have shown a positive result from the system. JV²M [GmGmGc04] is a game-based ITS that teaches the internal function of the java virtual machine and how source code is translated into object code by java. This paper gives a very high level idea of the ITS, there is no information about the real gameplay, how the source code can be compiled and executed by manipulating objects is not described. Therefore, it is not clear how educational content is presented to a player. As well, there is no indication of how difficulty levels affect the game. Additionally, the pedagogical agent is not described very well, there are no details about when and how it decides to help a player or even what sort of information is included in the models or how are they constructed. No evaluation of the ITS is presented and as a consequence it is hard to estimate the potential of JV²M. Prime Climb [Con04] is an educational game with an integrated educational agent that provides adaptive hints. The game is aimed at Grades 6 and 7 students and it is based on the concept of number factorization. An evaluation of the effects of the pedagogical agent was conducted using two versions of the game: with and without the pedagogical agent. Correlation studies show which hints provide more gain in learning.

The overall results show that the agent helps to improve the effectiveness of the game. The approach in this game is interesting as it is one of the rare attempts to create an effective game with the aid of a probabilistic model. The use of a dynamic Bayesian network for constructing the student model is effective. The limitations come from the nature of the design of the game and its relationship to the student model, resulting in an inaccurate model. In contrast, our proposed system includes an innovative game module, and for each game an effective local student module as well as a tutoring module, which are absent from traditional ITSs.

3. Proposed Intelligent Tutoring System for computer games

In the proposed system, we follow the same terminology as that used in a traditional ITS but the components are organized following a different architecture. A major difference is in the interface component, which in our case is composed of a game module responsible for tailoring the educational content and interacting with a student. Additionally, it provides feedback to the system for organizing subsequent interactions. The game module can be composed of an arbitrary number of games, with the feature that each game has its local student and tutoring modules. This specialized design approach is necessary because each game must be based on a concept, and gameplay must be designed with this concept in mind, so that each interface action is interpreted by the system as a glimpse of a student's knowledge in solving a particular problem. Thus, each local module (domain, student and tutoring) must be based on the gameplay and the concept, instead of adopting the global system defaults. This is also the reason for omitting a global domain model, since each game must deal with the domain expertise by itself. Besides each local module, the system maintains global student and tutoring modules as shown in Figure 1.

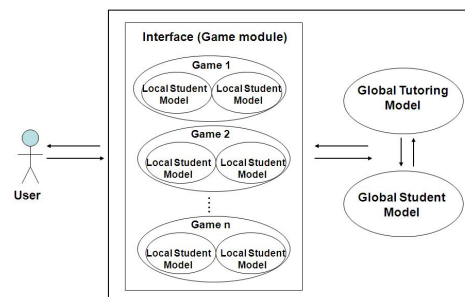


Figure 1: Game-based ITS components.

3.1. Student module

The global student module maintains a set of skills and concepts as metrics for assessing a student. The general structure of the global student module is a Bayesian network.

This network has two sets of nodes: skill and concept. Different concept and skill nodes estimate specific concept or skill knowledge. Each node represents a binary value named MASTERY that has two possible values: TRUE or FALSE. A probability of 1 for the TRUE value ($p(MASTERY = TRUE) = 1$) indicates that the student has totally mastered the skill or concept. The global student module is updated whenever a student finishes playing a game. While the local student module is designed for individual student sessions, the global student module serves as the norm for student assessment.

The local student module is structured according to the characteristics of each game, with the restriction that it must inherit the concept and skill nodes from the global module for updates. Besides concept and skill nodes, the local module also has binary-valued action nodes that represent the occurrence of interface actions. Thus, the structure of the network implies that each interface action is conditioned by the current concept or skill level. Therefore, it is critical to efficiently design the gameplay so that each interface action reflects the strategy a student is following to solve a problem. Whenever an interface action occurs, an action node is inserted and the network is updated following the Bayesian update rules. Given the properties of dynamic Bayesian networks to model situations that evolve over time (e.g., the paced interaction in games), the local network is a dynamic Bayesian network. In this configuration each time slice represents a relevant interface action in the game. At the end of this action, the probability values for the network are “rolled up” following the node hierarchy to summarize the probabilities in the root nodes (skill and concept nodes). These probabilities are then used as priors for the next action (Figure 2). Once a session has ended, skill and concept nodes contain the up-to-date probability values in the network. Thus, the next session is ready to start using these values. When a game ends, concept and skill nodes propagate their values to the global network, where data is ready to be used for the next game and inferences can be made about a student’s current level of knowledge.

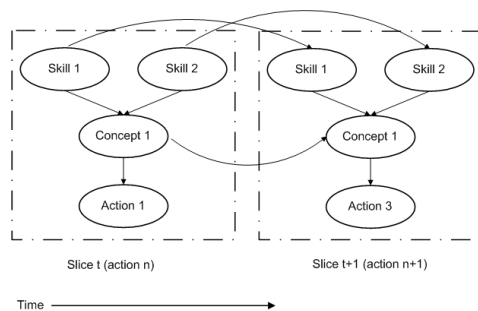


Figure 2: Example of dependencies in the local dynamic Bayesian network.

3.2. Tutoring module

The global tutoring module has two functions:

- Session planning: Plan a session for a student based on his or her skills and concepts reported by the student module. According to a predetermined policy (specified as an input to the system by the instructor or a default plan), the session can focus on the weakest skills of a student, focus on the weakest concept, or take a somewhat balanced approach.
- Reinforce concepts: Providing hints at the beginning of each game and between games (based on previous performance). This is achieved by displaying a screen (text, animation, diagram) that explains the concepts and skills needed before a game. This tutoring is independent of the game to be played.

The local tutoring model provides tutoring between sessions or difficulty levels, and between attempts in the game.

3.3. Game module

The game module serves as an interactive medium for presenting the educational content and interfacing between a student and the ITS. This module can be composed of one or more games. These games should be designed effectively as suggested in published literature on educational game design [Fis05] [UHT02]. Additionally, the game must be structured into small entities called levels, where each level presents a goal to the student. Game difficulty level must be increased between levels and as students learn. Priority should be given to educational content-dependant factors. However, game-dependant factors should not be ignored because they contribute to maintain the challenge and excitement, and thus engagement.

4. ITS prototype implementation

In the global student module of our prototype implementation, there are three basic skill nodes: math, physics and gaming. Initial nodes are generated once a student starts a session in the system by creating or loading a profile. Prior probabilities for root nodes in the modules are initialized to a value of 0.5 if previous information on a student’s level of knowledge is not available (as in the case of the current prototype implementation). However, these values can be computed if enough information about a particular student is available prior to using the system for the first time. In the case of non-root nodes, prior probabilities can be designed with the aid of an expert in the topic. These prior probabilities can also be computed if enough data is available from student performance.

4.1. Game examples

Three game examples on Physics topics have been developed in the current prototype.

4.1.1. Balance.

This game is based on the moments concept (Figure 3). A moment is a quantity representing the magnitude of a force influencing the rotation of an object. The rotation depends on the magnitude of the force and the perpendicular distance of the force from the pivot. The object is represented in the game as a bar lying on a wheel that acts as the pivot for the system. At the sides of the bar there are two boxes representing two forces acting on the beam. The user can control the movement of the bar and change the force acting on one side of the beam by catching balls of different weights. The objective at each level is to balance the moments acting on the beam. Increases in concept-dependant difficulty levels are achieved by moving the pivot closer to one of the ends of the bar or by adding one or more boxes to the beam.



Figure 3: Balance game screen.

4.1.2. Lightbeam.

This game is based on the concepts of light reflection and refraction on planar surfaces (Figure 4). The objective of the game is to hit targets on the right side of the screen by using the laser gun of a spaceship. Each difficulty level presents a new target and a different challenge to hit it. At the same time, a student must defend his or her own ship from incoming enemy ships attacking from different angles. The student can manipulate the angle of the gun (by moving the mouse around) and fire the laser by clicking the left button on the mouse. For the initial levels, only one surface and reflection properties are considered. Difficulty increases as more surfaces are introduced and refracting surfaces appear.

4.1.3. BBall

This game is based on the concept of projectile motion (Figure 5). To play this game the student must be familiar with this concept as well as the linear equations of motion. The objective at each level in the game is to score a basket. For initial levels the basket is at the same height as the player's hands, resulting in the simplest form of the projectile motion

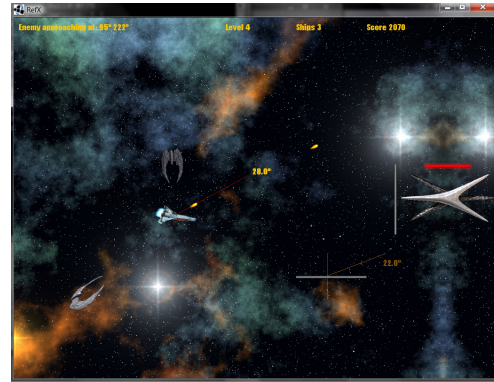


Figure 4: Lightbeam game screen.

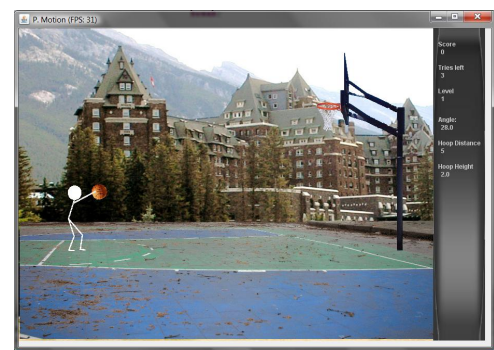


Figure 5: BBall game screen

problem. The student can face different challenges: to provide a right angle for the player's hands, position the player at the correct distance from the basket, or to make the throw with the correct force. These elements may be combined for more complex situations. Difficulty is gradually increased by varying the height of the basket, introducing a situation where the player is jumping or introducing air resistance.

4.2. Analysis of the nodes structure

In this section, we use the Balance game as an example to explain how the nodes are structured and updated.

In the local student module of the Balance game, there are three nodes inherited from the global student module and a number of action (Evidence) nodes (Figure 6). These nodes record the current skill level in the moments concept, algebra skills related to the moment concept and gaming skill level for the game. Action nodes record any relevant interface movements in the game, e.g. catching or avoiding a ball. Action nodes for the Balance game are summarized in Figure 7. Table 1 is the Conditional Probability Table (CPT) for a Catch node. In this table, ϵ is the probability of having a slip (i.e., occasionally having a wrong answer even when skills and concepts are mastered), p_{guess} is the probability

of guessing the correct answer given that none of the skills or concepts are mastered; and p_guess is the probability of making an educated guess (i.e., guessing the correct answer when just some of the concepts or skills are mastered). Action nodes may appear as child nodes of concept and skill nodes. For the Catch node example, only the moment node is affected by the appearance of a new action, algebra and forces skills are updated as well because they are parents of the moment node.

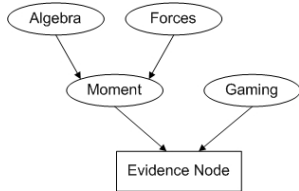


Figure 6: Bayesian network for the balance game.

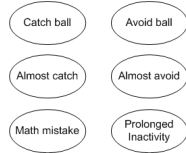


Figure 7: Evidence (Action) nodes in the balance game.

Table 1: CPT for catch node when it is activated.

Moment	Gaming	p(CATCH=TRUE)
FALSE	FALSE	$p_guess = 0.2$
FALSE	TRUE	$p_guess = 0.4$
TRUE	FALSE	$p_guess = 0.7$
TRUE	TRUE	$1 - \epsilon = 0.9$

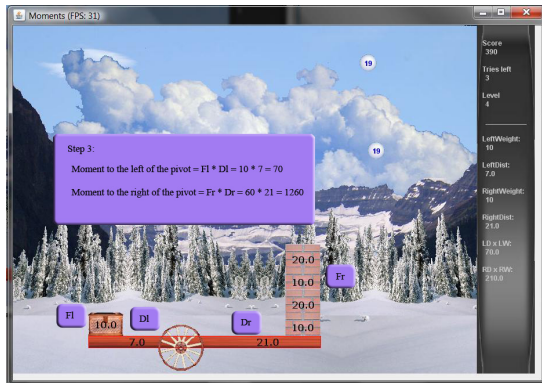


Figure 8: Interactive tutorial during the Balance game.

In the local tutoring module for the Balance game, instructions are provided for the skill and concept nodes defined

Table 2: Tutoring actions given for the moments concept

Impact	Specificity	Hint
Text	Weak	The moments on each side of the pivot must be equal.
Text	Average	A moment is the product of the magnitude of the force and its distance to the pivot.
Text	Strong	To balance a beam the sum of all moments at each side of the pivot must be equal ($Fl * dl = Fr * dr$)
Animation	Weak	Shows an animation indicating where the moments are
Animation	Average	Shows an animation indicating the product of a force and its distance to the pivot
Animation	Strong	Shows an animation indicating the calculation of the moments on a beam using the formula
Interactive tutorial	Weak	Step-by-Step interactive tutorial on seeing the effects of moments on one side of the beam
Interactive tutorial	Average	Step-by-Step interactive tutorial on seeing the effects of moments on both sides of the beam
Interactive tutorial	Strong	Step-by-Step interactive tutorial on balancing the beam by using the formula

in the student module that represent a degree of mastery in gaming, algebra, forces and moment. Whenever the value at these nodes drops below a pre-defined threshold in their level of mastery (0.3 in the current implementation), the tutoring module intervenes and provides feedback to the student (Figure 8). For each node there are three levels of specificity (i.e., weak, average and strong). Additionally, the tutoring module can present these hints in three different forms: text, animation and interactive tutorial. Table 2 shows the possible hints for the moments concept at all specificity and impact levels.

4.3. Immersion and VR game examples

To enhance our system, Virtual Reality (VR) techniques can be employed to make the game interface more appealing. Two examples have been implemented. Figure 9 shows several snapshots of the VR game *smashball*. In this game the sensor tracks the position and orientation of the racket used, so that users can interact with the game by only using the racket. Figure 10 depicts a training "game" to help users learn how to weld precisely. In the game, the users can feel fully immersed in the virtual environment. These human

computer interface techniques can be further explored in the future.



Figure 9: Smashball game with a long range signal amplifier hanging from the roof.

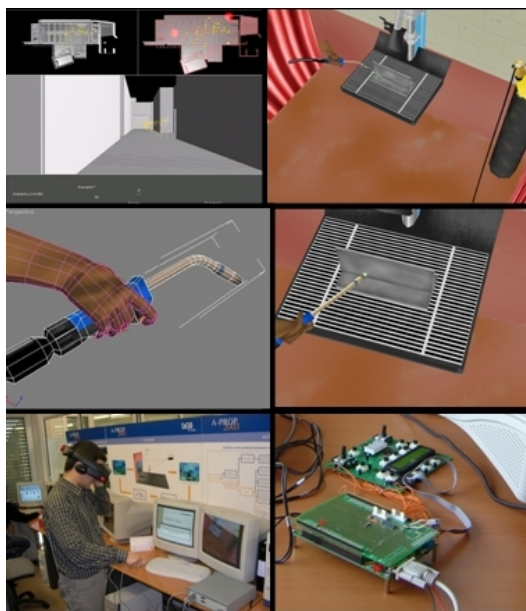


Figure 10: The VR welding simulator.

5. Evaluation

5.1. Experimental procedures

The experiments were designed for validating the feasibility of the proposed ITS in improving learning performance and evaluating its accuracy (*i.e.*, the effectiveness of the assessment). We used the Balance game, and a total of ten university students with mixed genders was divided into two groups: experimental and control, with five students in each group. The control group played games in the system without feedback and hints. The experimental group used the

complete version of the system. This setting allowed us to compare students' performance with and without the adaptive student and tutoring capabilities in the system. Both groups followed the following steps:

1. **Pre-test:** students answered a pen and paper exam and questionnaire. The exam was based on the concept and skills used in the game.
2. **Formative:** students in both groups used the system for twenty minutes.
3. **Post-test:** students answered a pen and paper exam and questionnaire. Similar to the pre-test, the exam was based on the concept and skills used in the game.

Data was collected from both pre and post tests for each student. Additionally, a log was saved that contains detailed information on the interaction with the system.

5.2. Experimental results

To analyze the effectiveness of the proposed ITS, scores from the pre and post tests were compared. Both pre and post tests were marked on a 1-10 range. For each student, the difference between the post-test score and a pre-test score was called a 'growth' or 'gain.' It was expected that both groups improved their pre-test scores. Meanwhile, students in the experimental group were expected to achieve bigger gains compared to students in the control group. Table 3 shows the results for the gain parameter. The experimental group showed a mean gain of 1.4 with a standard deviation of 0.8. The control group showed a smaller gain, but still on the positive side as expected. To verify the statistical significance of the performance data, a two-tailed t-test was performed. The resulting t (2.3837) had a p -value of 0.0445, which is statistically significant (p -value $<$ 0.05). This test confirms that the gain improvement was not a result of chance and there was enough evidence to accept the hypothesis (gain in experimental group $>$ gain in control group).

Table 3: Gain results ($t = 2.3837$, $p(2 - tail) = 0.0445$).

Group	Mean	Std. deviation
Experimental	1.4	0.8
Control	0.4	0.49

Is the improved performance a result of the tutoring capability? One can argue that students in the experimental group might have had an advantage in knowledge prior to the study, leading to the increased gain for their group. To disprove this hypothesis, a similar two-tailed t-test was performed on the pre-test scores. Table 4 shows the results of this comparison. The experimental group had a lower mean for the pre-test score, *i.e.*, students in this group performed worse than those in the control group. However, results from the t-test showed that any difference in score was not statistically significant for this study ($p >$ 0.05).

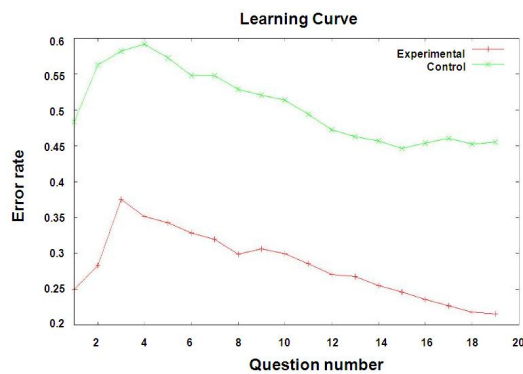
Table 4: Pre-test results ($t = -0.6396$, $p(2\text{-tail}) = 0.5401$).

Group	Mean	Std. deviation
Experimental	7.8	1.6
Control	8.4	1.36

Furthermore, difficulty level may be suspected to influence the results. For both groups, the mean difficulty level reached was calculated and a two-tailed t-test was computed as well. Table 5 shows these values, both groups reached similar difficulty levels and the t-test indicated that there was no significant difference ($p > 0.05$). One can observe that even when no tutoring system was active, students could still find their way through the levels in the game; however, the improved performance in the game was larger when tutoring was active as the gain results show.

Table 5: Difficulty level reached ($t = 1.06$, $p(2\text{-tail}) = 0.3201$).

Group	Mean	Std. deviation
Experimental	8.8	1.44
Control	8.0	0.89

**Figure 11:** Group comparison.

Alternatively, results can be analyzed by comparing the learning curves of both groups. A learning curve is a graph that shows the learning rate of a student over time (*i.e.*, a graph that shows question or problem solving progression against an error-rate) [MKMM05]. Summarizing for all students, Figure 11 shows the learning curves for both experimental and control groups. From this graph it can be observed that students in the experimental group showed a smaller error rate through all the questions. At the beginning the error-rate increased, but as soon as the tutoring system started to intervene, the error-rate was reduced. For the control group it can be observed that even with no hint the error-rate was reduced, but the learning rate was smaller than the

experimental group. Additionally, the error-rate of the control group converged to a positive value; instead of approaching the zero value as in the experimental group. These graphs are very useful in tracking student performance over time, and for planning new tutoring strategies.

In general, the experimental results shows that the proposed ITS is reliable in assessing students' skill levels and is effective in improving learning performance.

5.3. Discussion of limitations and possible future direction

Similar to other intelligent systems, our ITS requires a sufficiently large training set (collected from realistic student records) to initialize the probability values in the nodes. This can be cumulated and refined in the ITS over time but initial estimates have to be provided. A common shortfall in all game-based educational tools is how to separate the computer game playing skill from the subject knowledge. A student may score better because (s)he is proficient in manipulating a computer mouse or gadget. This is normally overcome by a warm up session or some pre-test to make the necessary adjustment to the final score if only subject knowledge is to be assessed.

In the proposed ITS, using the action nodes is a powerful mechanism to distinguish students' skills at a finer scale. For example, a student may understand the concept of moment but make a careless mistake in algebra. Another student may be totally ignorant on the topic. It is not fair if both students are assigned a zero mark because the final answers are wrong. Our ITS uses the action nodes in the dynamic Bayesian Network to interpret a student's intention which can indicate random moves or careless mistakes. Nevertheless, to accurately define and develop such an intelligent Game module is not trivial. By proposing this intelligent game system for education, our goal is to inspire initiatives from multiple disciplines including education, computing science and psychology, to make educational games more effective and beneficial.

Several open problems remain to help in the improvement of the evaluation approach. In addition to increasing the number of participants, focus should also be directed to assessing the importance of culture and gender in educational games, exploring different topics and types of gameplay, and exploring multiplayer and collaborative scenarios. Additionally, evaluations with students from different backgrounds must be performed, and long-term impact of the system on students needs to be studied.

6. Conclusion

Computer games have the potential for creating an attractive educational environment. To take advantage of their full potential in improving learning performance, games must be designed to illustrate subject knowledge, be adaptive to dif-

ferent student's capabilities and provide an effective form of assessment of students' understanding of a subject. In this paper, a system was proposed for improving learning performance based on computer games. In the system, gameplay was designed by putting the educational content at the core. This is achieved by monitoring a student's activities on the interface, which is recorded in the action node hierarchy. Computer and gaming skills are filtered by the game module, leaving the student module to focus on the educational content. The advantage of this design is to accurately distinguish among different skills. The tutoring module is effective in guiding and reinforcing an educational concept on a student as a game is played. The effectiveness of the tutoring is supported by the student module which adaptively records the current skill level. By developing and evaluating a prototype using physics topics, experimental results show that learning performance can be improved and a student's knowledge can be assessed reliably.

References

- [CAC*05] CORREIA N., ALVES L., CORREIA H., ROMERO L., MORGADO C., SOARES L., CUNHA J. C., ROM AO T., DIAS A. E., JORGE J. A.: Instory: a system for mobile information access, storytelling and gaming activities in physical spaces. In *Proceedings of the ACM SIGCHI* (2005), pp. 102–109. 2
- [Con04] CONATI C.: Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game. In *IUI* (2004), ACM Press, pp. 6–13. 2
- [EH05] EBNER M., HOLZINGER A.: Successful implementation of user-centered game based learning in higher education: An example from civil engineering. *Computers & Education* 49 (2005), 873–890. 2
- [FH98] FRANK R. K., HANISCH F.: Web based teaching of computer graphics: Concepts and realization of an interactive online course. In *SIGGRAPH* (1998), Addison Wesley. 2
- [Fis05] FISCH S. M.: Making educational computer games educational. *IDC: Interaction Design and Children*. (2005), 56–61. 3
- [GCV98] GERTNER A. S., CONATI C., VANLEHN K.: Procedural help in andes: Generating hints using a bayesian network student model. In *National Conference on Artificial Intelligence* (1998), pp. 106–111. 2
- [Gee03] GEE J. P.: What video games have to teach us about learning and literacy. *ACM Computers in Entertainment* 1, 1 (2003). 1
- [GmGmGc04] GÓMEZ-MARTÍN M. A., GÓMEZ-MARTÍN P. P., GONZÁLEZ-CALERO P. A.: Game-driven intelligent tutoring systems. In *In Proceedings of the Third International Conference on Entertainment Computing (ICEC* (2004), Springer, pp. 108–113. 2
- [KB04] KRISHNASWAMY A., BARANOSKI G. V. G.: A biophysically-based spectral model of light interaction with human skin. *Comput. Graph. Forum* 23, 3 (2004), 331–340. 2
- [KK05] KIM H., KIM S.: An integrated course based on educational games. *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)* (2005). 2
- [KP95] KJELLD AHL L., PRIME M.: A study on how depth perception is affected by different presentation methods of 3d objects on a 2d display. *Computers & Graphics* 19, 2 (1995), 199–202. 2
- [KS04] KJELLD AHL L., SANTOS B. S.: Visual perception in computer graphics courses. *Computers & Graphics* 28, 3 (2004), 451–456. 2
- [KTK05] KIRKLEY S. E., TOMBLIN S., KIRKLEY S.: Instructional design authoring support for the development of serious games and mixed reality training. In *Proceedings of Interservice/Industry Training, Simulation, and Education Conference (IITSEC)* (2005), pp. 1–11. 2
- [KV04] KATSIONIS G., VIRVOU M.: A cognitive theory for affective user modeling in a virtual reality educational game. *IEEE International Conference on Systems, Man and Cybernetics* (2004), 1209–1213. 2
- [LH93] LYVERS R. F., HOROWITZ B. R.: A unique instructional tool for visualizing equipotentials and its use in an introductory fields course. *IEEE Transactions on Education* 36, 2 (1993), 237–240. 2
- [MKMM05] MARTIN B., KOEDINGER K. R., MITROVIC A., MATHAN S.: On using learning curves to evaluate ITS. In *Int. Conference on AI in Education* (2005), pp. 419–426. 7
- [MN05] MAIA R., NETTO M.: Work in progress - a distributed approach to a learning management system using multi-agent technology. In *Frontiers in Education* (Oct. 2005), pp. F4G–T1A. 2
- [OSZ00] OWEN G. S., SUNDERRAMAN R., ZHANG Y.: The development of a digital library to support the teaching of computer graphics and visualization. *Computers & Graphics* 24, 4 (2000), 623–627. 2
- [Pre01] PRENSKY M.: *Digital Game-Based Learning*. McGraw-Hill, 2001. 1
- [Squ05] SQUIRE K. D.: Changing the game: What happens when videogames enter the classroom? *Innovate* 1, 6 (2005). 1
- [SSR07] SUNG K., SHIRLEY P., ROSENBERG B. R.: Experiencing aspects of games programming in an introductory computer graphics class. In *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education* (New York, NY, USA, 2007), ACM, pp. 249–253. 2
- [Stu03] STUBBS K.: Kana no senshi (kana warrior): A new interface for learning japanese characters. *CHI: Conference for Human-computer Interaction* (2003). 2
- [UHT02] UMETSU T., HIRASHIMA T., TAKEUCHI A.: Fusion method for designing computer-based learning game. *Proceedings of the International Conference on Computers in Education (ICCEŠ02)* (2002). 3
- [VDBNR98] VOLBRACHT S., DOMIK G., BACKE-NEUWALD D., RINKENS H.-D.: The 'city game': An example of a virtual environment for teaching spatial orientation. *Journal of Universal Computer Science* 4, 4 (1998), 461–465. 2
- [Won93] WONG S. C.-H.: A quick prototyping approach to educational software development - the hypermedia experience. *IEEE TENCON* (1993), 78–84. 2
- [Yan05] YANG H.-C.: A general framework for automatically creating games for learning. *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALTŠ05)* (2005). 2
- [YHT05] YU K.-C., HSIAO H.-S., TSAI F.-H.: The implementation and evaluation of educational online gaming system. *Information Technology: Research and Education, 2005. ITRE 2005. 3rd International Conference on* (June 2005), 338–342. 2