Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
editor@iberamia.org
Asociación Española para la Inteligencia Artificial
España

Inteligencia Artificial 20(60), 1-19
doi: 10.4114/intartif.vol20iss60pp1-19

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Intelligent tutoring Module for a 3Dgame-based science e-learning platform

Donfeng, Liu[1,2,3] , Luis Barba-Guamán[2], Priscila Valdiviezo-Díaz[2], Guido Riofrio[2]
[1]Scoool of Information Engineering, Guandong University of Technology, Guangzhou, China
liudf@gdut.edu.cm
[2]Departamento de Ciencias de la Computación y Electrónica, Universidad Técnica Particular de Loja, Loja, Ecuador {lrbarba, pmvaldiviezo,geriofrio}@utpl.edu.ec
[3]Proyecto Prometeo de la Secretaria Nacional de Ciencia, Tecnología e Innovación (SENECYT), Quito, Ecuador

Abstract The three-dimensional (3D) game-based intelligent science tutoring system (GIST) is an e- learning platform for science. The individual complexity of 3D games and conventional intelligent tutoring systems (ITS) results in extra complexities in system design and development of GIST. It is significant to develop practical GIST, not just to seek the powerful ones. The main contribution of this paper is the lightweight modelling of the intelligent tutoring module based on the brief-desire-intention (BDI) framework. The tutoring module, which is implemented by integration of the BDI framework into a game actor, can be able to suggest to each student specific learning tasks based on his/her learning histories. As a study case, algebra-based physics are intentionally chosen as the learning contents in this 3D game-based learning system, since the most of existing 3D game-based learning systems only focused on the qualitative understanding of physical concepts. Our proposed modelling is only based on the BDI reasoning mechanism, so it can be easily extended to obtain practical GIST by using standard game engines.

Resumen El juego serio en tres dimensiones (3D), basado en un tutor de sistema inteligente (GITS) para una plataforma e-learning de ciencias. La complejidad de un juego en 3D y de los sistemas de tutoría inteligente hacen difícil el desarrollo de un GIST, es decir, el desarrollo práctico de este. La principal contribución de la investigación es un modelo ligero de un módulo de tutor inteligente basado en la estructura de conciso-necesidad-intensión(BDI). El módulo tutor, se implementó para integrar la estructura (BDI) en el actor de juego, este puede ser capaz de sugerir a cada estudiante una tarea específica basado en su historial de aprendizaje. Un caso de estudio es la física basada en el álgebra, se escogió intencionalmente como un contenido de aprendizaje en este juego serio en 3D, ya que la mayoría de estos sistemas de aprendizaje solamente se enfocan en la comprensión cualitativa de los conceptos físicos. Nuestro modelo propuesto solamente se basa en un mecanismo de razonamiento del BDI, de esta manera puede ser fácilmente ampliado y obtener un GIST mediante el uso de motores de juego estándar.

Keywords: Intelligent Tutoring System, 3D Game-based Learning System, Reasoning modelling.
Palabras clave: Sistema Tutor Inteligente, Sistema de Aprendizaje basado en juegos 3D, modelado de razonamiento.

1 Introduction

An intelligent tutoring system (ITS) [39] is a computer-based instructional system that is able to offer specific and personalized contents to the students in some intelligent ways, that is, the system can make the decision which is

ISSN: 1137-3601 (print), 1988-3064 (on-line)

the more suitable content for each student. The conventional ITS have provided enough learning and tutoring functionalities, but the system environments are considered unexciting.

The development of serious games [49] indicates a new learning style, and therefore the game-based ITS become an important development direction of ITS. Game-based learning environment in ITS can provide students a perceptual intuition to the complex world, so it is especially appropriate for middle school students. In many cases, the learning scenes are not easily obtained in real world because they are costly or dangerous. The three-dimension (3D) virtual environment is well known in enhancing the immersion and interesting for learning processes. Therefore, 3D game-based ITS, which integrate the 3D game technologies into ITS, are receiving more and more attentions in computer-assisted learning.

3D games and ITS are both very complex, even if building a small prototype for them is a very time-consuming process. Therefore, e-learning communities have recently started to emphasize the need to make e-learning systems more practical rather than more powerful [10]. This practicability of e-learning systems can be extended to having three implications: (1) the objectives of the system are suited to learners; (2) the functionalities of the systems do not need to be complete, but are specially designed to realize the objectives of the system, and are appealing to the learners; (3) from the viewpoint of developers, the best choice of development methods and tools is to use those that commonly accepted and employed, so that it is easy for us to collect enough human and material resources, and it is helpful to the stability and extensibility of the systems.

There is a big demand for 3D game-based ITS, so it is very meaningful to find practical ways we can design and develop 3D game-based ITS with limited complexities. In this paper, we propose to develop a 3D Game-based intelligent science tutoring system (GIST) based on the BDI reasoning principle and an industry-level game engine. In addition, we report our experience in applying the problem-solving tutoring strategy for use in BDI paradigm. Finally, to test and evaluate our conceptual designs, a prototype game system was built for the subject of classic Newtonian kinematics. GIST is a part of long-term project, and this presented result is the first phase of work.

This paper is organized as follows: In Section 2, we will briefly review the technological background concerning to GIST. In the section 3, we will address the design of our e-learning platform GIST. The section 4 will provide an introduction about implementation of GIST. Finally, the section 5 will present some conclusions and future aims.

2 Background

2.1 Intelligent Tutoring System

As a means of providing cost effective personalized tuition, ITS have being studied by educational and computer science researchers since the 80s of last century [2]. With decades of research and development, ITS now have been built for various scientific domains such as physics [36], [22], [19], mathematics [9], [1], [41], chemistry[3], computer programming [16], [42], engineering [7], [26], medicine [13], [40], and so on. These studies showed that virtual manipulative through the ITS can significantly improve the learning.

A typical ITS has four basic modules [48]: the domain module, the student module, tutor module, and the communication module. The tutor module performs as an excellent human tutor with the support of the domain module. Based on these modules, ITS is able to provide precise feedbacks when mistakes are made and able to present new topics when the student is ready to learn. The biggest obstacle for the development of ITS is not in the computer hardware, but in the effective designing of the interactive learning environments [7].

2.2 Serious Game and Science Education

Serious games, loosely interpreted to mean almost any systems that use modern game technologies for the non-entertainment purposes, hold great potential for significant impact on education [30]. Simulation and serious games have unique strengths in description and representation of natural phenomena, so they are considered as the new powerful tools to increase the efficiency of science instruction and the absorption of the science learning. At the beginning of this century, education began to realize that 3D games can provide an ideal infrastructure for dynamic learning environments. Numerous studies showed that game-based learning can increase learning

motivation [6], [8], [24], [28]. Some researchers investigated various instructional strategies in the game-based learning [31], [11], [14], [44], [50]. Leisure game industry has established a series of unified standards in the product process, but for serious games for learning the situation still suffers from fragment and lack of coherence, therefore lots of efforts [31], [25], [5], [47], [15], [43], [51] were invested to study the system design and development of the game-based learning system platforms. Up to now, a broad range of curricula have been redesigned for game-based learning environments [24], [28], [44], [35], [12].
Compared with commercial entertainment games, game-based learning systems have two different points: (1) need more powerful supports of artificial intelligence, so the development of serious games is more complex and difficult; (2) the development of serious games is left largely to independent designers and research organization, meaning almost there is no companies involved into this field.

2.3 BDI based Reasoning Process

A BDI model represents the knowledge and the reasoning processes of an individual or entity in a certain domain, task or scenario, which shows up as the philosophical cognitive layer of a virtual human for the modelling of the human behaviour [27]. The basic BDI framework [17] is shown in Fig. 1. Beliefs are facts representing a virtual human's representation of the state or knowledge of the world. Virtual humans may obtain such beliefs from sensing. Desires are activity goals or some desired states. A virtual human may have multiple desires. Intentions refer the activity plans that a virtual human owns to achieve his goals. A Knowledge Area (KA) in BDI is a chunk of procedural plan knowledge which will achieve a specific goal in a given context. A set of Knowledge Areas (KAs) provides hierarchical methods to accomplish system goals.

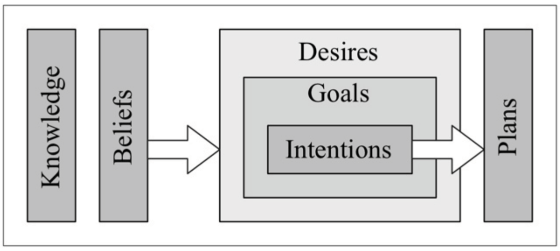


Figure 1. Belief-Desire-Intention model for reasoning procedure.

Several works used the BDI framework in ITS. Giraffa et al [20], [21] designed the student and tutor models based on BDI agents. Fernando et al. (2012) designed an intelligent tutoring module of ITS. This module, which is based on BDI agents, has ability to recognize each student and to obtain information about his/her learning progress.

2.4 Pedagogical Strategy based on Problem Solving

Problem-based learning is a tutoring and learning strategy that challenges students to seek solutions to real-world problems. It involves students in an active, student-centred learning environment. In [46] it has analyzed the combination of problem-based learning method with a game environment, and found that game resources could enhance the student's problem solutions [37] presented systematic reviews of problem-based learning in medical education.
An open-ended performance task [29] prepares the students to answer a question what, why, and how, hence not only engages them in the deeper understandings they have acquired, but it also mirrors the kind of tasks often performed in the working world. Researches [45], [32] have shown that providing students with appropriate worked example problems, which are deep structure similar to the original one, can facilitate problem solving. In addition [32], designed adapted problem posing tasks to probe students' understanding of concepts. This adapted mode of problem posing is significantly different from that of the original ones [31], [34]. It may test a specific conceptual knowledge, and is quite suited to the algebra-based problems when represented in the form of equations.

3 GIST

In this section, we provide an overview of GIST and how it supports the functionality of problem-solving learning based on lightweight BDI modelling. The GIST provides a student with 3D virtual environment in which learning contents can be created according to the student's current understanding of the subject matter, and the students are asked to solve the problems by various interactive methods, such as GUI input and 3D scene operations. After submitting the solution, the GIST feedbacks and provides learning scores for the students.
The significant attraction of GIST lies in its description and representation of learning contents. Different with conventional ITS, GIST employs 3D scene to describe the learning environments. This includes formulating a representation of the subject material, selecting and sequencing concepts from that representation. When we design and implement the tutoring with GIST, special attention has been paid on the context of the 3D game-based ITS. GIST can benefit students with rich information by providing the immersed virtual game environment, including 3D visual scenes, animations, sounds and physical effects. The architecture of GIST is shown in Fig. 2. It consists of a game subsystem and a tutoring subsystem. The game subsystem holds a game manager, which is the core in managing four components. The input component manages user input from mouse and keyboard. Graphical user interface (GUI) component manages the game main window, including the choosing of learning task, saving learning results, etc. The game component organizes the 3D game environment, including static scene, physical effect, animation, and sound effect, etc.

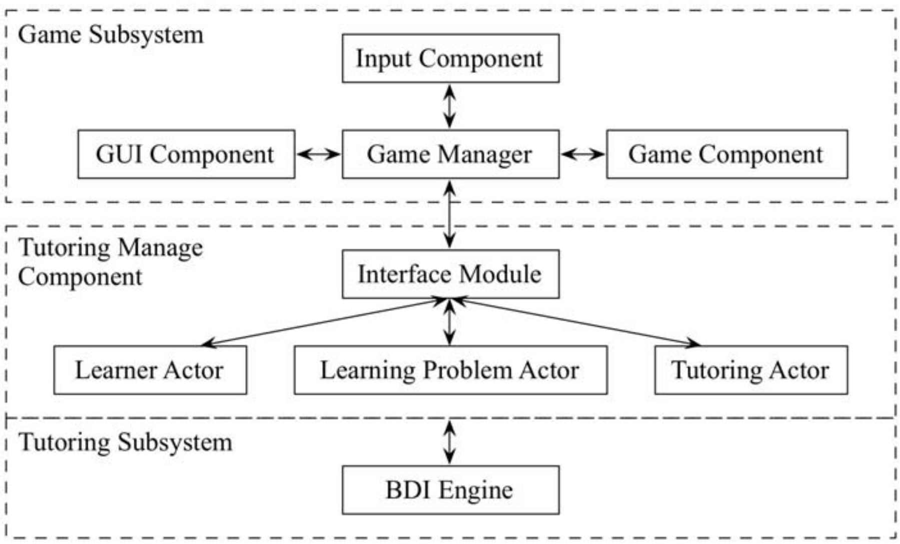


Figure 2. System components of GIST

An Intelligent Tutoring System must have two aspects in tutoring. Firstly, the system must be able to deduce the student's current understanding of the subject matter and adopt individualized instruction to the student's needs. Secondly, the system must be able to apply suitable tutoring strategies that reduce the difference between the expert and the student performance. The tutoring subsystem in GIST is designed to realize these two aspects above, which consists of a BDI engine and a tutoring component. The tutoring component is actually a learning management subsystem, used to manage learning task actor or learning problem actor (LPA), learning actor (LA), and tutoring actor (TA), as shown in Fig. 2. Fig. 3 describes a typical learning process in GIST. The following will introduce the main functional modules of GIST.

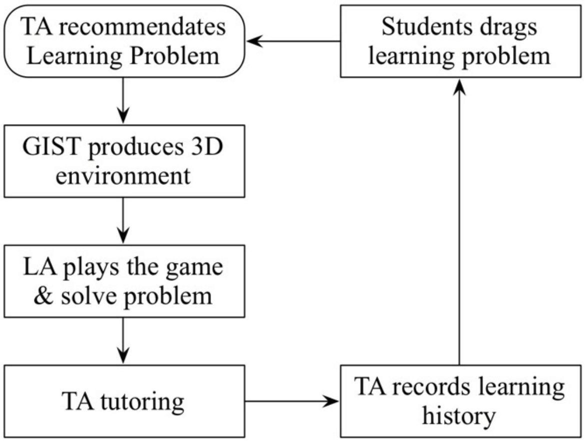


Figure 3. Learning flowchart in GIST

3.1 Graphical User Interface Component

GIST has been carefully designed in GUI to realize various functionalities. The login GUI is used for the learners to register and login, and the language GUI used to language choice among English, Chinese and Spanish. Here, we focus on the introduction of the work GUI of GIST. This interface is divided into two main parts: the 3D work space in the center and menu area in the side. Fig. 4 shows a snapshot of the main-menu, sub-menu, and 3D work space of GIST.

The work space displays the 3D scenario environment, presented in the center of the GUI, which will include necessary objects needed by learning problems, as well as the scenario Non- Player Characters (NPCs). Students can navigate around the environment and operate the objects. The work space is the primary channel through which the student interacts with the scenario environment, GIST has a great feature that all the learning contents are visualized in the GUI, so it is easy for students to understand this system and to choose learning problems. Five boxes from the most left to the right in the main menu indicate five various learning subjects: mechanics, electromagnetism, optics, thermology, and special topics of modern physics. Each box is textured with a simple photo to represent the content, so it is can be easily understood by the students. When the mouse points to one of these boxes, a sub-menu is displayed in the left, which shows the sub-learning fields of the subject represented by the box. For example, when the mechanics box is pointed, the learning fields of measuring, kinematics, simple machines, dynamics and gravity are shown in the sub-menu. There are five tool buttons in the right of the main menu, with which we can execute a simulation, open a chat window, save a learning result, open the learning history, and go back to other GUIs.

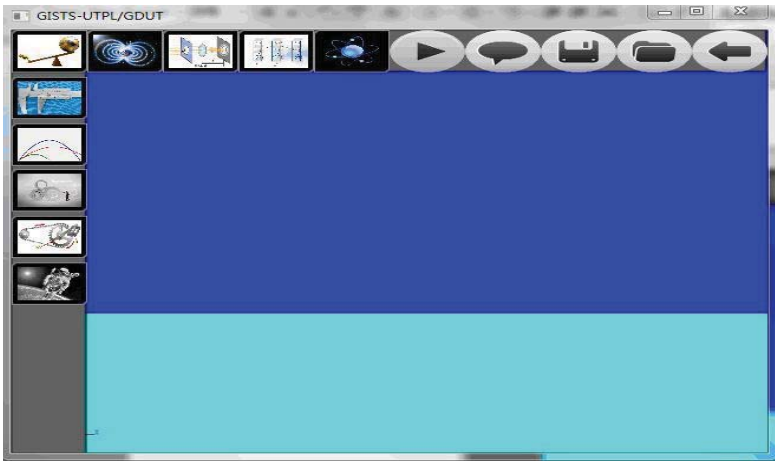


Figure 4. Main-menu, sub-menu, and 3D work area of GIST.

A matrix of boxes can be seen in the right-down area when we point to a box in the sub-menu (see Fig. 5). Each box in this matrix represents the specific learning problem and the corresponding 3D scene available to the students. The photo in each box is employed to visualize the learning problem and can be dragged out from the matrix into the 3D work space. By operating like this, a 3D learning scene is shown in the work space.

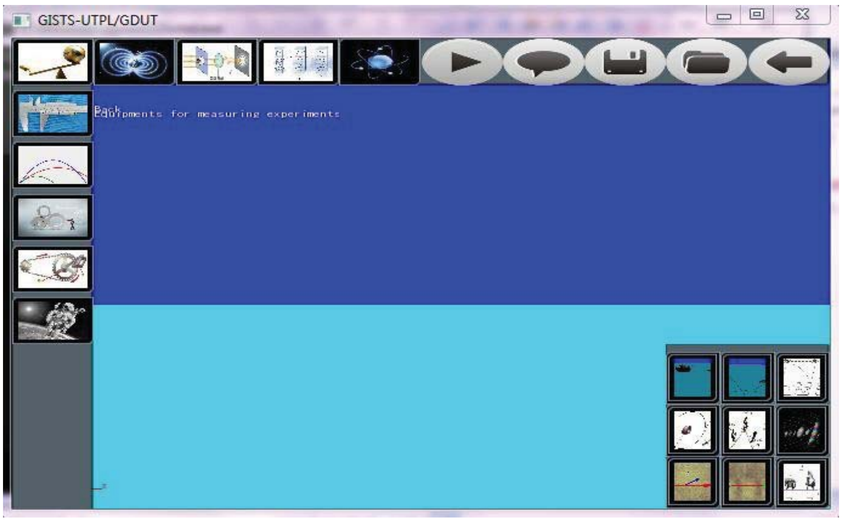


Figure5.Learning problem selection view of GIST.

In addition to the GUIs introduced above, there are other message pop windows used by the learning actor, learning problem actor and tutoring actor. These will be introduced in the following sections.

3.2 Tutoring Manage Component

3.2.1 The Learning Actor

It is necessary to collect four categories of information to describe the main characteristics of a student. The first is basic individual information, such as the name and grade. The second is knowledge level, that is, what the learner knows about the domain. The third is learning histories. The fourth is learning strategies the learner adapts.

The learning module in GIST, consisting of a learner actor (LA) and a learning problem actor (LPA), is designed to store these pieces of information and realize the corresponding functionalities.
LA in GIST only stores the first category of the variables. Each LA has his database called Map which can store a series of Learning Problem Actors. A Map which is empty with Learning Problem Actors shows the learner is new one. Once a learner finishes his learning problem which corresponds to a Learning Problem Actor, the Map will store the learning results. At the same time, LA behaves like the game player in the system, through which all the interactions between the student and the system are carried out by the keyboard and mouse. Unlike the conventional ITS, GIST does not employ the complex modelling for the learning actor's knowledge level. The learning experience in the past, which is described by a list of learning problem actors (LPAs), is used to determine the knowledge level of a student and to support the new learning activities. For example, the system will recommend a student with the learning content based his learning histories.
LA can save and inquiry its learning history, and is updated at the end of each problem. For each learning activity for a problem, some important data, such as learning title, learning date, obtained score, and appraisement description, are saved into a learning history database. When LA enters the state of learning activity, the system will immediately display the corresponding learning history, see Fig. 6. These historical data are considered as the learner's current knowledge level, from which the tutoring actor in GIST recommends the learning problem to students for their next learning activity. This is the first round of intelligent choice of learning contents that is conducted prior to the learning activity. The concerted choice algorithm will be described in the section of tutor actor. Since there is no learning history in GIST for the new learner, the tutoring actor will hint that he is able to choose any subjects.

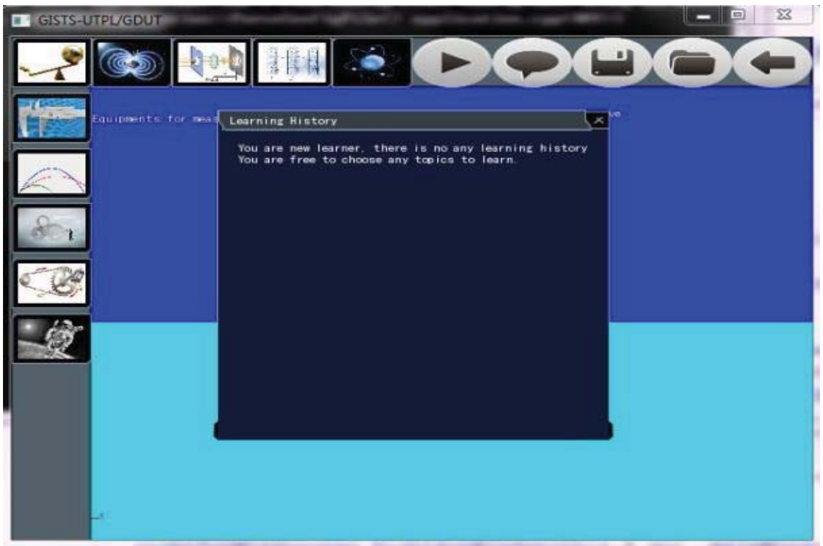


Figure 6. Learning history view of GIST

3.2.2 Learning Problem Actor

The pedagogical strategies of question-based learning are adopted in current GIST. This method is best suited for domains that require the learner to reason, such as physics, chemistry, etc. A learning problem actor (LPA) in GIST has three roles: (1) expressing the learning content;
(2) real-time producing various graphic use interface (GUI) and 3D scene to be used for the interaction between learners and GIST (see an example in Fig. 7), displaying the textual learning contents and learning results, data-input, and figure-based data output; (3) recording the learning results for this learning problem.
All learning problem actors (LPAs) in a relatively independent knowledge subsystem have a parent-child relationship, thus facilitating the concept of nested tasks and task dependencies. This modelling method for the

learning problems is helpful to the intelligent choice of learning content for the student based on his knowledge level. Fig. 8 shows an example of this relationship in the 2D kinematics.

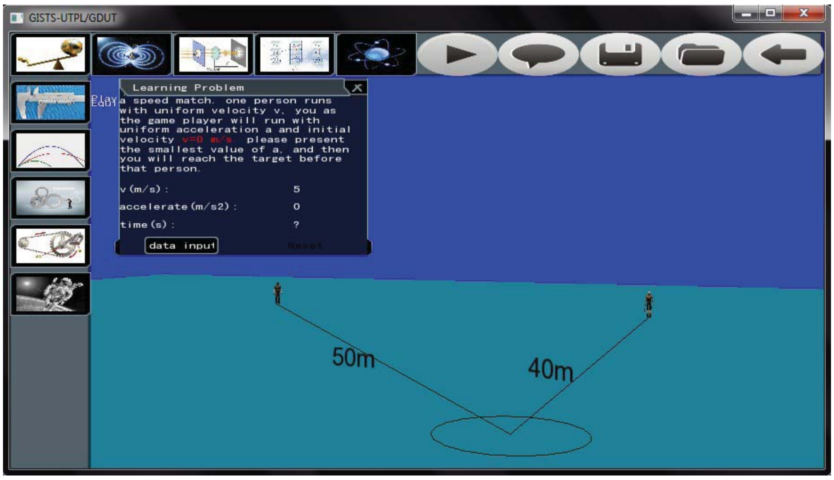


Figure7.Textual presentation of learning problem and the corresponding 3D environment.

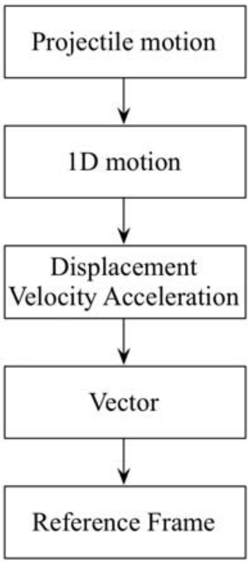


Figure8.Father-child relationship of learning problem actors in 2Dkinematics.

3.2.3 Tutoring Actor

The tutoring actor (TA) is the actual brain of GIST, so it must have the abilities to monitor the learner actor's behaviours, to assess individually the student's capacity, and to specify each student what to learn and how to do that. When a learner can not complete his learning task correctly, this model should present the corresponding explanation or hints and intervene the learning process. Therefore, tutoring is a complicated decision and reasoning process. We discuss in details the tutoring actor as follows.

The tutoring process in GIST begins with the monitoring over the learning content choice of a student. Considering the simplification and extensibility, the physics knowledge is classified in three levels. The top level is outlined in the main menu, and contains five top subjects: mechanics, electromagnetism, optics, thermology and special topics of modern physics. The second level is shown in the sub-menu, which presents the various learning topics of a top subject. An example in mechanics, its second level includes six learning topics: measurement, force, kinematics, dynamics, energy and simple machines. The third level is outlined in the matrix table in right-down area of GUI (see Fig. 5), which gives specific learning problems for each topic in the second level. The learning content choice flowchart is shown in Fig. 9. When a student logins into the system, GIST will check his learning history. If there is no learning history record in the system, the student is considered a new user and is free to choose any problems in the problem matrix. If the system has learning history for this student, then check if he has passed the latest learning problem. If passed, then recommend the father problem of the latest learning problem to student, otherwise, suggest the student to continue the old problem. The algorithm process in Fig. 9 is executed by the BDI engine. GIST provides two phases of monitoring over learning content choice. The first phase introduced above is the initial choice, and the second phase is the detailed choice, which will be described in the later.

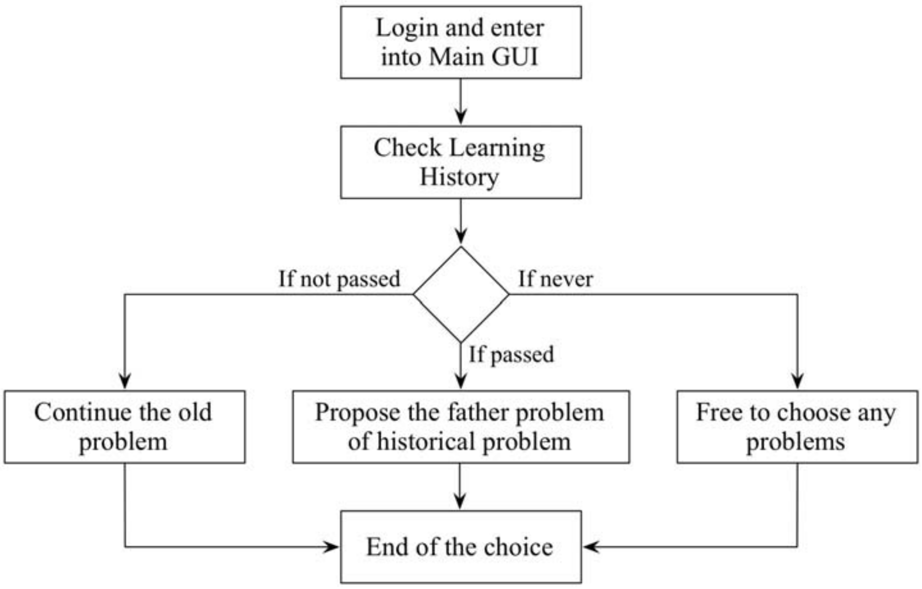


Figure9.Flowchart for initial choice of the learning content.

After the guided choice of learning content, the student can drag the learning problem from the matrix table into 3D work space to produce the 3D learning environment and the corresponding textual problem which is shown in a pop window (see Fig. 7). At this moment, TA steps into the stage to supervise the student’s problem solving. The tutoring strategy is described as following.

Tutoring Strategy

The tutoring strategy used by TA in GIST is based on the problem solving. As a computer- assisted instructional tool, the tutoring actor in GIST can give immediate feedback to students and allow automatic grading for the student, and then save learning records. Automatic grading and the learning records will be used for the next learning assignments. The tutoring flowchart is illustrated in Fig. 10. The tutoring process has three basic stages which are associated with the three modes of problems used in GIST: performance mode, problem posing mode, and worked- example mode (see the background section).

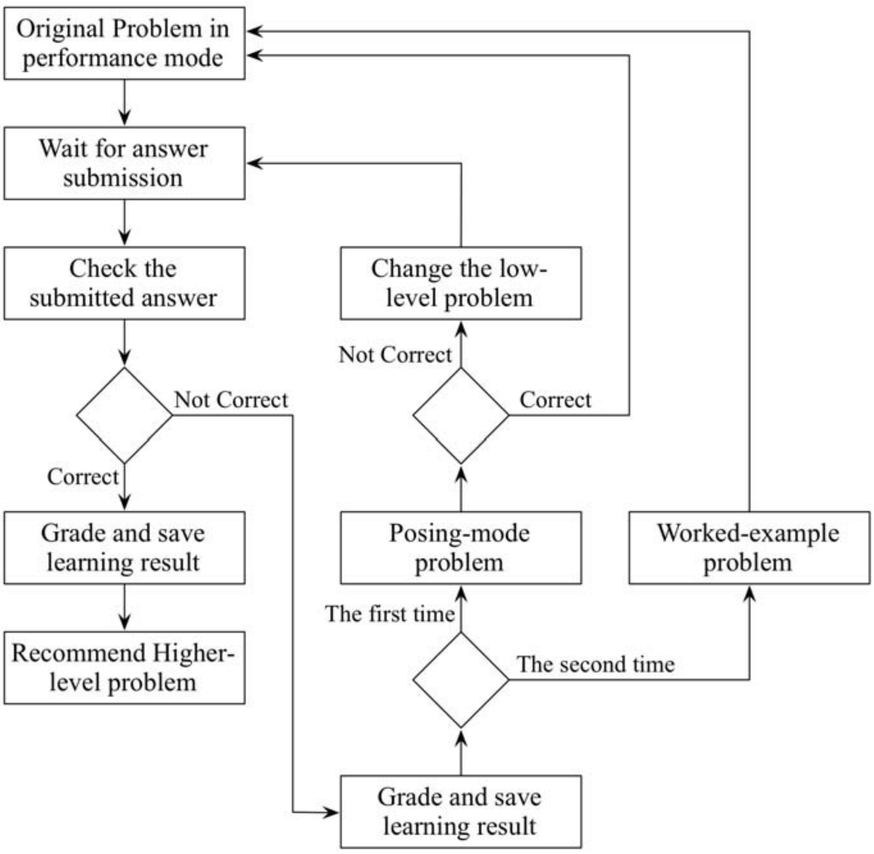


Figure10.Tutoring flowchart of the tutor actor.

We take the projectile problem as the example to introduce the tutoring process. The student is first presented with an open-ended performance problem, and asked to operate the tank to shoot the target in the 3D scene. For that, the student has to use his physics knowledge and calculate the correct horizontal distance between the tank and the target and the correct shooting angle of the gun. The actual values of these data are shown in the problem window when the student operates the tank, moving tank forward and backward, to adjust its position and shooting angle. The shooting action by the student means submitting the answer to TA for the judgment. Once the tutor actor receives the submission, it compares the submitted answer with the correct one in the system, and then presents the learning result through three modes of messages: textual explanation, figure displaying and sound attentions. If the student's answer is not correct, he can require help for tutoring (see Fig. 11).

Figure11. Questioning and tutoring view of GIST.

In the performance mode, the student is asked to solve a problem presented by GIST without any help and hints, and then submit his answer. As an example, the problem of projectile motions in performance mode is shown in Fig. 12, which is presented to the student tied together with the 3D scene.

Operate the tank in the 3D scene to shoot the target(box). You only have one chance to do that, so please calculate the exact values of the shooting angle and the horizontal distance between the tank and the target. The bullet has mass 5Kg and initial velocity of 100m/s.

Figure 12. Projectile problem in performance mode

If the answer is incorrect, GIST helps the student by asking a further task which is based on the mode of problem posing [32]. In this stage, the student is given a scenario which is similar to the original one. In this mode, the student is asked to construct a problem around the scenario that is based on certain physical principles. The problem-posing task which is concerned with the original one is shown in Fig. 13. In this task, all questions are specially designed to test the basic concepts. If the student cannot correctly answer all these questions, then it is impossible for the student to continue the original problem. TA will hint the student to change the learning content. This is the second round learning choice in GIST, contrasted with the first round one introduced above.

A ball in desk with height of 2m is shooting in the horizontal direction with initial speed 10m/s. Which question, when added to the stare above, will make a solvable problem that requires all of the following equations to solve?

$$s = v \times t$$

$$s = v_0 \times t + \frac{1}{2} a \times t^2$$

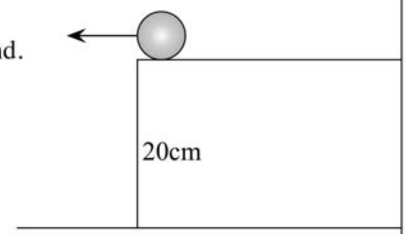
$$\vec{v} = v_x \vec{i} + v_y \vec{j}$$

1) What is the initial velocity in horizontal direction?
2) What is the initial velocity in vertical direction?
3) What is the flight time?
4) What is the flight distance in the horizontal direction?

Figure 13. Projectile problem in problem posing mode

After the learning of all the subjects above, TA will hint the student back to continue to solve the original problem. If there are still difficulties for the student in this stage, TA will present the worked-example task [32] to the student. The worked example problem includes a full solution and is available as a resource for students to use while solving his own original problem. The problem in this mode would be deep structure similar to the original problem, see Fig. 14 as an example.

A ball with mass 10kg on the desk is shooting at initial velocity of 10m/s. Calculate the velocity of the ball when it hits the ground.



Solution

When the ball leave from the desk, the ball is forced by weight force only. The object will keep constant velocity motion in X direction and constant acceleration motion in Y direction.

1) flight time t

$$h = \frac{1}{2} \times g \times t^2 \quad \Rightarrow \quad t = \sqrt{\frac{2h}{g}} = 2s$$

2) velocity in X direction

$$v_x = v_{0x} = 10m/s$$

3) velocity in Y direction

$$v_y = v_{0y} + g \times t = 0 + 10 \times 2 = 20(m/s)$$

4) total velocity

$$v = \sqrt{v_x^2 + v_y^2} = 10\sqrt{5}m/s$$

Figure 14. Worked example problem as adequate scaffolding to the original unsolved problems

BDI-Based Reasoning modelling of the Tutoring Strategy

As pointed above, the tutoring actor needs a complicated decision and reasoning procedure to complete the tutoring work. The BDI model is adopted as the cognitive module of the tutoring actor. In the cognitive layer, the tutoring actor has specific tutoring tasks for a specific student, and its top goal can break down to a series of lower-level goals and eventually to actions at the bottom. Each goal maps to a single plan which either adopts sub-goals or executes actions.

After the student logins into the system and choose the learning problem, TA enters into its position to keep eyes on the student's learning behaviours. The KA hierarchy in TA for the tutoring process is shown in Fig. 15. KAs begin with checking the student's status. Here, the important data which should be collected is the name of the chosen problem. With this name, KAs then display the problem content on a window with textual form and play sounds for attentions.

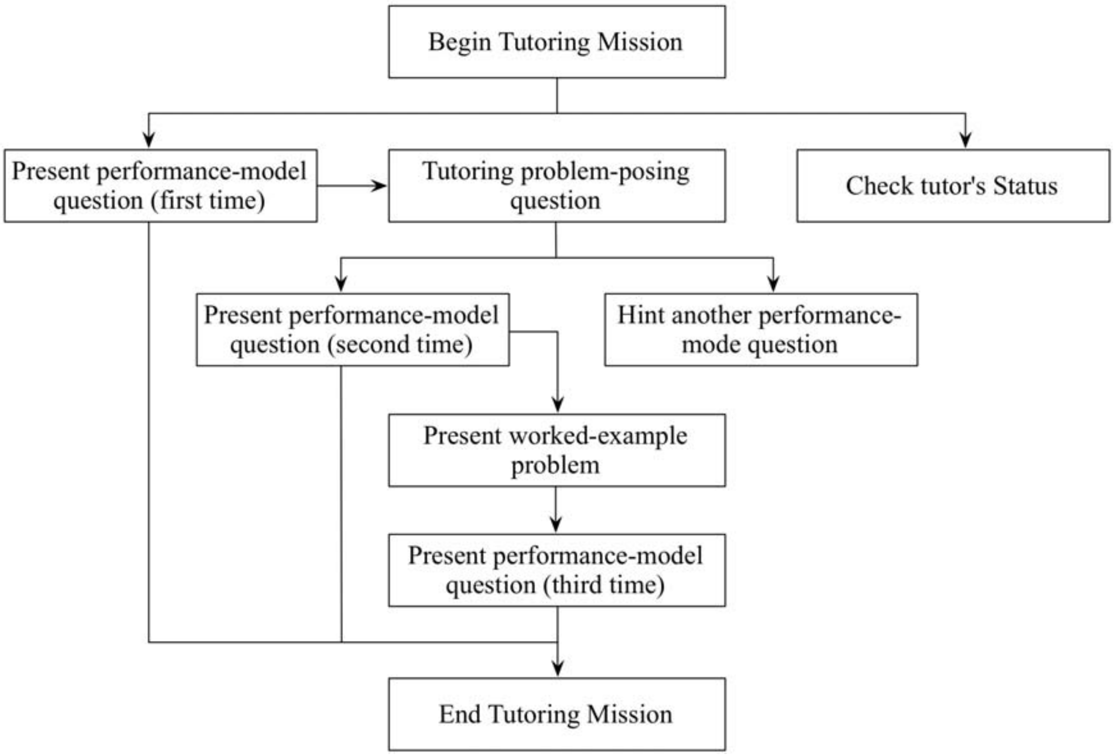


Figure 15. KA hierarchy of TA tutoring process for a problem

Each learning problem is produced by the BDI engine through a textual file. There are three benefits doing like this: The first is that all the initial data in a problem can be varied by the random values, so the problem can be different at each time when student choose to learn; The second is that the problem can be revised by the educators who is not familiar with programming; The third is that it is not necessary to recompile the system. The student can also press the "reset" button on the problem window to explicitly randomize the learning problem.

3.2.4 Interface Module

The complex learning and tutoring processes in GIST require the information of all game actors in 3D virtual environments. Considering the extendibility of GIST and the efficient communication between the game engine and the BDI engine, we design an interface module to bridge these two engines, so that the system complexity can be reduced in a degree, and it is helpful in improving the system development cooperation among game designers, BDI developers and science educators.

3.3 Game Scenarios and Game Levels

The creation of a 3D serious game scenario is more complex than one developed for a classroom case study. Like commercial video games, the learning problems and their corresponding game scenarios in GIST are broken into game levels. The learning manage component in GIST allows the student to progress to higher levels (increasingly more complex knowledge competencies) only once the lower level learning knowledge and competencies have been achieved.

4 Implementation of GIST

Delta3D (http://www.delta3d.org) is employed to develop the 3D game part of GIST, and the component framework of Delta3D is also used to construct the GIST's architecture (see Fig. 2). Each performance-mode problem corresponds to a game level, which is realized by a Map of Delta3D.

In this paper, we chose the BDI (Belief-Desire-Intention) architecture model of UMPRS [27] to define the cognitive layer of TA. Although there are several more advanced BDI engines are available to us: Jadex [4] and JACK [23], UMPRS BDI framework still has advantages: UMPRS is based on the SRI procedural Reasoning System (PRS), which is well suited for use in an agent architecture because it allows the agent to pursue long goals by adopting predetermined procedures based on context and not blindly following prearranged plans. In addition, because of its extensible design and its open C++ source code, UMPRS can be easily integrated into 3D game engine and provide a customable framework.

An example of BDI file, which is used to produce the projectile problem in middle-school physics, is shown in Fig. 16. This KA achieves a sub-goal named "problem", with four specific plans: (1) showing the textual content of the problem; (2) randomizing the velocity of the bullet; (3) calculating the correct answer used to the sequence appraisement; (4) playing the sound to give attention.

In order to implement the tutoring process shown in Fig. 10, TA is designed to have three parameters to control its three tutoring stages and to transfer among these stages.

```
KA{
        NAME
                "present text introduction"
        DOCUMENTATION
                "This is the KA to present an introduction to the learning contents"
        PURPOSE:
                ACHIEVE problem;
        CONTEXT
                FACT learning_task_complete "False";
        BODY:
                ASSIGN $content_0
                "please calculate the horizontal distance between the muzzle and target and the
                adjust the position of the tank to hit the target.g=10.0m/s2, bullet mass m=5kg,
                muzzle height h=1.70m";
                EXECUTE add_problem $content_0;
                EXECUTE float_random $v0;
                FACT bullet_angle $angle;
                ASSIGN $height 1.7;
                EXECUTE calculate_distance $distance $v0 $angle $height;
                EXCECUTE give_answer_to_learningactor $distance;
                ASSIGN $soundFile "introduction_projectile__01_wav";
                EXECUTE play_text_sound $soundFile;
}
```

Figure 16. KA for producing a projectile problem

5 Conclusions and Future Work

The design and development of 3D game-based intelligent science tutoring systems are very complex, thus these systems still have not been used by the general population. In order to achieve their practical application, we need to make trade-off between powerful functionalities and system effectiveness. The main contribution of this paper is the presentation of a prototype design of 3D game-based intelligent science tutoring system (GIST). In this design, we adopt a lightweight modelling of the intelligent tutoring module based on the brief-desire-intention (BDI) framework. The main task of the tutoring module is to suggest to each student specific learning tasks based on his/her learning histories. The system architecture is only dependent of a standard open source game engine.

For example, a game actor is designed to execute the functionalities of tutoring module. Doing this can let us have a coherent development process, so benefit us to reduce the design complex. An interface module is introduced to bridge the game engine and BDI engine, so that different researchers from game designers, BDI developers and science educators can cooperate easily and effectively. In addition, experiment-based (or operation-based) problems are usually considered to be especially suited to 3D games, so we make a try in this paper to integrate algebra-based physics problems into 3D game-based learning systems.
The main future work of GIST lies in two aspects: the first is to improve the intelligence of the tutoring actor. The second is to integrate the simulation functionalities and the assembly/disassembly of scientific Instruments into the learning process, so that increasing the capabilities of GIST to deal with a wider range of real learning problems.

Acknowledgements

The program “Prometeo” and Secretaría de Educación Superior, Ciencia, Tecnología e Innovación from Ecuador, sponsored this work.

References

[1] V. Aleven, B. M. McLaren, and J. Sewall, “Scaling up programming by demonstration for intelligent tutoring systems development: An open-access web site for middle school mathematics learning,” IEEE Trans. Learn. Technol., vol. 2, no. 2, pp. 64–78, 2009.

[2] J. R. Anderson and E. Skwarecki, “The automated tutoring of introductory computer programming,” Commun. ACM, vol. 29, no. 9, pp. 842–849, 1986.

[3] K. Barker, V. K. Chaudhri, S. Y. Chaw, P. Clark, J. Fan, D. J. Israel, S. Mishra, B. W. Porter, P. Romero, D. Tecuci, and others, “A Question-Answering System for AP Chemistry: Assessing KR&R Technologies.,” in KR, 2004, pp. 488–497.

[4] L. Braubach, A. Pokahr, and W. Lamersdorf, “Jadex: A BDI-agent system combining middleware and reasoning,” in Software agent-based applications, platforms and development kits, Springer, 2005, pp. 143–168.

[5] D. Burgos, C. Tattersall, and R. Koper, “Re-purposing existing generic games and simulations for e-learning,” Comput. Human Behav., vol. 23, no. 6, pp. 2656–2667, 2007.

[6] J. C. Burguillo, “Using game theory and competition-based learning to stimulate student motivation and performance,” Comput. Educ., vol. 55, no. 2, pp. 566–575, 2010.

[7] B. P. Butz, M. Duarte, and S. M. Miller, “An intelligent tutoring system for circuit analysis,” IEEE Trans. Educ., vol. 49, no. 2, pp. 216–223, 2006.

[8] J. Cai and S. Hwang, “Generalized and generative thinking in US and Chinese students’ mathematical problem solving and problem posing,” J. Math. Behav., vol. 21, no. 4, pp. 401–421, 2002.

[9] W. Canfield, “ALEKS: A Web-based Intelligent Tutoring System,” Math. Comput. Educ., vol. 35, no. 2, pp. 152–158, 2001.

[10] W.-C. Chang, T.-H. Wang, F. H. Lin, and H.-C. Yang, “Game-based learning with ubiquitous technologies,” IEEE Internet Comput., vol. 13, no. 4, pp. 26–33, 2009.

[11] K.-E. Chang, L.-J. Wu, S.-E. Weng, and Y.-T. Sung, “Embedding game-based problem-solving phase into problem-posing system for mathematics learning,” Comput. Educ., vol. 58, no. 2, pp. 775–786, 2012.

[12] D. B. Clark, B. C. Nelson, H.-Y. Chang, M. Martinez-Garza, K. Slack, and C. M. D'Angelo, "Exploring Newtonian mechanics in a conceptually-integrated digital game: Comparison of learning and affective outcomes for students in Taiwan and the United States," Comput. Educ., vol. 57, no. 3, pp. 2178–2195, 2011.

[13] R. S. Crowley and O. Medvedeva, "An intelligent tutoring system for visual classification problem solving," Artif. Intell. Med., vol. 36, no. 1, pp. 85–117, 2006.

[14] O. J. P. dit Dariel, T. Raby, F. Ravaut, and M. Rothan-Tondeur, "Developing the Serious Games potential in nursing education," Nurse Educ. Today, vol. 33, no. 12, pp. 1569–1575, 2013.

[15] M. D. Dickey, "Engaging by design: How engagement strategies in popular computer and video games can inform instructional design," Educ. Technol. Res. Dev., vol. 53, no. 2, pp. 67–83, 2005.

[16] H. H. Emurian, H. K. Holden, and R. A. Abarbanel, "Managing programmed instruction and collaborative peer tutoring in the classroom: Applications in teaching JavaTM," Comput. Human Behav., vol. 24, no. 2, pp. 576–614, 2008.

[17] F. A. M. Fonte, J. C. Burguillo, and M. L. Nistal, "An intelligent tutoring module controlled by BDI agents for an e-learning platform," Expert Syst. Appl., vol. 39, no. 8, pp. 7546–7554, 2012.

[18] F. A. Mikic-Fonte, J. C. Burguillo-Rial, M. Llamas-Nistal, and D. Fernández-Hermida, "A BDI-based intelligent tutoring module for the e-learning platform INES," in 2010 IEEE Frontiers in Education Conference (FIE), 2010, p. T1J–1.

[19] A. S. Gertner and K. VanLehn, "Andes: A coached problem solving environment for physics," in International conference on intelligent tutoring systems, 2000, pp. 133–142.

[20] L. M. M. Giraffa, M. da C. Móra, and R. M. Viccari, "Modelling the MCOE Tutor Using a Computational Model," in Advances in Artificial Intelligence: 14th Brazilian Symposium on Artificial Intelligence, SBIA'98, Porto Alegre, Brazil, November 4-6, 1998. Proceedings, F. M. de Oliveira, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 50–60.

[21] L. M. M. Giraffa, M. Móra, and A. Zamberlam, "Working on student models (really) based on mental states," in Computational Intelligence and Multimedia Applications, 2001. ICCIMA 2001. Proceedings. Fourth International Conference on, 2001, pp. 379–383.

[22] S.-G. Han, S.-G. Lee, and G.-S. Jo, "Case-based tutoring systems for procedural problem solving on the www," Expert Syst. Appl., vol. 29, no. 3, pp. 573–582, 2005.

[23] N. Howden, R. Rönnquist, A. Hodgson, and A. Lucas, "JACK intelligent agents-summary of an agent infrastructure," in 5th International conference on autonomous agents, 2001.

[24] F. Ke, "Computer-game-based tutoring of mathematics," Comput. Educ., vol. 60, no. 1, pp. 448–457, 2013.

[25] F. Ke, "A qualitative meta-analysis of computer games as learning tools," Handb. Res. Eff. Electron. gaming Educ., vol. 1, pp. 1–32, 2009.

[26] V. Krishnamoorthy, B. Appasamy, and C. Scaffidi, "Using intelligent tutors to teach students how APIs are used for software engineering in practice," IEEE Trans. Educ., vol. 56, no. 3, pp. 355–363, 2013.

[27] J. Lee, M. J. Huber, E. H. Durfee, and P. G. Kenny, "UM-PRS: An Implementation Of The Procedural Reasoning System For Multirobot Applications," in in Conference on Intelligent Robotics in Field, Factory, Service, and Space (CIRFFSS, 1994, pp. 842–849.

[28] J. C. Lester, H. A. Spires, J. L. Nietfeld, J. Minogue, B. W. Mott, and E. V Lobene, "Designing game-based learning environments for elementary science education: A narrative-centered learning perspective," Inf. Sci. (Ny)., vol. 264, pp. 4–18, 2014.

[29] M. Chun, "Taking teaching to (performance) task: Linking pedagogical and assessment practices," Chang. Mag. High. Learn., vol. 42, no. 2, pp. 22–29, 2010.

[30] M. A. Honey, M. Hilton, and others, Learning science through computer games and simulations. National Academies Press, 2011.

[31] J.-C. Marty and T. Carron, “Observation of collaborative activities in a game-based learning platform,” IEEE Trans. Learn. Technol., vol. 4, no. 1, pp. 98–110, 2011.

[32] F. Mateycik, N. S. Rebello, and D. H. Jonassen, “FACILITATING CASE BASED REASONING IN PHYSICS PROBLEM SOLVING.”

[33] L. C. McDermott and E. F. Redish, “Resource letter: PER-1: Physics education research,” Am. J. Phys., vol. 67, no. 9, pp. 755–767, 1999.

[34] J. P. Mestre, “Probing adults’ conceptual understanding and transfer of learning via problem posing,” J. Appl. Dev. Psychol., vol. 23, no. 1, pp. 9–50, 2002.

[35] C. Mills and B. Dalgarno, “A conceptual model for game-based intelligent tutoring systems,” Proc. 2007 Australas. Soc. Comput. Learn. Tert. Educ., pp. 692–702, 2007.

[36] L. S. Myneni, N. H. Narayanan, S. Rebello, A. Rouinfar, and S. Pumtambekar, “An interactive and intelligent learning system for physics education,” IEEE Trans. Learn. Technol., vol. 6, no. 3, pp. 228–239, 2013.

[37] A. J. Neville, “Problem-based learning and medical education forty years on,” Med. Princ. Pract., vol. 18, no. 1, pp. 1–9, 2008.

[38] R. A. I. Dovan, N. T. HEFFERNAN, J. D. GOBERT, and E. Joseph, “MilyΓs World: Math game involving authentic activities in visual cover story,” in AIED 2009: 14 th International Conference on Artificial Intelligence in Education Workshops Proceedings, 2009, p. 125.

[39] I. Roll, V. Aleven, B. M. McLaren, and K. R. Koedinger, “Improving students’ help-seeking skills using metacognitive feedback in an intelligent tutoring system,” Learn. Instr., vol. 21, no. 2, pp. 267–280, 2011.

[40] S. Suebnukarn and P. Haddawy, “COMET: A collaborative tutoring system for medical problem-based learning,” IEEE Intell. Syst., vol. 22, no. 4, pp. 70–77, 2007.

[41] M. Tsuei, “Using synchronous peer tutoring system to promote elementary students’ learning in mathematics,” Comput. Educ., vol. 58, no. 4, pp. 1171–1182, 2012.

[42] B. Vesin, M. Ivanović, A. KlašNja-MilićEvić, and Z. Budimac, “Protus 2.0: Ontology-based semantic recommendation in programming tutoring system,” Expert Syst. Appl., vol. 39, no. 15, pp. 12229–12246, 2012.

[43] M. Waalkens, V. Aleven, and N. Taatgen, “Does supporting multiple student strategies lead to greater learning and motivation? Investigating a source of complexity in the architecture of intelligent tutoring systems,” Comput. Educ., vol. 60, no. 1, pp. 159–171, 2013.

[44] J. J. Vogel-Walcutt, L. Fiorella, and N. Malone, “Instructional strategies framework for military training systems,” Comput. Human Behav., vol. 29, no. 4, pp. 1490–1498, 2013.

[45] M. Ward and J. Sweller, “Structuring effective worked examples,” Cogn. Instr., vol. 7, no. 1, pp. 1–39, 1990.

[46] S. J. Warren, M. J. Dondlinger, J. McLeod, and C. Bigenho, “Opening The Door: An evaluation of the efficacy of a problem-based learning game,” Comput. Educ., vol. 58, no. 1, pp. 397–412, 2012.

[47] W. Westera, R. J. Nadolski, H. G. K. Hummel, and I. G. J. H. Wopereis, “Serious games for higher education: a framework for reducing design complexity,” J. Comput. Assist. Learn., vol. 24, no. 5, pp. 420–432, 2008.

[48] L.-H. Wong, C. Quek, and C.-K. Looi, “TAP: a software architecture for an inquiry dialogue-based tutoring system,” IEEE Trans. Syst. Man, Cybern. A Syst. Humans, vol. 28, no. 3, pp. 315–325, 1998.

[49] M. A. Zielke, M. J. Evans, F. Dufour, T. V Christopher, J. K. Donahue, P. Johnson, E. B. Jennings, B. S. Friedman, P. L. Ounekeo, and R. Flores, "Serious games for immersive cultural training: creating a living world.," IEEE Comput. Graph. Appl., vol. 29, no. 2, pp. 49–60, 2009.

[50] D. Liu, P. Valdiviezo-Díaz, G. Riofrio, Y.-M. Sun, and R. Barba, "Integration of Virtual Labs into Science E-learning," in Procedia Computer Science, 2015, vol. 75.

[51] Calderon-Cordova, C., Guajala-Michay, M., Barba-Guaman, R., and Quezada-Sarmiento, P. "Design of a machine vision applied to educational board game" in Proceedings of 2016 International Conference on Information Technology and Science. 2016.