Artificial Intelligence in Education
C.-K. Looi et al. (Eds.)
IOS Press, 2005

Computer Games as Intelligent Learning Environments: A River Ecosystem Adventure

Jason TAN, Chris BEERS, Ruchi GUPTA, and Gautam BISWAS
Dept. of EECS and ISIS, Vanderbilt University
Nashville, TN, 37235, USA
{jason.tan, chris.beers, ruchi.gupta, gautam.biswas}@vanderbilt.edu

Abstract. Our goal in this work has been to bring together the entertaining and flow characteristics of video game environments with proven learning theories to advance the state of the art in intelligent learning environments. We have designed and implemented an educational game, a river adventure. The adventure game design integrates the Neverwinter Nights game engine with our teachable agents system, Betty's Brain. The implementation links the game interface and the game engine with the existing Betty's Brain system and the river ecosystem simulation using a controller written in Java. After preliminary testing, we will run a complete study with the system in a middle school classroom in Fall 2005.

Keywords: educational games, video game engines, teachable agents, intelligent learning environments

Introduction

Historically, video and computer games have been deemed counterproductive to education [1]. Some educators, parents, and researchers believe that video games take away focus from classroom lessons and homework, stifle creative thinking, and promote unhealthy individualistic attitudes [1,2]. But many children find these games so entertaining that they seem to play them nonstop until they are forced to do something else. As a result, computer and video games have become a huge industry with 2001 sales exceeding $6 billion in the United States alone [3].

Research into the effects of video games on behavior has shown that not all of the criticism is justified [3]. State of the art video games provide immersive and exciting virtual worlds for players. They use challenge, fantasy, and curiosity to engage attention. Interactive stories provide context, motivation, and clear goal structures for problem solving in the game environment. Researchers who study game behavior have determined that they place users in *flow states*, i.e., "state[s] of optimal experience, whereby a person is so engaged in activity that self-consciousness disappears, sense of time is lost, and the person engages in complex, goal-directed activity not for external rewards, but simply for the exhilaration of doing." [4]

The Sims (SimCity, SimEarth, etc.), Carmen Sandiego, Pirates, and Civilization are examples of popular games with useful educational content [3]. However, the negative baggage that has accompanied video games has curtailed the use of advanced game platforms in learning environments. Traditional educational games tend to be mediocre drill and practice environments (e.g., MathBlaster, Reader Rabbit, and Knowledge Munchers) [5]. In a recent attempt to harness the advantages of a video game framework for learning 3D mathematical functions, a group of researchers concluded that doing so was a mistake. "By telling the students beforehand that they were going to be using software that was game-like in nature, we set the [computer learning environment] up to compete against commercial video games. As can be seen by the intense competition present in the commercial video game market, the students' high expectations are difficult to meet." [6].

What would we gain by stepping up and facing the challenge of meeting the high expectations? Integrating the "flow" feature of video games with proven learning theories to design learning environments has tremendous potential. Our goal is to develop learning environments that combine the best features of game environments and learning theories. The idea is to motivate students to learn by challenging them to solve realistic problems, and exploit animation and immersive characteristics of game environments to create the "flow" needed to keep the students engaged in solving progressively more complex learning tasks.

In previous work, we have developed Betty's Brain, a teachable agent that combines learning by teaching with self-regulated mentoring to promote deep learning and understanding [7]. Experiments in fifth grade science classrooms demonstrated that students who taught Betty showed deep understanding of the content material and developed far transfer capabilities [8]. Students also showed a lot of enthusiasm by teaching Betty beyond the time allocated, and by putting greater effort into reading resources so that they could teach Betty better.

A study of game genres [9] has led us to adopt an adventure game framework for extending the Betty's Brain system. We have designed a game environment, where Betty and the student team up and embark on a river adventure to solve a number of river ecosystem problems. Their progress in the game is a function of how well Betty has been taught about the domain, and how proficient they are in implementing an inquiry process that includes collecting relevant evidence, forming hypotheses, and then carrying on further investigations to support and refine the hypotheses. This paper discusses the interactive story that describes the game structure and the problem episodes.

1. Learning by Teaching: The Betty's Brain System

Our work is based on the intuitively compelling paradigm, *learning by teaching*, which states that the process of teaching helps one learn with deeper understanding [7]. The teacher's conceptual organization of domain concepts becomes more refined while communicating ideas, reflecting on feedback, and by observing and analyzing the students' performance. We have designed a computer-based system, Betty's Brain, shown in Fig. 1, where students explicitly teach a computer agent named Betty [10]. The system has been used to teach middle school students about interdependence and balance in river ecosystems. Three activities, *teach, query, and quiz*, model student-teacher interactions. In the teach mode, students teach Betty by constructing a concept map using a graphical drag and drop interface. In the query mode, students can query Betty about the concepts they have taught her. Betty uses qualitative reasoning mechanisms to reason with the concept map. When asked, she uses a combination of text, speech, and animation to provide a detailed explanation of how she derived her answer. In the quiz mode, students observe

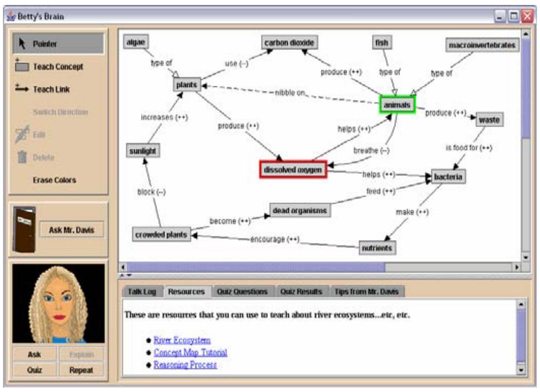

Figure 1. Betty's Brain Interface

how Betty performs on pre-scripted questions. This feedback tells students how well they have taught Betty, which in turn helps them to reflect on how well they have learned the information themselves. To extend students' understanding of interdependence to balance in river ecosystems, we introduced temporal structures and corresponding reasoning mechanisms into Betty's concept map representation. In the extended framework, students teach Betty to identify cycles (these correspond to feedback loops in dynamic processes) in the concept map and assign time information to each cycle. Betty can now answer questions like, "*If macroinvertebrates increase what happens to waste in two weeks?*" A number of experimental studies in fifth grade science classrooms have demonstrated the effectiveness of the system [8].

The river ecosystem simulation, with its visual interface, provides students with a window to real world ecosystems, and helps them learn about dynamic processes. Different scenarios that include the river ecosystem in balance and out of balance illustrate cyclic processes and their periods, and that large changes (such as dumping of waste) can cause large fluctuations in entities, which leads to eventual collapse of the ecosystem. The simulation interface uses animation, graphs, and qualitative representations to show the dynamic relations between entities in an easy to understand format. Studies with high school students have shown that the simulation helps them gain a better understanding of the dynamics of river ecosystems [11]. This has motivated us to extend the system further and build a simulation based game environment to create an entertaining exploratory environment for learning.

2. Game Environment Design

Good learning environments must help students develop life-long learning and problem solving skills [12]. Betty's Brain, through the Mentor feedback and Betty's interactions with the student-teacher, incorporates metacognitive strategies that focus on self-regulated learning [8]. In extending the system to the game environment, we hope to teach general strategies that help students apply what they have learnt to problem solving tasks. The River Ecosystem Adventure, through cycles of problem presentation, learning, teaching, and problem solving, is designed to provide a continual flow of events that should engage students and richly enhance their learning experience (see Fig. 2). Students are given opportunities to question, hypothesize, investigate, analyze, model, and evaluate; the six phases of the scientific inquiry cycle not only help students acquire new knowledge, but develop metacognitive strategies that lead to generalized problem solving skills and transfer [13].

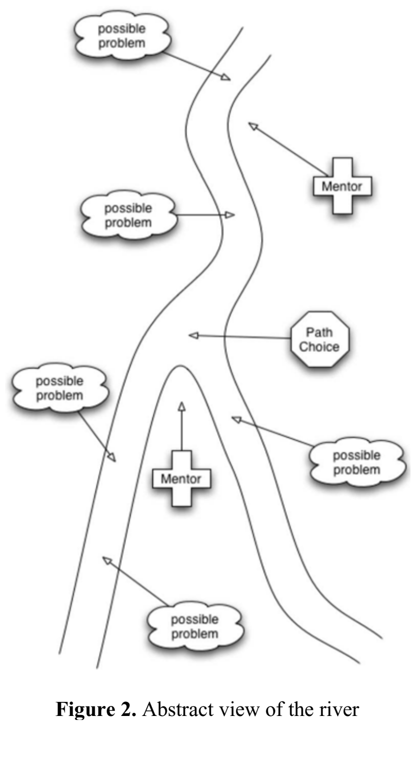

Figure 2. Abstract view of the river

The game environment is set in a world where students interact with and solve problems for communities that live along a river. The teachable agent architecture is incorporated into the game environment. The student player has a primary "directorial" role in all phases of game play: learning and teaching, experimenting, and problem solving. In the prelude, students are introduced to the game, made familiar with the training academy and the experimental pond, and given information about the eco-system problems they are likely to encounter on the river adventure. The *learning and teaching phase* mirrors the Betty's Brain environment. The student and Betty come together to prepare for the river adventure in a training academy. Like before, there is an interactive space (the concept map editor) that allows the player to teach Betty using a concept map representation, ask her questions, and get her to take quizzes. Betty presents herself to the student as a disciplined and enthusiastic learner, often egging the student on to teach her more, while suggesting that students follow good self-regulation strategies to become better learners themselves. Betty must pass a set of quizzes to demonstrate that she has sufficient knowledge of the domain before the two can access the next phase of the game. Help is provided in terms of library resources and online documents available in the training academy, and Betty and the student have opportunities to consult a variety of mentor agents who visit the academy.

In the *experiment phase*, Betty and the player accompany a river ranger to a small pond outside of the academy to conduct experiments that are geared toward applying their learnt knowledge to problem solving tasks. The simulation engine drives the pond environment. The ranger suggests problems to solve, and provides help when asked questions. Betty uses her concept map to derive causes for observed outcomes. The ranger analyzes her solutions and provides feedback. If the results are unsatisfactory, the student may return with Betty to the academy for further study and teaching. After they have successfully solved a set of experimental problems, the ranger gives them permission to move on to the adventure phase of the game.

In the *problem-solving phase*, the player and Betty travel to the problem location, where the mayor explains the problem that this part of the river has been experiencing. From this point on, the game enters a real-time simulation as Betty and the student attempt to find a solution to the problem before it is too late. The student gets Betty to approach characters present in the environment, query them, analyze the information provided, and reason with relevant data to formulate problem hypotheses and find possible causes for these hypotheses. The student's responsibility is to determine which pieces of information are relevant to the problem and communicate this information to Betty using a menu-driven interface. Betty reasons with this information to formulate and refine hypotheses using the concept map. If the concept map is correct and sufficient evidence has been collected, Betty generates the correct answer. Otherwise, she may suggest an incorrect cause, or fail to find a solution. An important facet of this process involves Betty explaining to the player why she has selected her solution. Ranger agents appear in the current river location at periodic intervals. They answer queries and provide clues, if asked. If Betty is far from discovering the correct solution, the student can take Betty back to the academy for further learning and teaching. The simulation engine, outlined in section 2, controls the state of the river and data generated in the environment. A screenshot of the game scenario is shown

Figure 3. Screenshot of the game

in Fig. 3.

As the simulation clock advances, the problem may get worse and it becomes increasingly urgent for Betty and the student to find a solution. A proposed solution is presented to the mayor, who implements the recommendation. Upon successfully solving and fixing the problem, the team is given a reward. The reward can be used to buy additional learning resources, or conduct more advanced experiments in the pond in preparation for future challenges. The challenges that the students face become more complex in succession.

2.1. Game Engine Selection

In order to accomplish our goal of combining the advantages of current video game technology and an intelligent learning-by-teaching environment, we looked at several adventure/RPG game engines. Most of these game engines provide a variety of scripting tools to control the characters, the dialog structures, and the flow of events in the game. In our work, we felt that a game engine that provides an overhead view of the environment would be most suitable for the student to direct Betty's movements and actions in the world, rather than game engines that provide a first-person point-of-view. This led us to select the Neverwinter Nights game engine from BioWare Corp. [14] as the development environment for this project. The game environment, originally based on the popular game, Dungeons and Dragons, includes the Aurora Toolset, a sophisticated content development toolkit that allows users to create new weapons and monsters, as well as new scenarios and characters using scripted dialogue mechanisms. The toolset has been very successful and has spawned many free user-created expansions.

2.2. Development Process

The Aurora Toolset uses a unique vocabulary for content creation. The adventure is created as a *module* containing all the locations, areas, and characters that make up the game. The module is divided up into regions or *areas* of interest. Each area can take on unique characteristics that contribute to different aspects of the game. The primary character in the game (the student) is the *Player Character (PC)*. A number of other characters not directly under the control of the PC can be included in the adventure. They are called the *Non-Playing Characters (NPC)*. In the River Adventure, Betty has an unusual role of being a NPC who is often controlled by the PC. Each individual problem scenario, the training academy, and the pond define individual areas, and the mentor agents, the rangers, and all other characters in the game environment are NPCs placed in the appropriate areas. Some NPCs can migrate from one area to another.

3. Implementation of the Game Environment

One of the benefits of the Neverwinter Nights game engine is that it can be implemented using a client-server approach. This allows us to separate the simulation engine, Betty's AI-based reasoners, and the other educational aspects of the game from the Neverwinter Nights interface. The underlying system based on the Betty's Brain system with the added functionality (described in Section 3) can then be implemented on the server side, as illustrated in Fig. 4.

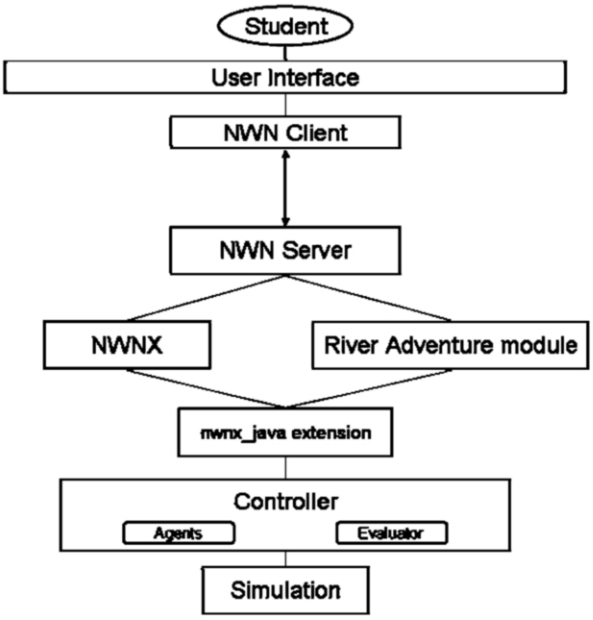

Figure 4. The game environment architecture

A representation of the world is presented to the player by the game engine through the game interface on the client system. The player interacts with the system using a mouse and keyboard to control the movements of his own character and Betty (they move together), click on items of interest (to perform experiments, collect data, check on the concept map, etc.), and to initiate dialog with other NPCs. These define the set of actions that are programmed into the game engine. When students perform an action, it is communicated to the game engine. The game engine controls the visual representation of the world, renders the necessary graphics, and maintains the basic state of the environment and all the characters.

On the server side, the *River Adventure module* describes the location and appearance of each NPC, the details of each area (what buildings and items are present in each scene), how each area connects to other areas, and the overall flow of the game from one level to the next. The Aurora toolset provides a powerful scripting engine used to control the NPC's actions, and other aspects of the module. However, to fully implement the Betty's Brain agent architecture, the river ecosystem simulation, and other more complicated aspects of the system, we utilize the "Neverwinter Nights Extender" (NWNX) [15]. NWNX allows for extensions to the Neverwinter Nights server. In our case, we use the nwnx_java extensions which implements an interface to Java classes and libraries. This allows us to incorporate aspects already implemented in the Betty's Brain system with less effort. The controller and the simulation, implemented in Java, can now be integrated into the River Adventure module. As described in Section 2, the simulation engine uses a state-based mathematical model to keep track of the state of river system as time progresses. Details of this component are presented elsewhere [11], so we do not repeat it here. The rest of this section focuses on the design of the controller, and the updates we made to Betty's reasoning mechanisms to enable her to perform diagnosis tasks.

3.1. The Controller

The controller, made up of the agent architecture and the evaluator, is the core of the intelligent aspects of the game implementation. Additionally, the controller maintains the current state of the game and determines what aspects of the world are accessible to the player. The evaluator assesses the performance of Betty and the student and is used to determine what scaffolding is necessary, as well as maintaining the player's score.

The controller leverages our previous work on multi-agent architecture for learning by teaching systems [8]. Each agent has three primary components: (i) the pattern tracker, (ii) the decision maker, and (iii) the executive. Betty, the mentors and rangers, and all of the significant NPCs in the game world have a corresponding agent within the controller. The pattern tracker monitors the environment, and initiates the decision maker when relevant observable patterns occur. The decision maker takes the input from the pattern tracker and determines what actions the agent should take. Finally, the executive executes these actions, and makes the necessary changes to the environment. Depending on the agent, this could include movement, dialog generation, or a specialized activity, such as making inferences from a concept map or generating help messages. NPC dialogues are generated by retrieving the correct dialog template and modifying it based on the decision maker's output. The controller relays new information resulting from the agents' actions through the nwnx_java plugin to the game module, and also updates the simulation as necessary.

Separate from the agent architecture, the evaluator is the part of the controller that assesses the student's performance and adjusts the game accordingly. The evaluator analyzes the results of the simulation as well as the student's past actions to determine how the game will progress. It takes into account what aspects of the problem the student has yet to complete and sends this information to the game module. The decision makers associated with the mentor agents use this information to determine what level of help the mentors should give the student. If certain aspects of the problem remain unsolved for an extended period of time the mentors can give additional help.

3.2. Betty's extended reasoning mechanisms

Problem solving in the game hinges upon Betty's ability to determine the root cause of a problem given the symptoms and current conditions. Betty's concept map has to be correct and sufficiently complete for her to generate a correct answer. The reasoning mechanism in the existing Betty agent focuses on forward reasoning. It allows Betty to hypothesize the outcome of various changes to the environment. For example, she may reason that if the number of plants in the river increases, then the amount of dissolved oxygen will increase. In the game environment, Betty needs to reason from given symptoms and problems, and hypothesize possible causes. To achieve this, the reasoning mechanism had to be extended to allow Betty to reason backward in the concept map structure. The combination of the forward and backward reasoner defines a diagnosis process [16] that was added to Betty's decision maker. The diagnosis component also gives Betty the capability of choosing the most probable cause when there are multiple possibilities of what is causing the problem in the river. Betty and the student can reflect on this information to decide on what additional information they need to determine the true cause for the problem they are working on.

4. Discussion and Future Work

In this paper, we have designed a game environment that combines the entertainment and flow provided by present day video games with innovative learning environments that support deep understanding of domain concepts, the ability to work with complex problems, and also develop metacognitive strategies that apply across domains. The Neverwinter Nights game interface and game engine are combined with the river ecosystem simulation to create a river adventure, where students solve a series of river ecosystem problems as they travel down a river. The learning by teaching component is retained, and incorporated into the game story by creating an initial phase where the student learns domain concepts and teaches Betty in a training academy. Components of the river adventure have been successfully tested, and preliminary experiments are being run on the integrated system. Our goal is to complete the preliminary studies this summer, and run a big study in a middle school classroom in Fall 2005.

Acknowledgements: This project is supported by NSF REC grant # 0231771.

References

[1] Provenzo, E.F. (1992). What do video games teach? *Education Digest*, 58(4), 56-58
[2] Lin, S. & Lepper, M.R. (1987). Correlates of children's usage of video games and computers. *Journal of Applied Social Psychology*, 17, 72-93.
[3] Squire, K. (2003). Video Games in Education. *International Journal of Intelligent Simulations and Gaming*, vol. 2, 49-62.
[4] Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optical Experience*. New York: Harper Perrennial.
[5] Jonassen, D.H. (1988). Voices from the combat zone: Game grrlz talk back. In Cassell, J. & Jenkins, (Ed.), *From Barbie to Mortal Combat: Gender and Computer Games*. Cambridge, MA: MIT Press.
[6] Elliot, J., Adams, L., & Bruckman, A. (2002). No Magic Bullet: 3D Video Games in Education. *Proceedings of ICLS 2002*, Seattle, WA.
[7] Biswas, G., Schwartz, D., Bransford, J., & The Teachable Agents Group at Vanderbilt University. (2001). Technology Support for Complex Problem Solving: From SAD Environments to AI. In Forbus & Feltovich (eds.), *Smart Machines in Education*. Menlo Park, CA: AAAI Press, 71-98.
[8] Biswas, G., Leelawong, K., Belynne, K., et al. (2004). Incorporating Self Regulated Learning Techniques into Learning by Teaching Environments. in *The 26th Annual Meeting of the Cognitive Science Society*, (Chicago, Illinois), 120-125.
[9] Laird, J. & van Lent, M. The Role of AI in Computer Game Genres. http://ai.eecs.umich.edu/people/laird/papers/book-chapter.htm
[10] Leelawong, K., Wang, Y, Biswas, G., Vye, N., Bransford, J., & Schwartz, D. (2001). Qualitative reasoning techniques to support learning by teaching: The teachable agents project. *Proceedings of the Fifteenth International Workshop on Qualitative Reasoning* , San Antonio 73-80.
[11] Gupta, R., Wu, Y., & Biswas, G. (2005). Teaching About Dynamic Processes: A Teachable Agents Approach, Intl. Conf. on AI in Education, Amsterdam, The Netherlands, in review.
[12] Schwartz, D. & Martin, T. (2004). Inventing to Prepare for Future Learning: The Hidden Efficiency of Encouraging Original Student Production in Statistics Instruction. *Cognition and Instruction*. Vol. 22 (2), 129-184.
[13] White, B., Shimoda, T., Frederiksen, J. (1999). Enabling Students to Construct Theories of Collaborative Inquiry and Reflective Learning: Computer Support for Metacognitive Development. *International Journal of Artificial Intelligence in Education*, vol. 10, 151-182.
[14] BioWare Corp. (2002). *Neverwinter Nights* and *BioWare Aurora Engine*.
[15] Stieger Hardware and Softwareentwicklung. (2005). NeverwinterNights Extender 2
[16] Mosterman, P. & Biswas, G. (1999). Diagnosis of Continuous Valued Systems in Transient Operating Regions. *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans*, Vol. 29(6),554-565.