QuizMASter - A Multi-Agent Game-Style Learning Activity

Mark Dutchuk[1], Khalid Aziz Muhammadi[2], Fuhua Lin[3]

[1] Vancouver Island University, Canada
Mark.Dutchuk@viu.ca
[2] Department of Energy, Government of Alberta, Canada
khalidaziz@gmail.com
[3] School of Computing and Information Systems, Athabasca University, Canada
oscarl@athabascau.ca
http://io.acad.athabascau.ca/~oscar/

Abstract. This paper describes a research project in progress of developing a Multi-Agent System-based educational game QuizMASter for e-learning that would help students learn their course material through friendly competition. We explore the use of perceptive pedagogical agents that would be able to determine the learner's attitudes; to assess learners' emotional states through examining learner's standing, response timing, and history, and banter; and to provide appropriate feedback to students in order to motivate them.

Keywords: Agents, edutainment, virtual game-show, game-based learning, e-learning, virtual learning environment, pedagogy.

1 Introduction

We are working on a research project of developing a Multi-Agent System-based educational game QuizMASter that would help students learn their course material through friendly competition. We explore the use of software agents in educational applications – specifically the use intelligent software agents to provide appropriate feedback to students in order to motivate them.

Conceptually, QuizMASter is designed similar to a TV game show, where a small group of contestants compete by answering questions presented by the game show host. Contestants score points by correctly answering questions before their opponents do. Questions are drawn from a Learning Management System database and presented to players one question at a time. The answer given, along with the length of time taken to respond, is transmitted back to a central agent. Scores will be tallied, and the feedback on a player's standing will be provided to motivate the player.

In QuizMASter, students naturally take the place of game contestants, however the game-show host has been replaced with an intelligent software agent. By studying the reaction of students to the game, and by altering the feedback provided by the game show host, we hope to determine the degree of success the agent has at motivating the player.

2 Related Work

Games are a popular way of enhancing learning, and many researchers have found educational games to be beneficial learning tools [1][2]. Paper [3] provides an excellent overview of game-based learning.

While some argue over the benefits to educational games, Crawford (1982) asserts that games are the *natural* way to learn, and that what we would normally consider "conventional" methods are, in fact, recent unproven teaching methods [4].

Early research by Malone (1981) [5] identified three important characteristics of games that motivate people to play them: Fantasy, challenge, and curiosity. However, later work by Okonkwo and Vassileva (2001) [6] showed that while educational games can be successful at motivating students to play them, their effect on actual learning might be small. Nevertheless, many researchers have used Malone's work as the basis for educational game design.

Kirriemuir et al. (2004) note that, among other reasons, educational games often fail because they are too *simplistic* when compared with commercial video games, too repetitive, and the tasks do not support progressive understanding [7]. Some researchers have attempted to address these issues through the use of intelligent software agents [8]. Intelligent agents are capable of behaviors that can be used to add an element of unpredictability and complexity to game-play. Agents, especially those that present an anthropomorphic persona, have been shown to be highly effective in engaging the student [8].

One well-known study of agents in a learning environment was Herman The Bug / Design-A-Plant [9]. 'Herman' was a virtual insect character capable of 30 behaviors and more than 150 audio responses, who would guide students in the design of a plant suited to an extra-terrestrial environments. Herman was capable of giving advice using both verbal and animated behaviors. It was found that the use of this animated agent had significant benefits to both motivation *and* performance.

MCOE (Multiagent Co-operative Environment) is an educational game that simulates a lake environment [10]. As human players play the game, they learn the effects of pollution and try to control it. MCOE includes purely reactive agents such as fish and plants, and cognitive agents including a tutoring agent and a virtual 'ecologist'. Here, the term '*cognitive'* refers to the ability of the agent to perceive its environment whereupon it may be able to change its behavior accordingly.

Other examples of agent-based educational games include TALENT [11], REAL [12], and Ask&Answer [13]. TALENT is a multi-player game designed to teach programming to University students. A pedagogical agent assumes the role of Mentor to supply guidance, hints, and motivation to students as they work through various programming problems. TALENT's agent forms a model of the learner based on the learner's progress and achievement, and is able to adapt its behavior accordingly [11]. REAL (Reflective Agent Learning Environment) [12] is an agent-based learning environment provides a framework for simulation game scenarios. REAL provides an Expert agent that contains the knowledge about the system being simulated, and a Reflective agent to model what the learner 'knows' about his or her environment. A Pedagogical agent compares the Reflective agent's knowledge with that of the Expert agent's and adjusts it teaching strategy. Finally, a communication agent is used to

handle actual human-computer interaction. In Ask&Answer [13], teams of students are presented with questions that they must answer. In doing so, they interact with an agent named CNPC (Capricious Non-Player Character) that knows the answers to the questions and is willing to hand out important clues that would help a particular team win. However, the agent must be emotionally 'stroked' in order to maintain loyalty to a team. If a particular team fails to satisfy the agent, it will change its loyalties giving advantage to the other team. Given this behaviour, an important component of an overall winning strategy is to maintain a strong social bond with CNCP. As such, Ask&Answer provides researchers with data on how students form and maintain social relations with an artificial entity.

3 Our Approach

Our approach is based on the framework for building classrooms and online learning environments proposed by Bransford et al. (2000), which suggests that learning environments should be knowledge, student, assessment, and community centered [14] (Bransford, et al., 2000), and that game-based learning environments should be enjoyable. There are three primary design decisions in our approach:

1. TV Game Show Format.
Few would deny that TV Game shows are popular. Since their first appearance in the 40's and 50's, TV game shows have attracted a large and steady audience. Examples such as 'The Price is Right' have endured for over 35 years [15].

While there are many educational games available, many are targeted to a younger learner. We believe the TV Game Show format will appeal to an older student, where intellectual stimulation generally holds a higher priority than flashy graphics and/or 'shoot-em-up' action. By putting the emphasis on mental challenge, we hope to avoid the simplicity problems noted by Kirriemuir & McFarlane (2004) [7], while still providing the motivational elements of fantasy, challenge, and curiosity identified by Malone (1980) [2].

2. The Use Of Pedagogical Agents To Provide Feedback and Motivation
By using pedagogical agents we hoped to avoid the simplicity problems mentioned above. Central to this idea is the agent designed for the role of game-show host. In QuizMASter, as in real life, one of the primary roles of the game show host is to keep to game interesting.

Conati & Klawe (2002) assert that, with respect to pedagogical agents in an educational game scenario, it is fundamental that the educational interventions be delivered within the spirit of the game, by characters that (i) are an integral part of the game plot; (ii) are capable of detecting students' lack of engagement, in addition to lack of learning; (iii) know how to effectively intervene to correct these negative emotional and cognitive states [16].

We addressed point (i) by casting our pedagogical agent as the host of a TV game show. Students will easily identify the role played by our host agent, and this should

contribute to the fantasy element of the game. On point (ii) we looked for ways to assess the student's interest and engagement in a manner that was consistent with the game. Three opportunities presented themselves: One, is the standing of the student in the game; two, through the analysis of timing information related to a student's response to questions; and three, by engaging the student in short conversations typical of what a game-show host does between rounds of game-play. On point (iii) we saw an opportunity to favorably *alter* the student's attitude through short conversations between the Host agent and the contestant.

3. The Use Of The Moodle LMS

The Moodle LMS (http://www.moodle.org) provides two important functions for QuizMASter. First, we use Moodle's user management system to maintain user accounts and control access to learning resources. Second, we use Moodle's quiz module as the source for our game questions. Thus, QuizMASter could be integrated in any course where a Moodle quiz is available[1].

Assessing Contestant Emotional State

Within the QuizMASter game environment, we identified three opportunities to gain information on the contestant's emotional state:

- **Contestant Standing**: We assume that a contestant that is winning is sufficiently engaged, and that those in last place are somewhat unhappy with, or uninterested in their performance in the game. Based on such assumptions we use the contestant's current standing as a factor when calculating their attitude.
- **Response Timing and History**: QuizMASter records the contestant's response time to the questions. These can then be analyzed and compared with the other players. Responses that are consistently too fast, too slow, or contain too many wrong answers, might indicate a lack of engagement, or insufficient knowledge on the subject. Currently, QuizMASter only uses this information to display the order in which contestants correctly answered questions, based on their response time.
- **Banter**: Research by Ward & Tsukahara (2003) suggests it is possible to design agents that can infer a user's state from their conversations [17]. In QuizMASter, we will examine the interaction between the host and the player during what we will call 'banter' sessions.

Anyone who has watched TV game shows is probably familiar with the short discussions or 'banter' that the host carries out with each of the contestants. For example, the host may tease a contestant about their work, their family, etc. In the case of QuizMASter, banter provides an important source of information as to the contestant's emotional state and their attitude towards the game-play.

---

[1] Currently, only true/false, multiple choice, and fill-in-the-blank question types are supported.

4 The Architectural Design

QuizMASter consists of a Host agent and two or more Player agents - one for each contestant or team of contestants playing the game. Agents are implemented in Java and the JADE (**J**ava **A**gent **DE**velopment framework) agent development platform (http://jade.tilab.com/). JADE has become a de facto standard in the multi-agent system community. QuizMASter depends on Moodle's database structure for user management and quiz questions. (See Fig. 1).

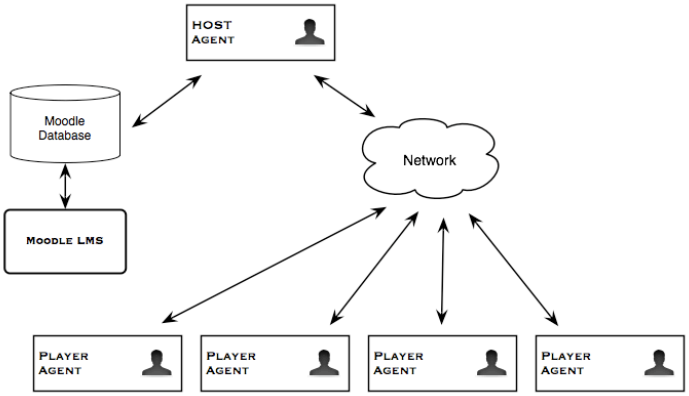

Fig. 1. The Software Architecture of QuizMASter.

4.1 Player Agent

It is the responsibility of the Player agent to:

- Assess and maintain the contestant's emotional state
- Receive and display questions
- Calculate response timing
- Send Response objects back to the Host agent

During game-play, the Player agent will display the new question, and begin a local timer[2]. To score points, the contestant must answer the question first, before other human players, and before the timer expires. When the human player answers the question, or the allotted time has expired, the Player agent will report the player's response and the timing data to the Host agent.

[2] Although the timing appears as a single 'game time' for all contestants, the actual response timing is calculated at the Player agent. This eliminates the possibility that a contestant might be disadvantaged because of transmission delays in sending a response back through the network. Under normal network conditions any timing discrepancies between contestants should be indistinguishable.

The Player agent interface is responsible for displaying information sent from the Host agent (see Fig. 2). In addition to the questions and answers themselves, the contestant's current standing in the game and the number of questions left in the round are displayed. The Player agent also displays the face of the game show host, the conversational text produced by the host agent and the various players, and provides the means for each contestant to contribute to the conversation using an 'instant messaging' style text interaction.

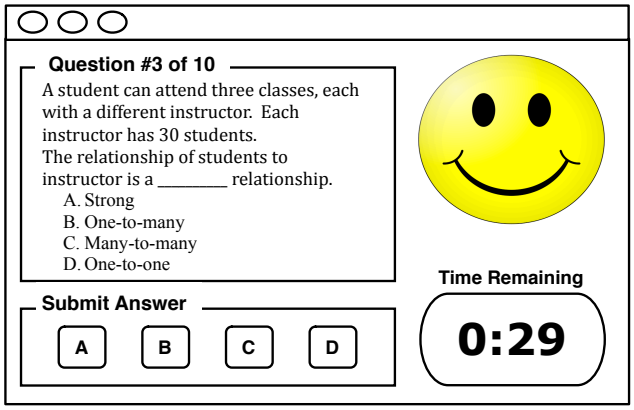

Fig. 2. Prototype Player agent interface presenting a question

4.2 The Host Agent

The QuizMASter Host agent assumes the role of a TV game show host. Consistent with that role, our Host agent provides the following functionality:

- Presentation of questions
- Feedback to the contestants
- Engage in banter sessions with contestants
- Attempt to 'bond' with the contestant, by displaying appropriate emotion.
- Maintain an high level of interest/excitement
- Scorekeeping

Banter Session
Periodically throughout the game (typically after every 2-3 questions) the Host agent will engage in 'banter' with a contestant. The contestant is chosen based on:

- The contestant with the lowest 'emotion' factor
- The winner after the current 'round' of play
- Random selection when no clear data is available

However, the software attempts to talk with all contestants. Once a contestant has participated in a banter session, they are removed from the pick list. Once all contestants have participated, the list is repopulated.

Bonding
Each Host agent attempts to bond with its contestant by celebrating their success, and 'sharing the pain' of an incorrect response. This is accomplished by presenting an appropriate face to the contestant. In an attempt to maintain a positive attitude, the Host agent will normally display a 'happy' face. However:

- Between the time that a new question has been presented and the time the contestant responds, the Host agent will present a 'thinking' face.
- If the contestant answers correctly the Host agent presents a 'happy' face.
- If the contestant answers incorrectly, or a timeout occurs, the Host agent will temporarily present a 'sad' face. This will return to a happy face after a short period of time.

The 'Dramatic Pause'
Consistent with game-show formula, a 'dramatic pause' is introduced into the feedback system before revealing the correct answers to the questions. At the beginning of the game, this pause is minimal. As the game progresses, the pause is increased.

Scoring Subsystem
A scoring subsystem is responsible for processing Response objects, calculating scores and standings for current game. Response objects contain the answers contestants have supplied to a Question, along with timing and emotional state information. Currently, scores are not written back to the Moodle database. In fact, they do not persist beyond the active QuizMASter session.

The Host agent user interface provides for choosing Moodle quiz activities as the source for game-play, inviting contestants, and displaying game status. Fig. 3 shows our prototype Host agent interface during game-play.

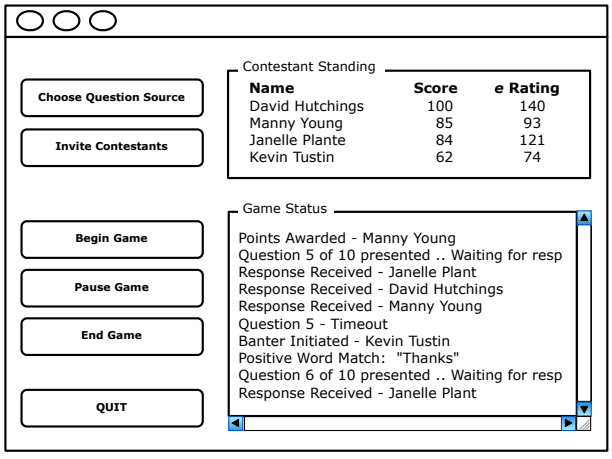

Fig. 3. Prototype Host agent interface during game-play.

5 Implementation

The Player agent class implements the student model, and is the interface through which contestants interact with the system. An instance of the Player agent runs on the contestant's hardware (typically a PC or laptop) and connects to the server-side components via a LAN or Internet connection. Up to four instances of the Player agents may be active at any one time. The Host agent interface provides the ability to select a course to draw the quiz activity from, and invite other players to join. The instructor of a course creates quizzes and stores them in the Learning Management System ahead of time.

A standard Moodle installation provides the required database functionality needed by QuizMASter. Moodle's Question Bank subsystem allows one to add, edit, delete, and import questions to be used by Moodle's Quiz subsystem. Username and password are checked against the user table in the Moodle database, and access and permissions to resources are consistent Moodle's defined roles of 'Teacher', 'Course Creator', 'Administrator', and 'Student'. Java classes were implemented to retrieve appropriate fields from the User, Course, Quiz, Question, and Answer tables of the Moodle database. MySQL was used as the database server, but any Moodle-supported database could be used with very minor alterations to the QuizMASter codebase.

We assess the contestant's emotional state by considering their current standing in the game – first place, second-place, etc., and through the banter sessions. In a typical banter session, the host agent will choose a contestant to chat with, and initiate the chat using a set of pre-defined questions and/or statements. The response from the contestant is compared against two sets of keywords. A response containing words that match those in the 'positive' set of words is an indication of a positive attitude towards the game; a response containing words that match those in the 'negative set of words is an indication of a negative attitude. In our naïve initial implementation, these sets contain only a few words. As we study game-play over many sessions, additional words will be added to both lists.

Initially QuizMASter assumes that all contestants are equally interested in playing the game and assigns an initial value of $e = 100$ to each contestant. During game-play:

- A value of 10 is added for each 'positive' word matched during banter
- A value of 10 is subtracted for each 'negative' word matched during banter
- Contestants in first place, or tied for first place are assigned $e * 1.00$
- Contestants in second place, or tied for second place are assigned $e * 0.80$
- Contestants in third place, or tied for third place are assigned $e * 0.70$
- Contestants in last place are assigned $e * 0.60$

The host agent alters its behavior by preferring to engage in banter sessions with the contestant with the lowest value of e. Through this, we hope to positively affect the contestant's attitude toward the game.

6 Conclusions and Further Research

We have presented QuizMASter, an educational game based on a TV game-show format that uses pedagogical agents to provide relevant and motivating feedback to participants in a game-based learning activity. The current version of QuizMASter provides basic perception and feedback systems to assess a player's attitude during game-play and provide appropriate responses. It has allowed us to identify and study several implementation issues. However, there is considerable room for improvement:

- Currently our prototype Host agent provides text-only interactions. FreeTTS[3] speech synthesis software and Sphinx-4[4] voice recognition software are being considered to provide natural language communications between contestants and the QuizMASter host. This would provide a less distracting interface for contestants, while permitting QuizMASter to identify keywords directly from the contestant's speech.
- QuizMASter could implement a considerably more sophisticated Banter subsystem by using the A.L.I.C.E.[5] chat bot or similar technology. Using A.L.I.C.E., one could provide context-specific game conversations by creating a custom set of AIML categories.
- Software is currently available that is able to detect emotions based on facial recognition. Software such as this could supplement the conversation-based perception subsystem used by the current version of QuizMASter.

Certainly the most significant improvement planned over the next year is to implement QuizMASter in Sun's Wonderland[6] 3D virtual environment. The Wonderland environment will improve our system substantially by adding graphics, animation, and sound elements. These features will go long way to satisfying the educational requirements identified by Malone (1981), Bransford et al. (2000), and Conati & Klawe (2002). Within the Wonderland environment, avatars will represent QuizMASter contestants, and the game show host will be implemented as an Animated Pedagogical Agent (APA). One can envision the ability to choose between a number of TV game show scenarios that would load alternate background graphics to simulate a particular game such as 'Price is Right'[15] or 'Who wants to be a Millionaire?'[7]. Choosing a particular scenario would alter the scoring system's rules and feedback system to be consistent with the particular game being simulated.

QuizMASter in an Immersive Environment
Since reporting on our initial work, we have focused our attention on implementing QuizMASter in a virtual 3D immersive environment using Sun's Project Wonderland Virtual World Toolkit (https://lg3d-wonderland.dev.java.net). The current version of

[3] freetts.sourceforge.net
[4] http://cmusphinx.sourceforge.net
[5] http://www.alicebot.org/about.html
[6] https://lg3d-wonderland.dev.java.net/
[7] http://www.millionairetv.com/

Sun's toolkit has progressed to the point where, in addition to movement through the 3D environment, the avatar subsystem is able to convey simple gestures, and some rudimentary facial expressions. We intend to exploit these features to provide more natural and expressive interactions between QuizMASter's virtual game show host and the game's contestants. The 3D environment also provides the opportunity to create quiz questions and game interactions based on viewing and/or manipulating 3D objects in virtual space. We have elected to use a simple messaging system to bridge the gap between the 3D virtual environment, and the QuizMASter's JADE agent-based environment, and are currently developing the protocols necessary to support the various game scenarios such as question asking and answering, and emotional expression. We hope to have a working prototype running later this year.

References

1. Gee, J. P.: What video games have to teach us about learning and literacy, Computers in Entertainment, 1, 1 (Oct. 2003), 20-20. (2003)
2. Malone, T. W.: What makes things fun to learn? Heuristics for designing instructional computer games, In: Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems. SIGSMALL '80. ACM, New York, NY, 162-169. (1980)
3. Mitchell, A. & Savill-Smith, C.: The Use of Computer and Video Games for Learning. Learning and Skills Development Agency, London. (2004).
4. Crawford, C.: The Art of Computer Game Design. (1982). Out of print: available online at: www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html
5. Malone, T. W.: Toward a theory of intrinsically motivating instruction. Cognitive Science: A Multidisciplinary Journal, Volume 5, Issue 4, pp333-369, (1981)
6. Vassileva, J., & Okonkwo, C.: Affective Pedagogical Agents and User Persuasion. In: Proc. Universal Access in Human-Computer Interaction (UAHCI). New Orleans, USA 397-401, (2001)
7. Kirriemuir, J., & McFarlane, A.: Games and learning, Graduate School of Education, University of Bristol, (2004).
8. Baylor, A.: Beyond butlers: Intelligent agents as mentors, Journal of Educational Computing Research, vol. 22, pp373-382, (2000).
9. Lester, J., Callaway, B., Stone, B.: "Mixed initiative problem solving with animated pedagogical agents". In Haller, S. and McRoy, S.: Working Notes of AAAI97 Spring Symposium on Mixed-Initiative Interaction, pp98-104; Stanford, CA. (1997).
10. Giraffa, L. M. M., Viccari, R. M., and Móra, M. C.: (1998a), Modeling the MCOE Tutor Using a Computational Model. SBIA'98 – Springer-Verlag, Brazil, November, (1998).
11. Maragos, K., & Grigoriadou, M., Designing an Educational Online Multiplayer Game for learning Programming. In: Proceedings of the Informatics Education Europe II Conference (2007).
12. Bai, X. and Black, J. B.: REAL: an agent-based learning environment. Paper presented at the Agent-Based Systems for Human Learning Conference, Hakodate, Japan. (2006).
13. Iurgel, I. A. and Ziegler, M.: Ask & answer: an educational game where it pays to endear your capricious virtual companion. In Lecture Notes in Computer Science, T. Panayiotopoulos, J. Gratch, R. Aylett, D. Ballin, P. Olivier, and T. Rist, Eds. Springer-Verlag, London, 15-24, (2005).

14. Bransford, J. D., Brown, A.L. and Cocking, R. R.: Eds. How People Learn: Brain, Mind, Experience and School, Committee on Developments in the Science of Learning, NRC Commission on Behavioral and Social Sciences and Education, Washington, D.C., National Academy Press. (2000)
15. The Price is Right, http://en.wikipedia.org/wiki/The_Price_is_Right
16. Conati, C., & Klawe, M.: Socially Intelligent Agents in Educational Games. In Socially Intelligent Agents - Creating Relationships with Computers and Robots, Kluwer Academic Publishers. (2002).
17. Ward, Nigel & Wataru Tsukahara: A Study in Responsiveness in Spoken Dialog. International Journal of Human-Computer Studies, 59:603--630 (2003) Online document, available at: http://citeseer.ist.psu.edu/ward03study.html