Using Game Days to Teach a Multiagent System Class

Leen-Kiat Soh
Department of Computer Science and Engineering
University of Nebraska
115 Ferguson Hall, Lincoln, NE 68588-0115 USA
E-mail: lksoh@cse.unl.edu

ABSTRACT
Multiagent systems is an attractive problem solving approach that is becoming ever more feasible and popular in today's world. It combines artificial intelligence (AI) and distributed problem solving to allow designers (programmers and engineers alike) to solve problems otherwise deemed awkward in traditional approaches that are less flexible and centralized. In the Fall semester of 2002, I introduced a new game-based technique to my Multiagent Systems class. The class was aimed for seniors (with special permission) and graduate students in Computer Science, covering some breadth and depth of issues in multiagent systems. One of the requirements was participation in four Game Days. On each Game Day, student teams competed against each other in games related to issues such as auction, task allocation, coalition formation, and negotiation. This article documents my designs of and lessons learned from these Game Days. The Game Days were very successful. Through role-playing, the students were motivated and learned about multiagent systems.

Categories and Subject Descriptors
Course Related, Courseware

General Terms
Design, Experimentation

Keywords
Active Learning, Game-based Learning, Multiagent Systems, Game Days

1 INTRODUCTION
Instruction in multiagent systems (MAS) has become increasingly important in engineering and scientific disciplines because this artificial intelligence technique represents a problem-solving paradigm that is closer to real-world human behavior because of its distributed and autonomous approach. Students in engineering may use multiagent systems to analyze network congestions; those in physics may use MAS to simulate molecule activity; and those in biology may use MAS to perform data fusion, for example.

SIGCSE'04, March 3–7, 2004, Norfolk, Virginia, USA.

An agent is an autonomous software program that *lives* in an environment, monitors the environment, reasons about the events it observes, acts in response to those events, and subsequently changes the environment [19]. Such a program may exist on a computing device, be embedded in a robot, or act on behalf of a user. It also has various degrees of the following attributes: reasoning, communication, learning, and mobility. A multiagent system (MAS) is a collection of such agents that may collaborate to solve a problem or set of problems, or may be in an adversarial relationship in areas of limited resources. A course in MAS usually covers the following topics: (1) foundations of Distributed AI, (2) the concept of agency, (3) MAS organization, (4) MAS communication, (5) MAS reasoning and emergent behavior, and (6) MAS learning.

Note that traditional paradigms have focused on predictable, rather static, and centralized approaches. However, problem solving using a multiagent system (MAS) encourages the students to anticipate failures (due to unpredictability), to plan for changing environments (due to lack of stability), and to distribute decision-making processes (due to decentralization and autonomous agents). MAS is a natural extension of traditional paradigms: the students not only design a single problem, but also must design *programs* to communicate and coordinate to accomplish tasks and an *environment* that supports those programs.

The idea of using games in MAS is not new—many MAS classes have involved programming projects based on games such as Robocup soccer. However, the idea of using *in-class games* to teach students about MAS is innovative and particularly appropriate. In this situation, the students themselves act as agents, instead of getting bogged down with the details of building a multiagent system in some programming language. They are autonomous decision-makers. They are able to communicate with other students, observe what other students are doing, and coordinate their activities. The classroom thus *automatically* becomes a multiagent system. The instructor becomes a monitor of the system, keeping track of the activities and upholding the rules in the multiagent system. In this way, the students can focus on the protocols, paradigms, and issues of MAS topics.

In the following, we first discuss some pedagogical principles underlying the Game Days. In Section 3, we introduce the designs and specifications of the Game Days, including post-game analyses and a game league. Next, we describe the qualitative evaluation of the Game Days on student pedagogy. Finally, we summarize the lessons learned from our experience before concluding.

2 BACKGROUND

2.1 Teaching and Learning Paradigms

The pedagogical principles that provide the foundation for our Game Days are active learning [2, 3], problem-based learning [5, 6, 7], peer (or collaborative) learning [8, 9, 10], and game-based learning [11, 12, 13, 14].

When learning is active, according to Silberman [3], students do most of the work studying ideas, solving problems, applying what they learn, and figuring things out by themselves. In addition, active learning is fast-paced, fun, supportive, and personally engaging. This underlies the assumption that to learn something well, it helps to hear it, see it, ask questions about it and discuss it with others.

Problem-based learning, according to the Center of Problem-Based Learning (CPBL) established by the Illinois Mathematics and Science Academy (IMSA) [4], is a curriculum development and instructional approach. It simultaneously develops problem-solving strategies, disciplinary knowledge bases and skills, by placing students in the active role of problem-solvers confronted with an ill-structured problem that mirrors real-world problems. Most of the problems in the Game Days mirror real-world problems, e.g., forming a group to build a house, negotiating to rescue hostages, bidding to obtain items of high utility, and allocating resources effectively among consumers.

Peer learning is an approach where students support each other in learning. For each Game Day, there are two levels of peer learning. First, each team has multiple team members. The team members work closely to study the open-ended problems for each Game Day, formulate a flexible strategy, and cooperate during the Game Days. Secondly, each team interacts with other teams on each Game Day and learns from others when discussing the results of the Fox and Hound games.

In game-based learning, students engage in the subject and the learning becomes fun. Students are generally motivated in playing games and learn through their effort in trying to win the games. Researchers have also noted other benefits from game-based learning. For example, Haas [12] suggests that game-based learning teaches students to follow directions and use social skills, review information and use cognitive skills, make decisions and live with the consequences, and become aware of new information. Black [10] also suggests that peer learning involved in game instruction allows discussion, reflection, and problem solving, as well as tolerance for self and others. Feezel [11] asserts that game-based instructions also encourage teachers to be creative and more effective. Yaman [15] concurs that games are highly effective in reinforcing learning because students are entertained, less stressed, and work together in teams. In addition, the instructors are able to identify difficult or poorly understood material through observable, immediate feedback from the students.

2.2 Teaching Multiagent Systems

Many instructors of MAS use game-based programming projects in their teaching. A google.com search (conducted July 22, 2003) on "Teaching Multiagent Systems" yield results in which almost all MAS classes that involve at least a programming project but none employing the idea of Game Days. These classes in multiagent systems have incorporated team-based programming projects using gaming testbeds such as Robocup [16, 17], Agent Rescue Emergency Simulator (ARES) [18], and Trading Agent Competition (TAC) [19] to deal with timing, resource sharing, cooperation, communication, and dynamic environments. However, the RoboCup and TAC environments are no longer effective in the classroom as students can cheat using information available on the World Wide Web [18]. Although ARES offers a flexible testbed for teaching MAS, the design is based on a static environment and does not deal with complex real-time issues. Moreover, the above testbed environments are usually semester-long and thus can only be offered once per semester. The setup usually does not encourage the teams to interact with each other pre-, during, and post-game. Hence, as MAS becomes more mainstream, there is a need for easy-to-implement teaching mechanism that involves active and peer learning among students. Our design of Game Days satisfies that need.

3 DESIGNS

There were four Game Days: (1) Auction Day, (2) Allocation Day, (3) Coalition Day, and (4) Negotiation Day. Each Game Day was allotted 75 minutes. Each student group had two students and had a team name.

Each team received a Game Day Package. The exact format of the Game Day Package was given to the students on-line before each Game Day. However, the actual values (utility values, amount of money, etc.) were given out as part of the Game Day Package only on each Game Day. So, the students could work on pre-game strategies using the on-line version beforehand.

I designed a Monitor Package for myself for each Game Day. In this Monitor Package, I had the actual values of every team. Also, I had tables with parameters (that I wanted to track) listed as columns. This Monitor Package allowed me to observe and record the activities conveniently during the games.

I graded each team based on two items: (1) Game Day Worksheets (50%) and (2) End-Of-Day Ranking. I gave customized worksheets to each team as part of the Game Day Package. On the worksheets were itemized rounds, tables, and blanks for the student to record their during-game actions. The students were also encouraged to submit their pre-game discussions and strategies at the end of the Game Day together with their worksheets. At the end of each Game Day, I evaluated each team on their Game-Day Performances and ranked them. Usually, the team that won would have all 50%, the second team would have 45%, and so on.

In the following, I briefly described the four Game Days. Readers are referred to my class website [1] for the detailed Game Day assignments. I used [19] as my textbook for the class.

3.1 Game Day 1: Auction Day

The objectives of Auction Day were to learn and familiarize with various auction protocols, to learn how to manage resources to obtain services/goods of high utility, and to learn how to observe the environment (e.g., the behavior of other agents) to support own decision making process. Each student team's goal was to obtain goods through bidding. Each team's key to winning the game was to obtain goods that were important to itself with the limited amount of resources that each team had.

3.2 Game Day 2: Allocation Day

The objectives of Allocation Day were to learn and familiarize with the various allocation mechanisms, to learn how to consider

or decide which task to perform, and to learn how to re-allocate tasks/resources better from observing the environment. At the implementation level, this Allocation Day also exposed students to how multi-threaded programming was needed for efficient and effective processing for an agent in this environment. Each team's key to wining the game was to solve as many problems as possible with as low costs (costs of tasks and re-allocations) as possible, while helping with as many other teams as possible in solving their problems.

3.3 Game Day 3: Coalition Day

The objectives of Coalition Day were to learn and familiarize with the various coalition formation mechanisms (coordination and communication), to learn how to manage resources to obtain services/goods of high utility, and to learn how to observe the environment (e.g., the behavior of other agents) to support own decision making process. At the implementation level, this Coalition Day also exposed the students to how multi-threaded programming was needed for efficient and effective processing for an agent in this environment. This Game Day focused particularly on three coalition formation mechanisms: blackboard, voting, and matchmaking (or facilitating). Each team's key to winning the games was to solve as many problems as possible while keeping as much money as possible.

3.4 Game Day 4: Negotiation Day

The objectives of Negotiation Day were to learn and familiarize with the various negotiation protocols, and to learn how to observe the environment (e.g., the behavior of other agents) to support own decision-making process. On Negotiation Day, students were required to participate in two types of negotiations. The first was an open, free market where each team was a monopoly on a unique product. Each team also needed to obtain goods from all other teams to solve their problems. The second type was a hostage rescue simulation where the kidnappers and police negotiated using some argument types. I scored the teams playing in the hostage rescue scenarios based on the number of argument types that they used.

3.5 Game Day Packages

The design of the Game Day Packages had the following common features: (a) a very brief, informal introduction, (b) a procedure or setup description, (c) a team-specific description of utilities, (d) a description of how the game was scored, (e) customized and tabulated worksheets to make things as convenient as possible for the students, and (f) an accounting of the items in the game package such as Monopoly paper money, paper tokens, placards, name tags, etc.

3.6 Monitor Packages

For each Game Day, I designed a Monitor Package. The objective of these packages was for me to track key parameters (bid values, transactions, etc.) easily during the games. They also provided me with team-specific utility values so I could resolve any questions or arguments quickly using them as references.

3.7 Post-Game Analysis

For the Post-Game Analysis of each Game Day, I carried out two tasks.

The first task was the evaluation of the Worksheets (Game Day Packages) that each team turned in. I double-checked all the transactions and the Monopoly paper money amounts of the teams to make sure that all monies and paper tokens were accounted for correctly. I also reviewed the videotape to resolve any conflicts I found in the worksheets. I also examined each team's pre-game strategies, in-game observations, and post-game analyses. I strongly encouraged the teams to come up with a set of pre-game strategies beforehand. I also encouraged each team to pay attention to what other teams were doing during games, as agents are required to observe their environments. At the end of each Game Day, I also required each team to speak for about 1 minute about their views of the Game Day and wrote their views down on the Worksheets. All these I took into account when grading the Worksheets.

The second task was my Post-Game Analysis, which comprised the following items: (a) *table of results* based on the parameters that I tracked during the games, (b) *declaration of winners and ranking*, (c) *discussion of operations* where I discussed the operational issues, (d) *general observations*, such as "Nobody posted too many messages on the blackboard." "Some teams were too aggressive—grabbing whatever they saw posted on the board. As a result, they were stuck with too many resources/services," (e) *team-specific observations* where I targeted specifically my comments for each team, such as "Team 1: This team came in without a pre-game strategy. They did learn during the second round that if they could not solve their own problems, then at least they could help other groups solve their problems. They were able to lower their cost very well in Round 1, Situation B," and (f) *lessons learned* where I drew conclusions and related the observations made earlier in the post-game analysis back to the design and research issues that I had covered in the class (about 4-5 lessons learned for each Game Day).

I handed my Post-Game Analysis of each Game Day back to the students immediately (the next day) and discussed some of the key points with them in class. I stressed to them that participating and winning a game was one thing; learning about multiagent systems from the game was another. I emphasized that the latter was the ultimate objective of the Game Days.

3.8 Game Day League

At the end of the semester, I tallied up the Game-Day Performances of the teams for all Game Days and announced the ranking for the Game Days League.

Together with the announcement, I also stressed to the students that the above ranking was simply how they performed on the Game Days, excluding their worksheet scores. This notion of a League was a good one as some teams felt very competitive and tried to do well to win the League.

4 EVALUATION

Before the semester ended, we did a survey about the Game Days. Students were asked to fill out the survey anonymously. The second question of our survey was based on a pedagogical ordering of five items, from the mastery, to the familiarity, and to the exposure of subjects or topics in MAS. Specifically, the questions were grouped into two sets. First, the students were asked to score the helpfulness of each Game Day. Second, the students were asked to score how the Game Days helped them in: in (a) helping me understand the concepts of MAS, (b) helping me understand the design issues of MAS, (c) helping me remember the issues/terms of MAS, (d) helping me appreciate what MAS is about, (e) helping me communicate better (English), and (f)

nothing. In the least, the goal of the class was to expose the usefulness of MAS to the students (d).

Table 1 shows the average scores for Questions 1 and 2. The students thought that Auction Day and Negotiation Day helped them the most, and Allocation Day and Coalition Day not as much. Auction Day and Negotiation Day were easier to play. The rules were simpler; computations were simpler; communication among group members of the same team was not needed as much; the environment was much less dynamic. With a less dynamic environment, the students were able to plan pre-game strategies well.

Q1	**average**
a. Auction day	4.50
b. Allocation Day	4.00
c. Coalition Day	4.17
d. Negotiation Day	4.33
Q2	**average**
a. Understand concepts	4.58
b. Understand design issues	4.00
c. Remember issues/terms	4.17
d. Appreciate MAS	4.08
e. Communicate better in English	3.75
f. Nothing	1.58

Table 1. Average scores for Questions 1 and 2 of Survey.

Most students thought that the Game Days helped them (a) understand the concepts of MAS (4.58/5.0), (b) understand the design issues of MAS (4.00/5.0), (c) remember the issues or terms of MAS (4.17/5.0), and (d) appreciate what MAS is about (4.08/5.0). Based on this survey, we conclude that the Game Days were very successful. Table 2 shows the ranking of the helpfulness of the Game Days and other assignments in the class to the students' learning and understanding of the topics in the class.

Items	**Score**
Game Days	6.33
Final Project	5.92
Topic Summaries	5.92
Lectures	5.58
Homework	4.33
Seminar	4.03
Exam	3.83

Table 2. Ranking of requirements in the class in terms of helpfulness.

In the Score column, the students scored the Game Days' usefulness at 6.33, way above the closest item (Topic Summaries and Final Project at 5.92). Overall, the students thought the Lectures, Game Days, Topic Summaries and the Final Project to be very useful, above 5.0 in a scale of 7.0.

5 LESSONS LEARNED

In this section I discuss the lessons learned from the Game Days.

(a) First of all, the Game Days that I have designed and conducted are natural role-playing games for students in multiagent systems. Each team is an agent and naturally, the class becomes a multiagent system. So the application of multiagent system-related problems to games is straightforward.

(b) Second of all, these role-playing games where students get to move around in class, form their own cliques, and discuss and argue loudly and energetically are very motivating. Students feel a sense of accomplishment. I believe that the face-to-face contacts during the games are a key factor to their enjoyment of the games.

(c) The size of the class has to be small enough. Judging from my experience, I do *not* recommend more than 8 teams in a class. Each team may have 2 to 4 students, however.

(d) Keep the games simple and easy to play. Make them as convenient as possible. Make everything as readily available as possible (tables, worksheets, paper tokens, etc.).

(e) Punish rule-breakers and reward rule-abiders fairly. Some teams are bound to break the rules of the games. Hopefully, they are caught during the games so the impact can be minimized. If not, penalize them post-game.

(f) It is important to make sure that the games are fair to everybody. Students are very particular about this. They want to compete and they want the games to be fair. And it is our responsibility as instructors to ensure that. So, when you assign individual utility values and costs, make sure that they are symmetrical. For games that are not symmetrical (such as consumer groups and provider groups for my Coalition Day), score them differently.

(g) Give the students their Game Day assignments early. For my Game Days, I gave the students the assignment at least one week before the Game Day. Make sure that a general version of the Game Day Package is available to the students. The students are generally motivated to do well on Game Days, and thus they *do* study the assignments a few days ahead of the Game Day, unlike what they do with other, more conventional assignments.

(h) Give more weight to the Game Days. In my class, the Game Days only accounted for 10% of the final grade. I realized in the end that the students learned much more from the Game Days and spent much time on the preparation for the Game Days that they deserved more than 10%. My recommendation for a semester of four Game Days is 15-20%.

(i) Encourage the students to come up with pre-game strategies, to perform in-game observations, to conduct post-game analyses, and to speak out at the end of the Game Day about the games. Give them enough time to share their views with the class. In my class, some teams enjoyed these requirements; some teams did not. Teams that did were better teams.

(j) Be flexible on the Game Days. Since you have to fit the Game Day into a class period, that is a really hard time constraint. Eliminate or shorten rounds that are going on too long. Some students immediately notice the problems with the design as they play the games. Acknowledge them by fixing the problems (if fixable) immediately. Use these fixes in your Post-Game Analysis.

(k) Be persistent and dedicated to your Post-Game Analyses. Games are just games if the students do not learn from them. So, when you analyze the outcomes of the games, give feedback specifically to each team and draw general observations. Most of all, discuss the lessons learned from the viewpoint of a multiagent system designer. Relate the lessons back to the topics or subjects taught in the class. Make the connections between the games and

the lectures for the students explicit through your Post-Game Analyses.

(l) Invite other faculty to visit your Game Days. Introduce the visitors to your class at the beginning of your Game Days. The students have a sense of pride and tend to do better to show the visitors that they are good and they have fun.

(m) Every student must be present on Game Days, especially for 2-member teams. I had two occasions where a student failed to show up and another showed up late. In both occasions, the team with only one student had to work doubly hard to cope with the "processing" of information and events in the games. I penalized harshly on those students who failed to show up or show up on time, and gave those students working on their own extra points.

(n) Be prepared to spend time pre-game and post-game when you first incorporate the Game Days into your class. Based on my experience, I spent probably 20 hours, for each Game Day, on defining the assignment, preparing the Game Day Packages and the Monitor Packages, designing the utility values and costs, and double-checking all numbers were correct. I spent another 5 hours or so on the Post-Game Analysis for each Game Day. Luckily, each Game Day, once designed, is re-usable. That means, in the future, I only have to spend, say, a few hours on pre-game design and packaging.

(o) Make your Post-Game Analyses available to the next class for their Game Days. Thee provide valuable insights for those future students.

6 CONCLUSIONS

In this paper, we have described a new approach to teaching a class in MAS: Game Days. We have designed and conducted four Game Days: Auction, Allocation, Coalition, and Negotiation. We have discussed each game day in terms of its design and specifications, and illustrated the design of the Game Day Packages and Monitor Packages. In addition, we have outlined the key topics in the Post-Game Analyses. Moreover, we have presented a qualitative evaluation based on a semester-end survey. The survey indicates that the students enjoyed the Game Days and agreed that the Game Days were helpful in their learning and understanding of the materials taught in the class. Finally, we have listed a set of agendas as lessons learned from my experience for all instructors who are interested in adopting Game Days for their MAS classes.

Our future work includes continued refinement of Game Days for the MAS class, and additional summative and formative evaluation of the students' learning of MAS topics.

REFERENCES

[1] URL: http://www.cse.unl.edu/~lksoh/Classes/CSCE475_896_Fall03/gamedays.html

[2] Bonwell, C. C. and J. A. Eison (1991). Active Learning: Creating Excitement in the Classroom, *ERIC Clearinghouse on Higher Education*, Document No. ED 340 272.

[3] Silberman, M. (1996). *Active Learning: 101 Strategies to Teach Any Subject*, Allyn & Bacon.

[4] URL: http://www.imsa.edu/team/cpbl/cpbl.html

[5] URL: http://www.samford.edu/pbl/search/whosearch.shtml.

[6] URL: http://www.mcli.dist.maricopa.edu/pbl/info.html.

[7] Boud, D., R. Cohen, and J. Sampson (2001). *Peer Learning in Higher Education: Learning from and with Each Other*, Kogan Page.

[8] Wills, C. E., and D. Finkel (1994). Experience with Peer Learning in An Introductory Computer Science Course, *Computer Science Education*, **5**(2):165-187.

[19] Tinto, V., A. Goodsell, and P. Russo (1993). Collaborative Learning And New College Students, *Cooperative Learning and College Teaching*, **3**(3):9-10.

[10] Black, R. L. (1992). Pharmacology Instruction: A Game Approach for Students, *Nurse Educator,* **17**(2):7-8.

[11] Feezel, J. D. (1993). Preparing Teachers through Creativity Games, *Proceedings of the Joint Meeting of the Southern States Communication Association and the Central States Communication Association*, April, Lexington, KY.

[12] Haas, M. E. (1988). Using Small Group Games in Social Studies, *Proceedings of the Annual Meeting of the National Council for Social Studies*, November, Orlando, Florida.

[13] Prensky, M. (2001). *Digital Game-Based Learning*, McGraw-Hill.

[14] Daniel, G. and K. Cox (2002). Computer Games in Education, *Web Tools Newsletter*, 25 June 2002.

[15] URL: http://learningware.com/whatsnew/gameswork.html

[16] Robocup initiative, URL: http://www.robocup.org/

[17] Buhler, P. and J. M. Vidal (2001). Biter: A Platform for the Teaching and Research of Multiagent Systems' Design using Robocup, *Proceedings of the International Robocup Symposium*.

[18] Bergen, M., J. Densinger, and J. Kidney (2003). Teaching Multi-Agent Systems with the Help of ARES: Motivation and Manual, *Proceedings of the Western Canadian Conference on Computing Education,* Courtenay, 2003.

[19] Weiss, G. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press.