Received April 9, 2015, accepted April 24, 2015, date of publication June 3, 2015, date of current version June 18, 2015.

Digital Object Identifier 10.1109/ACCESS.2015.2442680

Challenges and Opportunities in Game Artificial Intelligence Education Using Angry Birds

DU-MIM YOON AND KYUNG-JOONG KIM
Department of Computer Engineering, Sejong University, Seoul 143-747, Korea

Corresponding author: K.-J. Kim (kimkj@sejong.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (2013 R1A2A2A01016589).

ABSTRACT Games have been an important tool for motivating undergraduate students majoring in computer science and engineering. However, it is difficult to build an entire game for education from scratch, because the task requires high-level programming skills and expertise to understand the graphics and physics. Recently, there have been many different game artificial intelligence (AI) competitions, ranging from board games to the state-of-the-art video games (car racing, mobile games, first-person shooting games, real-time strategy games, and so on). The competitions have been designed such that participants develop their own AI module on top of public/commercial games. Because the materials are open to the public, it is quite useful to adopt them for an undergraduate course project. In this paper, we report our experiences using the Angry Birds AI Competition for such a project-based course. In the course, teams of students consider computer vision, strategic decision-making, resource management, and bug-free coding for their outcome. To promote understanding of game contents generation and extensive testing on the generalization abilities of the student's AI program, we developed software to help them create user-created levels. Students actively participated in the project and the final outcome was comparable with that of successful entries in the 2013 International Angry Birds AI Competition. Furthermore, it leads to the development of a new parallelized Angry Birds AI Competition platform with undergraduate students aiming to use advanced optimization algorithms for their controllers.

INDEX TERMS Computing education, Angry Birds, game, edutainment, program design, artificial intelligence (AI), game competition, game-based learning.

I. INTRODUCTION

Angry Birds is one of the most popular mobile games based on physics [1]. The goal in the game is to shoot a bird using a slingshot and thus damage pigs. Because the targets often remain inside a building structure, it is difficult to directly kill the pigs. Instead, the game requires a type of strategy that destroys parts of the buildings and indirectly damages the pigs. In this way, it is possible to clear a stage using a limited number of birds. Because the game is based on physics, the player needs to consider relationships among objects, building structure, and the properties of objects. Fig. 1 shows a screenshot of the mobile game.

Recently, there have been many artificial intelligence (AI) competitions as special events for international AI game conferences [2]. The competitions include a variety of game genres including first person shooting [3], car racing [4], Super Mario [5], Ms. Pac-Man, Angry Birds, real-time strategy [6], and board games. In the competitions, the organizers provide a software platform on which to build AI into the target games. In the development, participants can use any approach (including rule-based methods) to create strong artificial players for the games. Based on predefined rules of

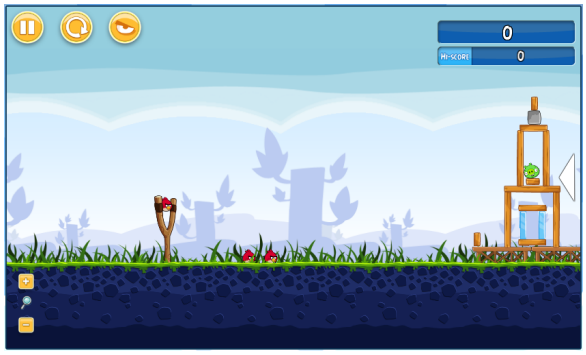

FIGURE 1. A screenshot of the Angry Birds game. The three red birds on the left are game player's weapons and can be shot using the sling. The green pig on the right is the target that the player has to clear. Because the game has a 2D physical engine, objects (wood, stone, and ice with the pig) can be broken and collapsed by shooting the birds.

the competition, entries compete with each other to win the challenge.

Although the main goal of the AI competitions is biased toward research (e.g., benchmarking and test bedding new techniques), the competitions have promising potential as educational resources [2], [7]. Because they are based on games, it is easy to attract student interest. Also, basic software tools (API and game SW[1]), manuals, and rules for the competitions have been opened to the public to increase the number of participants, and these can save much time for educators who intend to deploy the competition in their class. Finally, the students can be motivated to improve their work for subsequent international competitions.

Since 2012, such international competitions have been held for Angry Birds mobile games [8]. The goal of the competition is to build software that allows one to play Angry Birds games on unseen levels. The most recent program entailed analyzing the structures of buildings and finding the best shooting point to obtain the best score. Because the competition levels are not known, the program must be designed to generalize well on unseen test levels. There were 19 participating teams from 13 countries in the 2013 IJCAI[2] competition.

In the fall semester of 2013, we offered a project-based course for undergraduate students (especially juniors) with computer engineering majors based on the Angry Birds AI competition. The course was designed to attract students to advanced programming, including handling uncertainty, physics-based simulation, image processing, game content creation [9], and resource management. During the course, students (a team of 3∼4 members) experienced several internal competitions with different levels and rules. In this paper, we describe the course contents, designed based on the game AI competition, and the results of the course.

Based on our early experience [2], [7], it is ineffective to simply tell students that their goal is to build an entry for the next game AI competition and prepare the event for credit in this course. Instead, it is desirable to introduce step-by-step instructions and run internal competitions of gradually increasing difficulty, which allows students the chance to revise their programs. Because the international competition rules are a bit difficult for beginners, it is critical to introduce multiple milestones. Finally, their programs are evaluated over 2 days under the same conditions as the 2013 IJCAI Angry Birds AI competitions. In addition to the controller creation (mainly design and programming tasks), students also experienced the game contents generation (game levels) with our software to support the replacement of original levels in Angry Birds with user created one.

Although AI competitions can be very useful tools for education for both AI-related and non-AI courses, it is not straightforward to introduce them as a course project. In this paper, we attempt to share our experience in adopting the Angry Birds AI competition for the undergraduate project-based course. Although our case is for the Angry Birds AI competition, educators could use our template syllabus to extend their course to other game AI competitions (car racing, unreal tournament, Pac-Man, Super Mario, and so on).

[1]All source codes (including games) for competitions are usually open source. However, some competitions require the purchase of inexpensive commercial game SW (e.g., StarCraft and Unreal Tournament 2004).

[2]International Joint Conference on Artificial Intelligence.

II. BACKGROUND

A. ANGRY BIRDS

Angry Birds is one of the most popular 2D video games for mobile environments. Although the game was designed initially for mobile phones and tablets, it can be played on top of web browsers (for example, Chrome) in desktop environments. This game has a simple control method but complex game levels, which is a very attractive feature for game playing (easy to learn but difficult to master). Basically, one can score by shooting birds against pigs and other objects. Because this game has different types of birds, pigs, and objects, the game player should consider many factors to maximize the chance of killing all of the pigs. It is desirable to save shooting birds (remaining birds can be counted for bonus score) and playing time to maximize the total score (if the stage can be cleared quickly, other levels can be attempted given sufficient time). If a player fails to kill all of the pigs in the level, the player obtains a score of zero, so it is important to make trade-offs between the opportunity of killing all of the pigs and the time expended. There are many different levels of games and themes (a set of levels designed with special themes), and self-designed problems can be created for one's own enjoyment.

B. AI BIRD COMPETITIONS

For several years, there have been many different game AI competitions based on commercial or open-source games. These include 2D car racing simulator (TORCS open source car racing simulator), infinite super Mario (open source), Ms. Pac-Man, unreal tournament (first-person shooting), StarCraft, and so forth. In the case of commercial games, it is not easy to obtain support from the creator (company) allowing the building of AI for the games. For this case, the organizers built a specialized meta-interface to play the game with a customized AI. For example, a toolkit from the organizers continuously captures the screen of Ms. Pac-Man (source code is not available) and analyzes the inputs to provide current states (position of enemies, obstacles, scores, and the player). They also provide software codes to generate virtual key events (mouse click or key press) for playing the games using an AI module.

The Angry Birds AI competition has been organized since 2012 [8]. Because the competition does not obtain support from the game creator, they use the same protocol to run the game AI competition as for the Ms. Pac-Man competition. Angry Birds is run on top of the Chrome web browser, and

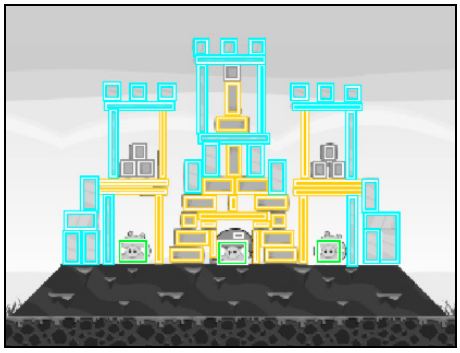

FIGURE 2. An example of a screenshot analyzed by the built-in image-processing library from organizers. It recognized 30 ice objects (cyan color), 22 woods (yellow), 7 stones (gray), and 3 pigs. However, there are actually 31 ice objects and no information is provided on the types of pig, which indicates that the image processing is not perfect.

the software from organizers captures the screens of games and identifies objects using simple image processing (Fig. 2). The goal of the competition is to obtain the total high scores for several unknown levels within a limited time. The 2013 events attracted 19 teams and reported that the winning team outperformed the best player in 2012.

There are several constraints that make the programming difficult. These include uncertainty caused by the image processing, and because the analysis is not perfect, some errors in inputs. It is also possible to have a small time delay caused by image processing time. In summary, the current state might differ slightly when the AI module actually shoots the birds. It is not easy to predict the outcome of the planned actions because the game is based on a physics engine (Box 2D). A small change in trajectory could result in different outcomes. Although there have been some attempts to use Box 2D for the competition AI module, the results have not been promising.

C. GAME-BASED LEARNING

The discipline of education has been researched for a very long time. Studies have focused on ways to motivate students because motivation leads to improvement in knowledge. For this reason, some researchers have used problem-based learning [10], with a team-based approach to problem solving to attain active student participation. Other researchers have included an additional element in learning that is fundamental to motivation: fun [11]. This is in an effective method that has brought about the use of a new word, 'edutainment,' and has encouraged many studies based on the combination of education and entertainment. However, some studies have concluded that playful learning is a more efficient approach than edutainment [12].

Video games are very interesting subjects for motivating students' learning because games have many interesting elements such as fantastic graphics, audio, and real-time interactions and can provide specific objectives in the game [13]. Before video game-based research, video games had generally been regarded simply as a type of entertainment and seen as a waste of time by adults. Nevertheless, video games have been proven by dozens of game researchers to provide not only an interesting but also an efficient learning tool [14], [15].

To date, there have been a number of studies on game-based education (Table 1). As shown in the table, both single-based and competition-based games are useful for teaching. In this paper, we applied AI competition-based game AI development to our program design course.

In the game AI competition-based courses, it is important to know the key property of the games used. For example, if the course adopts games with real-time constraints, a major issue of real-time AI development for games is that the AI agent should respond as soon as possible with many various game environments. In general, real-time games (such as action games, platform games, and racing games) require an AI with a fast response because of the short response time. On the other hands, the Angry Birds game requires AI focused on recognizing the environment and understanding the structure of objects rather than on computing speed.

Consequently, there are many AI competitions including the AI Birds competition [2], [16]. If some students or teachers do not want to use AI Birds, they can use other cooperative or competitive game AI competitions in software education. As we mentioned above, using the game AI competition approach is a good way to teach a program design course. For example, those who want to develop game AI can begin easily because AI can function with even a simple rules system. In addition, because game AI performance testing is similar to game play with the AI, it provides a joyful experience to students even during tests.

III. COURSE DESIGN

In this section, we introduce in detail our course design. First, the given objective of this course is to create an Angry Birds AI that can obtain the highest score. For completion of this objective, we discuss important issues to be faced during AI development such as strategy selection and time management. In the next section, we discuss the team assignment, evaluation method, and questionnaire contents of our course. In the final section, we introduce a measurement method for students' AI agents with customized game levels, because to win in the AI Birds competition, agents should obtain a highest score in not only known game levels (Poached Eggs, the first episode of Angry Birds) but also unknown game levels (customized levels).

A. ANGRY BIRDS AI-BASED PROGRAM DESIGN

In the Angry Birds game, it is difficult to obtain a high score despite the fact that it uses a simple interaction: just drag and drop. Because there are many types of obstructive breakable objects, unbreakable ground, birds, and pigs, it has a large number of solution cases. Furthermore, an AI has to catch all pigs with as small a number as possible of birds in the 2D physics engine-based environment.

TABLE 1. A list of previous research.

Authors	Ref	Game or tool	Course	Related competition	Summary
			Game-based teaching		
M. Overmars	[17]	Game Maker	Computer Science	-	He taught computer science through game design with Game Maker.
W. Chen *et al.*	[18]	The 'Game' framework	Object-oriented programming (OOP) laboratory course	-	They taught OOP with game programming.
X. Yao *et al.*	[19]	Japanese puzzles: Oekaki logics (お絵かきロジック)	Evolutionary Algorithms	-	The puzzle is an interesting and addictive game and is a suitable game for teaching.
H. C. Jiau *et al.*	[20]	A SIMulated Programming Learning Environment	Programming	-	He used self-motivation for programming learning with game-based simulation and metrics.
B. -S. Jong *et al.*	[21]	Peer interaction game	Operating Systems	-	They used game-based cooperative learning to improve learning motivation with a cooperative online game.
I. Paliokas *et al.*	[22]	PlayLOGO 3D	Programming	-	They taught LOGO programming to children aged 6–13 years based on 3D video games.
M. Chang *et al.*	[23]	A web-based multiplayer online role-playing game	Java Programming	-	He evaluated Java programming grade using a web-based multiplayer online role playing game.
J. D. Bayliss	[24]	Minecraft	Artificial Intelligence for Games	-	He taught game AI with Minecraft, which is a modding-allowed commercial sandbox-style game mod.
			Game competition-based teaching		
K.-J. Kim *et al.*	[7]	The Open Car Simulator (TORCS)	Introductory Programming	IEEE conf. on CIG Car Racing Competition	They taught C programming, using it to make AI drivers.
J. DeNero *et al.*	[25]	Pac-Man	Introductory Artificial Intelligence	Competition designed by the authors	They taught four topics (state-space search, multi-agent search, probabilistic inference, and reinforcement learning) of AI using Pac-Man.
M. E. Taylor	[26]	Generalized Mario	Computer Science & Artificial Intelligence	RL-competition	He taught reinforcement learning with a Mario game.

Note: The upper list of research applied a basic approach that is to use a game or a tool for teaching. The lower list used competition-based gaming to teach. This method has a synergistic effect in that the game promotes a competitive spirit among students.

An AI that considers bird type-based potential energy, the force of gravity, shooting speed and angle, and object structure analysis can obtain a higher score than one that does not. In this paper, we choose vision, game level analysis, time management, playing strategy, and uncertainty handling as the most important issues for making an AI.

1) VISION

The basic AI agent provided by the AI Birds competition recognizes current game status through real time-based in-game visual images, such as human gamers, instead of internal game data (Fig. 2). Basically, the provided built-in API can recognize all kinds of bird and object types, such as red birds, yellow birds, stone, wood, and so forth, but it provides only simple information and sometimes recognizes a wrong object (it often determines a wrong or missed bird type, recognizes a menu or background as an object, or misses a sling position). Furthermore, it cannot recognize unbreakable ground obstacles (dark brown color) or non-orthogonal objects.

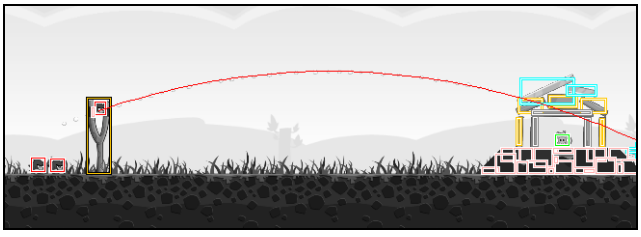

FIGURE 3. This image shows the trajectory of a shooting with enhanced vision. The large rectangle at upper-right represents a tilted ice block (this client of version 1.23 cannot recognize tilter, unlike version 1.30). The small blocks at bottom-right represent recognized obstacle ground (this function was added by the students of team Simple Birds).

Fortunately, students can improve the vision performance of an agent by directly modifying the AI source code (Fig. 3) For example, they enhanced the vision using the built-in vision API for a zoom-in command and pixel analyzing in the sling area (Fig. 4). Another example is a ground obstacle. A developer can make a vision system that can recognize an unbreakable object using a unique RGB color

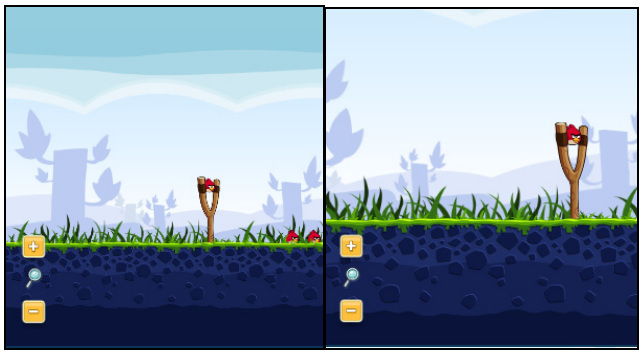

FIGURE 4. These images show a zoom function in the Angry Birds game. The left image is the default zoom-out state. The AI agent generally uses this state because it provides an overview of the current level. However, when a game is played with a large-size level, it can be difficult to recognize birds. In this case, a zoom function can make it easier to recognize bird type (right).

value (dark brown). By such methods, the agent has the potential to overcome these problems. However, it should manage computing power sources because AI Birds runs in limited time.

2) GAME LEVEL ANALYSIS

To obtain a high score, the agent needs to analyze the state of the current game level in consideration of the physics engine. An agent should address the strengths and weaknesses of the five types of birds (red, blue, yellow, white, and black) and five types of objects (ice, wood, stone, ground and TNT) used in the game level. For example, the blue bird is strong against ice blocks and can duplicate to three birds with similar directions with a secondary mouse-button click (click or tap while flying the bird). The yellow bird is strong against wood blocks and acquires more speed by a secondary click, and the TNT block provides additional impact to surrounding objects when it breaks.

Furthermore, an after effect caused by shooting damage can provide additional damage to all surrounding objects by the physics engine, so an agent can use this effect if shooting trajectories are planned by structure analysis. Generally, a shooting bird has higher damage power when it is shot with a low angle or when it uses a special ability and can thus produce damage more effectively when it crashes into the vertex of building blocks. The special ability of each bird (except the red bird) is an important function for a high score because it produces a different result according to the timing and position used (Fig. 5).

3) TIME MANAGEMENT

Time management is another important issue in the Angry Birds. The 2013 AI Birds competition ran 10 levels for 15 min in the qualifying round and for 30 min in the final round (qualifying: one and a half min per level; final round: three min per level). All agent behavior for visual analysis, i.e., zooming in/out in game screen (for better interpretation of game objects), changing level, and so forth, expends some

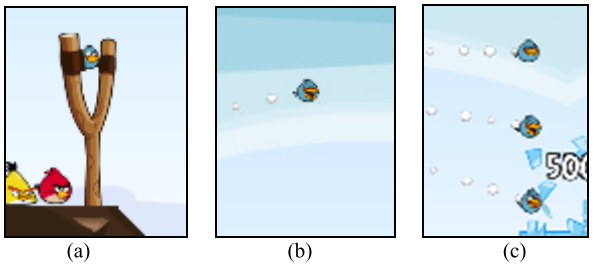

FIGURE 5. All birds except the red bird have a special ability. An agent checks the next shooting bird on the sling (a), and the flying bird (b) uses its special ability with a mouse tap (c). The special ability of a blue bird is to divide into three birds.

working time, and if the agent cannot clear a level, a score in the level will not be recorded. This means that an agent that has behaviors with minimized unnecessary behaviors will have a greater chance to be a winner in the competition.

For example, when the agent shoots a bird rapidly to clear the level, the best timing to take a screen shot for visual analysis is one time management issue. In this case, a general answer is 'it should take a screenshot as soon as possible in the limited time.' However, this does not work in the real game because an effect from a previous shooting of a bird still influences surrounding objects, and it causes the agent to wait until all objects are stable for the screenshot. Having only a screenshot with stable objects however can provide the correct state of the current game level to the agent. Thus, an agent should consider screenshot timing for time management.

4) PLAYING STRATEGY

The basic objective of Angry Birds is to catch all of the pigs and get the highest scores in a level, but the AI Birds competition objective is a little different. Its objective is to obtain the highest sum of scores within a limited time, and this objective is more difficult to achieve than the basic objective. To catch all of the pigs is a precondition for getting a score on the level. In this situation, an agent having caught all pigs with a low score is not very good, but it is better than an agent obtaining a score of 0 by having one more pigs remaining (the latter would be wasting time if there is no mechanism to exploit the failure experience).

Usually, retaining a higher number of birds and breaking a higher number of objects is a good basic approach for getting a high score. However, in some cases, although the agent can clear the level with a lesser amount of birds, breaking as many objects as possible results in a higher score. In the competition, the winner will be decided by the highest sum of scores, so an agent with some lower scores on some levels can be the winner if the agent has the highest sum of scores. This case means different strategies are required to analyze each level.

5) HANDLING UNCERTAINTY

The visual analyzer of an agent can recognize a different result in the same screen shot. Sometimes, although an agent

TABLE 2. Summarized course schedule.

WEEK	Activity	Remarks
1 st	Lecture	Introduction
2–4 th	Student activity	Following step-by-step instructions to start the entry making and running the naive agent (sample controller)
5 th	Basic benchmarking	levels 1–21 in Poached Eggs
6 th	Student activity	-
7 th	1st round competition	5 levels in Poached Eggs and 5 unknown levels (from teaching assistant)
8 th	No class	-
9 th	Lecture	How to create levels for Angry Birds
10–12 th	Student activity	Review of last year competition entries' document Game contents creation (levels)
13 th	Basic benchmarking	levels 1–21 in Poached Eggs
14 th	Student activity	
15 th	2nd round competition	10 levels created by students
16 th	3rd round competition	10 levels of 2013 AI Birds qualification round

This table shows a summary of our course schedule. The 5th week's activity is a basic benchmarking, but the benchmarking was actually done in a day. We did not test with a mid-term because we wanted to make students focus on AI bot development. For this reason, most of the remaining time was used for student activity, agent development, in the course.

aims at a specific point, the shooting bird does not fly to that same aiming point, which makes it difficult to predict the result of an AI agent in a real competition game. Furthermore, competition participants do not know which kinds of levels or what difficulty of levels will be run in the competition because the competition does not announce game levels in qualifying or final rounds.

For these reasons, participants cannot be certain that the AI agent will perform as it did in their self-testing. Of course, in anticipation of these problems, an AI agent can be made prepared for uncertainty by trying to play many times and updating the agent with various game levels.

B. COURSE SYLLABUS

We taught this program design course (course name is ''Intensive Programming Design'') with 13 undergraduate students (computer engineering major) in computer labs. The course progressed for a total of 48 hours, twice per week for 16 weeks (Table 2). When beginning the course, we organized 13 students into 4 teams and instructed each team to make an AI Birds agent (programming in Java with API and samples provided from the competition website). We used 6 hours to lecture, 6 hours to benchmark AI, and had 3 rounds of internal competition (for evaluation). Each team analyzed and promoted its AI through benchmarks and internal competitions, and each obtained a final score of the agent's performance from the three internal competitions. After the course, we surveyed students using a questionnaire.

1) TEAM ASSIGNMENT

All 13 third-year undergraduate students, including 11 males, were assigned to one of 4 teams (only one team had 4 students). Every team member was chosen randomly because although a team made up of friends can sometimes produce a better result, we worried about friendship being an obstacle to improving individual learning ability and wanted to encourage cooperation among formerly unacquainted persons. These 4 teams were named Plan A+ (4 students), Angry Harvard, Simple Bird, and Angry Programmer.

2) EVALUATION

Each team's AI agent was ranked by the sum of scores obtained after 3 rounds of internal competition. Before the competitions, teams tested the performance of their agent with levels 1 to 21 of Poached Eggs (Poached Eggs is a basic episode provided by the Angry Birds game).

The first competition ran with 5 levels in Poached Eggs and 5 unknown customized levels. Students did not know the contents and order of the 10 levels until the competition. The second competition ran with 10 customized levels that were chosen randomly from among 13 customized levels (students created). We did not announce the content of the 13 levels but also did not deny discussion among team members. The final round had a closed run with 10 levels of the 2013 AI Birds qualification round (the levels are created by organizers and not open to the public). In the final round, all agents ran twice with the same 10 levels and attained higher scores between the former and the latter per each level.

3) QUESTIONNAIRE

After the course, we collected student's comments using a questionnaire. The main contents of the questionnaire related to evaluating the game AI and learning based program design course. The questionnaire had a total of 17 questions, including 8 five-point scale questions, one three-point scale question, and 6 free-writing questions. Through the questionnaire, we expected to learn how well students understood the course, students' opinions about game-based learning, and whether individual learning ability and game experience were of help to the course.

C. CUSTOMIZED LEVEL CREATION AND APPLYING FOR ANGRY BIRDS

Angry Birds, the original game of AI Birds, has many types of levels, but most of the levels are quite difficult not only for AI agents but also for human players because the levels are designed with the objective that more than half of the

levels should have difficulties challenging to human players, including high-level gamers. For this reason, all agents used just 1–1 to 1–21 levels of the basic Poached Eggs episode as benchmarking. In the course, 21 levels can be used to benchmark but not to test AI agents because using only a few known levels can lead to a poor quality agent (not generalize well on unknown levels).

For example, a poor-performing agent can also archive a high score with some 21 levels by luck, and a high-performing agent also can be specialized to get a high score with only 21 levels, similar to 'overfitting' in statistics and machine learning [27], [28]. In this case, a game-level editor can be one solution to making a better AI agent by creating various levels. Unfortunately, AI Birds (competition website and organizers) does not provide such. So, we created other kinds of levels for training and testing using a different method. We introduced to students our method for creating customized levels and applying them with the Chrome browser and a web-browser-based Angry Birds game.

1) CUSTOMIZED GAME-LEVEL CREATION METHOD

First, to create customized levels, it is important to know about the structure of Angry Birds levels. According to the result of our self-analysis, a game level created by JavaScript Object Notation (JSON) consists of various information (camera scale, position, sling position, birds, pigs, and so on) in a table-type structure. However, because directly writing these contents in plain text is an inefficient approach, we used another method to make the contents.

The approach that we used is to convert a customized Lua script [29] formatted level to a JSON formatted level after making the level using the Angry Birds Level Editor v0.1b website [30]. The web-based level editor can generate Lua script formatted customized level content. However, we made a conversion program for converting Lua format to JSON format because the contents of the Lua-format game level have different grammar, such as some object's names and measurements, than the JSON-format contents.

Our conversion program solved the problem using rescaled measurements with specific values and information based on self-analysis from the Lua format and JSON format game levels. Through this process, we created a conversion program that can convert an Angry Birds game level from a web-based level editor, and an Angry Birds game with a created level can be played using the program. However, the level editor has no option for controlling the shooting order of birds. Thus, the shooting order has to changed by directly editing a bird's part of the converted JSON file using text editor.

2) CUSTOMIZED GAME LEVEL APPLING METHOD

Basically, the web-based Angry Birds game created in JavaScript needs to receive game data from the server. Thus, the game provides a full download of game contents for a short loading time. However, the downloaded game contents based on a customized game level modification are too difficult because the downloaded content data were encrypted by some kind of security method. Hence, we used a different method for applying customized levels to a real game: the packet sniff technique with modified data (the packet sniff is a hacking technique). The method follows: 1. Set up an HTTP proxy server on a client PC for packet capture. 2. Remove cached data of the game content data from the Chrome browser. 3. Find the URL of the game data (found when connected to a game level). 4. Remember the URL and remove the cached data one more time to prevent the use of cached data. 5. When reconnecting some game level, send JSON-formatted customized level data instead of the original URL's data using a proxy server by packet changing. 6. Chrome browser will now read and run the customized level instead of the original data from the URL (Fig. 6).

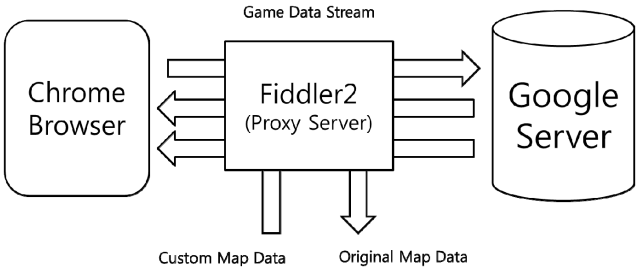

FIGURE 6. The Fiddler2, which is an HTTP-protocol-based proxy server program, can monitors and change internal packets in an installed computer, so one can change the original level's content from Google server to a customized level file using the program.

In the work, we used Fiddler2 [31] as a proxy server with Chrome browser's incognito mode, which makes it unnecessary to remove the cached data step.

IV. RESULTS AND STUDENT FEEDBACK

In this section, we show our results and analyses. During the course, each team's agent was developed and obtained a higher score than in its previous condition. We first describe each team's final agent and the agents' results in the three competitions (Fig. 7). Then, we analyze the final competition and questionnaire results. Before we analyzed the questionnaire results, we could not be certain whether the questionnaire results could be influenced by the final rank, so we checked the correlation between each team's rank and the questionnaire results. In this research, we used the Spearman rank correlation coefficient method instead of the Pearson method because of the small number of participants and the rank.

A. DESIGN PROGRESS OF AI AGENTS

All teams made their agent based on the provided Naive AI agent program (sample controller from the organizer). The following descriptions are a summary of each team's final agent.

- *Naive Agent (Sample Controller, Top 8 Players in 2013 Competition):* The agent, provided by the AI Birds basic program, uses random approaches to solve the problems.

FIGURE 7. The photo of classroom in the internal competition day.

It aims at one of the targets randomly and chooses shooting angles between the highest and lowest angle to the target. After aiming, it shoots a bird to the angle with a random tap. Although it's a sample controller, it was ranked as one of top 8 players in the 2013 competitions.

- *Plan A+:* The agent of team had the objective of clearing levels with a score of more than 80% of the level's theoretical highest score. If it obtains a lower score, it will retry. It used the provided basic API to recognize bird type. Although the API sometimes makes a poor result, the agent can precisely recognize the type of shooting bird using a zoom-in command and check the highest *Y*-position of birds in all detected birds. The agent calculates shooting trajectories based on shooting power and avoidance of crossing land. It also uses two different strategies according to the type of bird and object.
- *Angry Harvard:* The agent of team used an attack strategy using weak point analysis. Generally, most building block structures have physical weaknesses that differ with object type, so developing a weak point approach can yield a high score with a minimum shooting count for the agent (remaining birds provide additional scoring). The tap time of a shooting bird is decided based on flying time, except for the black bird and white bird. The agent also considers characteristics of objects, such as ice, wood, stone, and so forth. On the first time, it plays levels sequentially and after playing the final level replays lower-score levels.
- *Simple Birds:* The agent of team basically shoots with a low angle, except for the black bird and white bird. However, if land lies between the bird and target, it shoots with a high angle; and if an object lies between the bird and target, it shoots with a low angle. It also can recognize ground and objects, so when it locates a hard object on the shooting trajectory it can change trajectory. It sets tap time by considering the type of shooting bird, sling position, and obstacles.
- *Angry Programmer:* The agent of team used a different strategy. It focused on the special abilities of the types of birds. Its highest priority is to clear game levels, not to obtain the highest score in each level. The agent plays a not-played or failed level first, but when it gets 3 failures on a level, it passes the level. The agent used an approach of reinforcing Naive agent performance by reducing randomness and optimizing trajectories.

B. LEVELS USED FOR EVALUATION

In the first round, 5 customized levels were chosen (created from teaching assistant): the 1–7, 1–10, 1–14, 1–15, and 1–21 levels of the Poached Eggs episode. The levels of the Poached Eggs episode use only red, blue, and yellow birds, but other customized levels and the 2013 AI Birds competition additionally use black birds, white birds, and TNT objects (explosive objects). The most customized levels are not difficult but are cleared with well-used special abilities. In the first round, an agent should hit TNT in all levels that have TNT objects to obtain a high score.

The levels of the second round were a chosen 10 levels out of 13 customized levels made by 13 students. Most of the created levels have diverse features and difficulty without game balancing because students were not constrained in making a customized level. These levels have more difficulty than the first 21 levels of the Poached Eggs episode and required a high-level understanding to clear each game level. For example, in some levels, an agent can obtain a high score by shooting birds into a gap in ground objects rather than into a pig or object. Other levels require an agent, which has physics engine-based understanding in order to use unbreakable ground objects (Fig. 8).

Final round levels were 10 chosen levels from the qualification levels of the 2013 AI Birds competition (the levels were replicated from screenshots of the competition by authors). The snapshots have been provided only to the participants. The levels have balanced game difficulties, from easy to difficult, and most levels have a physics-engine-based solution with a high score. Thus, using these levels is good way to evaluate the performance of agents.

C. PERFORMANCE OF AI AGENTS

Table 3 shows the total scores for the three competition rounds in the course. The final score of each agent is recorded by AI performance in 10 levels of the Angry Birds game. Each level has different objects, highest score, and usable bird number and type, and the levels are more difficult nearer the final level.

According to Table 3, the agents had a different rank in every round, suggesting that the agent is specialized to the level. However, especially, the agent of team Plan A+ maintained a high score without jittering. If an agent can get a high score at any level, it would be concluded that the agent is generalized and has high performance AI (i.e., it is a well designed program).

FIGURE 8. Samples of custom levels created by students. (a) A basic-type level with TNT objects. (b) A level that needs to analyze the physical relationship between TNT and stones. (c), (d) Levels requiring an attack of a TNT object and avoidance of other objects.

We used 10 levels from the 2013 AI birds qualification round in the final round. These levels had a proper difficulty for testing AI performance; it was difficult to obtain a high score in each level. In the final round, we gave 30 min to each agent, unlike the real competition (in the 2013 IJCAI competition, 15 min for the ten levels). The results showed that the agent of team Plan A+ had the highest performance in the course, and if we ignore playing time (or make the agent faster), the agent could have obtained a rank of 5 in the qualification results of the 2013 AI Birds competition. The score of Plan A+ was 513,730, and the scores of the 1st and 4th bots were 584,600 and 532,310 in the qualification rounds of the AI Birds 2013 competition [32].

TABLE 3. Scores and levels cleared in the three competition rounds (bold means the best score in each row).

	Plan A+	Angry Harvard	Simple Bird	Angry Programmer
Bench mark	725,490 (19/21)	**836,000** (21/21)	428,400 (11/21)	753,110 (20/21)
1st Round	349,140 (6/10)	395,210 (7/10)	**648,290** (9/10)	198,220 (4/10)
2nd Round	511,260 (7/10)	**602,240** (8/10)	289,780 (4/10)	236,810 (4/10)
3rd Round	**513,730** (9/10)	380,410 (7/10)	377,790 (6/10)	253,460 (4/10)

These scores are the sum of all the level's scores in each level. The benchmark score is measured from Poached Eggs 21 levels. (2013 competition winner - Beau Rivage was 952,390 and naïve agent was 818,920 scores in the benchmark. Although the benchmark is an important indicator on the performance of agents, it is more important to show better performance on unseen maps) The first round score is evaluated with 5 levels in Poached Eggs and 5 customized levels, the second round with 10 levels created by students, and the final round with qualification levels in the 2013 AI Birds competition. The numbers in parentheses are a count of the levels cleared of the 10 total levels.

D. MAJOR QUESTIONNAIRE ANALYSIS

We now present the results of our questionnaire analysis from students in this course (the survey was done in the next week after the final round). Table 4 shows the questions in the survey. We first analyzed 10 scale-based questions before considering some other extra questions. We also analyzed the relationship between the final-score-based team rank and questionnaire results before performing the numerical questionnaire analysis. According to the result, 12 out of 13 students participated in this survey. The exception was one Angry Programmer team member.

Only two students among the participants (both male) had no Angry Birds game experience. All questions were related to course evaluation subjects. For example, Question 2 was used to discover whether game-playing experience could lead to a higher performance AI agent in the game. However, there was no significant relationship in this study because only two of the participants were inexperienced. Questions 6 and 8 were used to discover whether a game-based program design lecture is helpful for other programming lectures.

For a more complete analysis of the relationships, we used Spearman's rank correlation coefficient [33], [34]. We used average value and rank in the analysis based on each team's final score. The results (Table 5) showed that Q-4 and Q-9 had a strong positive correlation between answer and rank and that Q-7 had a weak positive correlation (a lower value is

TABLE 4. Contents of questionnaire.

No.	Question	Selection
	Major Question	
Q-1	What is your gender?	Male / Female
Q-2	Had you ever played the Angry Birds game before this course?	Yes / No
Q-3	Did you find the Angry Birds game based course interesting?	5 / 4 / 3 / 2 / 1
Q-4	Did you think the final agent had a higher performance than the earlier one?	5 / 4 / 3 / 2 / 1
Q-5	Did you satisfactorily meet the content of the game-based course?	5 / 4 / 3 / 2 / 1
Q-6	Did the course help to your program design study?	5 / 4 / 3 / 2 / 1
Q-7	If you have had at least one programming design lecture, how do rate this lecture compared to others? (If you have not had any lectures, please skip this question.)	5 / 4 / 3 / 2 / 1
Q-8	Do you think to the learning contents of this course are useful for other programming lectures?	5 / 4 / 3 / 2 / 1
Q-9	Did you produce a satisfactory result in the final round?	5 / 4 / 3 / 2 / 1
Q-10	If you did not produce a satisfying result, do you think you could improve your AI agent with more time? (Only those who answered 1~3 on Q-9, should answer this question.)	5 / 4 / 3 / 2 / 1
	Extra questions	
Q-11	Which round is most interesting: the first, middle, or final round?	
Q-12	What are three important issues for making an AI agent (e.g., visual recognition, attack strategy, time management, etc.)?	
Q-13	What are three difficult hurdles in the progress of this lecture (e.g., coding, debugging, ideas, AI design, etc)?	
Q-14	Do you think your team communicated well with each other?	
Q-15	What is the most satisfying part of this lecture?	
Q-16	What is the most unsatisfying part of this lecture?	
Q-17	Extra comment	

This table shows 11 major scale-based questions from 17 total questions. The number scores mean 5-strongly agree, 4-agree, 3-neither agree nor disagree, 2-disagree, or 1-strongly disagree. We also used the remaining 6 minor questions in our analysis. These 6 questions are free-writing questions and are not directly related to the evaluation of their AI agent.

better in rank and a higher value is better in answer). In other words, if the agent of a team obtained high performance, team members thought that their agent improved more and they felt more satisfaction in the final round. However, the response of Q-7 shows a weak relationship with team rank.

Additionally, we analyzed the results based on individual results rather than team results using the Spearman method. All participated students received a grade from their teams in this team-based course. In this case, we wanted to know if team's correlation value is representing their member's values.

The individual-based result shows that Q-4 and Q-9 have high correlation with team rank, like the team result, but that all remaining questions have only low correlation. This analysis means that game-based learning has similar or better study efficiency irrespective of team member rank than other learning approaches. However, the low average of Q-8 shows that this course will not be especially helpful to the student in the other programming course. In a similar manner, we also analyzed the following.

TABLE 5. Scaled questions-based analysis.

No.	Question's score	Team based p	Individual based ρ
Q3	4.58 ±0.67	0.4	0.30227
Q4	4.5 ±0.67	1	0.76125
Q5	4.58 ±0.51	0.8	0.27873
Q6	4.08 ±0.90	0.4	0.10794
Q7	4 ±0.74	0.210879	0.22967
Q8	3.75 ±0.97	0.4	0.12666
Q9	3.75 ±1.36	1	0.92051
Q10	4 ±1.10	-	-

This table shows a summary of the statistical analysis for questions 3 to 10. The values in the 3rd and 4th columns indicate Spearman rank correlation coefficient based on the score of the final round. The team-based p (3rd value) is the analyzed correlation between each team's rank and average score, and the individual-based p is analyzed with the team's rank and individual scale score.

According to the result, all participants answered with high positive responses for the Angry Birds game-based program design course and were satisfied the contents of the course (Q-3, Q-5). They also answered with a positive response for this course compared to others, but it was not as high as the previous (Q-7).

Next, we analyzed the learning effect of the course. The students answered that they strongly agreed that the final agent had a higher performance than its earlier version (Q-4). They also answered that they agreed that this course helped their program design study, but not significantly (Q-6). In other words, the students were satisfied and rated their AI agent as having a higher final result (most students agreed to improve their final agents, irrespective of the final result). The results also showed that they thought that the game-based learning had a significant interest factor and helped in their study of program design, but they did not feel that the benefit would extend to other programming courses. However, four students who had answered with unsatisfying final results in Q-9 (5 students with 1–3 points) responded that they could make a higher scoring agent if they had more attempts. This indicates that the course generally had a positive impact on developing and challenging the spirit of students.

E. EXTRA QUESTIONNAIRE ANALYSIS

We analyzed potential issues and problems of this course with an extra questionnaire (Q-11–Q-17). In Q-11, 7 students answered with final round, 5 students answered with middle round, but no student answered with first round, probably because of all students felt that their early agent had poor performance. This question also seems to have a correlation with the answer of Q-4. In Q-12, 10 students answered with attack strategy, including target selection, 6 students answered with visual recognition, and 4 students answered with time management.

One curious point is the inclusion of visual recognition in the answers because an agent of the AI Birds competition has a provided basic API, and competition participants can focus on creating agent behavior, such as attack and time management strategies. However, despite providing the visual recognition API, the result indicates that all students

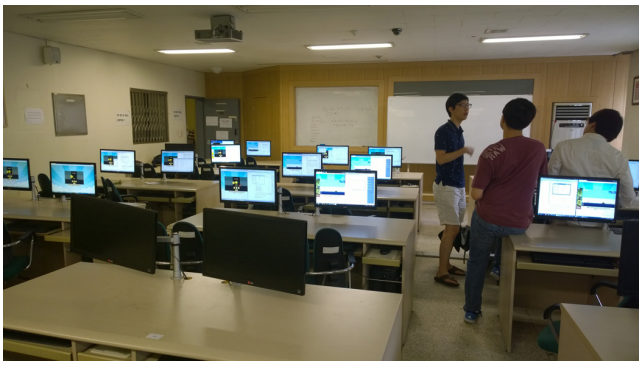

FIGURE 9. Parallelization of Angry Birds AI controller's evaluation on multiple machines for optimization (each machine evaluates different levels and the final results are merged into a master computer).

could not use their energies efficiently to make an agent because of a imperfect API (most game AI agents can access original raw data from the game system, but visual-recognition-based games such as the AI Birds competition have such problems). AI Birds organizers are also aware of the problem of the imperfect visual recognition API, so they are providing an improved basic AI Birds agent (Ver. 1.3) in the 2014 competition, which includes ground recognition [35].

In Q-13, 10 students answered with idea and design for making an AI agent and 6 students answered debugging and exception handling. We did not worry about the former issue because the issue is a major subject of our program design course. However, one major opinion of the latter issue is a big problem: students had to use a large amount of time for agent testing. Unfortunately, the latter problem cannot be avoided in the game-based course using game program that has no game simulation. For example, the evaluation of the 21 levels from the Poached Eggs requires about 1 hour (3 min per each level). It means that simple parameter change can take one hour to see the result. The solution of the problem is to use parallel testing (each machine evaluates one single level) by distributing AI program to multiple machines with different levels. If we use 21 machines, it can return the total scores within 3 min. It can allow students to use advanced optimization techniques (for example, genetic algorithm).

Recently, the Plan A+ team (with authors) implemented the parallelized version of Angry Birds competition platform (Fig. 9). It distributes Angry Birds levels and a set of parameters into multiple machines. Using the system, they are able to optimize some parameters of their rule-based systems with advanced techniques. They're now using the parallel platform for optimization of their 2014 competition entry.

Nevertheless, students answered positively to the other questions about whether they were interested in game-based learning. They had fun progressing through the course, despite AI development being a difficult subject, and they wanted us to continue to run a course such as this one. On the other hand, they answered negatively about having only a short amount of time compared to the difficulty of the objective. They complained that they could not form teams with their friends because of the random assignment of team members, and so forth. Fortunately, most team members answered that they had good communication with each other in the end.

V. CONCLUSIONS

Program design is an important skill to become a developer, but learning program design is difficult for undergraduate students because it is very complex and can be rather boring. We thought that game-based learning would provide motivation for these students as well as a positive learning experience. Thus, we tried to apply the approach in our teaching of a program design course by having students create an AI agent based on the AI Birds competition. The use of open game AI competition platform can be beneficial for educators. It can significantly reduce the time and resource required to define problems, development of games, and benchmarking sets.

In this work, we propose to use Angry Birds AI competition platform for intensive programming course. It includes strategic decision-making, time management, vision, and uncertainty handling problems. During the course, students experienced the development of entries for the competition and the creation of game contents. The benchmarking results from the 2013 Angry Birds AI competition have been quite useful to indicate the current progress of students' works. They realize that the gap between international competition entries and their works is not too big in the final round. It promotes students to prepare the submission of entries for the international game AI Competitions in 2014 (at this moment, the Plan A+ team prepares the submission for the 2014 competition.)

Consequently, the results showed that the approach could help students with their learning, maintain motivation for the course, and encourage student team work and lift their spirits. All students had a positive response to the teaching approach used in the course. These analysis results encourage our belief that game-based learning is a good educational approach for students. If we continue to improve the approach with the assistance of the analyzed student feedback, we can teach certain courses in a way that is different from the typical boring or difficult content, which will make student's learning more interesting and motivating.

ACKNOWLEDGMENTS

The authors would like to express thanks to the 2013 Angry Birds AI Competition organizers (Jochen Renz, XiaoYu (Gary) Ge, and Stephen Gould). Kyoung-Jun Ahn, Hyun-Jin Kim and Dong-Hyun Jang successfully supported our class as teaching assistants.

REFERENCES

[1] *Angry Birds*. [Online]. Available: http://www.angrybirds.com/, accessed Nov. 17, 2014.

[2] K.-J. Kim and S.-B. Cho, "Game AI competitions: An open platform for computational intelligence education," *IEEE Comput. Intell. Mag.*, vol. 8, no. 3, pp. 64–68, Aug. 2013.

[3] P. Hingston, "A Turing test for computer game bots," *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 3, pp. 169–186, Sep. 2009.
[4] D. Loiacono *et al.*, "The 2009 simulated car racing championship," *IEEE Trans. Comput. Intell. AI Games*, vol. 2, no. 2, pp. 131–147, Jun. 2009.
[5] J. Togelius, N. Shaker, S. Karakovskiy, and G. N. Yannakakis, "The Mario AI championship 2009–2012," *AI Mag.*, vol. 34, no. 3, pp. 89–92, 2013.
[6] S. Ontanon, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game AI research and competition in StarCraft," *IEEE Trans. Comput. Intell. AI Games*, vol. 5, no. 4, pp. 293–311, Dec. 2013.
[7] K.-J. Kim and S.-B. Cho, "Experience on running a small-size simulated car racing tournament in an introductory programming course," in *Proc. Int. Conf. Converg. Hybrid Inf. Technol.*, 2010, pp. 200–206.
[8] *Angry Birds AI Competition*. [Online]. Available: http://aibirds.org, accessed Nov. 22, 2014.
[9] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Trans. Affective Comput.*, vol. 2, no. 3, pp. 147–161, Jul./Sep. 2011.
[10] P. L. Schwartz, *Problem-Based Learning*. Evanston, IL, USA: Routledge, 2013.
[11] M. Prensky, *Digital Game-Based Learning*. Saint Paul, MN, USA: Paragon House, 2007.
[12] M. Resnick, "Edutainment? No thanks. I prefer playful learning," *Associazione Civita Rep. Edutainment*, 2004. [Online]. Available: http://web.media.mit.edu/~mres/papers/edutainment.pdf
[13] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper Perennial, 2008.
[14] M. Griffiths, "The educational benefits of videogames," *Edu. Health*, vol. 20, no. 3, pp. 47–51, 2002.
[15] L. A. Annetta, "Video games in education: Why they should be used and how they are being used," *Theory Pract.*, vol. 47, no. 3, pp. 229–239, Jul. 2008.
[16] *Game AI Competition Portal*. [Online]. Available: http://cilab.sejong.ac.kr/gc, accessed Feb. 3, 2014.
[17] M. Overmars, "Teaching computer science through game design," *Computer*, vol. 37, no. 4, pp. 81–83, Apr. 2004.
[18] W.-K. Chen and Y. C. Cheng, "Teaching object-oriented programming laboratory with computer game programming," *IEEE Trans. Educ.*, vol. 50, no. 3, pp. 197–203, Aug. 2007.
[19] S. Salcedo-Sanz, J. A. Portilla-Figueras, E. G. Ortiz-Garcia, A. M. Perez-Bellido, and X. Yao, "Teaching advanced features of evolutionary algorithms using Japanese puzzles," *IEEE Trans. Educ.*, vol. 50, no. 2, pp. 151–156, May 2007.
[20] H. C. Jiau, J. C. Chen, and K.-F. Ssu, "Enhancing self-motivation in learning programming using game-based simulation and metrics," *IEEE Trans. Educ.*, vol. 52, no. 4, pp. 555–562, Nov. 2009.
[21] B.-S. Jong, C.-H. Lai, Y.-T. Hsia, T.-W. Lin, and C.-Y. Lu, "Using game-based cooperative learning to improve learning motivation: A study of online game use in an operating systems course," *IEEE Trans. Educ.*, vol. 56, no. 2, pp. 183–190, May 2013.
[22] I. Paliokas, C. Arapidis, and M. Mpimpitsos, "PlayLOGO 3D: A 3D interactive video game for early programming education: Let LOGO be a game," in *Proc. 3rd Int. Conf. Games Virtual Worlds Serious Appl. (VS-GAMES)*, May 2011, pp. 24–31.
[23] M. Chang and Kinshuk, "Web-based multiplayer online role playing game (MORPG) for assessing students' Java programming knowledge and skills," in *Proc. 3rd IEEE Int. Conf. Digit. Game Intell. Toy Enhanced Learn.*, Washington, DC, USA, Apr. 2010, pp. 103–107.
[24] J. D. Bayliss, "Teaching game AI through Minecraft mods," in *Proc. IEEE Int. Games Innov. Conf. (IGIC)*, Sep. 2012, pp. 1–4.
[25] J. DeNero and D. Klein, "Teaching introductory artificial intelligence with Pac-Man," in *Proc. Symp. Edu. Adv. Artif. Intell.*, 2010, pp. 1–5.
[26] M. E. Taylor, "Teaching reinforcement learning with Mario: An argument and case study," in *Proc. 2nd Symp. Edu. Adv. Artif. Intell.*, 2011, pp. 1737–1742.
[27] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.
[28] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surv.*, vol. 27, no. 3, pp. 326–327, Sep. 1995.
[29] *Lua Script*. [Online]. Available: http://www.lua.org, accessed Aug. 27, 2014.
[30] *Angry Birds Level Editor v0.1b. (It Revived and Tweaked by Apache Thunder. Original Version Created by Forcer)*. [Online]. Available: http://www.battlefieldsingleplayer.com/apachethunder/angrybirds/, accessed Jul. 3, 2014.
[31] *Fiddler2*. [Online]. Available: http://www.telerik.com/fiddler, accessed Mar. 10, 2014.
[32] *The Score of Qualification Rounds in AI Birds 2013 Competition*. [Online]. Available: https://aibirds.org/past-competitions/2013-competition/results.html, accessed Nov. 12, 2013.
[33] J. H. Zar, "Spearman rank correlation," in *Encyclopedia of Biostatistics*. New York, NY, USA: Wiley, 2005.
[34] E. S. Pearson and B. A. S. Snow, "Tests for rank correlation coefficients III. Distribution of the transformed Kendall coefficient," *Biometrika*, vol. 49, nos. 1–2, pp. 185–191, Jun. 1962.
[35] *AI Birds Agent Basic Program Version 1.3*. [Online]. Available: https://aibirds.org/Software/AngryBirds/abV1.3.zip, accessed Jan. 2, 2014.

DU-MIM YOON received the B.S. degree in digital content and the M.S. degree in computer engineering from Sejong University, in 2010 and 2012, respectively, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering. He has been involved in several media art projects and education based on Arduino and 3-D printing. His research interests include game, robotics, and physical computing. He was a finalist in the Angry Birds AI Competition with the Hungry Bot Entry in 2013.

KYUNG-JOONG KIM received the B.S., M.S., and Ph.D. degrees in computer science from Yonsei University, in 2000, 2002, and 2007, respectively. He was a Post-Doctoral Researcher with the Department of Mechanical and Aerospace Engineering, Cornell University, in 2007. He is currently an Associate Professor with the Department of Computer Science and Engineering, Sejong University. His research interests include artificial intelligence, game, and robotics.