Development and Evaluation of a Game-Based Bayesian Intelligent Tutoring System for Teaching Programming

Danial Hooshyar¹, Rodina Binti Ahmad², Minhong Wang³, Moslem Yousefi⁴, Moein Fathi², and Heuiseok Lim¹

Abstract
Games with educational purposes usually follow a computer-assisted instruction concept that is predefined and rigid, offering no adaptability to each student. To overcome such problem, some ideas from Intelligent Tutoring Systems have been used in educational games such as teaching introductory programming. The objective of this study was to advance Online Game-based Bayesian Intelligent Tutoring System (OGITS) to enhance programming acquisition and online information searching skills, thus improving students’ ability in web-based problem solving through board games. The study sample comprised 79 college students in introductory level Computer Science classes. Qualitative and quantitative data were then gathered. Results of this study revealed generally favorable opinions about OGITS. As OGITS targets individual knowledge acquisition of computer programming and web-based

¹Department of Computer Science and Engineering, Korea University, Seoul, The Republic of Korea
²Department of Software Engineering, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
³Faculty of Education, The University of Hong Kong, Hong Kong
⁴School of Civil, Environmental and Architectural Engineering, Korea University, Seoul, The Republic of Korea

Corresponding Author:
Heuiseok Lim, Department of Computer Science and Engineering, Korea University, Seoul, The Republic of Korea.
Email: limhseok@korea.ac.kr
problem-solving skills, it offers a suitable learning environment for students both as a stand-alone course and as a supplement to traditional classroom settings.

**Keywords**
game-based environment, intelligent tutoring system, Bayesian network, online information searching, computer programming

**Introduction**

The importance of cultivating students’ problem-solving skills has been stressed by many researchers (Caviglia & Delfino, 2015; Hooshyar, Ahmad, Yousefi, Yusop, & Horng, 2015; Hwang & Kuo, 2011; Yu, She, & Lee, 2010). In recent decades, a number of approaches and methods have been suggested to understand students’ behavior concerning web-based problem solving. For example, Meta-Analyzer, a learning environment built by Hwang, Tsai, Tsai, and Tseng (2008), can be used to help educators evaluate student behaviors concerning online problem solving. A subsequent study by Chen (2010) has used online problem-solving tasks to analyze student learning behavior in a pair of music appreciation courses and found that such tasks can enhance both learning and advanced cognitive skills of students. Other investigations have addressed difficulties related to finding information online. One study has researched challenges that new Internet users face in finding and using information (Tsai, 2009). However, other studies have shown that learners frequently lose time or morale in attempting to find solutions to complicated problems without online support or guidance (Hargittai, 2006; Li & Kirkup, 2007). Therefore, it is not surprising that many researchers have found that providing students with guidance can enhance their performance in web-based problem-solving tasks (Hwang & Kuo, 2011; Kauffman, Ge, Xie, & Chen, 2008; Zamani & Shoghlabad, 2010).

The effectiveness of online educational games has drawn great attention in recent years. Multiplayer online games particularly can provide multiple benefits when they are utilized in education (Sourmelis, Ionnou, & Zaphiris, 2017; Yang & Huang, 2013). For example, massive multiplayer online role-playing games continue to be popular as they can reinforce motivation and creativity while allowing skill development such as problem solving (Voulgari, Komis, & Sampson, 2015). Since games with educational purposes usually follow computer-assisted instruction concepts that are predefined and rigid, they are not equally suitable to each student (Mills & Dalgarno, 2007). Some ideas in Intelligent Tutoring Systems (ITS) have been borrowed and applied in educational games for teaching introductory programming (Zapusek & Rugelj, 2013).
This has two benefits. First, it addresses the problem of ITS which is losing learning motivation and enthusiasm when learners do not receive enough timely guidance or interaction. Second, it addresses the problem of educational games by giving students a learning environment adapted to their personal needs. To this end, we developed an Online Game-based Bayesian Intelligent Tutoring System (OGITS) in the form of a competitive board game by revising the classic table game “Snakes and Ladders” to enhance students’ learning in computer programming and online information searching. Monitoring student’s progress and modeling prerequisite information are required to customize guidance and learning materials to each student. Decisions have to be made based on uncertain or incomplete information. Therefore, a Bayesian network was employed for the decision-making process in the OGITS system. This study had the following research question:

How do students evaluate the OGITS? What are their perception, attitudes, and views about learning computer programming and online information searching while using the system?

**Literature Review**

**Web-Based Problem Solving**

The ability to solve problems is the foremost among various skills required to adapt to a living environment. Such ability comprises six skills: identifying the essence of the problem, establishing steps to solve it, establishing techniques to solve it, selecting pertinent information, assigning relevant resources, and overseeing the problem-solving procedure (Kuo, Hwang, & Lee, 2012; Lin, 2017). Because computing is so popular, many studies have indicated that a web-based learning activity (or web-based problem solving) can greatly enhance students’ ability of problem solving (Hwang & Kuo, 2011). Generally, students must define keywords in order to answer questions about a given problem. From these keywords, potential web resources are identified, suitable webpages are discovered, and pertinent information is extracted and summarized. Although the usefulness of this web-based problem solving has been demonstrated in recent experimental studies (Kuo et al., 2012; Rae & Samuels, 2011), very few systems have been developed to improve students’ learning performance in online information searching skills, which in turn can improve students’ ability in web-based problem solving and programming courses. Although online information searching is a prominent approach to improve students’ problem-solving ability, it could backfire in the form of complicated problems. For example, students are prone to be lost and confused as a result of not having proper
guidance (Ferreira & dos Santos, 2009; Hargittai, 2006). Therefore, some form of learning guidance is essential.

**Educational Computer Games**

Because technologies of computing and communication have grown so rapidly in sophistication and use, researchers have long expected that learning will become increasingly technologized in centrally featured educational computer games (Hooshyar, Yousefi, & Lim, 2017; Johnson et al., 2014; Prensky, 2005). To examine the efficiency of computer games with educational purposes for different study fields, many research studies have been carried out in the past decades. The potential of applying such games to the learning process of students to improve their learning performance has been indicated by numerous researchers (Hooshyar et al., 2016; Hwang & Kuo, 2011; Malliarakis, Satratzemi, & Xinogalos, 2017). A number of researchers have determined what learning improvements that educational computer games might offer students (Peterson, 2016; Wang & Chen, 2010). Many studies have suggested that educational computer games can inspire students to have increased interest and motivation in learning (Beier, Miller, & Wang, 2012; Ventura, Shute, & Zhao, 2013). Particularly, multiplayer online games can provide multiple benefits when they are utilized in education (De Lope & Medina, 2016; Sourmelis et al., 2017; Yang & Huang, 2013). For example, massive multiplayer online role-playing games continue to be popular as they reinforce motivation and creativity while allowing skill development such as problem solving and communication (Voulgari et al., 2015). More recently, educational games for programming (Malliarakis et al., 2017) and distributed pair programming systems (Tsompanoudi, Satratzemi, & Xinogalos, 2016) have been developed to help students deal with difficulties and improve their performance. Educational games succeed mainly by motivating students through their immersive qualities while distributed pair programming systems succeed by supporting collaboration between students. As shown in the literature review, online educational games are subjects of primary interest to researchers in this field. However, advancing the efficacy of such games remains a significant challenge.

**OGITS: An Online Game-Based Bayesian Intelligent Tutoring System**

OGITS was developed with the aim to improve novice programmers’ skills in learning computer programming and online information searching. The architecture of OGITS is shown in Figure 1. This multiplayer competition board game (Salen & Zimmerman, 2003) comprises board game interfaces such as Snake & Ladder (see section 3.2) and Tic-tac-toe (see section 3.3) games to
enhance programming acquisition and online information searching skills, respectively. Bayesian interface assessment, adaptive guidance, and knowledge bases will be thoroughly explained in the following sections.

After answering predesigned questions, the Bayesian network algorithm (see section 3.1) is updated and learners are directed to the entry page of OGITS along with its navigational menu that describes study goals (Figure 2). Learners have the choice of starting a game of Snakes and Ladders (refer to the following subsection) or reviewing study materials by clicking on the light icon. There are 29 nodes in the system. Each node stands for a programming concept. All nodes are connected to each game board square. The navigational menu presents the programming concepts in different colors. Each color corresponds to the user’s mastery of the given concept. For example, a red light signals that the concept is “unknown and not yet available to the user.” A green light indicates “unknown concept that the user is ready to learn,” while a yellow light indicates that the concept is already known to the user. When a user clicks on the colored icon, they gain access to learning material that corresponds to the concept. They are then given a Tic-tac-toe quiz for a single player (TRIS-Q-SP) game button (see section 3.3). Students can either play TRIS-Q-SP dependently to update the Bayesian network, navigational menu, and their level of knowledge or be redirected by the Snakes and Ladders game to play TRIS-Q-SP as a penalty.
Bayesian Networks in OGITS

Guiding a student successfully through these concepts requires two things. First, it is necessary to model the prerequisite information. Second, it is required to monitor the student’s progress on each concept. Obviously, this involves creating systems that can make decisions based on uncertain or incomplete information. One formal framework for uncertainty management that can facilitate the aforementioned two tasks is Bayesian networks (Wong & Butz, 2001). It utilizes probability theory as a formal framework for uncertainty management.

A Bayesian network comprises directed acyclic graph (DAG) and relevant conditional probability distributions (CPDs). Given probabilistic conditional independencies encoded in DAG, CPD can produce a joint probability distribution (JPD). Bayesian networks thus function as economical configurations of a JPD and tools for semantic modeling. To calculate probabilities of variables (given that other variables had specific values), several inference algorithms are proposed for Bayesian networks. Using this method, a set of initial computer programming concepts are pinpointed, with a node in the graph corresponding to each concept. When knowledge of one concept is a precondition for grasping another, a directed edge is added to connect these concepts (nodes). In this manner, a DAG is manually assembled with the help from a programming textbook. For example, the “If-Statement” concept presupposes understanding concepts of “assignment” and “logical operators” (Figure 3). A slight portion of the DAG that OGITS uses is illustrated in Figure 3, and OGITS’ programming concepts in their entirety are shown in Figure 4.
In building a Bayesian network, one must subsequently establish a CPD for each node on the basis of its parents. For variable $a_i$ with parent set $P_i$, a CPD $p(a_i|P_i)$ has the property that for each configuration (instantiation) of variables in $P_i$, the sum of probabilities of $a_i$ is 1.0. The parent set of “If-Statement” node is demonstrated in Figure 3, while the corresponding CPD $p($If_Statement$|Assignment$, Logical_Operators$)$ is shown in Table 1.

We should note that results of student pretests can provide all CPDs for DAG. For each question, the concept being measured was initially identified. The concept was labeled as known or unknown depending on whether or not the student accurately answered the corresponding question. At this point, one
could calculate the probability of whether a given concept was known (e.g., \( p(a_i = \text{known}) \)). Furthermore, one could calculate the probability of the student accurately answering both concept \( a_i \) and its prerequisites \( P_i \) in the equation \( p(a_i = \text{known}, P_i = \text{known}) \). Thus, every CPD in the Bayesian network could be calculated once one has ascertained the desired CPD \( p(a_i = \text{known} | P_i = \text{known}) \) from \( p(a_i = \text{known}, P_i = \text{known}) \).

### Snakes and Ladders Game

Snakes and Ladders game is considered a classic board game. Its major attraction is that some locations on the board display snakes and ladders. A dice is required to take turns and play the game. According to total cells on the board, the board is numbered from 1 to 36 in OGITS. In our proposed system, each square on the game board is connected to all 29 nodes or programming concepts (Figure 3). Once students land on a particular square, the corresponding task (either a mini game or a multiple-choice question) is triggered. Design purposes of the mini-game and multiple-choice questions are to provide supplementary materials and carry out online information searching tasks, respectively. A student’s game score will increase when they answer a question correctly. They are allowed to throw the dice and move forward in the game based on the outcome. Accumulated scores of all students on the board are shown during the gaming process as shown in Figure 2 to inform peers of each student’s gaming status. In case of individual student, if the time is up or the learning task is completed, the game ends. The first game winner is the one who reaches number 36 on the board before others. The next winner is the one who reaches it next.

The system will guide students to search for relevant information on the web if they fail to answer a question correctly on a square without a snake or ladder as illustrated in Figure 5. If the student lands on a snake square, he or she is required to respond to the question before moving down the snake’s tail. In case the student answers the question wrong, as a penalty, he or she will move down the snake’s tail and automatically get redirected to the corresponding concept in

### Table 1. CPD Corresponding to the “If-Statement” Node in Figure 3.

<table>
<thead>
<tr>
<th>Parent nodes</th>
<th>If-Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>Logical_Operators</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Known</td>
<td>Known</td>
</tr>
<tr>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td>Known</td>
</tr>
<tr>
<td></td>
<td>Unknown</td>
</tr>
</tbody>
</table>

Note. CPD = conditional probability distribution.
the navigational menu based on the question asked as each square in the game is connected to all 29 nodes. In addition, in case of landing on a ladder square, the player cannot move up the ladder if they respond to the question wrong. They will also be redirected to the corresponding concept on the navigational menu. Thus, the Bayesian network decides exactly which unknown concept students should be referred to. By doing so, the student is directed to one of these concepts with a green light as yellow light indicates concepts already “known” in the Bayesian network while red light concepts are labeled as “unknown and not ready to learn.” Therefore, their prerequisite concepts must be learned first (Figure 6). After going through learning materials, the student will be asked to play TRIS-Q-SP so that he or she can return to the main Snakes and Ladders game with his or her opponents (Figure 7). It is worth noting that the Bayesian network automatically estimates these lights based on pretest questions and game playing.

**Tic-Tac-Toe Game**

Figure 7 illustrates Tic-tac-toe quiz for single player (TRIS-Q-SP), an online formative assessment mini-game. Once a user begins a new game, the online mini-game starts with Step 1 (Figure 7). Rules for TRIS-Q-SP are the same as those for standard Tic-tac-toe. A player wins by being the first to place three pieces side-by-side in a column, row, or diagonal. However, TRIS-Q-SP departs from conventions of Tic-tac-toe in several respects. For instance, after a turn, a user is given a multiple-choice test selected at random from the database of...
Figure 6. Learning materials on the “assignment” concept and the game button.

Figure 7. The three steps of playing TRIS-Q-SP.
learning materials (Figure 7, Step 2). Here is where the rules of the game are altered. For the sake of having clear stakes and game play, the player must correctly answer the question in order to place their piece on the grid (Figure 7, Step 3-2). An incorrect answer will result in his or her opponent’s piece being placed instead (Figure 7, Step 3-1). This alteration to the rules of Tic-tac-toe serves to incentivize the player to put careful consideration into question answering.

Playing TRIS-Q-SP is one way of providing the system with feedback and accordingly updating the Bayesian network. Figure 6 illustrates the play button and learning materials of the “Assignment” concept. When a player is victorious and the Bayesian network is updated, the navigational menu then shows the given concept as “known.” However, the concept remains shown as “unknown” if the player loses. The player is encouraged to review relevant conceptual material. He or she will be able to view the right response to the quiz. The Bayesian network is also updated.

Research Method

This study aims to develop and evaluate game-based Bayesian ITS named OGITS. After first designing and developing OGITS, it was then evaluated by its implementation in real classroom and results were then analyzed. Because random sampling posed difficulties, convenience sampling method (Fraenkel, Wallen, & Hyun, 2006) was employed in this study. In other words, users were chosen based on their availability.

The experiment comprised a pair of introductory Computer Science (CS) classes. A total of 79 first-year undergraduates participated (31 females and 48 males) in this study. None of them had any previous familiarity with computer programming. The experiment took place in the fall semester of 2014–2015 in three classrooms in two universities in Malaysia. Quantitative and qualitative data were collected from these students to address research questions of this study.

Procedure

First, Computer Science course content was developed and integrated into OGITS. This was made of Variables, Assignments, and Control Structures. OGITS was also presented to students prior to the experiment. At one university, two distinct classes and two tutors were selected. In the other university, only a single class and one tutor were selected. These students then received lectures regarding all three programming concepts via OGITS at a computer lab while they were watched by three tutors and two researchers. The study lasted for 6 weeks. Students worked in the computer lab for 4 hr/week (24 hr total). OGITS in its current construction presents one new C++ programming
concept every 2 weeks for a total of three new C++ programming concepts in 6 weeks.

**Instruments**

To evaluate learning improvement in both computer programming and online information searching skills, student scale was used as a quantitative data collection tool. A Likert-type scale consisting of 15 items was used. It was developed with reference to the literature on scales employed for evaluating adaptive and intelligent online educational environments (Brown, 2007; Mustafa & Sharif, 2011). The research also employed qualitative data collection methods. These students were given an interview guide to present various aspects and results of the research process.

**Development of Scales**

During the development process of scales, development phases generally took place in experimental or theoretical processes. In the case of this study, the sample size of 79 students did not allow for development via experimentation. Thus, theoretical process was applied to the scale’s development instead. This entailed seeking expert opinions following operational steps suggested by Yurdugül (2005). In the scale’s initial development, researchers filled its form. A Likert scale offering five choices was developed with 15 items. These choices were: Strongly Disagree (1), Disagree (2), Undecided (3), Agree (4), and Strongly Agree (5). Following the theoretical development, the validity of the content had to be ascertained (Cakiroglu, Guven, & Akkan, 2008; Yurdugül, 2005). Subsequent operations were performed based on the theoretical scale form.

- First, a field expert group was assembled. It consisted of four professors in Software Engineering and six postgraduate students from the same department whose area of study pertained to computer and online education.
- Based on literature review and expert advice, 15-item candidate scale forms were developed. Experts were called upon to present items in such a way to make them easily grasped. Revisions were included in final revisions of the scale items. Likewise, experts were consulted on whether items in the scale were adequately aligned with factors to be measured. This resulted in the inclusion of all 15 items. Experts graded items in terms of Content Validity Rates (CVR) of scale items and Content Validity Index (CVI) they produced. Evaluation results included “Necessary,” “Beneficial but Inadequate,” and “Unnecessary.”
- Once the expert group had filed their forms, these were gathered and points were calculated for each item.
Here is the CVR value for all scale items:

\[ CVR = \frac{N_N}{N/2} - 1 \]  

where \( N \) is the total number of experts assessing the scale and \( N_N \) is the number of experts labeling an item “necessary.” Given 10 experts, a minimum CVR value was set to 0.51 with significance level of \( \alpha = .05 \) (Veneziano, 1997). Because all 15 items of the scale received CVR values in excess of 0.51, none was excluded.

Following CVR value assessment, the arithmetical mean for each item’s CVR was calculated. The CVI value of the scale was found to be 0.75. As this CVI value exceeded the value of 0.51 set for 10 experts, the scale’s content validity was statistically significant. Hence, the scale was finalized in this manner.

This scale of 15 items pertaining to the study’s content was given to the 79 students of this study. Reliability analysis was then performed. Results of this analysis showed that one of these items reduced the scale’s Cronbach’s \( \alpha \) value. Therefore, the scale was employed with all 15 items.

**Interview Guides**

Students’ opinions concerning OGITS were solicited in details to gain insight into our research questions. Hence, semistructured interview guides comprising the following eight questions were used:

1. To what extent do you think using OGITS according to your learning style affected your study of computer programming?
2. When you answered questions incorrectly, OGITS prompted you to locate relevant material on the web or presented you with pertinent content to review. What effect did this have on your learning?
3. What is your opinion of using games (such as Tic-tac-toe) with feedback as a mechanism for studying the course concepts and principles in OGITS?
4. Would you say that, because of OGITS, learning can happen without instructors?
5. What effect did OGITS have on the process of observing your weaknesses and strengths?
6. Did OGITS influence your views concerning basic programming concepts?
7. What is your opinion concerning studying these subjects via OGITS or teachers?
8. Would you like to study other programming subjects through a system like OGITS?

Data Analysis

OGITS was evaluated in terms of learning by means of the developed scale, from which quantitative data were gathered and their responses were calculated in terms of frequency distribution and percentage rates. In addition, each item’s mean was calculated. Qualitative data were analyzed through content analysis with the aim to explain the present data. Data were analyzed within the following phases: encoding data, setting themes, defining and grouping data with respect to themes, and analysis of results (Yıldırım & Șimşek, 2005).

Results

Quantitative Data Analysis

Student responses to the OGITS learning assessment scale were analyzed by means of descriptive statistical methods. Results of this analysis including average values, percentage rates, and frequency distributions for each item are summarized in Table 2.

Among 15 items in the scale, the 1st, 3rd, and 2nd items demonstrated the highest rates of positive opinion (i.e., the sum of Strongly Agree and Agree; 75.4%, 75.3%, and 74.1%, respectively), while the 8th, 11th, and 9th items received positive view percentages of 67.9%, 66.7%, and 65.5%, respectively. These topped the positive view percentages among items in the scale. The 7th and 15th items received rates of positive opinions at 49.4% and 41.9%, respectively, clearly reflecting largely neutral opinions.

One item on the scale, the 14th, was phrased as a negative statement. For this item, the sum of Strongly Disagree and Disagree opinions was 81.5%. Given that both the question and answers were negative, this value represented the percentage of students who disagreed with the notion that learning with OGITS was boring. Overall, the frequency distribution of items generally demonstrated positive assessments of OGITS except for three items (two mostly neutral, one negatively phrased [Item 14]). The cumulative average of items was 66.3%.

Qualitative Data Analysis

Twenty-eight students were randomly selected from the 79 students who worked with OGITS for interview. One student who offered contradictory answers was
### Table 2. Student Scale for Evaluation of OGITS.

<table>
<thead>
<tr>
<th>No.</th>
<th>Items</th>
<th>( \bar{x} )</th>
<th>( f )</th>
<th>%</th>
<th>( f )</th>
<th>%</th>
<th>( f )</th>
<th>%</th>
<th>( f )</th>
<th>%</th>
<th>( f )</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Appropriate learning materials offered by OGITS facilitated my learning</td>
<td>3.80</td>
<td>5</td>
<td>6.2</td>
<td>4</td>
<td>4.9</td>
<td>9</td>
<td>11.1</td>
<td>45</td>
<td>55.6</td>
<td>16</td>
<td>19.8</td>
</tr>
<tr>
<td>2</td>
<td>OGITS guided me to search for relevant information on the web if I had difficulty to answer a question correctly. This contributed to my online information searching skills</td>
<td>3.81</td>
<td>6</td>
<td>7.4</td>
<td>6</td>
<td>7.4</td>
<td>7</td>
<td>8.6</td>
<td>38</td>
<td>46.9</td>
<td>22</td>
<td>27.2</td>
</tr>
<tr>
<td>3</td>
<td>When I answered questions incorrectly, OGITS provided me with pertinent content to review which contributed to my learning computer programming</td>
<td>3.95</td>
<td>5</td>
<td>6.2</td>
<td>5</td>
<td>6.2</td>
<td>8</td>
<td>9.9</td>
<td>32</td>
<td>39.5</td>
<td>29</td>
<td>35.8</td>
</tr>
<tr>
<td>4</td>
<td>I could understand the programming concepts and the subject better by learning pertinent learning materials in OGITS</td>
<td>3.67</td>
<td>6</td>
<td>7.4</td>
<td>12</td>
<td>14.8</td>
<td>11</td>
<td>13.6</td>
<td>42</td>
<td>51.9</td>
<td>8</td>
<td>9.9</td>
</tr>
<tr>
<td>5</td>
<td>I would not forget the knowledge I gained by OGITS</td>
<td>3.52</td>
<td>8</td>
<td>9.9</td>
<td>9</td>
<td>11.1</td>
<td>12</td>
<td>14.8</td>
<td>34</td>
<td>42</td>
<td>16</td>
<td>19.8</td>
</tr>
<tr>
<td>6</td>
<td>OGITS helped me learn programming concepts independently of teacher</td>
<td>3.54</td>
<td>9</td>
<td>11.1</td>
<td>9</td>
<td>11.1</td>
<td>10</td>
<td>12.3</td>
<td>32</td>
<td>39.5</td>
<td>19</td>
<td>23.5</td>
</tr>
<tr>
<td>7</td>
<td>The number of tasks available in OGITS for basic concepts of C++ programming language is enough</td>
<td>3.39</td>
<td>4</td>
<td>4.9</td>
<td>15</td>
<td>18.5</td>
<td>20</td>
<td>24.7</td>
<td>26</td>
<td>32.1</td>
<td>14</td>
<td>17.3</td>
</tr>
</tbody>
</table>

(continued)
<table>
<thead>
<tr>
<th>No.</th>
<th>Items</th>
<th>x</th>
<th>f</th>
<th>%</th>
<th>f</th>
<th>%</th>
<th>f</th>
<th>%</th>
<th>f</th>
<th>%</th>
<th>f</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Giving me a goal to be perused (i.e., winning the games) is a good idea</td>
<td>3.66</td>
<td>7</td>
<td>8.6</td>
<td>9</td>
<td>11.1</td>
<td>8</td>
<td>9.9</td>
<td>35</td>
<td>43.2</td>
<td>20</td>
<td>24.7</td>
</tr>
<tr>
<td>9</td>
<td>Learning basic concepts of C++ and task completion by engaging in Snakes and Ladders and Tic-tac-toe game simultaneously is a good idea and could improve my motivation</td>
<td>3.78</td>
<td>4</td>
<td>4.9</td>
<td>5</td>
<td>6.2</td>
<td>17</td>
<td>21</td>
<td>31</td>
<td>38.3</td>
<td>22</td>
<td>27.2</td>
</tr>
<tr>
<td>10</td>
<td>Tips and solution provided by OGITS facilitated my learning of programming concepts</td>
<td>3.56</td>
<td>8</td>
<td>9.9</td>
<td>7</td>
<td>8.6</td>
<td>13</td>
<td>16</td>
<td>35</td>
<td>43</td>
<td>16</td>
<td>19.8</td>
</tr>
<tr>
<td>11</td>
<td>OGITS helped me to recognize my strengths and weaknesses</td>
<td>3.61</td>
<td>7</td>
<td>8.6</td>
<td>9</td>
<td>11.1</td>
<td>9</td>
<td>11.1</td>
<td>37</td>
<td>45.7</td>
<td>17</td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>OGITS was a good help in learning this subject</td>
<td>3.68</td>
<td>7</td>
<td>8.6</td>
<td>10</td>
<td>12.3</td>
<td>13</td>
<td>16</td>
<td>20</td>
<td>24.7</td>
<td>29</td>
<td>35.8</td>
</tr>
<tr>
<td>13</td>
<td>OGITS enabled me to develop more positive attitude concerning programming concepts</td>
<td>3.61</td>
<td>4</td>
<td>4.9</td>
<td>14</td>
<td>17.3</td>
<td>12</td>
<td>14.8</td>
<td>28</td>
<td>34.6</td>
<td>21</td>
<td>25.9</td>
</tr>
<tr>
<td>14</td>
<td>Learning with OGITS was boring</td>
<td>2.1</td>
<td>22</td>
<td>27.2</td>
<td>44</td>
<td>54.3</td>
<td>7</td>
<td>8.6</td>
<td>2</td>
<td>2.5</td>
<td>4</td>
<td>4.9</td>
</tr>
<tr>
<td>15</td>
<td>I would like to learn other subjects of computer programming through a system similar to OGITS</td>
<td>3.06</td>
<td>13</td>
<td>16</td>
<td>16</td>
<td>19.8</td>
<td>16</td>
<td>19.8</td>
<td>21</td>
<td>25.9</td>
<td>13</td>
<td>16</td>
</tr>
</tbody>
</table>

Note. OGITS = Online Game-based Bayesian Intelligent Tutoring System.
excluded from the evaluation. Qualitative data were assessed in accord with survey findings. Student opinions concerning OGITS were ordered in the same manner as interview questions. Particularly noteworthy points concerning the OGITS scale are presented below:

- **Students offered generally positive sentiments concerning the use of OGITS in learning computer programming.** Based on Item 1 shown in Table 2, percentages of negative, neutral, and positive opinions were 11.1%, 11.1%, and 75.4%, respectively. Given these rates, percentages of negative or unsure responses did not override generally positive opinions of these students.

This impression was supported by data garnered from interviews. Most students declare that OGITS is an ideal way for learning computer programming in that it both caters educational experience to personal preferences of each student while enhancing their motivation via gaming. One student remarked, “I do not think our tutor can provide all students with individualized types of feedback and learning materials at the same times. Moreover, OGITS tests our knowledge of computer programming frequently while offering us feedbacks and tips.” Another student commented, “I mostly get distracted or lose myself in thought during tutor conversations. However, using OGITS, I can focus more and pay more attention to learning materials.” A few students voiced the idea that adding a comparative feature to OGITS, in which students could see each other’s performance, would enhance competition between users. Significantly, a number of students strongly agreed that OGITS enhanced both their knowledge of computer programming and their online information searching skills.

- **Students declared that, when they experienced difficulty, it was helpful when OGITS prompted them to look up pertinent information on the web or presented them with the requisite course content.** As shown in Table 2, Items 2 and 3 measure student’s satisfaction concerning this feature of OGITS. Although negative and neutral percentages responding to this claim were 14.8% and 12.4% for negative and 8.6% and 9.9% for neutral, respectively, the percentages of positive responses are generally high, at 74.1% and 75.3%. Given these rates, negative and neutral percentages were unremarkable compared with the high percentages of positive responses.

Data attained from interviews confirmed this impression. These students universally agreed that being guided to find pertinent information on the web or review requisite course materials was an effective way to resolve their struggles with difficult question or topic. Students commented that it helped them grasp fundamental programming concepts when they were directed to pertinent
learning programming concepts in the form of games was fun and I really enjoyed it when OGITS automatically directed me to either search for relevant information on the web or relative learning materials while I had difficulty in answering a particular question.

Although the majority of students felt that this direction helped their learning by informing them with other perspectives, a small minority of students declared that it did little to enhance their learning.

- Students praised the game aspect of OGITS, in which games of Snakes and Ladders and Tic-tac-toe were combined with feedback and guidance to help them understand programming concepts and principles. In Table 2, Items 9 and 10 depict student satisfaction concerning the use of these games-plus-guidance in learning the basics of programming. On these claims, the negative and neutral opinion percentages were 11.1% and 18.5% negative and 21% and 16% undecided, but positive opinion percentages are high: on Item 9, 65.5% positive responses concerning learning through games, and on Item 10, 62.8% positive responses concerning the feedback, guidance, and content review OGITS offered within games. These rates clearly indicated that most students held favorable opinions concerning these games.

Interview data also supported this impression as most of these students declared that the games (Snakes and Ladders, Tic-tac-toe) and connected support and guidance had a beneficial effect on their study of programming concepts. One student said, “The games along with various types of feedbacks and tips helped me learn the programming concepts better in a fun way, therefore, I think it is perfect to learn in this way.” Another student commented, “I learned various programming concepts step-by-step with detailed feedbacks and tips which facilitated my comprehension. Additionally, it improved my motivation.” However, one student argued that the guidance offered lacked descriptiveness: “Feedback and learning materials for some concepts were not sufficient. I asked for more assistance from the teacher. I assume they must be presented in a more descriptive way.”

- Students also indicated a belief that OGITS could free them from needing a teacher to learn. Table 2 presents Item 6, reflecting students’ opinions concerning studying computer programming without a teacher. Negative and neutral opinion percentages concerning this claim were 22.2% and 12.3%, respectively. However, 63% of students answered Item 6 positively. Thus, the
majority of students expressed positive opinions concerning OGITS promoting independence in learning.

The interview data echoed this impression. Most students declared that, thanks to OGITS, independent learning of the topic was feasible. One student voiced this opinion by saying, “without the teacher, we can still learn. To me, there is no difference between OGITS and teacher, I consider OGITS as my teacher.” A second student added, “I think we can learn without a teacher around. We could learn the subject without having any prior knowledge.”

- **Students also agreed that OGITS helped them more clearly see their own strengths and weaknesses.** Table 2 illustrates Item 11 which solicits students’ opinion concerning this issue. Percentages of negative and neutral responses to this claim were 19.7% and 11.1%, respectively. However, percentage of positive responses to this claim was 66.7%. These rates showed that most students responded favorably to the claim that OGITS promoted critical self-understanding.

The following seven themes were discerned in responses to Item 11’s question about the efficacy of OGITS in helping students recognize their own strengths and weaknesses: motivation, learning approach, confidence, learning characteristics, learning responsibility, subject, and independent learning.

1. Students declared that games, feedback, and course materials enhanced their motivation in learning C++ programming basics.
2. One student commented, “I come to understand that learning at my own pace along with my learning style is better for my learning.”
3. One student remarked, “I realized things that I can and cannot do while teacher is not around. My confidence is enhanced in these regards.”
4. Some students commented that they recognized how OGITS improved and eased their learning.
5. Students declared that OGITS helped them realize the extent to which they are responsible for learning the requisite concepts to complete their studies.
6. Most students remarked that OGITS helped them realize their weak areas within the subject.
7. Regarding this fact, one student remarked, “Thanks to the tips, feedbacks, and learning materials in OGITS, I come to know that I can learn independently of teachers.”

- **Students demonstrated positive views of the impact of OGITS on their outlook on C++ programming concepts.** Table 2 presents Item 13 which addresses this matter. Percentages of negative and neutral opinion for this time were 22.2% and 14.8%, respectively. However, the percentage
of positive opinion for this item was 60.5%. Thus, most students held positive opinions concerning the effect of OGITS on their outlook on C++ programming concepts.

Six students declared that OGITS had no effect on their views concerning C++ programming concepts. Of these students, three said they liked C++ programming already and their views underwent no change. Another student said that she or he never liked C++ programming. OGITS did not change such dislike. Twenty-one students announced that OGITS had a positive effect on their attitude toward C++ programming. However, 12 of these declared that the system helped them view C++ programming concepts in a new light. Regarding this, one student declared, “I realized that learning C++ programming is fun with the help of OGITS.” Another student commented, “I always thought that it is impossible to learn C++ programming without a teacher. However, after learning by OGITS, I realized that some subjects or concepts of C++ programming can be learn without a teacher.” A third student described an even more positive effect of OGITS on his or her views of C++ programming: “I already liked C++ programming. After being exposed to OGITS, I like it even more owing to the fact I can learn independently on my own.”

- **Students demonstrated more interest in lectures received via OGITS than that via a tutor.** A majority of these students said they would prefer to learn a subject through OGITS. Of these, 14 declared a preference for OGITS because of individualized content and motivational incentives it provides. Regarding this, a student remarked, “I would rather using OGITS owing to the responsibility I feel in answering these questions. Using this system, our progress depends on us and feels like everything is under control.” A second student declared,

> During lectures, if we do not learn the learning materials at first or second time, the lecturer will have no time to repeat over and over for us. However, using OGITS, we can take our time and learn at our own pace. We can even return and redo.

Six students voiced a preference for a teacher over OGITS because of their particular study habits and the possibility of dialogue. Regarding this, a student stated, “Sometimes the system cannot answer us as our teacher does. We are somehow bounded to the computer and it cannot provide us with more when needed.” Seven students indicated that their ideal combination would be to receive lectures from both OGITS and a teacher.

- **Some students articulated somewhat positive opinions concerning the use of a system akin to OGITS for studying other subjects within C++ programming.** Negative, neutral, and positive opinion percentages concerning this claim
were 35.8%, 19.8, and 41.9%, respectively. These rates reflected the fact that students had mixed opinions on the possibility of learning other C++ programming subjects through such a system. Some were positive, while others thought it impossible. While there is little agreement between students on this subject, the rate of students who would reject such a system is low.

Again, data culled from interviews confirmed this impression. Ten students doubted that more difficult concepts (class, files, etc.) could be taught through a system like OGITS. However, they agreed that simpler concepts (control structures, arrays, etc.) could lend them to being taught in such a system. Thirteen students declared that they would like to tackle other C++ programming concepts by means of a system like OGITS. They voiced a preference for OGITS because it could cater the learning environment to the needs of the user, empower students, encourage permanent knowledge, and enhance their motivation by means of games. Moreover, the majority of students agreed that OGITS assisted their online information searching abilities. Four students said that pairing OGITS with a teacher would make it an optimal learning environment.

**Discussion and Conclusions**

Replicating human instruction in individualized one-to-one instruction can present great challenges in research on the design of educational environments in some ways. Successful e-learning depends particularly on the traditional architecture of ITS and new ways to model and enhance a student’s learning process, performance, and engagement. For this reason, OGITS was developed in order to improve programming acquisition and online information searching skills, thus improving students’ ability of problem solving. Because these fundamental skills require active learning which diverges from theoretical learning, the OGITS approach holds great promise. If it is done correctly, OGITS can help students master essential concepts of computer programing. At the same time, it can enhance their online information searching and encourage them to partake in online assessments so that they can learn more efficiently. OGITS encourages users to explore online educational content in a more interactive way by using games (Snakes and Ladders and Tic-tac-toe) while also providing an adaptive and individualized system. It steers students through problem-solving activities in a way that is both enjoyable and instructive, providing players with educational goals and corresponding content sequences. While many researchers have studied the effect of educational computer games on users’ motivation and educational accomplishments, OGITS makes the use of online information searching educational and fun.

Quantitative and qualitative data garnered from students were used to uncover users’ attitudes and opinions concerning OGITS. They were also used
to demonstrate how OGITS affected users’ progress in computer programming and online information searching skills. Our findings concerning OGITS are significant because they can contribute to assessments of the role and future of game-based ITS in educational system. In interviews, students indicated that OGITS helped them study in a manner well-suited to their learning and expertise. They felt that they had developed their programming and online information searching skills. Data gathered from the scale indicated that 75.4% of students agreed with the claim that OGITS aided their learning by providing appropriate contents, while 74.1% and 75.3% of students identified that OGITS improved their computer programming and online information searching skills, respectively. The qualitative data reinforced that students’ comprehension of programming concepts was improved when OGITS responded to their incorrect answers by presenting them with appropriate course material or directions for finding pertinent information on the web. Notably, some students never received such guidance from OGITS because they answered all questions correctly. Some studies in the literature have suggested that personalized learning can directly enhance student satisfaction. For instance, Niemi, Nevgi, and Virtanen (2010), Iacovides, Aczel, Scanlon, and Woods (2012), Schiaffino, Garcia, and Amandi (2008), and Brown, Brailsford, Fisher, and Moore (2009) have demonstrated that learners are satisfied when they are given contents suited to their learning styles and individual abilities. Findings of this study are consistent with results of these studies.

Users also declared that guidance and feedback helped them recognize their weak spots. This is in agreement with a previous study showing that adaptive game-based learning environment is useful because it can provide users with guidance and feedback tailored to their expertise and learning style (Hatzipanagos & Warburton, 2009). Papanikolaou, Mabbott, Bull, and Grigoriadou (2006) have concluded that the majority of learners positively view course content and feedback given to them by the system.

Students also recognized their ability to learn in the absence of a teacher after using OGITS. The sixth item in the scale clearly suggested this since its percentage of positive opinions was 63%. The belief that learning could take place without a teacher was echoed in the qualitative data as a number of students commented that they had come to understand that OGITS could make independent learning possible. The success of future systems like OGITS depends on whether students are willing to and able to learn other C++ programming subjects through such a system. The 15th item on the scale showed that percentages of negative, neutral, and positive responses to this topic were 35.8%, 19.8%, and 41.9%, respectively. Given these rates, there is no clear consensus among students regarding this issue. However, the rate of students expressing no interest in studying with this kind of system was low. Qualitative data derived from interviews illuminated this range of opinions. Some students declared that
many subjects could not be studied with a system like OGITS. Others indicated their willingness to try out systems like OGITS because they could offer a learning environment tailored to individual educational needs. In addition, they could empower students, enhance their motivation, and promote an atmosphere of discovery. The remainder of students expressed a preference for using such systems in combination with human teachers.

As shown by the gathered data, students identified a number of positive attributes of OGITS. It increased their motivation by means of games, similar to the work conducted by Sung and Hwang (2013). However, OGITS not only promotes the idea of navigating online learning materials and updating the Bayesian network by applying an online game-based formative assessment, it also seeks to enhance online information searching skills to improve students’ ability in web-based problem solving. In addition, it increases their self-confidence and capacity for independent learning, helps them identify their learning style, strengths, and weaknesses, promotes responsibility in the learning process, guides them through the learning process, and makes the learning environment enjoyable and memorable.

Declaration of Conflicting Interests
The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the ICT R&D program of MSIP/IITP in the Republic of Korea (grant number 2016 [B0101-16-0340]) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP; grant number R1610941).

References


Author Biographies

Danial Hooshyar is a research professor at the Faculty of Computer Science and Engineering, Korea University, Korea. His primary area of research is artificial intelligence in education and technology-enhanced learning.

Rodina Binti Ahmad is a professor at Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. Her primary area of research is Requirements Engineering and Organizational analysis and modeling.

Minhong Wang is a professor at Faculty of Education, The University of Hong Kong, Hong Kong. Her primary area of research is Knowledge Visualization, e-
Learning Design and Analysis, STEM Education, and Technology in Higher Education.

Moslem Yousefi is a research professor at the School of Civil, Environmental and Architectural Engineering, Korea University, Korea. His primary area of research is artificial intelligence and its application in the real-world problems including design and management of energy systems and educational data mining.

Moein Fathi is currently a master student at Faculty of Computer Science and Information Technology, University of Malaya, Malaysia. His primary area of research is artificial intelligence in education and technology-enhanced learning.

Heuiseok Lim is a professor of educational data mining at the Faculty of Computer Science and Engineering, Korea University, Korea. His main research interests are artificial intelligence in education (e.g., educational data mining), information retrieval, and natural language processing.