Teaching Boolean Logic through Game Rule Tuning

Jui-Feng Weng, Shian-Shyong Tseng, *Member*, *IEEE*, and Tsung-Ju Lee

Abstract—The Boolean logic is a tool to model the algebra of truth values. It is an essential subject in Computer Programming and Digital Electronics. While teaching the Boolean logic, teacher usually gives examples to elaborate the Boolean logic formulation. However, the physical meaning of the Boolean logic is difficult for students to imagine without the realization to real-world case. The classical Pac-Man game which has clear Boolean logic game rules and simple game scenario is chosen as the teaching aid. Accordingly, two learning activities, the Boolean logic realization activity and game scenario formulation activity, are proposed. Our idea is that through game rule tuning in the learning activities, the students can comprehend how the changing of game rule (Boolean logic) affects the game scenario. To manipulate the game scenario for teaching specific learning objectives of Boolean logic, the Game Rule of the original Pac-Man game is defined and seven versions of modified game scenarios provided as illustrations are appropriately organized for students to practice one Boolean logic concept each time. The experimental results showed the proposed activity can have higher learning achievements compared to the traditional lecturing.

Index Terms—Game rule realization, Boolean logic, game rules tuning, Scratch, game-based learning.

1 Introduction

The Boolean logic is the logical calculus with algebra of truth values 0 and 1 representing false and true, respectively. Numerous subjects studied in schools, such as describing scientific theorem in Mathematics and Physics or causal statements in History and Literature use the Boolean logic as the formulation language. There are also many problems about decision making or rule designing in our daily lives, which can use the skill of Boolean logic to help clarify the possible effects for different situations.

Since the Boolean logic is a tool for representing the cognitive principles or rules, the Truth table, Logic gate, and Venn diagram approaches are usually used in traditional lecturing to visualize the behavior of different operators and the results of various logic expressions. However, these traditional approaches lack meaningful scenario to connect students' experiences, the mappings between Boolean logic and real-world case spaces are usually difficult for students to imagine and comprehend. Teachers usually need to explain the concepts with real-world cases including the formulation from the real-world cases to Boolean logic space and the realization from the Boolean logic expressions to the real-world space. It motivates this paper of applying the game-based learning approach to assist students in manipulating and observing the relationship of two spaces.

With our survey, some practitioners used the puzzle games for training the Boolean logic [1], [2], [3]. It seems that the game is interesting and more acceptable for students to realize the abstract concepts through the real-world examples. However, the game rules of the traditional puzzle games are predefined with the fix set of Boolean expressions. It is not easy for teachers to collect related games or modify the rules to teach the designated Boolean logic operations. How to design a game-based activity which can manipulate the game rules and the difficulty to meet the specific teaching objectives is a challenging and important issue.

With our observation, a game is composed of game rules, game script, and game roles. The popular game such as Pac-Man involves well-designed game rules, simple game scenario, and characteristic game roles to interest players. In this paper, we define the grammar of game rules to represent the Boolean logic expressions space and the game scenarios to represent the real-world case space. For example, the rule "Ate_Pill AND Touched_Ghost → Score" includes the conjunction and implication operators for one scoring scenario that "if the Pac-Man ate the pill and touched the ghost then the player gets one point of score." Thus, the students can practice the realization of Boolean logic from the game rules to the game scenario and vice versa. Accordingly, our goal aims to provide several modified game scenarios from original Pac-Man game to stimulate students to comprehend the specific Boolean logic in game rules in terms of teacher's teaching objectives. Therefore, how to model and manipulate the rules of the game cases is our first concern.

To solve the problem, the context-free grammar is applied to represent the Boolean logic expressions of game rules. Therefore, the rule-tuning learning activity is defined as modifying the original set of Boolean logic expressions to the new set of expressions. And the difficulty of each rule-tuning activity can be defined as the minimum number of Boolean logic operations needed to be modified from the

- *J.-F. Weng and T.-J. Lee are with the Department of Computer Science, National Chiao Tung University, EC447 KDELab, No. 1001, Ta Hsueh Road, Taiwan 300, R.O.C.*
  E-mail: roy@cis.nctu.edu.tw, freeman1217@gmail.com.
- *S.-S. Tseng is with the Department of Information Science and Applications, Asia University, EC447 KDELab, No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan, R.O.C. E-mail: sstseng@asia.edu.tw.*

Manuscript received 9 Dec. 2009; revised 17 Apr. 2010; accepted 26 Aug. 2010; published online 14 Oct. 2010.
For information on obtaining reprints of this article, please send e-mail to: lt@computer.org, and reference IEEECS Log Number TLTSI-2009-12-0193.
Digital Object Identifier no. 10.1109/TLT.2010.33.

1939-1382/10/$26.00 © 2010 IEEE Published by the IEEE CS & ES

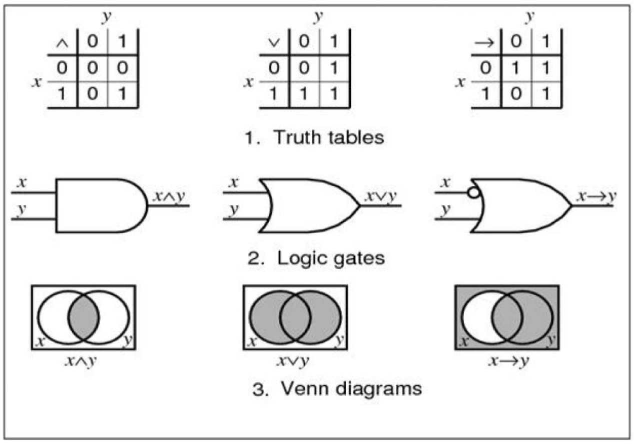


Fig. 1. Traditional teaching tools of Boolean logic.

original game rules. The proposed approach has been applied to the 43 junior high school students. The finding is surprising that students were actively joined the learning activity even for those passive students. Although some students could not correctly tune the game rules, the activities engaged their attentions and they were willing to try again.

The main contributions of this paper are that we proposed a novel perspective to smoothly apply game-based learning to the teaching of Boolean logic. The model of game rules can support teachers easier manipulating the difficulty of game-based learning activities for specific teaching objectives.

In the following, the preliminaries about traditional Boolean logic instructions are introduced in Section 2. The issues about realization in logic learning and our ideas of applying game-based learning activity are given in Section 3. The model defined for manipulating the difficulties of game-based learning activities is proposed in Section 4. The experimental evaluations and findings are discussed in Section 5. The concluded remark is given in Section 6.

2 Preliminaries

This section describes the issues of traditional platforms used in Boolean logic instructions. The recent game-based learning approaches are also surveyed to improve the students' interest and effectiveness of logic learning.

2.1 Traditional Boolean Logic Instructions

The Boolean logic was developed by George Boole in 1840s. The basic Boolean operations includes conjunction $x \wedge y$ (AND), disjunction $x \vee y$ (OR), and complement or negation $\neg x$ (NOT). The implication $x \rightarrow y$ can combine with above basic operations to express the well-known De Morgan's laws equivalent. In traditional Boolean logic instruction, the Truth table, Logic gate, and Venn diagram are the well-known approaches to demonstrate and elaborate the operations of the Boolean logic [4] as shown in Fig. 1. To assist the students' practice, the Boolean operations, interactive tools, such as electronic circuit simulator [5], and interactive Venn diagram exercises [6] are developed. However, the traditional instruction tools can only support students to be proficient with the abstract Boolean logic expression. How to formulate and realize the Boolean logic to real-world case is difficult for students.

With our survey, the Boolean logic has been used in several puzzle games for the training of students' logic reasoning. The Binary Color Device game [1] allows players to practice the Boolean algebra by applying a sequence of Boolean operations on given numbers to produce specific objective numbers. The Logic Puzzle [2] provides a series of English expressions and asks players to conclude the properties of each subject. The classic Wolf Sheep and Cabbage logic game [3] allows players to reason and find a solution from every combination of subjects. Although there are several practical case studies of game and logic learning, these games are basically developed for fun. How to further apply the pedagogical theory to the game platform is still a challenging and important issue.

2.2 Game-Based Learning Approaches

The computer games with exciting interactive activities and interesting multimedia provide a way to motivate students to learn actively and interestingly. With the growth of computer and learning technology, there are several approaches to apply computer games to the learning activities.

Applying the game as simulation platform allows students to comprehend the cause and effect of the concepts through the trial and error process, and understand the learning achievements with visualized charts or scores. For example, the Farmtasia [7], [8] simulated the biology, chemistry, and economics of farm management, and the Fish Tank [9] simulated the nitrogen cycle in an aquarium. Applying game as assignment [10], [11] can motivate students engaging in the assignments with interesting game scenario and game context, such as the scoring, adventure, and competition properties. Applying game creation project [12] can assist students to easily understand the abstract knowledge with interesting game context. In this paper, we aim to apply the game creation project to support the students' learning. However, the programming is too difficult for the junior high school students. Therefore, providing an easy-to-use tool and the well-designed learning activity is important to apply the game-based creation approach.

2.3 The Interactive Tools to Assist the Logic Learning

The game-like interactive tool to support the teaching of abstract knowledge is an interesting research field to promote the philosophy of learning by doing or game to learning. The Logo [13], the turtle graphics, is one of the oldest libraries used to introduce computing concepts to beginners in the early 1970s. The concept of a "turtle" that can move across a 2D plane with a pen, which can be positioned on or off the ground, and thus, may leave a trace of the turtle's movements. Programming the turtle to draw different patterns can be used to introduce general computing skill, such as procedural operations, iteration, and recursion. The process of computing can be easily realized for students by mapping onto the turtle world.

The other tool is Karel the Robot [14], [15], [16], [17], [18]. The programming is based on a virtual robot, named as Karel, in a 2D grid world with walls around and beepers to collect. Similar to the turtle graph, it provides a metaphor for students to easily understand the physical meaning and execution effect of the abstract object-oriented programming by observing the changes of objects' behavior in the 2D grid world.

The BlueJ [19] is an educational programming tool specially designed for students to understand the concepts of object-oriented programming by visualizing the properties, such as classes, objects, and methods, etc. The BlueJ tool provides the class diagram for students to define classes and their relations. Next, the execution screen supports strong graphical feedback to visualize how objects behave to reflect the designed class diagram.

The Greenfoot [20], [21], [22], [23] is the object-oriented programming tool that supports the concept of learning within given scenarios and storytelling, e.g., it supports several predefined game scenarios as learning samples and students can directly interact with visualized objects called actors to build up a game based on teacher's assignment. Currently, classical computer applications including robot control, ant colonies ecological simulation, the lifts simulation, and turtle graphics, etc., are provided.

The Scratch [24], [25], [26] developed by Resnick from MIT Media Lab is another game creation platform based on the self-defined object-oriented programming language. It should be noticed that the Scratch programming tool provides building-block-based interface with low barrier and high expressive programming power. With the easy three steps including image, program, and share, the children can easily create their own games or animations and upload them to the Scratch online website to share their works with other community members.

The Alice [27] from CMU provides an object-oriented 3D animation programming tool for the introductory computer science courses. The properties and behaviors of 3D objects can be learned by editing or programming for the animation. To further make computer programming more attractive for middle school girls, the Storytelling Alice [28] was proposed. Three major improvements were made in the comparison of Storytelling Alice to generic Alice: the high-level animation by analyzing the storyboard that the girls created, the library of 3D models to help spark idea, and the story-based tutorial through Stencil technique.

In summary, several high-level interactive media creation tools were proposed to assist in realization of the abstract programming logic with interesting game design or storytelling. However, given the single and fixed learning scenario, it may not be suitable for teachers to teach the required objectives. Besides, the important defect of the game-based learning is that students may just play it as a game but not really learn the subjects. Thus, how to appropriately organize the learning activities with well-designed game material becomes an important and interesting issue.

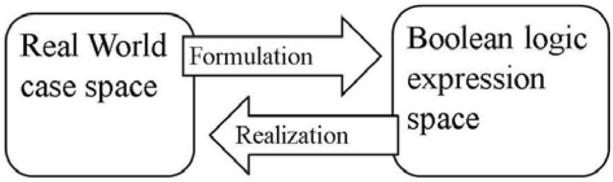


Fig. 2. The formulation and realization process of two spaces.

3 Game-Based Learning through Game Rule Tuning

In this section, the issues of teaching Boolean logic are shown and our idea of game-based learning through game rule tuning is proposed.

3.1 The Issues

As shown in Fig. 2, there are two spaces for teaching the Boolean logic, which are Boolean logic expression space and real-world case space. For example, the real-world case shows that "if two switches are connected in a series circuit, then they light only when both the two switches are turned on," and the students should be able to formulate the Boolean logic expression for the circuit as "Switch_A $\wedge$ Switch_B $\rightarrow$ Light." The traditional truth table can help students to describe the Boolean logic expression through elaborating all-possible combinations of values and each operation result. However, the realization and formulation between Boolean logic expressions and real-world case are usually difficult for students to imagine.

It seems that the growth of game-based learning technology can help the students to easily catch the point through their interesting game as real-world learning examples. Therefore, our idea aims to use the Pac-Man game scenario as the realization of real-world case space, and the formulation of game rules as the Boolean logic expression space. Through providing different modified Pac-Man game scenarios, students are able to realize and comprehend the effects of different combinations of Boolean logic expressions.

Since the game cases should be able to assist a teacher in teaching the specific Boolean logic concept, two issues of the game-based learning that need to be discussed are as follows:

- Teaching Boolean logic realization: How to manipulate the difficulty of Boolean logic to assist students in easily observing the effects of specific logic operations.
- Teaching real-world case formulation: How to manipulate the dissimilarity among different modified game scenarios to choose appropriate difficulty of game case for students' formulation practice.

In summary, the above issues direct the need for the modeling of game-based learning activity for appropriately manipulating the teaching objectives and difficulty.

3.2 The Learning Contexts

To support the teaching of Boolean logic and solve the issues above, the game-based learning with game-rule-tuning activity is proposed. The game-rule-tuning activity

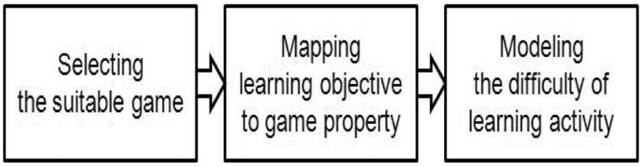

Selecting the suitable game
Mapping learning objective to game property
Modeling the difficulty of learning activity

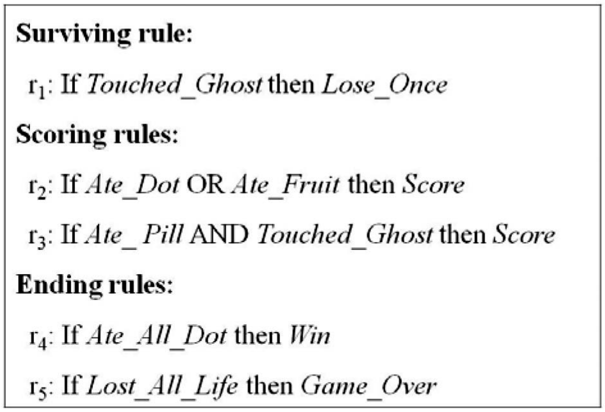
Fig. 3. The game-rule-tuning activity design process.

design process is shown in Fig. 3. The initial process of selecting the suitable game is the most important process. For example, the logic learning should select the puzzle game focused on logic training, the management learning should select the strategy game focused on decision making, the clinical learning should select the simulation or action game focused on reaction training, etc. Next, the process of mapping learning objective to game property instantiates the abstract learning objective to the meaningful game scenarios or obvious scores. The process of modeling difficulty of learning activity allows the teachers to provide suitable tasks following the pedagogical need. In this paper, the classical game Pac-Man is chosen as the teaching material and the Scratch programming tool is used as the learning platform.

3.2.1 The Game Context

The elements of the Pac-Man game can be defined as follows: The main game scenario of the Pac-Man is that a player controls Pac-Man through a maze, eating pac-dots. When all dots are eaten, Pac-Man is taken to the next stage. Four ghosts including *Blinky, Pinky, Inky, and Clyde* roam the maze, trying to catch Pac-Man. If a ghost touches Pac-Man, a life is lost. When all lives have been lost, the game ends. Thus, the game objects include Pac-Man, ghost, pac-dot, power pill, and fruit. To simplify the discussion, the game rules which are embedded on the specific game object are controlling the behavior of the object. The definition of elements of the game rule is as follows:

Definition 1 (The Elements of the Game Rule). *The game rule of a game is defined as* $G = (O, S, R)$, *where the O is a set of game objects representing the scene objects and roles in the game scenario, the S is the actions in the game scenario including the change of events and status, and* $R = \{r_1, r_2, r_3, \ldots, r_m\}$ *is a set of game rules with left side of compound event representing trigger conditions and right side the status changing actions.*



Surviving rule:

r_1: If *Touched_Ghost* then *Lose_Once*

Scoring rules:

r_2: If *Ate_Dot* OR *Ate_Fruit* then *Score*

r_3: If *Ate_ Pill* AND *Touched_Ghost* then *Score*

Ending rules:

r_4: If *Ate_All_Dot* then *Win*

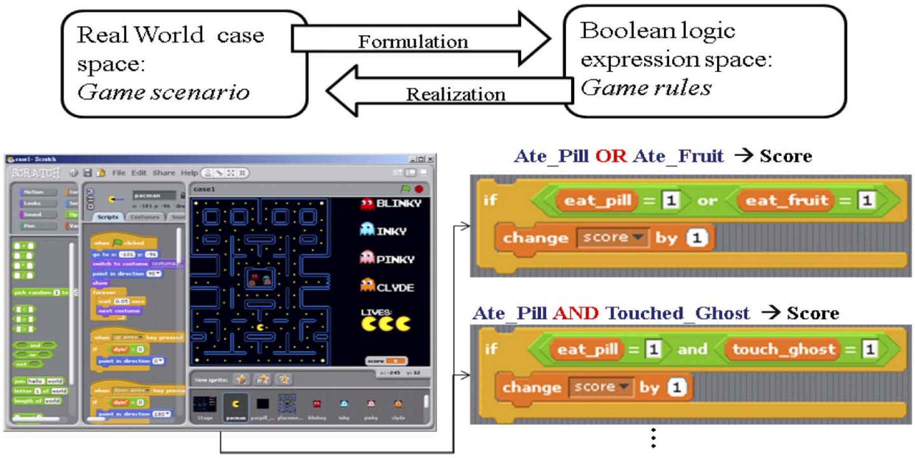
r_5: If *Lost_All_Life* then *Game_Over*

Fig. 4. The game rules controlling the scenario of *Pac-Man.*

In the Pac-Man game, the $O = \{Pac\text{-}Man, Ghost, pill, fruit, pac\text{-}dot\}$, the $S = \{Touched\_Ghost, Ate\_Pill, Ate\_Fruit, Ate\_Dot, Ate\_All\_Dot, Lose\_Once, Score, Win\}$, and the R are shown in Fig. 4. Each game rule is composed with left-side compound conditions and right-side actions. There are three types of rules corresponding to different actions. The r_1 is the surviving rule, the r_2 and r_3 are the scoring rules, and the r_4 and r_5 are the ending rules.

3.2.2 The Platform Context

The Scratch [21], [25] open source programming tool was developed by Resnick [26] from MIT Media Lab in 2007 for students to easily create games and share their creations. The Scratch is used as the learning platform in the Boolean logic learning. The Scratch programming tool is based on the self-defined object-oriented programming language with the support of the logical operators including the conjunction, disjunction, and negation. As shown in Fig. 5,


Real World case space:
Game scenario
Formulation
Realization
Boolean logic expression space:
Game rules
BLINKY
INKY
PINKY
CLYDE
LIVES:
Ate_Pill OR Ate_Fruit → Score
Ate_Pill AND Touched_Ghost → Score
⋮

```
if < <eat_pill = 1> or <eat_fruit = 1> >
  change score by 1
```

```
if < <eat_pill = 1> and <touch_ghost = 1> >
  change score by 1
```

Fig. 5. The Scratch programming tool.

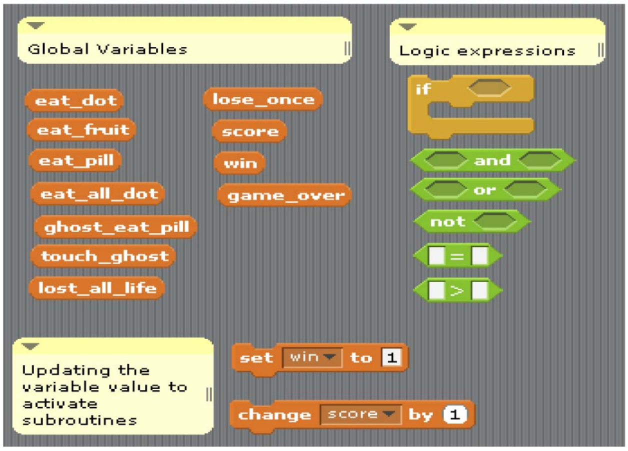


Fig. 6. The Scratch components for game rule tuning.

the WYSIWYG (what you see is what you get) interface of the Scratch programming tool allows students to easily compare the game scenario (the left part of Fig. 5) and game rules (the right part of Fig. 5).

To provide the game-rule-tuning activity, the events and subroutines of original *Pac-Man* game, such as *Ate_Power_Pill, Touched_Ghost, and Score etc.* are controlled by the corresponding global variables. For example, the game events of *Ate_Power_Pill* can be obtained from the global variable *eat_pill* in a period of time and the subroutines of *Score* can be activated by reconfiguring the global variable *score*. Therefore, the students can tune the logic expressions of game rules, using the Scratch components including global variables, if-statements, logical operations, and variable operations as shown in Fig. 6.

3.3 How to Use the Game to Teach the Boolean Logic

According to the learning context introduced above, the game-rule-tuning activities involving Boolean logic realization activity and real-world case formulation activity are proposed to teach the Boolean logic as shown in Fig. 7.

The process from right to left, i.e., the game rules of the *Pac-Man* game are provided in the Boolean logic expression, for example, *"If Ate_Power_Pill AND Touched_Ghost then Score*." Then, the quest of game rule tuning with new operation, such as "*changing logical AND to logical OR*" is given to students. Next, students can obtain the new game with new rule, *"If Ate_Power_Pill OR Touched_Ghost then Score"* in the modified Pac-Man game. Students can play the game and analyze the change of the difficulty degree and interesting degree of the game.

The process from left to right, the real-world case formulation activity, is similar to the process above. The only difference is that the quest in tuning the game rules stage provides new scenarios, such as *"tuning the game rule to allow the Pac-Man to get score when touching the ghost without eating the power pill."*

With the Pac-Man game rule defined in Definition 1, the compound conditions can be extended by various combinations of events to derive different difficulty levels of game-rule-tuning quests. In order to allow students realize the

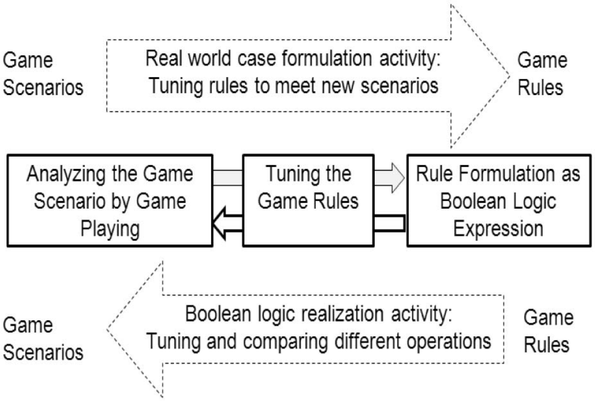

Game Scenarios
Real world case formulation activity: Tuning rules to meet new scenarios
Game Rules
Analyzing the Game Scenario by Game Playing
Tuning the Game Rules
Rule Formulation as Boolean Logic Expression
Game Scenarios
Boolean logic realization activity: Tuning and comparing different operations
Game Rules

Fig. 7. Game-rule-tuning activities.

complex logic expressions with meaningful game scenario, the derived quests are based on the scenarios of the surviving rules, scoring rules, and ending rules. Therefore, extending the logic expressions can be interpreted by the patterns of *adding more constrains with logical AND operation* and *providing more options with logical OR operation* in the game scenario. For example, the scoring rule can be modified from the original expression, *"If (Ate_Dot* OR *Ate_Fruit*), then *Score"* to the complex conditions: *"If ((Ate_Dot* AND NOT *Lose_once)* OR (*Lose_once* AND *Ate_Pill* AND *Ate_Dot*) OR (*Ate_Dot* AND *Ate_Fruit*)), then *Score."* The new scenario means *"the Pac-Man will not be able to eat the pac-dot to score while loses once except eating the pill or fruit to recover."* The complex conditions can be modeled and taught in the game.

It should be noticed that while providing the game-rule-tuning activity, the quests given to the students should be appropriately designed, since too many modifications may confuse the students and result in misinterpreting the effects of the Boolean logic operations. Therefore, the formal model of the game cases to support teacher measuring the different difficulty levels of game rules is proposed in the next section.

4 Modeling the Difficulty of Game-Rule-Tuning Activity

The difficulty of a game for students to learn the Boolean logic game rule tuning in here means how complex the rule expressions are. In other words, the more operators used in the expression, the more difficult it is. In the following, the formal expression of the rule and the rule set of a game are defined.

Definition 2 (Game Rule Boolean Logic Expression). *The game rule Boolean logic expression is a context-free grammar as follows: Let* $\sum = \{a_i|\forall i\}$ *be a set of terminal symbols representing the propositions of Boolean expression with S being the start symbol and N a nonterminal symbol. In each game rule, the implication operator* ($\rightarrow$) *appears once, where the left side represents the conditional propositions and the right side represents the game action propositions. Thus, the grammar of game rule Boolean logic expression R is as follows:* $S \rightarrow (N \rightarrow N), N \rightarrow N \wedge N \mid N \vee N \mid \neg(N) \mid a$.

Definition 3 (The Disjunctive Normal form of Game Rule). *A Boolean logic expression or a formula is in disjunctive normal form (DNF) if it is a disjunction of clauses, where a clause is a conjunction of literals, for example,* "$(X1 \wedge Y1) \vee (X2 \wedge Y2) \vee \cdots \vee (Xn \wedge Yn)$."

Example 1 (The Game Rule of Scoring in Context-Free Grammar). For the game rule, *"If (Ate_Pill AND Touched_Ghost) OR Ate_Fruit then Score,"* let P, T, F, and C annotate the facts *Pac-Man Ate_Pill, Touched_Ghost, Ate_Fruit,* and *player Scored,* respectively. Then, the Boolean logic expression of the game rule is represented as "$(P \wedge T) \vee F \rightarrow C$."

The property of every Boolean logic expression can be converted into an equivalent formula that is in DNF. With the defined grammar of game rule, the difficulty of Boolean logic expression can be defined as how many operators are used in the expression. The more operators used, the more difficult it is for students to learn. The definition of difficulty of Boolean logic expression is as follows:

Definition 4 (Difficulty of Boolean Logic Expression). *The difficulty measurement* $f : L(G) \rightarrow \mathbb{R}$ *is defined as* $f(e) = \sum w_i a_i$, *where* $e \in L(G)$,

- a_1 is the number of conjunction operators in the DNF of e,
- a_2 is the number of disjunction operators in the DNF of e,
- a_3 is the number of negation operators in the DNF of e, and
- a_i is the number of steps converting e into DNF.

The difficulty of a Boolean logic expression is computed via weighted sum of frequencies of distinct logic operators. It provides the flexibility for teachers, when teaching different concepts/subtopics to students, such as the usage and meaning of different operators. For example, when teaching the conjunction operator, the teacher can give the conjunction operator more weight. That is, by increasing the value a1.

To avoid the ambiguity between different forms of one propositional logic expression, the disjunctive normal form is used. For advanced topics covering derivations or equations, such as De Morgan's laws, distributive laws, and associative laws, the difficulties are involved in equations. However, our system did not support this kind of concept directly. Instead, we can use two games representing the left side and the right side, respectively. In these cases, we can also use DNF to measure its difficulty (complexity). For measuring the difficulty of eliminating double negation, we can estimate the converting process by counting the number of steps to DNF.

Definition 5 (Game Rule Case). *A game rule case is a set of Boolean logic expressions, which are modified from Pac-Man game rules.*

Definition 6 (Dissimilarity of Game Rule Cases). *The dissimilarity of two propositional logic expression e1 and e2 is defined as* $d(e_1, e_2) = \sum_{c1} \text{Min}\{\text{diff}(c1, c2) \mid \forall c1, c2, \text{ a clause of DNF of e1 and e2}\}$, *where* $\text{diff}(c_1, c_2)$ *is the number of different propositions between* c_1 *and* c_2. *Thus, the dissimilarity of two cases of games A, B is* $\sum \min\{d(a, b) \mid \forall\, a \in A, b \in B\}$.

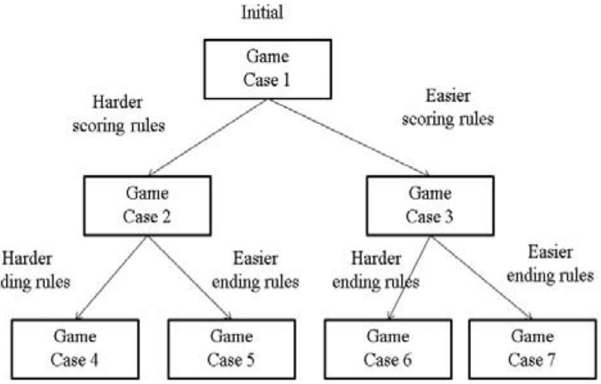


Fig. 8. Game cases for game-rule-tuning activities.

With the defined dissimilarity function, different game cases (Cases 2-7) with respect to the original game case (Case 1) can be ordered by the difficulties and dissimilarity as shown in Fig. 8. The closer to the Case 1, the easier the game-rule-tuning activity is. The computing of the dissimilarity of game rule cases is shown in Example 2.

Example 2 (Dissimilarity of Game Rule Cases). This example introduces the dissimilarity among Case 1, Case 2, and Case 3 in Fig. 6. First, the game rules of three cases are shown.

Case 1: The initial game rules represent the scenario that Pac-Man touched the Ghost will lose and ate all dots will win. The Pac-Man can score by eating the pill, fruit, or killing the Ghost after eating the pill.

- e1 : $Touched\_Ghost \rightarrow Lose\_once$
- e2 : Ate_Pill OR $Ate\_Fruit \rightarrow Score$
- e3 : Ate_Pill AND $Touched\_Ghost \rightarrow Score$
- e4 : $Ate\_All\_Dot \rightarrow Win$

Case 2: Modify e2 to $e2'$ and $e2''$ with two conjunction expressions. The new scenario is that the Ghost becomes weak when Ghost eats the power pill, and thus, the Pac-Man can kill the Ghost to score.

- e1 : $Touched\_Ghost \rightarrow Lose\_once$
- $e2'$: $Ate\_Fruit \rightarrow Score$
- $e2''$: $Touched\_Ghost$ AND $Ghost\_Ate\_Pill \rightarrow Score$
- e3 : Ate_Pill AND $Touched\_Ghost \rightarrow Score$
- e4 : $Ate\_All\_Dot \rightarrow Win$

Case 3: Modify e3 to $e3'$ with one disjunction expression. The new scenario is that eating the fruit can also kill the Ghost.

- e1 : $Touched\_Ghost \rightarrow Lose\_once$
- e2 : Ate_Pill OR $Ate\_Fruit \rightarrow Score$
- $e3'$: (Ate_Pill OR Ate_Fruit) AND $Touched\_Ghost \rightarrow Score$
- e4 : $Ate\_All\_Dot \rightarrow Win$

Let P, T, F, and A be the propositions of Ate_Pill, Touched_Ghost, Ate_Fruit, and *Ate_All_Dot*, respectively; L, C, and O are the propositions that players *Lose_Once, Score,* and *Win*; G denotes that *Ghost_Ate_Pill*. The transformation of expressions into DNF can replace the implication statement $(\mathrm{P} \vee \mathrm{F}) \rightarrow C$ with $\neg(\mathrm{P} \vee \mathrm{F}) \vee \mathrm{C}$ and remove the parentheses using the De Morgan's law: $(\neg\mathrm{P} \wedge \neg\mathrm{F}) \vee \mathrm{C}$. For some case, the distributive law should be further applied to obtain the DNF. The original case rules above can be converted into the disjunctive normal form of the corresponding Boolean logic expressions as follows:

Case 1: Initial

- $\mathrm{e}_1{:}\neg\mathrm{T} \vee \mathrm{L}$
- $\mathrm{e}_2 : (\neg\mathrm{P} \wedge \neg\mathrm{F}) \vee \mathrm{C}$
- $\mathrm{e}_3 : \neg\mathrm{P} \vee \neg\mathrm{T} \vee \mathrm{C}$
- $\mathrm{e}_4 : \neg\mathrm{A} \vee \mathrm{O}$

Case 2: Modify e_2 to $\mathrm{e}_{2'}$

- $\mathrm{e}_1{:}\neg\mathrm{T} \vee \mathrm{L}$
- $\mathrm{e}_{2'}{:}\neg\mathrm{F} \vee \mathrm{C}$
- $\mathrm{e}_{2''}{:}\neg\mathrm{T} \vee \neg\mathrm{G} \vee \mathrm{C}$
- $\mathrm{e}_3{:}\neg\mathrm{P} \vee \neg\mathrm{T} \vee \mathrm{C}$
- $\mathrm{e}_4{:}\neg\mathrm{A} \vee \mathrm{O}$

Case 3: Modify e_3 to $\mathrm{e}_{3'}$

- $\mathrm{e}_1{:}\neg\mathrm{T} \vee \mathrm{L}$
- $\mathrm{e}_2 : (\neg\mathrm{P} \wedge \neg\mathrm{F}) \vee \mathrm{C}$
- $\mathrm{e}_{3'} : (\neg\mathrm{P} \wedge \neg\mathrm{F}) \vee \neg\mathrm{T} \vee \mathrm{C}$
- $\mathrm{e}_4 : \neg\mathrm{A} \vee \mathrm{O}$

For examples, the dissimilarity between e_2 and $\mathrm{e}_{2'}$ are computed as

$$\begin{aligned}\mathrm{d}(\mathrm{e}_2, \mathrm{e}_{2'}) &= \min\{\mathrm{diff}(\neg\mathrm{P} \wedge \neg\mathrm{F}, \neg\mathrm{F}),\\ &\quad \mathrm{diff}(\neg\mathrm{P} \wedge \neg\mathrm{F}, \mathrm{C})\} + \min\{\mathrm{diff}(\mathrm{C}, \neg\mathrm{F}), \mathrm{diff}(\mathrm{C}, \mathrm{C})\}\\ &= \min\{1, 3\} + \min\{2, 0\} = 1 + 0 = 1.\end{aligned}$$

Based on this dissimilarity measure of two Boolean logic expressions, we can compute the dissimilarity of two cases of games by Definition 6. The dissimilarity between Case 1 and Case 2 is 1 and the dissimilarity between Case 2 and Case 3 is 6.

5 Evaluation of the Game-Rule-Tuning Activity

This section shows the experimental results of evaluating the learning achievement between students with game-rule-tuning activity and students with traditional lecturing. The discussions and findings of the game-rule-tuning activities are provided as follows:

The teaching scope of the experiment is the Boolean logic expressions including conjunction $x \wedge y$ (AND), disjunction $x \vee y$ (OR), and negation $\neg x$ (NOT) and implication $x \rightarrow y$. This study compared two educational applications on Boolean logic realization concepts that were identical in terms of learning objectives, and differed in that the experimental group applied the game-rule-tuning activity, whereas the control group followed the traditional activity of drawing truth table and logic gate. Based on the overview above, there are two hypotheses for this experiment as follows:

1. The pretest and posttest of students' scores have significant improvement in experimental group.
2. The posttest scores of students in experimental group have significantly greater achievements than students in control group.

TABLE 1
Students Joined the Experiment

Groups	Size	Term	Boys / Girls
Control group	27	2009-2010	12 / 15
Experimental group	43	2009-2010	20 / 23

To evaluate the hypotheses that we made, there are 70 random selected ninth Grade students (14-15 years old) in Minzu Junior High School, Taipei, Taiwan, who participated in the experiment in semester 2009-2010. There are 27 students in control group and 43 students in experimental group. It should be noticed that initially, there were also 43 students in control group who participated in the pretest. Among them, 16 students missed the posttest due to attending another school activity. Therefore, these students are excluded from the experimental result analysis. The 70 experiment samples and their background are shown in Table 1.

Based on the concept of logical-mathematical intelligence mentioned by Gardner [29], the mathematic ability and logic ability are usually applied together for investigating issues logically. To make sure that students selected in two groups have the same level of logic ability, the pretest has been applied. The pretest results show that students in two groups, which have the same level of mathematical scores also have the same level of logical ability.

The experimental tool of the learning achievement evaluation is the self-designed test sheet for Boolean logic realization. As shown in Table 2, there are three sections of test items with scenarios of *Pac-Man rule*, *poker game*, and *school regulation*. Students should answer true or false and explain correctly for the statements. The test sheet is applied in pretest and posttest without giving students answers after the tests.

5.1 Experiment Evaluations

In the evaluation of students in the control group and the experimental group, the comparison of learning improvements and the comparison of learning outcomes have been conducted.

5.1.1 Comparison of Learning Improvements

The scores and the evaluations of the learning improvements between pretest and posttest are shown in Table 3. To evaluate the difference between two groups, the paired two-sample t-test for means of scores is applied. A statistically significant score is found for the pretest and posttest scores in experimental group with $\mathrm{t}(42) = 3.5391$ and $\mathrm{p} = 0.0010$. For the control group, the scores of pretest and posttest are not found to be statistically significant with $\mathrm{t}(26) = 0.2934$, $\mathrm{p} = 0.7715$. Consequently, the t-test results of the experimental groups suggest significant differences but the

TABLE 2
The Test Sheet for Boolean Logic Realization

Please Answer True (O) or False(X) and explain your answer	Answer	Reason
Let P: *Ate_Pill* **, T:** *Touched_Ghost*, **F:** *Ate_Fruit*, **A:** *Ate_All_Dot*, **L:** *Lose_once*, **S:** *Score*, **W:** *Win* **Rule 1:** $\mathbf{T \rightarrow L}$ **Rule 2:** $\mathbf{A \vee F \rightarrow S}$ **Rule 3:** $\mathbf{P \wedge T \rightarrow S}$ **Rule 4:** $\mathbf{A \wedge \neg L \rightarrow W}$		
T_1. If the Rule 2 is rewritten as **“$A \wedge F \rightarrow S$”** then it is easier to score.	X	
T_2. If the Rule 4 is rewritten as **“$(A \wedge \neg L) \vee P \rightarrow W$”** then it is more difficult to win.	X	
T_3. If the Rule 3 is rewritten as **“$P \wedge T \wedge F \rightarrow S$”** then it is more difficult to score.	O	
T_4. Although Pac-Man touched the ghost, it will win after eating all dots.	X	
T_5. If the Rule 3 is rewritten as **“$P \vee T \rightarrow S$”** then it is easier to win.	X	
The poker game: a 52-card deck is equally divided into face-down stacks for all players. Each player take turns to place the card face-up to the playing surface and announce the number from 1 to 13 in order. While “the announcing number is the same with card number”, all players should slap this card and the slowest one lose this round.		
T_6. Rewriting the rule as “**the announcing number is the same with card number AND the next player’s announcing number is the same with card number**” will have more chance to slap the card.	X	
T_7. Rewriting the rule as “**the announcing number is the same with card number OR the announcing number is the same with last player’s card number**” will have more chance to slap the card.	O	
Let P: *Have tissue* **, H:** *Have hat*, **B:** *Have Book*, **K:** *Be Punished* **The school ask students to follow the regulation: “$\neg P \vee \neg H \vee \neg B \rightarrow K$”**		
T_8. If the regulation is rewritten as $\mathbf{\neg (P \wedge H \wedge B) \rightarrow K}$, then the regulation becomes stricter.	X	
T_9. If the regulation is rewritten as $\mathbf{\neg P \wedge \neg H \wedge \neg B \rightarrow K}$, then the regulation becomes more lenient.	O	
T_{10}. If the regulation is rewritten as $\mathbf{P \wedge H \wedge B \rightarrow \neg K}$, then the regulation becomes stricter.	X	

control group has no significant differences. Finally, we may conclude that only the experimental group has significant learning improvement.

5.1.2 Comparison of Learning Outcomes

The evaluations of the learning achievements of posttest are shown in Table 4. To evaluate the difference between two groups, the unpaired two-sample t-test for means of scores is applied. The t-test result for pretest shows that F-value is 1.35117 and p-value is 0.3774. Consequently, the t-test results of the two groups suggest no significant differences for scores of pretest at a confidence interval of 95 percent. The t-test result for posttest shows that F-value is 1.56083 and p-value is 0.01946. Consequently, the t-test results of the two groups suggest significant differences for the scores of posttest at a confidence interval of 95 percent. Finally, we may conclude that the experimental group has higher learning achievement than control group.

5.2 Findings and Discussion

From the experiments, the students with game-rule-tuning activity showed higher achievements for the Boolean logic realization. With our further observation, it seems that the students in experimental group performed better results in the test items of scenario *poker game* and *school regulation*. It supports that the students with game-rule-tuning activity may have better capability to apply the Boolean logic to the new problems. From the teacher’s feedback, the students in experimental group surprisingly engaged in the activity and had many discussions with peers and with their teacher.

TABLE 3
The Paired t-Test of Pretest and Posttest Scores for Control Group and Experimental Group

	Pretest		Posttest			
	Mean	**Std. Div.**	**Mean**	**Std. Div.**	**Mean diff.**	**Paired t-test**
Control group (size=27)	54.44	20.82	55.56	20.82	1.11	t(26)=0.2934, p=0.7715
Experimental group (size=43)	55.35	17.91	65.58	16.66	10.23	t(42)=3.5391, p=0.0010*

*p<.05

TABLE 4
The Two-Sample t-Test of Pretest and Posttest Scores for Control Group and Experimental Group

	Pretest		Posttest	
	Mean	**Std. Div.**	**Mean**	**Std. Div.**
Control group (size=27)	54.44	20.82	55.56	20.82
Experimental group (size=43)	55.35	17.91	65.58	16.66

Two-sample t-test	F=1.35117,
	p-value=0.3774
			F=1.56083,
			p-value=0.01946*	

*p<.05

In summary, our findings of applying the game tuning activity in the game-based learning can be concluded in following points:

- Although the game-based learning is interesting for students, the game is additive for students. Therefore, the teacher should control the progress of the teaching stages and provide a clear and well-designed learning goal.
- Since teaching the realization of Boolean logic is our main objective, choosing the well-known game with simple game scenario is better for students to quickly catch the point.
- The learning materials and learning platform should be prepared for students to avoid wasting time in getting familiar or while installing the preliminarily used tools.
- Understanding the application of Boolean logic affecting the interesting games is surprisingly attractive for students to actively engage in the learning activities.

Currently, the main challenge of the game-based learning or serious game design focused on how to connect the attractive game play to the pedagogical objectives. In our opinion, the game-based learning is beyond entertaining for the students. For the proficient players, the game playing of reasoning for how to follow the game rules to make a best decision in specific game status is also an exciting and interesting property of the game. This property is suitable for logic training. Through this work, we have some suggestions of applying game to learning. For the game platform selection, the game should be easy to manipulate and the factors of the game should be easy to observe. For the reinterpretation of the game, teachers should connect the game scenarios, roles, and rules with the knowledge in order to apply the pedagogical theory to the game-based learning activity. For activity control, the teacher should be able to control the learning progress by dividing the game playing into several rounds, and thus, the explanation can be provided before each round. For the activity design, the learning objectives of each activity are better to be limited in one or two specific concepts.

The effectiveness of the game-based learning for realization of basic operations of Boolean logic has been shown in our experiment. Currently, the Pac-Man game-rule-tuning activity only supports the learning of simple Boolean logic expressions for junior high school students. In the near future, we will further apply the game-rule-tuning activity for the complex logic expressions or higher order logic with quantifiers (existential and universal). Since the key point of game-based learning is the selection of the suitable game type, the simulation games or strategy games which contain more criteria for decision making can be selected as the game context for the learning of complex logic expressions.

ACKNOWLEDGMENTS

The authors wish to thank Kuan-Ying Huang of Minzu Junior High School, Taipei, Taiwan. This work was partially supported by the National Science Council of the Republic of China under contracts NSC 97-2511-S-468-004-MY3 and NSC 98-2511-S-468-004-MY3.

REFERENCES

[1] Binary Color Device from Cut-the-Knot, http://www.cut-the-knot.org/pythagoras/bicolor.shtml, Mar. 2010.

[2] Logic Puzzle from Puzzler Paradise, http://www.puzzlers paradise.com/page1034.html, Mar. 2010.

[3] Wolf Sheep & Cabbage from Plastelina Interactive Logic Games, http://www.plastelina.net/game1.html, Mar. 2010.

[4] Boolean Logic on Wikipedia, http://en.wikipedia.org/wiki/Boolean_algebra_(logic), Mar. 2010.

[5] Virginia Tech Computer Science Online Course, http://courses.cs.vt.edu/~csonline/MachineArchitecture/Lessons/Gates/index.html, Mar. 2010.

[6] D. Nute, Univ. of Georgia, Exercises for Venn Diagramming, http://donald.nute.ws/PHIL1500/venn/index.html, Mar. 2010.

[7] E.T.H. Luk, M.K.H. Wong, K.K.F. Cheung, F.L. Lee, and J.H.M. Lee, "Design and Implementation of Farmtasia: A Game Designed for the VISOLE Teaching Style," *Proc. First Int'l Conf. Technologies for e-Learning and Digital Entertainment*, pp. 566-571, 2006.

[8] J. Shang, M.S.Y. Jong, F.L. Lee, and J.H.M. Lee, "A Pilot Study on Virtual Interactive Student-Oriented Learning Environment," *Proc. First IEEE Int'l Workshop Digital Game and Intelligent Toy Enhanced Learning*, 2007.

[9] J. Tan and G. Biswas, "Simulation-Based Game Learning Environments: Building and Sustaining a Fish Tank," *Proc. First IEEE Int'l Workshop Digital Game and Intelligent Toy Enhanced Learning,* pp. 73-80, 2007.

[10] G. Sindre, L. Natvig, and M. Jahre, "Experimental Validation of the Learning Effect for a Pedagogical Game on Computer Fundamentals," *IEEE Trans. Education,* vol. 52, no. 1, pp. 10-18, Feb. 2009.

[11] M. Papastergiou, "Digital Game-Based Learning in High School Computer Science Education: Impact on Educational Effectiveness and Student Motivation," *Computers and Education,* vol. 52, no. 1, pp. 1-12, 2009.

[12] H.C. Jiau, J.C. Chen, and K.F. Ssu, "Enhancing Self-Motivation in Learning Programming Using Game-Based Simulation and Metrics," *IEEE Trans. Education,* vol. 52, no. 4, pp. 555-562, Nov. 2009.

[13] G. Lukas, "Uses of the LOGO Programming Language in Undergraduate Instruction," *Proc. ACM Ann. Conf.,* vol. 2, pp. 1130-1136, 1972.

[14] R. Pattis, J. Roberts, and M. Stehlik, *Karel the Robot: A Gentle Introduction to the Art of Programming,* second ed. John Wiley & Sons, 1994.

[15] J. Bergin, M. Stehlik, J. Roberts, and R. Pattis, *Karel++: A Gentle Introduction to the Art of Object Oriented Programming.* John Wiley & Sons, 1997.

[16] J. Bergin, M. Stehlik, J. Roberts, and R. Pattis, "Karel J. Robot: A Gentle Introduction to the Art of Object Oriented Programming in Java," Unpublished Manuscript, http://csis.pace.edu/~bergin/KarelJava2ed/Karel++JavaEdition.html, Sept. 2003.

[17] B.W. Becker, "Teaching CS1 with Karel the Robot in Java," *Proc. ACM SIGCSE,* vol. 33, pp. 50-54, 2001.

[18] D. Buck, D.J. Stucki, "JKarelRobot: A Case Study in Supporting Levels of Cognitive Development in the Computer Science Curriculum," *Proc. ACM SIGCSE,* vol. 33, pp. 16-20, 2001.

[19] M. Kölling, B. Quig, A. Patterson, and J. Rosenberg, "The BlueJ System and Its Pedagogy," *J. Computer Science Education,* special issue on learning and teaching object technology, vol. 13, no. 4, pp. 249-268, Dec. 2003.

[20] Greenfoot, http://www.greenfoot.org, Mar. 2010.

[21] M. Külling and P. Henriksen, "Game Programming in Introductory Courses with Direct State Manipulation," *Proc. ACM SIGCSE Conf. Innovation and Technology,* pp. 59-63, 2005.

[22] S. Cooper, W. Dann, and R. Pausch, "Teaching Objects-First in Introductory Computer Science," *Proc. ACM SIGCSE,* pp. 191-195, 2003.

[23] P. Henriksen, "A Direct Interaction Tool for Object-Oriented Programming Education," master's thesis, Maersk Mc-Kinney Moller Inst. for Production Technology, Univ. of Southen Denmark, 2004.

[24] Scratch, http://scratch.mit.edu, Mar. 2010.

[25] Scratch Programming Handbook, http://www.redware.com/scratch, Mar. 2010.

[26] M. Resnick, "Sowing the Seed for a More Creative Society," *Learning and Leading with Technology,* vol. 35, no. 4, pp. 18-22, 2007.

[27] Alice, http://www.alice.org, Mar. 2010.

[28] C. Kelleher and R. Pausch, "Using Storytelling to Motivate Programming," *Comm. ACM,* vol. 50, no. 7, pp. 58-64, 2007.

[29] H. Gardner, *Frames of Mind: The Theory of Multiple Intelligences.* Basic Books, 1993.

Jui-Feng Weng received the BS and MS degrees from the Department of Computer and Information Science, National Chiao Tung University, Taiwan, in 2000 and 2002, respectively. He is currently working toward the PhD degree at the Department of Computer Science, National Chiao Tung University, Taiwan. His current research interests include e-learning, knowledge engineering, expert systems, and data mining.

Shian-Shyong Tseng received the PhD degree in computer engineering from National Chiao Tung University in 1984. From 1983 to 2009, he was on the faculty of the Department of Computer and Information Science at National Chiao Tung University. From 1991 to 1992 and 1996 to 1998, he acted as the chairman of the Department of Computer and Information Science. From 1992 to 1996, he was the director at the Computer Center, Ministry of Education, and the chairman of Taiwan Academic Network (TANet) Management Committee. In December 1999, he founded the Taiwan Network Information Center (TWNIC) and was the chairman of the board of directors from 1999 to 2005. He was the dean in the College of Computer Science, Asia University, from 2005 to 2008. He is currently a vice president of ASIA University and the chairman of the board of directors of TWNIC. He is an editor-in-chief of the *International Journal of Digital Learning Technology* and an editor of the *International Journal of Fuzzy Systems*, the *Journal of Internet Technology*, and the *International Journal of Computational Science*. He is also a co-editor-in-chief of the *Asian Journal of Health and Information Science*. He was named an Outstanding Talent of Information Science of the Republic of China in 1989. He obtained the 1992, 1994, and 1995 Outstanding Research Awards of the National Science Council of the Republic of China. He was the winner of the 1990, 1991, 1998, and 2000 Acer Long Term Awards for outstanding MS Thesis Supervision and the winner of 1992 and 1996 Acer Long Term Awards for outstanding PhD Dissertation Supervision. He was also awarded the Outstanding Youth Honor of R.O.C. in 1992. His current research interests include expert systems, data mining, computer-assisted learning, and Internet-based applications. He has published more than 100 journal papers. He is a member of the IEEE and Phi Tau Phi Societies.

Tsung-Ju Lee received the BS degree in mathematics from TungHai University in 2000 and the MS degree in applied mathematics from National Chiao Tung University, Taiwan, in 2002. He is currently working toward the PhD degree in the Department of Computer Science, National Chiao Tung University, Taiwan. His current research interests include machine learning, data mining, and various applications, especially in network security, e-learning, and software testing.