

# Effects of Cognitive Styles on Computational Thinking and Gaming Behavior in an Educational Board Game

Jung-Chuan Yen <sup>\*</sup>, Wei-Chi Liao <sup>\*</sup>

## Abstract

The aim of the study was to examine the effects of different cognitive styles on the elementary students' computational thinking and gaming behavior in an educational board game for computer programming. This quasi-experimental design study lasted four weeks with 3 hours per week. The subjects were 25 field-independent and field-dependent 6th grade students (assessed by the Group Embedded Figures Test) who participated in a programming foundation course. An ANCOVA and U-test analysis were performed on the definitive test data. The results showed that the field-independent learners achieved significantly improved computational thinking over field-dependent learners in the educational board game, and although no significant difference was found in gaming behavior between them, the field-independence group demonstrated more learning behavior related to the execution of complex thinking. This study suggests to provide differentiated instruction for learners of different cognitive styles, should be more to enhance the effectiveness of programming performance and positive gaming behavior.

*Keywords:* cognitive styles, programming education, computational thinking, gaming behavior

## 1 Introduction

In recent years, elementary schools in Estonia, Singapore, Austria, the United Kingdom, and the United States have started to provide computer programming courses aimed at equipping students with computational thinking. Many other countries have followed suit by incorporating computer programming into their educational policies; this makes the subject matter increasingly popular for research in primary school information technology education.

In Taiwan, information technology education has been reformed several times, with its curricular focus shifting from operation-related content to higher-order thinking skills [1]. The curricular reform underwent four stages of orientation: programming skills; software and hardware application; problem-solving and computer science; and computational thinking. Computational thinking is one of the goals of the curriculum of 12-year compulsory education scheduled to be implemented in Taiwan in 2019. Under the curriculum, students will learn to apply this thought process and information technologies to solve problems, work as a team, and express themselves. All these skills are in line with the core competencies that the curriculum is designed to

---

<sup>\*</sup> National Taipei University of Education, Taipei, Taiwan

promote: systematic thinking; problem-solving; planning, execution, and innovation; symbol utilization and communication; and technology, information, and media literacy.

Programming course is designed to teach students how to use programming language algorithms and design techniques to solve specific problems, allowing them to develop the logical and abstract thinking skills and, therefore, cognitive schemas for problem-solving in different fields [2], [3]. With the rapid development of information technology, the paradigms and learning models of programming languages are increasingly varied. However, what exactly should be taught in the elementary school programming courses? Numerous researchers and experts have argued that rather than introducing programming languages, structures, and design methods, such courses should focus on hands-on skills including decomposition, pattern recognition, abstraction, and algorithms to equip students with logical and problem-solving skills [3]. In other words, the literacy about how to use information technologies to solve problems in daily life, named as computational thinking, is considered to be the focus that should be taught [4], [5]. Based on this argument, current programming education features the use of robotics, toys, board games, and maker practices in its curriculum to provide higher-order thinking skills has become more and more popular.

A cognitive style refers to the model that a person typically uses to process cognitive tasks, such as perception, memorization, thinking, and problem-solving, and it can be seen as encompassing the personality traits related to comprehension, memorization, and cognitive processing [6]. In the context of programming education, cognitive styles may differ in terms of learning outcomes, problem-solving strategies, transfer of learning, design and presentation of teaching materials, and learning behavior.

Based on the aforementioned rationale, this study investigated the effects of cognitive styles on the computational thinking and gaming behavior of learners participating in board games for programming education. Specifically, this study addressed two objectives:

- (1) To compare learning outcomes in board games for programming education between learners with different cognitive styles (i.e., field dependence vs. field independence); and
- (2) To investigate differences in gaming behavior (i.e., the total number of cards played, the number of bugs, the number of tactic cards played, and the number of programming cards played) in board games for programming education between field-dependent and field-independent learners.

## 2 Literature Review

### 2.1 Programming education with computational thinking

As specified in the elementary information literacy curriculum for Taiwan's upcoming 12-year compulsory education system, programming education is aimed at familiarizing students with typical information systems, equipping them to solve simple everyday problems through information technologies, teaching them to logically describe approaches to problem-solving, and providing courses that cover the functions and uses of programming tools and the fundamentals of programming.

Bayman and Mayer [7] divided program design knowledge into conceptual and strategic knowledge. With strategic knowledge, people can develop problem-solving abilities by analyzing problems and transforming problem-solving steps into algorithms. Moreover, by learning programming on the basis of familiar problems, learners can develop problem-solving abilities that are transferable to other subjects [8]. Papert suggested that programming teachers teach students how to think, rather than what to think [9]. Resnick et al. who developed Scratch (a

programming language popular in elementary schools), suggested that schoolchildren can learn programming in their daily lives through interactions, games, sharing their creations, or social occasions that allow to them to solve problems [10].

The concept of “computational thinking” has become popular with the view that claimed by Wing [2] which is “one of the daily life skills that everyone needs, rather than just being a programming skill used only by computer scientists.” However, there is still no consensus on this definition so far. Later on Wing [11] have developed the definition and expressed computational thinking as a thinking process that includes expression of the solutions and how to solve problems effectively with computers. ISTE and CSTA [12] supports this statement and further interprets computational thinking skill as a reflection of algorithmic thinking, creative, logical thinking and problem solving skills. In other words, the skill of computational thinking meaning that it not only lets the computer understand the way to solve problems, but also helps people to understand the solutions and problems more deeply. Therefore, most countries in the world regard computational thinking as the main practice and innovation connotation of K-12 science education.

## **2.2 Cognitive styles**

A cognitive style is used to process cognitive tasks, and it comprises the personality traits related comprehension, memorization, and cognitive processing [13]. Cognitive style also represents the characteristic mode of functioning shown by individuals in their perceptual and thinking behavior during the decision-making process [14]. In the field of educational psychology research, cognitive styles can be categorized into field dependence and field independence [15]. A field-dependent person is strongly affected by an organizational field, and his/her cognitive style is vulnerable to external stimuli or visual structures that make it difficult to discriminate between cognitive objectives in a situation. A field-independent person can use a cognitive mechanism to identify objectives in a complex, obscure, or ambiguous situation [16]. Differentiating between field independence and field dependence is normally a continuous phenomenon with no absolute boundaries; it is a relative measure of difference.

Students with different cognitive styles have many differences in the performance of subject learning. Research suggested that students who are field-dependent succeed better with socially oriented learning tasks, for example cooperative learning whilst field-independent students rather work on abstract and less socially oriented assignments [17]. An empirical study discovered that field-dependent learners scored lower than field-independent learners in a condition mismatched with their preferred manipulation [18]. The research outcome conducted by Lee showed that field-independent learners tended to be internally driven in contrast to field-dependent learners who rely on the external forces to perform a task [19]. On the other hand, field-dependent learners’ performance deteriorated when received an instructional strategy that contradicted with their cognitive styles. Therefore, teaching to suit students’ cognitive styles is essential with regard to learning effectiveness.

In summary, this study assumes that cognitive style may affect students' learning outcomes, such as computational thinking, problem-solving strategies, and exhibit different gaming behaviors in the process of playing the educational board game.

## 2.3 Computer Science Unplugged

Computers are not the only means of developing programming skills. Drawing on their extensive experience in programming education, a group of scholars claimed that computers may be the reason for distracting students [20]. Many essential concepts of computer science are related to problems encountered in daily life, and students can learn about computer science away from the computer. They proposed a series of “unplugged (not computer-mediated)” activities which use games, magic tricks and competitions to show children the kind of thinking that is expected of a computer scientist. In recent years, unplugged learning activities for computer science have gradually been valued by grassroots teachers. In addition to using paper and pencil, they also developed teaching strategies that combine plugged and unplugged approach [21]. Among these, educational programming board games with similar characteristics have also become increasingly popular.

The board games, which comprise not only educational programming content but also gaming elements, provide students with knowledge of programming as well as gaming enjoyment. In Taiwan, there are many popular programming board games; these included Robot Turtles and Bee-Bot, which introduce simple programming processes into the games and are suitable for children of preschool age; and Code Monkey Island and Code Master, which teach the concepts of sequences, loops, conditional operators, and logical operators to children aged 8 years or older. Something these four programming board games have in common, we believe, is providing the fundamental concepts of programming through gaming, allowing students to solidify their thinking and develop reasoning and problem-solving abilities. Researchers suggested that exposure to learning content embedded in a board game can improve students’ learning motivation, prompting them to complete learning exercises repeatedly and unconsciously during the game, gain knowledge unknowingly, and enhance their learning outcomes [22].

## 3 Methods

### 3.1 Participants

This study recruited Grade 6 students from an elementary school in Taipei, Taiwan. A total of 25 students were enrolled after excluding those who did not meet the inclusion criteria. Of the participants, 12 were boys (48%), and 13 were girls (52%).

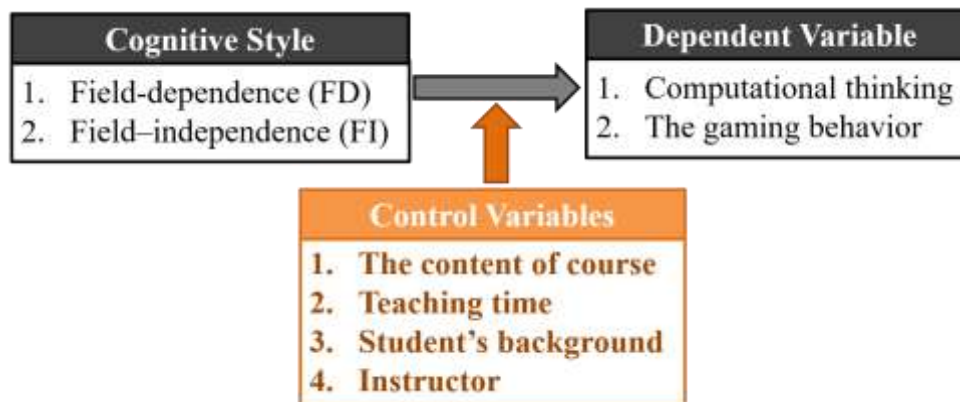


Figure 1: The research framework of this study

A quasi-experimental design was adopted to divide the participants into field-dependence and field-independence groups. The participants partook in a 2-week experimental course based on a programming board game; the course spanned 3 hours per week. The dependent variables in the study design were the pretest and posttest scores of participants' knowledge of programming and the gaming behavior they adopted when learning about programming. Figure 1 depicts the research framework of this study.

### 3.2 Research Procedure

In this quasi-experimental design, all participants were instructed by the author. Before the experiment began, the Group Embedded Figures Test (GEFT) was administered to the participants. The participants were divided into field-dependence and field-independence groups according to the test results. The participants subsequently took a pretest of their prior knowledge and computational thinking of programming. The teaching material used in the experimental course was "King of Pirates," an educational programming board game. The course comprised five lessons, totaling 200 minutes. After the course ended, the participants took a posttest of their computational thinking. Figure 3 shows the situation that all students are playing the game intently.



Figure 2: All students are playing the game intently

### 3.3 Research Instruments

The instruments used in this study include the GEFT tests, pretest and posttest of knowledge and computational thinking of programming, educational programming board game, and self-developed game record form. The GEFT, comprised of 18 complex figures, measures the cognitive style "field independence". Within each complex figure is embedded a previously seen simple figure. The participants' task was to identify the embedded figures that were hidden in the complex figures. A score was obtained by counting the number of correctly traced embedded figures. Scores range from 0 to 18, with higher scores indicating more field-independence. The GEFT mean score was employed as the cutting point for field-dependence / field-independence group classification (Cronbach's  $\alpha = .82$ ) [23].

To compare learning outcomes of the educational programming board game between participant groups, pretests and posttests were designed to measure students' computational thinking of programming. The tests comprised fill-in-the-blank questions, which helped to eliminate biases caused by guesswork. Both tests, which were reviewed by an academic specializing in information education and two elementary school senior teachers, had expert validity.

The educational programming board game used in this study was King of Pirates, developed

by Papacode (official website: <http://www.papacode.com.tw/>). This board game, which features scenarios of pirates competing for treasure and covers the fundamentals of programming, makes programming education entertaining and equips learners to think logically. Players in the game are required to use movement and magic cards; in the process, they learn the basics of programming such as sequence, loops, logical operators, and conditional operators. Figure 3 shows the ocean map and all cards used in the game. Moreover, the game is made more enjoyable through the use of tactic cards such as “increased whirlpools” and “suspend another player for one round.”



Figure 3: Educational Programming board game - King of Pirate

The game record form used in the game was designed by the author. The form recorded the player code, types of card played, locations of bugs, and final location of the pirate ship. During the game, the player preceding the dealer was required to complete the game record form for the dealer, filling the form with the card codes according to the cards played. At the end of the game, the author examined each participant’s game record form to calculate the total number of cards played, the number of bugs, the number of tactic cards played, and the number of programming cards played (Figure 4). Afterwards, limited by the sample size and do not follow the normal distribution, a non-parametric statistics was employed to examine participants’ gaming behavior.

Game-Play Record Statistic

Group	No.	Cognitive Styles	Cards Played	Bugs	Tactic Cards Played	Computational Thinking Programming Cards Played		
						Loops	Logical Operators (And/Or)	Conditional Operators (If/else)
1	S01	Field-independence	9	4	2	2	0	4
	S02	Field-dependence	8	2	1	0	2	0
	S03	Field-dependence	8	0	3	2	0	2
	S04	Field-independence	8	3	2	3	1	1

Figure 4: The self-developed game record form used in this study

## 4 Results and Discussion

The purpose of this study was to examine the effects of different cognitive styles on the ele-

mentary students' computational thinking and gaming behavior (i.e., total number of cards played, number of bugs, number of tactic cards played, and number of programming cards played) in the educational programming board game. Data collected in the game were examined through an analysis of covariance (ANCOVA) to yield statistics on learning outcomes. An independent-sample U test was used to assess their gaming behavior. The results of the aforementioned analyses are discussed as follows.

#### 4.1 Descriptive statistics for the research variables

Table 1 summarizes the descriptive statistics of field-independent and field-dependent participants' learning outcomes and gaming behavior in the programming board game. In both pretests and posttests, the field-independence group (mean pretest score: 36.92 of 100; mean posttest score 69.62 of 100) outperformed the field-dependent group (mean pretest score: 28.33; mean posttest score 49.58), likely because field-independent learners tend to immerse themselves in learning, have greater intrinsic motivation, and learn more efficiently [16].

Table 1: Descriptive statistics for the research variables

Learning Outcomes	FD(N=12)		FI(N=13)		ALL(N=25)	
	M	SD	M	SD	M	SD
<i>Computational Thinking</i>						
Pretest	28.33	24.80	36.92	26.89	32.80	25.22
Posttest	49.58	28.64	69.62	12.49	60.01	23.15
<i>Gaming Behavior</i>						
Total number of cards	9.44	4.64	9.22	3.83	9.33	4.01
Num. of bugs	0.89	1.45	1.67	1.32	1.28	1.37
Num. of tactic cards	1.78	1.85	1.78	1.09	1.78	1.44
Num. of program cards	2.00	1.87	3.78	3.07	2.89	2.56

#### 4.2 Field-independent students have better learning performance

An ANCOVA was performed on posttest scores, with pretest scores used as a covariate to control for the influence of participants' prior knowledge. An intragroup regression coefficient homogeneity test was performed on the pretest scores of both participant groups to ensure no significant difference was present. The result shows that the  $F$  value was 2.939, whereas the  $p$  value was .101 ( $> .05$ ), indicating no significance and accepting the null hypothesis. Therefore, the covariate (the pretest score) and the dependent variable (the posttest score) corresponded to the hypotheses of the intragroup regression coefficient homogeneity test for ANCOVA, prompting ANCOVA to be further conducted.

Next, a one-way ANCOVA was conducted to assess differences in the posttest score between participant groups (Table 2). By controlling for the influence of the pretest score, significant differences were found in the posttest score between the groups ( $F = 4.480, p = .046 < .05$ ). After post hoc comparison, the results suggested significant improvements in learning outcomes on computational thinking for field-independence group, with a mean deviation of 16.18.

Table 2: One-way ANCOVA of posttest of cognitive style students' computational thinking

Source	SS	df	MS	$F$	$p$	$\eta^2$
Covariance (pretest)	3106.36	1	3106.36	8.770	.007	.285
Between group effect	1586.22	2	1586.22	4.480	.046*	.169

In-group (error)	7789.63	22	354.07
Total	103400.00	25	

\* $p < .05$

Based on the statistical results, the use of board games as a teaching material for programming courses can yield noticeable learning outcomes in learners with different cognitive styles. Previous studies have indicated that in complex gaming scenarios, field-independent learners have stronger learning motivation and are more capable of acquiring clues to solving problems, and more likely to participate in learning activities, generally have higher intrinsic motivation and show more efficient learning outcomes [24], this proposition is supported by the results of this study. However, because King of Pirates both introduces programming concepts and exhibits board game elements, further research should be conducted to ascertain whether the concepts or the gaming elements is attributable for the improved learning outcomes achieved by the field-independent participants in this study.

### 4.3 Field-independent students seem showing more computational thinking and more trial-and-error behavior

Table 3 shows the non-parametric statistics (Mann-Whitnet U test) results of students' gaming behavior between the participant groups.

Table 3: The Mann-Whitnet U test result of students' gaming behavior

Gaming Behavior	FD (N=12)		FI (N=13)		Significances	Summary
	M	SD	M	SD		
Total number of cards	9.44	4.64	9.22	3.83	.976	FD≈FI
Num. of bugs	0.89	1.45	1.67	1.32	.190	FD<FI
Num. of tactic cards	1.78	1.85	1.78	1.09	.666	FD≈FI
Num. of program cards	2.00	1.87	3.78	3.07	.222	FD<FI

The groups showed no significant difference in gaming behavior (Table 3); yet, there were noteworthy findings about their gaming behavior. First, there existed slight differences in the total number of cards played and number of tactic cards played between the groups. This was either because field-independent and field-dependent participants showed no marked differences in their game involvement (the total number of cards played) and gaming tactics (the number of tactic cards)

Second, the programming cards used in the game "IF... Else," "And/Or," and "Loop" served to enhance players' knowledge of programming, and players who used these three cards could gain an advantage. Moreover, players who used common cards (e.g., "Move Forward," "Move Backward," "Turn Right," and "Turn Left") were more likely to encounter a lower number of bugs (the number of bugs related to the number of points deducted for logic errors in the combination of cards that occurred when a player used cards to advance toward a target). Although U-test results indicated no significant difference in the number of programming cards played and the number of bugs between the participant groups, the field-independence group seems to occur more use of the program card and more bugs behavior (3.78 for the number of programming cards played and 1.67 for the number of bugs). This result is consistent with the phenomenon observed by the research team during the teaching process.

## 5 Conclusions

This study compared learning outcomes and gaming behavior in an educational programming



board game between learners with different cognitive styles. Conclusions derived from the analysis of the findings are as follows. First, in the programming board game, field-independent learners achieved significantly improved learning outcomes over field-dependent learners. Second, although there is no significant difference was found in gaming behavior between field-independent and field-dependent learners, the field-independence group demonstrated more learning behavior related to the execution of complex thinking and more trial-and-error behavior. These conclusions indicate that to provide differentiated instruction for learners with different cognitive styles, should be more to enhance the programming learning performance and positive play behavior.

## Acknowledgement

This study was sponsored by the Ministry of Science and Technology of Taiwan, project number: MOST 106-2511-S-152-001 & 107-2511-H-152-009.

## References

- [1] Y. C. Lin and J. J. Wu, "Computational thinking and the information technology course for primary and secondary schools," *Pulse of Education Electronic Journal from National Academy for Educational Research*, no. 6, Jun. 2016.
- [2] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33-36, Mar. 2006.
- [3] S. Grover, and R. Pea, "Computational thinking in K-12: A review of the state of the field," *Educational Researcher*, vol. 42, no. 1, pp. 38-43, 2013.
- [4] S. Y. Lye and J. H. L. Koh, "Review on teaching and learning of computational thinking through programming: What is next for K-12?" *Computers in Human Behavior*, vol. 41, pp. 51-61, Dec. 2014.
- [5] Y. C. Lin and J. J. Wu, "Computational thinking and the information technology course for primary and secondary schools," *Pulse of Education Electronic Journal from National Academy for Educational Research*, no. 6, Jun. 2016.
- [6] A. Theodoropoulos, A. Antoniou, and G. Lepouras, "How do different cognitive styles affect learning programming? Insights from a game-based approach in Greek schools," *ACM Transactions on Computing Education*, vol. 17, no. 1, 2017.
- [7] P. Bayman and R. E. Mayer, "Using conceptual models to teach BASIC computer programming," *Journal of Educational Psychology*, vol. 80, no.3, pp. 291-298, 1988.
- [8] G. Salomon, and D. N. Perkins, "Rocky roads to transfer: Rethinking mechanism of a neglected phenomenon," *Educational psychologist*, vol. 24, no. 2, 113-142, 1989.
- [9] S. Papert, "The children's machine: Rethinking school in the age of the computer," Basic Books. New York, NY 10022-5299.

- [10] M. Resnick, et al. "Scratch: programming for all," *Communications of the ACM*, vol. 52, no. 11, pp. 60-67, 2009.
- [11] J. M. Wing, "Computational thinking benefits society," *40th Anniversary Blog of Social Issues in Computing*, 2014.
- [12] ISTE & CSTA. "Operational definition of computational thinking for K-12 education," Retrieved from <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf>.
- [13] H. Morgan, *Cognitive styles and classroom learning*. Westport, CT: Praeger, 1997.
- [14] W. C. Burnett, "Cognitive style: A meta-analysis of the instructional implications for various integrated computer enhanced learning environments," Doctoral dissertation, Indiana University of Pennsylvania, Pennsylvania, 2010.
- [15] K. Dragon, "Field dependence and student achievement in technology-based learning: A meta-analysis," Master thesis, University of Alberta, Alberta, 2009.
- [16] H. A. Witkin, C. A. Moore, D. R. Goodenough and P. Cox, "Field-dependent and field-independent cognitive styles and their educational implication," *Review of Educational Research*, vol. 47, no. 1, pp. 1-64, Mar. 1977.
- [17] O. N. Saracho, "Research directions for cognitive style and education," *International Journal of Educational Research*, vol. 29, no. 3, pp. 287-290, Aug. 1998.
- [18] K. Fullerton, "The interactive effects of field dependence-independence and internet document manipulation style on student achievement from computer-based instruction," Ed.D. Dissertation, University of Pittsburg, 2000.
- [19] J. Lee, "The effects of information conveying approaches and cognitive styles on learners' structural knowledge and perceived disorientation in a hypermedia environment," Ph.D. Dissertation, Indiana University, 2000.
- [20] T. Bell, J. Alexander, I. Freeman and M. Grimley, "Computer science unplugged: School students doing real computing without computers," *The New Zealand Journal of Applied Computing and Information Technology*, vol. 13, no. 1, pp. 20-29, 2009.
- [21] Z. S. Sie and G. S. Yan, *Mixed plug-in and unplugged Information Science for Taiwan, Handbook*. Taipei: Software Liberty Association Taiwan, 2017.
- [22] M. Kordaki, and A. Gousiou, "Digital card games in education: A ten year systematic review," *Computers & Education*, vol. 109, pp. 122-161, 2017.
- [23] H. A. Witkin, "A manual for the embedded figures tests," Consulting Psychologists Press, 1971.
- [24] S. C. Kong, M. M. Chiu, and M. Lai, "A study of primary school students' interest, collaboration attitude, and programming empowerment in computational thinking education," *Computers & Education*, vol. 127, pp. 178-189, 2018.