The Effects on Young Students' Computational Thinking in CS Unplugged Activities

[1]Siyu Zha, [2]Ruiwen Peng, [2]Fei Zhang
[1]*Beijing Normal University*, [2]*Tsinghua University*
[1, 2]*China*

Abstract

Computational thinking is viewed as a critical and necessary skill in the 21st century that everyone should learn. In addition, six years ago, Wing had argued for adding this new competency to every child's analytical ability as a vital element of science, technology, engineering, and mathematics (STEM) learning. This study aims to analyze an unplugged computer course effects on young students' computational thinking, especially focuses on their programming attitude and computational concepts and explore the relationship between logical reasoning ability and computational concept. This study investigates the development of 25 young students' computational thinking in 1st to 2nd grade through a "computer science unplugged course", which is a curriculum that has been developed for many years and has the goal to teach computational thinking without using computers. It adopted action research and collected data through questionnaire. The results showed that there is a strong connection between logical reasoning ability and computing concept. Moreover, there was little difference in the attitude toward programming between gender in young students.

1. Introduction

Computer programming education is widespread around the world. It has a direct connection with computational thinking which has received considerable attention recently. Wing [12] believed that computational thinking was the process of applying problem-solving, system design, and human behavior understanding of basic concepts about computer science. And researchers stated that the computational thinking has five elements of abstraction, generalization, decomposition, algorithm, and debugging [1]. Although there was still no precise definition of computational thinking, the three-dimension framework of computational thinking constructed by Brennan and Resnick [3] including computational concepts, computational practices, and computational perspectives have been recognized by many researchers. The computational concepts refer to the concepts used by the designer when programming; the computational practices refer to the practice that the designer develops in programming, and the computational perspectives refer to the designer's formation which regards to the world around them and their ideas. This theoretical framework helps researchers conduct practical research on computational thinking. Through these experiences, they are expected to establish a computational identity and achieve digital empowerment [13].

2. The definition of computing thinking

Computational thinking (CT) is a fundamental skill for students. Wing [12] believed that computational thinking was the process of applying problem-solving, system design, and human behavior understanding of basic concepts about computer science. It included a series of thinking activities that cover the computer science and is an essential skill for every child beside reading, writing, and arithmetic. Some researchers supported this view by claiming that CT was an essential skill as reading, writing, and other basic language arts skills, pointing out that "programming is a language for expressing ideas [6]. In addition, there are also define one state that the CT has five elements of abstraction, generalization, decomposition, algorithm, and debugging [1]. Although there was still no precise definition of computational thinking, the three-dimension framework of CT constructed by Brennan and Resnick [3] covers computational concepts, computational practices, and computational perspectives have been recognized by many researchers.

In the review of Lye and Ko [5], it was mentioned that most of the existing study focused on computational thinking regarding computational concepts, computational practices, and computational concepts in the K-12 education. The computational concepts refer to the concepts used by the designer when programming; the computational practices refer

to the practice that the designer develops in programming, and the computational perspectives refer to the designer's formation which regards to the world around them and their ideas, this theoretical framework help researchers conduct practical research on computational thinking. In the dimension of CT perspectives, extended from the original three perspectives in Brennan and Resnick [3], they are the important attributes for achieving the goal of nurturing creative problem solvers in the digital world. It is therefore important for learners to experience expressing, connecting, and questioning in the process of programming. Through these experiences, they are expected to establish a computational identity and achieve digital empowerment [13].

Moreover, the evaluation of computational thinking is also a part of the practice of computational thinking. Grover and Pea [4] conducted a six-week course on "Promoting Fundamentals of Computational Thinking" in secondary schools, focusing on the basics of computing and related capabilities in computing thinking in the form of Scratch-based task teaching. Sáez-López, Román-González, and Vázquez-Cano [8] proposed a computational thinking measurement scale for 28 elementary and middle school students. The content of the test is a computational concept, and part of the computational practice. The evaluation of computational thinking was mostly concentrated on computational concepts and computational practices. Besides, In the study of Atmatzidou, Demetriadis and Nika [2] employed an appropriate CT model for exploring students' CT skills development in two different age groups and across gender and founded that students reach eventually the same level of CT skills development independent of their age and gender, but girls appear in many situations to need more training time to reach the same skill level compared to boys. What maybe are the causes of the difference between gender?

2.1. The cultivation of computing thinking through CS unplugged

Programming for K-12 can be traced to the 1960s when Logo programming was first introduced as a potential framework for teaching mathematics [11]. Papert [7] claimed that the Logo programming experience could develop powerful intellectual thinking skills among children. To better adapt to this digital age, Kong [13] propose a seven-principle framework to design the curriculum in K-12 to promote computational thinking (CT) through programming. After Logo, the use of programming to teach computational thinking skills in K-12 was not extensively reported. However, in the recent years, it is fuelled by the availability of easy-to-use visual programming languages such as Scratch and Alice have been modelled after aspects of Logo [10]. There are a lot of methods to culticate computing thinking by programming. However, as for the young students, it can also adopt CT without computer.

Computer Science Unplugged is a method that cultivate students' CT through the fun games and puzzles, students can understand the concept of computer science and improve students' interest in computer science, so that the concept of computational science can be well understood without turning on the computer. The unplugged computer science teaching activities were initiated by New Zealand's Tim Bell, Ian H.Written and Mike Fellows. According to the actual teaching, the teaching activities were designed and organized. These teaching activities not only let students understand. The concept of computer science is more important to enhance students' thinking ability. Unplugged computer science is now also a world-wide information technology popularization project, and innovative teaching activities are constantly being supplemented.

Unplugged computer science is suitable for primary and secondary school students in different countries and with different levels of knowledge. Unplugged computers can be studied anytime, anywhere without any restrictions. Students can understand the generation, principle and process of computer science concepts in the process of making games or puzzles, and understand the diversified ideas and methods of computer scientists designing various technologies of computers. In the information technology class, the computer is turned off, without the attraction of various software in the computer, the students' attention can be more concentrated, and the students will deeply think about the computer science knowledge contained in the activity during the process of making games or puzzles. It is not only satisfied with a certain software that learns to operate a computer. Such activities will have a positive impact on the daily learning of students.

2.3. The research and Practice of Computational Thinking

Researchers pay more attention to the use of skills and tools, but there is no agreement on how to design and implement teaching activities. For example, Calao [14] proposed to integrate the core ideas of computational thinking into other curriculum activities, which will achieve a win-win effect. In the research, they integrate the core concepts of computational thinking into the grade six mathematics classroom and find that in addition to

computational thinking. In addition to the improvement in the level, the students' mathematics scores are also significantly higher than the control group. Therefore, it is speculated that the computational thinking can not only improve the students' problem-solving ability, but also improve the students' academic performance to a certain extent.

In the review of Lye and Koh [5], it was mentioned that most of the existing study focused on computational thinking regarding computational concepts, computational practices, and computational concepts in the k-12. The framework was first developed in 2012 by the MIT Lifelong Kindergarten Research Group by years of research and programming activities which has three dimensions includes computational concepts, computational practices, and computational concepts. And it is used to guide the design, implementation, and evaluation of computational thinking courses soon [3]. The computational concepts refer to the concepts used by the designer when programming; the computational practices refer to the practice that the designer develops in programming, and the computational perspectives refer to the designer's formation which regards to the world around them and their ideas, this theoretical framework help researchers conduct practical research on computational thinking.

Moreover, the evaluation of computational thinking is also a part of the practice of computational thinking. Grover [4] conducted a six-week course on "Promoting Fundamentals of Computational Thinking" in secondary schools, focusing on the basics of computing and related capabilities in computing thinking in the form of Scratch-based task teaching. Román-González at al. [15] proposed a computational thinking measurement scale for 28 elementary and middle school students. The content of the test is a computational concept, and part of the computational practice. Above all the research, the evaluation of computational thinking was mostly concentrated in computational concepts and computational practices. However, in this study, we want to focuses on students' programming attitude and computational concepts and explore the relationship between logical reasoning ability and computational concept.

3. Research Aims

The aim of the study is to analyze the effects on young students' computational thinking through a "computer science unplugged course"and the specific objectives are:

(1) To analyze the attitudes of young students' differences between gender through computer science unplugged course.

(2) To assess the relationship between basic computer programming concepts and logical reasoning ability in young students.

4. Research Methods

4.1. Participants

The participants of this study were 25 elementary school students (18 boys and 7 girls) from Grade 1 to Grade 2. Eleven participants were from Grade 1, and fourteen of them were from Grade 2. All students have little exposure to programming before this.

4.2. Instructional design

The instructional design was adopted from the teaching guide provided by computer science unplugged with a gamification learning approach. It consisted of explain the binary with puzzle paper, understanding commands and algorithms through outdoor games, explain the way of parity in data error correction by playing card magic, combine the LEGO suite to learn the sorting algorithm and so on. The participants took each course around ten weeks. In order to assure the effects of this study, students were randomly assigned into two groups. The two group of students take classes on Saturday and Sunday mornings respectively.

4.3. Instruments

The questionnaire developed by Shim, Kwon, and Lee [9] which has two parts were used to evaluate participants' attitude towards programming and the programming tool's usability and edutainment. As shown in Table 1, there are 14 items about the attitudes towards programming. All choices were measured using the Likert 5-scale, which included five options from "completely disagree," "disagree," "agree," "more agree," to "completely agree," with the score of 1-5 respectively. The reliability coefficients of value, interest, and confidence are 0.849, 0.822, 0.93, and the KMO value was 0.926. Besides, it also adopted the questionnaire about computational concept, there are 9 items need to be filled in the blank and the reliability coefficient is 0.876, the KMO value is 0.892. And the other test is from Raven's Standard Progressive Matrices (SPM) which test the logical reasoning ability of young students.

Figure 1. Learning binary with puzzle paper

Figure 2. explain the way of parity in data error correction by playing card magic

Figure 3. understanding commands and algorithms through outdoor games

Table 1. Questionnaire about programming attitude

Dimensions	Questionnaire
Value	I think that I can resolve practical problems through programming. I think that I can live a successful life if I understand programming. I think that programming is important for constructing something new in tl future. I think that programming will have a critical role in my life. I think that today's experiential learning is as important as learning math ar science.
Interest	I want to ask questions of, or learn from, a person who is proficient programming whenever I see him or her. I want to befriend someone who is proficient at programming. I enjoy using the concepts learned from programming activities. I want to continue learning programming.
Confidence	I am confident of understanding contents related to programming. I am confident that I can be proficient at learning programming. I am confident that I can solve problems and tasks using programming. I am confident that I can be proficient at doing something usir programming. I am confident that I can simplify a problem, or make it less difficult, usir programming.

5. Results

5.1. Gender difference in programming attitude

An independent sample t-test was conducted to analyze whether there was a gender difference in students' programming attitude. The analysis results are shown in Table 2 that the male's score of the test was no significantly higher than female from the aspect of programming value (t (25) =-.844, p=0.407), programming interest (t (25) =0.597, p=.556) and the programming confidence (t (25) =0.643, p=0.527). This result is different from the study of Sáez-López et al., [8]. They conducted computer teaching for students in different grades of primary school and found that the development of students' computational thinking is significant regarding gender and grade.

Table 2. Programming attitude in gender

	Gender	*N*	*M*	*SD*	*t*	*p*
Value	Female	7	19.389	3.499	-.844	.407
	Male	18	20.714	3.534		
Interest	Female	7	16.867	2.854	.597	.556
	Male	18	21.722	21.199		
Confidence	Female	7	19.143	3.078	.643	.527
	Male	18	24.333	20.999		

Note: * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$

Table 3. Relationship between programming concepts and logical ability

	N	*M*	*SD*	*t*	*p*
SUM	25	6.320	1.060	1.060	0.010**
SPM	25	43.760	36.607	36.607	

Note: * p < 0.05; ** p < 0.01; *** < 0.001.

5.2. Relationship between programming concepts and logical reasoning ability

Table 3 reports the relationship between programming concepts and logical reasoning ability with p=0.010. Statistically, the stronger the student's logical reasoning ability, the higher the student's score on the programming concept, in which students can understand computational concepts well.

6. Conclusion

Computer science unplugged is the activities that has been developed for many years and has the goal to teach computational thinking without using computers and many of the activities include manipulatives such as cards or dice. This current study investigated the effects on young students' computational thinking without computers. It was showed that the computer science unplugged really did work on young students.

As for the attitude towards programming, the results showed that there is no significant difference in programming attitude between gender. Regarding the reason for this may be the content of this unplugged programming class is lively and uses interesting gamification teaching methods, which leads the participation of girls is also high. In order to reveal whether the students' logical reasoning ability is related to the learning and understanding of computational concepts in computational thinking, this study found that the correlation is higher based on the data collection and processing.

Moreover, as the correct rates for students in the algorithm, abstraction, and logic sections are 95%, 52%, and 76%, which revels students can understand the algorithm well. However, perhaps because of age, students' ability in abstraction is weak, which also provides reference and reference for teaching young students in programming.

7. References

[1] Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., and Zagami, J. (2016). A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Journal of

Educational Technology and Society, 19(3), 47–57.

[2] Atmatzidou, S., Demetriadis, S., and Nika, P. (2017). How Does the Degree of Guidance Support Students' Metacognitive and Problem-Solving Skills in Educational Robotics? Journal of Science Education and Technology, 1–16. https://doi.org/10.1007/s10956-017-9709-x

[3] Brennan, K., and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Retrieved December 14, 2017, from https://www.media.mit.edu/publications/new-frameworks-for-studying-and-assessing-the-development-of-computational-thinking/

[4] Grover, S., and Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. Educational Researcher, 42(1), 38–43. https://doi.org/10.3102/0013189X12463051 Kong, S.-C. (2016). A framework of curriculum design for computational thinking development in K-12 education. Journal of Computers in Education, 3(4), 377–394. https://doi.org/10.1007/s40692-016-0076-z

[5] Lye, S. Y., and Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

[6] National Research Council. (2010). Report of a Workshop on the Scope and Nature of Computational Thinking. National Academies Press.

[7] Papert, S. (1980). Mindstorms: children, computers, and powerful ideas. Retrieved May 22, 2018, from https://cn.bing.com/academic/profile?id=d041c558229147ac04a3b561314ae782andencoded=0andv=paper_previewandmkt=zh-cn

[8] Sáez-López, J.-M., Román-González, M., and Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. Computers and Education, 97, 129–141. https://doi.org/10.1016/j.compedu.2016.03.003

[9] Shim, J., Kwon, D., and Lee, W. (2017). The Effects of a Robot Game Environment on Computer Programming Education for Elementary School Students. IEEE Transactions on Education, 60(2), 164–172. https://doi.org/10.11 09/TE.2016.2622227

[10] Utting, I., Cooper, S., Kölling, M., Maloney, J., and Resnick, M. (2010). Alice, Greenfoot, and Scratch – A Discussion. Trans. Comput. Educ., 10(4), 17:1–17:11. https://doi.org/10.1145/1868358.1868364

[11] Wallace Feurzeig, and Seymour A. Papert. (2011). Programming-languages as a conceptual framework for teaching mathematics: Interactive Learning Environments: Vol 19, No 5. Retrieved May 12, 2018, from https://www.tandfonline.com/doi/abs/10.1080/10494820903520040

[12] Wing, J. (2006). Computational thinking. Communications of the ACM.

[13] Kong, S., (2016). A framework of curriculum design for computational thinking development in K-12 education. Journal of Computers in Education. 10.1007/s40692-016-0076-z.

[14] Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In Design for Teaching and Learning in a Networked World, Cham: Springer (pp.17–27)

[15] Román-González, M., Pérez-González, Juan-Carlos and Jiménez-Fernández, C., 2017. Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. Computers in Human Behavior, Volume 72, pp. 678-691.

