Title	Explorations of two approaches to learning CT in a game environment for elementary school students
Author(s)	Chiu-Pin Lin, Su-Jian Yang, Kuen-Yueh Lin, Chee-Kit Looi and Yen-Hua Chen

Lin, C.-P., Yang, S.-J., Lin, K.-Y., Looi, C.-K., & Chen, Y.-H. Explorations of two approaches to learning CT in a game environment for elementary school students. *J. Comput. Educ.* (2021). https://doi.org/10.1007/s40692-021-00203-x

Explorations of Two Approaches to Learning CT in a Game Environment for Elementary School Students

Abstract Collaboration can be a significant determinant in the teaching and development of computational thinking (CT) skills in school students. This paper presented a case study on how students learned CT skills through a coding game via two approaches: one using collaborative learning (through cognitive apprenticeship) and the other through individual learning. The quasi-experimental design was used in two classes (24 students in each class), and was divided into using one of the two approaches. Relevant data were collected through pre- and post-tests based on the E-game CT test. The research results indicated: 1) Both collaborative learning (through cognitive apprenticeship) and individual learning effectively enhanced the learning effects of students' development for CT skills in an E-game coding education platform; 2) In collaborative learning (through cognitive apprenticeship), students in lower-scoring groups could enhance their CT skills more effectively than those in higher scoring groups; 3) For those students who fell in the two lower-scoring groups, those in collaborative learning (through cognitive apprenticeship) had higher significant learning effects than those in individual learning; 4) Students with lower scores who programmed with a higher-scoring partner learned more than when they programmed alone. Overall, this study showed that the use of an E-game learning platform enabled the learning process of CT to become more interesting and challenging for students.

Keywords Coding Education · Cognitive Apprenticeship · Collaborative Learning · Computational Thinking

Introduction

In the 21st century information technology has been developing quite rapidly, be it in theory, techniques, systems and tools; information technology has been an important area influencing the development of human society. American scholar Wing (2006) introduces the idea of Computational Thinking (CT), regarding the skills of computer computational thinking as skills and competences everyone should be equipped with. For reading, writing and arithmetic, we should cultivate each child with analysis skills of CT. As an approach for thinking and acquiring new strategies to solve problems, CT can be commonly applied to every field. It can help people focus on important features of a problem and ignore insignificant details. Through the strong computational skills given by information technology, problem solving can become more efficient (Wing, 2006, 2008; Ying, 2016). After the idea of CT was raised, people care more about blending computer science into K-12 environments. Such grand occasions are driven by many factors, such as computer skills becoming requirements for the economic and technical labor force of the future (Chen et al., 2017).

Accordingly, courses incorporating some aspects of CT are prevalent in middle and elementary school education among advanced countries. For example, in 2015, the USA had the Every Student Succeed Act (ESSA) and in 2016, President Obama proposed the Computer Science for All Project. Programming plays an important role in all subjects (U.S. Department of Education, 2016; White House, 2016). The fact that the UK Ministry of Education changed the subject from ICT (Information and Communication Technology) to "Computing" suggests that the course will focus more on computer science rather than the operation and application of information technology (GOV.UK, 2013). To keep up with the trends in the age of information, each country emphasizes the cultivation of CT (indicating the importance of CT) through newly formed information technology courses including related concepts in general course guidelines (ACARA, 2013; CSTA, 2017; GOV.UK, 2013).

Based on these educational trends and the worldwide recognition of the importance of CT, we can see how crucial it is to cultivate children's CT skills from a young age (Wing, 2008). Despite this widespread interest, successful CT integration in compulsory education still faces unresolved issues and challenges (European Commission, 2016). Related studies point out that the most direct way to cultivate CT is through coding (Buitrago Flórez et al., 2017; Parmar et al., 2016). The interface for traditional coding requires the input of all kinds of grammatical instructions that include English, numbers and signs. Many students might find coding boring and tedious in this way, and it is hard to motivate their interest to learn more (Mannila et al., 2006). The use of appropriate and visualized coding software will trigger the students' interest in learning and reduce their sense of frustration (Brusilovsky & Spring, 2004). In these courses at K-12 grades, activities should be designed to provide more material/activities for different grade levels/learning areas by using programming tools rather than only focusing on the use of the tools (Dağ, 2019). The starting point of this research, the work presented in this paper, is a coding education game for teaching CT in elementary education. E-game (U-Generation Island Paradise) is an online coding education game which is an interactive teaching platform that was developed in Taiwan. It guides students through a graphical interface to do programming via building blocks and by following models. It can be used for game-based learning and inspire more advanced coding courses in the future. The literature on this research seems to be scarce, so it arouses our great interest.

In Taiwan, a number of initiatives addressing CT and coding/programming have been carried out, and teachers have started to combine some digital teaching material or games with CT skills. However, due to the lack of guidance mechanisms or appropriate learning strategies, it is difficult for students to achieve good academic performance. Collaboration is a significant determinant in teaching and developing of CT skills (Standl, 2016). Compared to the individualistic approach, collaboration can generate more benefits (Voogt et al., 2015). They can reduce frustration experienced, enhance student enjoyment, and promote positive attitudes in programming (Bishop-Clark et al., 2006; McDowell et al., 2002; Preston, 2005). Therefore, in order to understand how students learn CT skills through the E-game platform, this study uses two different approaches to further explore whether the differences in learning strategies will affect CT learning (especially students with low grades): collaborative learning (through cognitive apprenticeship) and individual learning. One of the purposes of the research is to compare the differences in the learning effects of the two approaches, and the second purpose is to compare the differences in the learning effects of the high and low groups of students. Besides, this research also examines students' learning attitudes in platform operation, learning motivation, problem solving, course content, collaborative learning, and individual learning.

Literature review

Computational thinking and programming

In the past, coding education was provided usually only in higher education. Although programming is extremely important, it is often abstract and hard for beginners to learn. Besides, most programming languages are not highly interactive. Beginners might easily encounter difficulties when learning programming (Felleisen et al., 2004). However, there have been advances in the development of languages which are easier for beginners to learn, and there is much research literature now in programming education. In order to familiarize students with programming, Scratch (developed by the Massachusetts Institute of Technology), Code.org® and Blockly (a graphic programming tool provided by Google) are mostly used for programming learning in junior high and elementary schools. These environments use visualized programming building blocks, meaning that students do not need to learn complicated text instructions. With certain logical concepts, they can complete simple coding by assembling programming building blocks. Therefore, these software and websites are suitable for students who have no programming experience. For example, the most prominent feature of Scratch is to turn difficult programming instructions into pieces of building blocks. As long as the learner matches the instructional building block to the corresponding programming concept and drags the block to the instruction editing area, (s)he can create a game or an animation. Thus, people of all ages are able to learn programming by simply dragging programming building blocks.

The process of programming mainly helps the learner to establish a problem-solving procedure through coding. Without the guidance of CT, it is relatively more complicated and difficult to directly compose a program to solve problems (Jenkins, 2002). CT emphasizes algorithmic processes. In algorithmic thinking, we need a problem-solving plan first before producing a problem-solving procedure accordingly. In short, programming makes CT more concrete (Liu, 2017).

Wing (2008) suggests that CT is a form of Analytical Thinking as well as a series of thinking activities using basic concepts of computer science to solve problems, design systems and understand human behaviors. CT helps individuals to construct a successful problem-solving model by abstraction, decomposition, composition, algorithm, assessment, and generalization, using computation or technology (Angeli et al., 2016; Shute et al., 2017; Wing, 2006). In abstraction, we decide on what details we need to highlight and what details we can ignore – underlies computational thinking (Wing, 2008). In decomposition, we break down a complex problem into smaller parts that are easier to understand and solve (CSTA, 2017; National Research Council, 2010; Rich et al., 2018). Pattern recognition looks for similarities in solutions among, within and problems (Burton et al., 2020; Figueiredo, 2017). Algorithmic thinking develops a step-by-step solution to the problem (Selby, 2014). These problem-solving methods are expected to be applied to problems that may be faced in the future (Angeli et al., 2016; Shute et al., 2017; Wing, 2006). These skills can be used by everyone, regardless of their area of knowledge, tasks or age (Figueiredo, 2017), and even young primary school students can engage in learning these CT skills (Rijke et al., 2018).

Many studies point out that "learning through programming" is a more effective way to learn CT (Dasgupta et al., 2016; Lye & Koh, 2014; Ying, 2016). Some studies have explored how to cultivate children's CT skills from a young age to develop cognitive learning and problem-solving skills (Grover & Pea, 2013; Kafai, 2016; Voogt et al., 2015). For example, Bers (2019) suggested in her study that there is a need to explicitly conceptualize pedagogical approaches for teaching computer science in the early years that embrace the maturational stages of children by inviting play and discovery, socialization, and creativity. Coding activities can engage children in thinking about powerful ideas from computer science, as well as other domains. In addition, some studies note that visualized programming languages can promote children's logical thinking, reasoning skills, learning motivation, creativity, individual and collaborative learning, problem-solving skills, etc. (Giordano & Maiorana, 2014; Gupta et al., 2012; Lewis, 2010; Lai, & Lai, 2012; Maloney et al., 2010; Wang et al., 2014).

E-game is an online interactive teaching platform game developed in Taiwan, including English, mathematics, science, coding, robots and other courses. Its system is based on HTML5. Therefore, it is suitable for e-learning on mobile devices. It uses the programming environment Code.org® and the Scratch 3.0 online version. The user can drag the graphic coding building blocks to complete coding. It covers sequences, loops, parallelism, events, conditionals and other coding concepts, and it designs different levels of courses step by step from easy to difficult. Thus, it may be suitable for cultivating elementary school students' CT programming skills.

Collaborative learning through cognitive apprenticeship

Since the rise of cognitive psychology, constructivists are starting to regard learning as a process for active knowledge construction. Hence, they combine the idea of cognition and traditional apprenticeship to develop cognitive apprenticeship, which emphasizes learning through real conditions and problem-solving oriented learning. They care about the "active cognitive process" in "learning from doing" (Brown et al., 1989).

Traditional apprenticeship is often a one-on-one technical teaching method, focusing on sharing the master's personal examples and experiences with the apprentice. There is no regular and systematic teaching. The master is the dominant leader of the whole learning process. It is a unidirectional teaching method between a superior and a subordinate. The apprentice might just imitate without innovation. Collins, Brown and Newman (1989) propose cognitive apprenticeship, suggesting two meanings: 1) It focuses on how the expert handles complicated skills and highlights the expert's process and situated learning. 2) It emphasizes cognitive and metacognitive levels rather than motor skills and processes stressed by traditional apprenticeship. Therefore, cognitive apprenticeship suggests that learning should be based on systematic learning activities and the experts' demonstrations so that the learner can understand the inner thinking process and possible solutions when encountering problems. Also the learner has the chance to actually operate and experience what they learn. In the course of hands-on operation, they can self-interpret, absorb, convert and use that knowledge. It emphasizes the interactive process of passing on experience and requests the learner to reflect upon themselves and their thoughts. Under the instructor's guidance, the learner's creativity can be motivated to create their own problem-solving skills (Wu & Lin, 2002).

The strategies of cognitive apprenticeship include six steps, namely, modeling, coaching, scaffolding, fading, articulation, reflection and exploration (Collins et al., 1989；Brown, 2017；Garcia-Cabrero et al., 2018). Among them, articulation, reflection and exploration form a set of learning strategies. After being coached, the learner will understand the key to his/her own problem-solving method. Then (s)he will try to explain it to others. The learner will compare his/her own problem-solving process with that of the expert's or peers' to understand the differences and further improve his/her own learning. Finally, (s)he will have achieved the learning objectives. The process of reflection can also be aided by technology. Eventually, the learner can independently explore and solve problems in new conditions. The teaching strategies and activities adopted by the study are as presented in Table 1.

Table 1 Cognitive Apprenticeship Teaching Strategies and Activities Adopted by the Study

Mode	Scaffolding	Modeling/Scaffolding	Coaching	Fading	Articulation	Reflection, Exploration
Expert (Master)	Setting a learning objective and contents	Modeling and motivating	Confirming teaching steps and actual teaching	Less instructions for operation and reducing the learner's sense of frustration	Discovering problems, guiding discussion, and integrating learning experience	Reviewing teaching effects and giving new challenges and positive feedback
Learner (Apprentice)	Reviewing what (s)he has learned and contents regarding the teaching topic	Observing and learning from the expert's modeling and teaching	Completing assigned activities under assistance and listening to advice modestly	Trying to complete assigned activities as independently as possible	Discussing with others what (s)he finds hard to understand, understanding the problem and trying to explain it to others	Comparing the differences, improving his/her learning, independently exploring and solving problems in new conditions
Interactive Mode	Individual activities	Collaborative learning	Collaborative learning	Collaborative learning	Collaborative learning	Collaborative learning and individual activities

Many scholars have found the application of cognitive apprenticeship in education to be quite successful. Garcia-Cabrero et al. (2018) indicate that learner-centered online virtual learning courses under cognitive apprenticeship teaching strategies, experts' online modeling, coaching and scaffolding are able to enhance students' learning efficacy and improve the contents and quality of online courses. In addition, studies have shown that students regularly perform better with pair programming than with solo programming in CT (Lye & Koh, 2014; Werner & Denning, 2009). Pair programming is a useful strategy to promote CT among students (Leow & Huang, 2021). The study of Larkins et al.

(2013) found that the cognitive apprenticeship framework, in conjunction with robotics, is an excellent way to build interest in STEM and develop skills in engineering, science, and CT.

With the progress of information technology, e-learning is now blended with cognitive apprenticeship. Accordingly, cognitive apprenticeship often becomes a teaching method to cultivate professionals and enhance digital learning effects. The teaching strategy of this study is collaborative learning (through cognitive apprenticeship) with two people in a group. The highly accomplished student will lead the lower accomplished student. Through oral and non-oral interaction, modeling and feedback, this study explores the learning efficacies of CT cultivation in students' coding education.

Method

This study explored the influence of coding on learning efficacy and attitudes through the cultivation of fifth graders' CT skills by using the E-game in two different approaches, collaborative learning (through cognitive apprenticeship) and individual learning. A quasi-experimental design was adopted.

Participants

In this study, 51 students without any CT training before from two fifth grade classes were divided into experimental group 1 and experimental group 2. Experimental Group 1 contained two students with special educational needs while Experimental Group 2 contained only one. During the research, all the students, including students with special educational needs participated in the experiment. However, the scores used for the experimental data were those of the 48 students excluding the students with special educational needs.

Based on the survey, 24 of the 51 students had computers or mobile devices at home for networking. For those without Internet access, because the school promoted e-learning, the students had tablet computers for individuals. Therefore, for students of individual learning at Experimental Group 2, they had sufficient hardware devices. Students of Experimental Group 1 used the collaborative learning (through cognitive apprenticeship) approach in which they worked in pairs. According to "E-game CT test", the grades of the pre-test were divided into higher score groups and lower score groups. Each group consisted of two students, one from the higher score group and one from the lower. There were 26 students in 13 groups. Experimental Group 2, which was composed of 25 students, used the individual learning approach.

Experimental procedure

Before receiving experimental teaching, both groups took a pre-test (E-game CT test). After the pre-test, they started receiving their five-week courses. After the experiment, the two groups then took a post-test (E-game CT test) and filled out a "Programming Learning Attitude Questionnaire." The teaching duration for both groups was the same. The experiment procedure is as presented in Table 2.

Table 2 Experiment Procedure Table

Phase	Description	Experimental Group 1 Collaborative Learning (through Cognitive Apprenticeship)	Experimental Group 2 Individual Learning
Pre-test	Both groups of students took a pre-test in written form.	They took a pre-test (E-game CT test).	
Scaffolding	· Teaching contents and syllabus · Setting a learning objective and contents	Five learning units： Unit 1 E-game environment introduction (35 minutes): The students would be familiar with how to log in and basic operations as well as functions including billboard, island and passports, group mode, etc. Unit 2 Lilija introduces programming to you (35 minutes): The programming building block concepts to be used included sequence and loops. Unit 3 What are loops? (35 minutes): The programming building block concepts to be used included until, loops, and sequence. Unit 4 The application of conditions and loops (35 minutes): The programming building block concepts to be used included if, until, loops, and sequence. Unit 5 Escaping from Mist Forest (35 minutes): The programming building block concepts to be used included if…else, if, until, loops, and sequence.	
Modeling/ Scaffolding	Motivating	Teacher Demonstrating: The teacher took examples as the starting point, such as traffic lights, lottery prize announcements, and automatic doors of convenience stores to explain how programming was applied in life through hands-on operation. The teacher would ask students why they needed to learn programming and timely present to them how emerging technology was applied in life. The teacher would also indicate the importance of programming in the current technology field. The teacher would take the example of Dako Island in E-game to introduce the environment of the website to the students and teach them the learning objectives. It was the first step in introducing the students to programming.	
Coaching	Development activities	The student would have their own computer to learn programming. The student would start an E-game and log in for games. The teacher would make sure the student enters the group mode set up by the teacher (teacher-student teaching mode).	The student would have their own computer to learn programming. The student would start the E-game and log in for games. The teacher would make sure the student enters the group mode set up by the teacher (teacher-student teaching mode).
Fading	· Less instruction · Reducing the learner's frustration · Collaborative in group 1 · Individual learning in group 2	The teacher would observe and record students' learning process from the system. The student would finish each learning activity according to the learning objectives. If she/he had any questions, the teacher with relatively higher score) can have a conversation with the student (student with relatively lower score). This stage was the core process to build up Cognitive Apprenticeship.	The teacher would observe and record Student's learning process from the system. The student would finish each learning activity according to the learning objectives. If she/he had any questions, she/he can look them up in the online E-game student manual.

Articulation	· Discovering problems, guiding discussion in group 1 · Individual learning in group 2	Students could ask the teacher if they still didn't understand. The teacher would timely provide guidance without directly giving them answers. The student could take a break from the game once they completed the goal of the week instead of moving to the next level.	If students still didn't understand, they could ask the teacher. The teacher would timely provide guidance without directly giving them answers. In this stage interaction and communication with a peer was not allowed. The teacher would record the students' questions. Students completing the goal of the week could still keep going to the next level by themselves.
Reflection, Exploration	Reviewing & feedback	5 minutes before the end of the class, the teacher would explain the key points of the day's learning. The teacher would tell the students in advance the progress of the next class but would remind them not to get ahead of the schedule. The teacher would orally praise the group with excellent performance in class.	5 minutes before the end of the class, the teacher would explain the key points of the day's learning. The teacher would explain common questions and remind the students of the progress of the next class. The teacher would orally praise the student with excellent performance in class.
Post-test	Two groups of students take a post-test in written form.	The students would take a post-test (E-game CT test) and fill out "Programming Learning Attitude Questionnaire."	

Instruments

E-game (U-Generation Island Paradise)

The study used the Dako Island (Saving Lilija) in E-game as teaching materials for students to learn CT. The E-game structure is designed by HTML5 to meet cross-platform and cross-vehicle standards. It is a programming learning platform integrated with role-play games. The teacher could analyze the students' learning conditions through the back-end management, understand their learning conditions in different stages and arrange proper courses according to their learning skills. In the learning activity, as shown in Figure 1, students could design a program to control the movement of forward, right, left, or return to their destination.

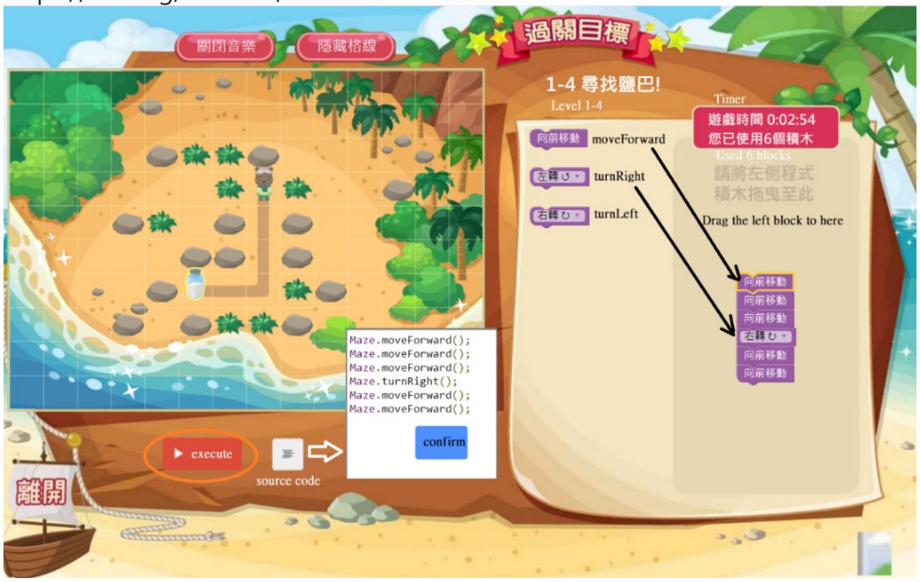

Fig. 1 E-game (U-Generation Island Paradise) course mode

E-game CT test paper

The researchers created the E-game CT test paper according to the contents of the course. Based on the Dako Island (Saving Lilija) unit in E-game, the test contains questions covering solutions used in the game in selections, blank filling, error finding, etc., including sequence, loops, condition judging, etc. We invited two computer science teachers to check the validity of the paper and corrected the test questions based on the comments they provided. Following that, the test was administered to 45 students in Grade 6 to analyze its reliability and to collect feedback on question comprehension. Finally, the author composed an official 27-question test, including three types: 14 questions for understanding and memory, 9 questions for resolving and simplification, and 4 questions for error finding. According to Ebel and Frisbie (1991), the level of difficulty and discriminability of good test items should be between 0.4–0.8 and 0.4–1, respectively. The overall discrimination index of the test paper was 0.408, the difficulty was 0.69, and the overall test reliability Cronbach's alpha value was 0.714, which were all within credible range.

Programming learning attitude questionnaire

The researchers also created the E-game "Programming Learning Attitude Questionnaire" based on the teaching process and contents. It invited two computer science teachers to check the validity of the scale and corrected the questions based on the comments they provided. The data was collected by dividing the questionnaire into six dimensions; platform operation, learning motivation, problem-solving, course contents, collaborative learning (only for Experimental Group 1) and individual learning (only for Experimental Group 2). The subscale Cronbach's alpha values were 0.772, 0.801, 0.787, 0.847, 0.828, 0.804, which were all within credible range.

Data on group interaction

In order to understand how participants collaborate in the CT course, a camera was set up in the classroom to record their interaction. In addition, we observed how students participate in class and content, and conducted in-depth semi-structured interviews to better understand this issue.

Results analysis

Analysis of learning achievement

Pre-test analysis

Before the experiment, the students in Experimental Group 1 and 2 would respectively take the "E-game CT test" pre-test. After the scores of each group are calculated, the researcher would analyze the scores under independent sample *t*-test through SPSS so as to collate the individual differences between the two groups of students. The results are presented as Table 3.

Table 3 The independent sample t-test chart of the two groups' pre-test scores

Test	Group	N	Mean	SD	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Pre-test	Experimental Group 1	24	52.50	21.40	-.687	.495	0.199
	Experimental Group 2	24	48.79	15.52			

The mean of Experimental Group 1 and 2 is 52.50 and 48.79 respectively. The Levene test of homogeneity of variance is not significant (F=3.656, p = .062 > .05), indicating that the discrete degree of these two groups does not have a significant difference. According to the *t*-value and significance hypothetically with equal variance, the result is not significant (t=-.687, p=.495 > .05). The effect size of Cohen's d is 0.199. Cohen (1988) gave a rule of thumb for the interpretation, this result in Cohen's d of 0.199 is a small effect size. Experimental Group 1's and 2's pre-test scores before the teaching experiment do not have a significant difference, meaning that before the experiment, the initial skills of the two groups are of a similar level.

Analysis on the pre-test and post-test

Experimental Group 1 and 2 use the scores of the "pre-test" and "post-test" as the paired variance respectively. The results under paired sample *t*-test analysis through SPSS are shown in Table 4. According to the data, the pre-test and post-test scores of Experimental Group 1 (*t*=-5.206, *p*<.001) and Experimental Group 2 (*t*=-8.415, *p*<.001) achieve a significant difference, the result in Cohen's d of 1.077 and 1.718 represents a large effect size. The correlation of Experimental Group 1's pre-test and post-test is 0.652. The students' average score improves from 52.50 to 70.71, making progress by 18.21 points. The correlation of Experimental Group 2's pre-test and post-test is 0.815. The students' average score improves from 48.79 to 65.96, making progress by 17.17. The pre-test and post-test paired sample of the two groups show significant positive correlation.

Table 4 The pre-test and post-test paired sample *t*-test chart of the two groups

Group	Test	N	Mean	SD	Progressed Score	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Experimental Group 1	Pre-test	24	52.50	21.40	18.21	-5.206	<.001	1.077
	Post-test	24	70.71	19.47				
Experimental Group 2	Pre-test	24	48.79	15.52	17.17	-8.415	<.001	1.718
	Post-test	24	65.96	16.99				

This section compares the differences in the learning effects of the two approaches. As shown in Table 4, both Experimental Group 1 and Experimental Group 2 make good progress after being taught with a significant difference in their performance. For both groups of students are under "Collaborative Learning (through Cognitive Apprenticeship)" or "Individual Learning," their learning efficacy in "E-game CT test" is effectively improved. However, as Experimental Group 1 in this study mainly focuses on collaborative learning factors, the study will analyze in detail the higher score and lower score students in Experimental Group 1.

The higher and lower scoring students are selected for paired sample *t*-test analysis in Experimental Group 1. Based on the post-test scores in Experimental Group 1, the top 27% students are then selected. 7 of the higher scoring students are selected for test analysis. The statistical result is shown in Table 5. The mean of the pre-test for higher scoring students is 77.29 and that of the post-test is 86.29. The *t*-value is -2.213 while significance *p*=.069>.05 (d=0.836), indicating that although the learning efficacy of the higher scoring students made progress, it does not achieve a significant difference.

Based on the post-test scores in Experimental Group 1, the bottom 27% students are then selected. 7 of the lower scoring students are selected for test analysis. The statistical result is shown in Table 5. The mean of the pre-test for lower scoring students is 26.00 and that of the post-test is 61.71. The *t*-value is -6.406 while significance *p*=.001<.05, the result in Cohen's d of 2.421 represents a very strong effect size, indicating that the learning efficacy of the lower scoring students achieves a significant difference.

Table 5 The Paired Sample *t*-Test Analysis Chart for the Learning Efficacy of Higher and Lower Score Students in Experimental Group 1

N=7	The Paired Sample *t*-Test of Higher Scoring Students' Pre-test and Post-test					
Test	Mean	SD	Progressed Score	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Pre-test	77.29	5.83	9	-2.213	.069	0.836
Post-test	86.29	10.45				
N=7	The Paired Sample *t*-Test of Lower Scoring Students' Pre-test and Post-test					
Test	Mean	SD	Progressed Score	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Pre-test	26.00	7.44	35.71	-6.406	.001	2.421
Post-test	61.71	21.81				

This section compares the differences in the learning effects of the high and low groups of students under the same method. According to the paired sample *t*-test analysis of the higher and lower scoring students in Experimental Group 1, under collaborative learning (through cognitive apprenticeship), both higher and lower scoring students made progress. Yet, the learning efficacy of the lower scoring students achieves a significant difference, making progress by 35.71 points. It is possible that the lower scoring students' initial skills are less adequate. After collaborative learning, under oral and non-oral interaction with the higher scoring students, they effectively construct scaffolds. With the support of their basic understanding, their cognitive development is enhanced, and so is their learning efficacy.

The analysis of the paired sample *t*-test on the higher and lower scoring students in Experimental Group 2 shows improvements in both higher and lower scoring students after individual learning in "E-game CT test." However, both higher and lower scoring students do not achieve a significant difference (see Table 6).

Table 6 The Paired Sample *t*-Test Analysis Chart for the Learning Efficacy of Higher and Lower Score Students in Experimental Group 2

N=7	The Paired Sample *t*-Test of Higher Scoring Students' Pre-test and Post-test					
Test	Mean	SD	Progressed Score	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Pre-test	72.29	6.85	8.57	-2.391	.054	0.904
Post-test	80.86	11.28				
N=7	The Paired Sample *t*-Test of Lower Scoring Students' Pre-test and Post-test					
Test	Mean	SD	Progressed Score	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Pre-test	34.71	9.88	7.29	-2.339	.058	0.884
Post-test	42.00	9.85				

Post-test analysis

This section compares the differences between the two approaches in the post-test learning effect. Comparing the post-test scores of Experimental Group 1, under "Collaborative Learning (through Cognitive Apprenticeship)" and Experimental Group 2, under "Individual Learning" in programming courses through the E-game, the researcher analyzes the results with an independent sample t-test through SPSS. The results are shown in Table 7.

Table 7 The Independent Sample t-test Chart for the Two Groups' Post-Test

Test	Group	N	Mean	SD	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Post-Test Score	Experimental Group 1	24	70.71	19.47	.900	.288	0.26
	Experimental Group 2	24	65.96	16.99			

As shown in Table 7, the mean of the students in Experimental Group 1's learning efficacy post-test score is 70.71 while that of Experimental Group 2 is 65.96, showing that the scores for Experimental Group 1 are higher than those of Experimental Group 2. The Levene test of homogeneity of variance is not significant (F=.148, p = .702 > .05). According to the *t*-value and significance under the hypothesis of equal variance, the results of both post-tests do not achieve a significant difference (t=.900, p=.288 > .05). The effect size of Cohen's d is 0.26. Cohen (1988) gave a rule of thumb for the interpretation, this result in Cohen's d of 0.26 is a small effect size. According to the above-mentioned results, the post-test scores of students under both collaborative learning (through cognitive apprenticeship) and individual learning in "E-game CT test" do not show a significant difference. However, because Experimental Group 1 is under collaborative learning (through cognitive apprenticeship) during the learning process, the higher scoring students would help the lower scoring students; it is a common teaching strategy. To figure out the learning efficacy of collaborative learning (through cognitive apprenticeship), this experiment will further analyze higher and lower scoring students.

The students in Experimental Group 1 and 2 are divided into higher scoring and lower scoring groups under selection ranked according to the top and bottom 27% of their post-test scores. Each group selects 7 higher and lower scoring students for independent sample t-test analysis. The statistical results are shown in Table 8.

Table 8 Independent Sample t-test Chart for the Post-test of Higher and Lower Scoring Students in Experimental Group 1 and 2

Test	Group	N	Mean	SD	*t*-value	Sig. (Two-tailed)	Cohen's d effect size
Post-test score of higher scoring group	Experimental Group 1	7	86.29	10.45	-.959	.357	0.499
	Experimental Group 2	7	80.86	11.28			
Post-test score of lower scoring group	Experimental Group 1	7	61.71	21.81	-2.180	.039	1.165
	Experimental Group 2	7	42.00	9.85			

This section compares the differences in the learning effects of the two approaches in the post-testing high and low groups of students. As shown in Table 8, the mean of the higher scoring students′ learning efficacy post-test in Experimental Group 1 is 86.29 while that in Experimental Group 2 is 80.86. The Levene test of homogeneity of variance is not significant (F=.044, p = .837 > .05). Based on the t-value and significance under the hypothesis of equal variance, both post-test results do not achieve a significant difference (t=-.959, p=.357 > .05). This result of Cohen's d as 0.499 represents a medium effect size. Namely, in the higher scoring students' post-test, students in both collaborative learning (through cognitive apprenticeship) and individual learning make progress in learning efficacy. However, these two approaches do not show significant differences. The mean of the lower scoring students' learning efficacy post-test in Experimental Group 1 is 61.71 while that of the Experimental Group 2 is 42.00. The Levene test of homogeneity of variance is not significant (F=.655, p = .434 > .05). Based on the t-value and significance under the hypothesis of equal variance, both post-test results do not achieve a significant difference (t=-2.180, p=.039 < .05). This result in Cohen's d of 1.165 represents a large effect size, meaning that in lower scoring students' post-test, collaborative learning (through cognitive apprenticeship) is apparently better than individual learning. The lower scoring students in collaborative learning (through cognitive apprenticeship) make greater progress (see Table 5). It is possible that in later levels of the E-game, there are more complicated questions requiring students to further understand the logic in the questions. Under collaborative learning (through cognitive apprenticeship), the students of higher and lower scores interact orally and non-orally. The lower scoring students are able to understand how to solve problems and further enhance their learning efficacy.

Analysis on learning attitude questionnaire

To understand the feelings and opinions of the two groups of students about the learning methods after the experiment. The researcher gives a learning attitude questionnaire through a Google form. The following are the statistical analysis on the results:

Platform operation

The dimension of "platform operation" is Question 1 to 5. The results are shown in Table 9. These questions focus on whether the interface and function of the E-game conforms to the students' learning schema and is easy to operate and learn.

Table 9 Statistical Analysis on Platform Operation

Questions \ Platform Operation	Experimental Group 1		Experimental Group 2	
	Mean	SD	Mean	SD
1. It is easy to log in to the E-game.	4.04	1.02	3.68	1.18
2. The operation interface of the E-game is clear.	4.00	0.87	3.84	0.90
3. I log in and use E-game at home.	2.52	1.33	3.24	1.74

4. I use E-game for online courses.	3.76	1.30	3.44	1.19
5. I learn programming building block courses with E-game.	3.80	1.35	4.04	1.06

Students in both collaborative learning and individual learning groups have positive feedback towards platform operation. As shown in Table 9, students in both Experimental Group 1 and 2 give nearly 3.5 points in general for platform operation, suggesting that most students are positive about platform operation. In "whether it is easy to log in the website," "whether the interface of the website is clear," and "whether one uses E-game for online course learning," we can see the scores in Experimental Group 1 are higher than those in Experimental Group 2. The group collaboration in Experimental Group 1 enhances interaction and in turn mobilize students' discussions. Yet, in "whether the user would log in and use the website at home," students in Experimental Group 2 have more willingness than those in Experimental Group 2. It might be because students in Experimental Group 1 encounter teaching limitations. They are reminded not to go ahead of the schedule so as to avoid being in different levels with other students in the next class and not be able to discuss together. Students in Experimental Group 2 do not have this limitation and are able to log in and learn more at home. Generally speaking, students in Experimental Group 1 feel better about the platform operation than those in Experimental Group 2 do.

Learning motivation

The dimension of "learning motivation" is Question 6-10 as in Table 10. These questions mainly explore whether the interface design of the E-game and course contents increase students' learning motivation.

Table 10 Statistical Analysis on Learning Motivation

Learning Motivation / Questions	Experimental Group 1		Experimental Group 2	
	Mean	SD	Mean	SD
6. It is fun to learn through E-game.	3.96	1.31	3.96	1.06
7. In comparison with general lectures, I prefer having an E-game class.	3.84	1.40	3.48	1.16
8. I like to clear more levels in E-game after class.	3.64	1.47	3.08	1.26
9. I hope to get a good grade in E-game.	4.36	1.11	4.04	0.98
10. I focus more in class than before.	4.00	1.00	3.80	0.96

In learning motivation, the ratings for the collaborative learning group were better than the individual learning group. As shown in Table 10, both Experimental Group 1 and 2 equally get 3.96 points in Question 6. In Question 7 "In comparison with general lectures, I prefer having an E-game class," Question 8 "I would like to clear more levels in E-game after class," Question 9 "I hope to get a good grade in E-game," and Question 10 "I focus more in class than before," the scores in Experimental Group 1 are better than those in Experimental Group 2.

According to the scores in Experimental Group 2's questionnaire, the results for the learning attitude questionnaire show double peaks. Students who are learning well through the E-game give more positive feedback while those encountering more obstacles give more negative feedback. The scores by Experimental Group 1 are in normal distribution. The feedback given by both higher and lower scoring students is more consistent, with most of them being positive. In Question 9, "I hope to get a good grade in E-game," Experimental Group 1 gives 4.36 while Experimental Group 2 gives 4.04, both of which is feedback that is above average. This might suggest that the E-game offers sufficient challenges and stimulation to the students. In order to improve their scores (using less building blocks or clearing a level faster), the students would clear the same level again to gain a better score. In the mean of Experimental Group 1, 4.36 points, 14 of the students give "5 points" (Strongly Agree), indicating that in the experiment course, the students are able to learn by continuously clearing levels and practicing to better themselves.

Problem-solving

The dimension of "problem-solving" is Question 11-15, as in Table 11. It mainly explores whether students are able to solve difficult problems by using the student manual or under the teacher's (master) guidance when encountering more difficult or complicated levels at E-game.

Table 11 Statistical Analysis on Problem-Solving

Problem-Solving / Questions	Experimental Group 1		Experimental Group 2	
	Mean	SD	Mean	SD
11. When encountering a problem, I break down the problem into several small questions.	3.76	0.93	3.68	1.38
12. I grasp the keywords of the problem to think about possible solutions.	4.00	1.29	3.60	1.12
13. I arrange steps of the possible solution in order.	3.92	1.35	3.96	0.79
14. I solve programming problems according to the problem-solving steps.	4.08	1.15	3.68	1.14
15. When a problem cannot be solved, I seek my classmate or teacher for help.	4.48	0.87	4.12	0.93

In the feedback for problem-solving, the collaborative learning group was better at problem-solving than the individual learning group. As shown in Table 11, Experimental Group 1 has higher scores in Question 11, 12, 14 and 15 than those of Experimental Group 2, indicating that students in Experimental Group 1 perform better in breaking down problems, arranging steps, systematic thinking, seeking help, etc. In Question 14, "I would solve programming problems according to the problem-solving steps," the higher scoring students would discuss with the lower scoring students about problem-solving strategies so that the lower scoring students would understand the logical thinking of problem solving and try to solve problems on their own. Through teaching, the higher scoring students can further ensure whether there is a better solution. In Question 15, "When a problem cannot be solved, I would seek my classmate or teacher for help," Experimental Group 1 and 2 have high scores, 4.48 and 4.12 respectively. Yet in the details of the questionnaire, 18 students in Experimental Group 1 give "strongly agree" feedback, higher than 9 students in Experimental Group 2. In class, some students dared not to ask the teacher questions. Under the guidance of collaborative learning, they are open to discussions with their peers, obviously higher than seeking help from the teacher. Therefore, in problem-solving, Experimental Group 1 has higher scores than Experimental Group 2.

Course contents

The dimension of "course contents" is Question 16-20, as in Table 12. It mainly examines whether the level design of the E-game meets the sequence and logic of programming education. After taking the course, the students are able to learn corresponding programming concepts from the levels.

Table 12 Statistical Analysis on Course Contents

Course Contents / Questions	Experimental Group 1		Experimental Group 2	
	Mean	SD	Mean	SD
16. In comparison to previous information courses, I like the E-game better.	4.12	1.19	3.84	1.43
17. After finishing the course, I still like to use E-game for other courses, such as English, mathematics, and science.	3.84	1.25	3.56	1.12
18. I take the initiative to search for more information and learn about E-game.	3.88	1.39	3.40	1.22
19. Through the course, I learn how to break down problems and think about solutions.	4.20	1.08	3.32	1.22
20. Through the E-game, I understand the logic of programming in life.	3.68	1.46	3.72	1.06

In course content ratings, the collaborative learning group was better than the individual learning group. As shown in Table 12, from Question 16-19, Experimental Group 1 has higher scores than Experimental Group 2. Besides programming, E-game also contains English, mathematics, science and other courses. Under the guidance of the website, students can extend their learning to other fields. That is also one of the purposes of the website. In Question 19, "Through the course, I can learn how to break down problems and think about solutions," Experimental Group 1 has 4.20 points, apparently higher than Experimental Group 2. It suggests that students think their skills to break down important questions and think systematically in class is enhanced more after discussions under group collaborative learning.

Collaborative learning

The dimension of "collaborative learning" is Question 21-25 (for Experimental Group 1 only) as in Table 13. It mainly explores students' learning attitude at the E-game under collaborative learning (through cognitive apprenticeship).

Table 13 Statistical Analysis on Collaborative Learning

Collaborative Learning / Question	Experimental Group 1	
	Mean	SD
21. I help my classmates to solve programming problems.	4.16	0.94
22. I speak out my opinion and confirm with my classmates if it is correct.	4.16	0.75

23. When our opinions are different, I discuss with my classmates to reach a consensus.	3.64	0.95
24. I cooperate with my classmates to clear levels in E-game.	3.92	0.86
25. I understand programming more easily after discussing with others.	3.88	1.13

According to the mean of Experimental Group 1 regarding the feedback for collaborative learning in Table 13, the students keep positive and affirmative attitudes toward collaborative learning (through cognitive apprenticeship), indicating that the students agree collaborative learning is helpful. Both higher and lower scoring students can interact much more under group collaborative learning than in traditional class. Question 22 "I would speak out my opinion and confirm with my classmates if it is correct" is an indicator that is difficult for traditional classes to make a distinction on. The composition of the students in a class is heterogeneous. It is difficult to hear the real thought of each student in class. However, through peer collaboration, it is faster and more effective for peers to communicate and solve problems with each other than to ask the teacher questions. This method helps students to understand their problems. Collaborative learning (through cognitive apprenticeship) substantially helps students with learning programming by E-game.

Individual learning

The dimension of "individual learning" is Question 26-30 (for Experimental Group 2 only) as in Table 14. It mainly examines the students' learning attitude to E-game using student individual learning manual as edited by the teacher.

Table 14 Statistical Analysis on Individual Learning

Individual Learning / Question	Experimental Group 2	
	Mean	SD
26. I find strategies to solve problems by using E-game student individual learning manual.	4.08	1.22
27. The E-game individual learning student manual helps me learn E-game courses better.	4.24	1.05
28. I like the feeling of controlling the course schedule on my own.	4.12	1.09
29. I study other courses (Chinese, mathematics) through individual learning.	4.16	1.28
30. After using the E-game individual learning student manual, I understand programming courses more easily.	3.64	1.60

As shown in Table 14, regarding the feedback of individual learning, most students in individual learning give positive feedback. The students feel positive about being able to control the course schedule on their own. When encountering problems, they can try to look them up in the materials (individual learning manual) or audio-visual material (website) and find solutions to their problems.

Analysis on group interaction

The students in collaborative learning (through cognitive apprenticeship) are divided into groups paired with one higher scoring student and one lower scoring student. Table 15 shows the interaction between members in the groups and the correlation with pre-test and post-test.

Table 15 Experimental Group 1 Higher and Lower Scoring Student Group Interaction and Pre-test and Post-test Analysis Chart

Group	Interaction in the two-people group		Higher Score Student				Lower Score Student			
	Group Type Communication Mode	Communica-tion Mode Schematic Diagram	Pre-test	Post-test	Progress	Mean Progress	Pre-test	Post-test	Progress	Mean Pro-gress
G1	The higher scoring student directly teaches the lower scoring student.	●→◎	85	91	6	9.33	20	42	22	2.67
G4			78	88	10		21	15	-6	
G11			61	73	12		47	39	-8	
G2	The lower scoring student would ask the higher scoring student immediately when they encounter a problem.	●←◎	80	97	17	-0.33	15	31	16	20
G3			79	65	-14		20	47	27	
G10			62	58	-4		42	59	17	
G5	Mutual communication between the two students.	●↔◎	77	91	14	16.16	31	65	34	40
G6			74	91	17		31	79	48	
G7			68	82	14		31	83	52	
G8			67	91	24		34	85	51	
G9			65	88	23		41	79	38	
G12			51	56	5		50	67	17	
G13	The two students do not communicate with each other.	●　◎	51	56	5	5	17	22	5	5

Notes. ●Higher scoring student ◎Lower scoring student

In these groups, the two students can have the best learning efficacy when they fully communicate with each other. The higher scoring student can make progress by 16 points while the lower scoring student can make progress by up to 40 points. When the higher scoring student directly teaches the lower scoring student, it is beneficial for the learning of the higher scoring student, making progress by more than 9 points. When the lower scoring student asks the higher scoring student immediately when encountering a problem, it is beneficial for the learning of the lower scoring student, who makes progress by 20 points. When the two students do not communicate with each other, there is little progress.

Table 16 shows the analysis on Group 8 (mutual communication mode)'s level clearing record. From level 3, 4, 6, 12, we can see that the lower scoring students have difficulties in these levels. They stay in level 12 for the longest, 208 seconds. When actively discussing with the higher scoring student after encountering difficulties, (s)he clears the level smoothly. In level 13, both the higher and lower scoring students encounter the same problem and stay for longer periods of time. However, both stay at the same level for a similar length of time. The interview record is analyzed as the following:

G8SH: We first looked for the same. Then we completed one part first to see how many times we need to repeat. Then we wrap it with building blocks repetitively.

G8SL: I would discuss with him how to do it. If he did not know, I would do it first on my own.

G8SH: We would discuss together to figure out how to clear the level.

G8SL: If I had problems, I would ask my classmate. He would answer my questions.

According to the above-mentioned talk, the students indeed communicated mutually with each other. The higher scoring student in Group 8 fails the level 30 times while the lower scoring student fails 57 times, less than the whole class's average failure number, 133 times. It suggests that in the group, they can first discuss difficult levels before solving problems in order to reduce errors. The progress the group makes is more prominent than other groups. We can infer that in collaborative learning, good communication and interaction together with active inquiry and guidance can make learning more effective (Level 13 see Figure 2).

Table 16 Statistical Analysis on Group 8 Level Clearing

Group 8	Higher Score Student				Lower Score Student			
Level	Number of Stars	Number of Errors	Error Duration (Second)	Number of Correct Answers	Number of Stars	Number of Errors	Error Duration (Second)	Number of Correct Answers
1	3	0	0	2	3	0	0	2
2	3	0	0	1	3	0	0	2
3	3	0	0	1	2	4	190	2
4	3	0	0	1	2	2	117	2
5	3	3	81	3	2	1	34	2
6	3	0	0	2	2	2	173	1
7	3	0	0	1	2	0	0	1
8	2	0	0	1	2	0	0	1
9	2	1	32	1	2	1	26	1
10	2	0	0	1	2	0	0	1
11	3	0	0	1	3	0	0	1
12	3	0	0	1	2	2	208	1
13	3	11	986	2	2	7	1442	1
14	3	3	363	2	2	1	319	1
15	3	2	100	3	1	8	288	1
16	3	0	0	1	3	0	0	1
17	3	3	61	2	2	3	118	1
18	3	0	0	2	2	2	211	1
19	3	0	0	1	1	3	351	1
20	3	0	0	1	1	2	324	1
21	2	0	0	1	2	2	111	1
22	2	7	1685	2	0	14	1226	0
Total	61	30	3312	33	43	57	4123	26
Average	2.773	1.364	150	1.500	1.955	2.591	656	1.182

Fig. 2 E-game—Saving Lilija Level Schematic Diagram (Level 13)

Conclusion and discussion

Effect of CT in E-game

Both collaborative learning (through cognitive apprenticeship) and individual learning could effectively enhance the learning efficacy of students' cultivation of CT in E-game. This study examined two approaches to learning programming using the E-game platform for a five-week course. The quantitative analysis on the pre-test and post-test of "E-game CT test" suggested that both collaborative learning (through cognitive apprenticeship) and individual learning could effectively enhance the learning efficacy of students' cultivation of CT in E-game. From a qualitative exploration using the E-game "Programming Learning Attitude Questionnaire," both groups of students showed positive feedback about collaborative learning and individual learning, indicating that the students' learning attitude was positive regarding learning CT in an E-game.

Effect of CT in E-game among students with different achievements

In collaborative learning (through cognitive apprenticeship), the lower scoring students could enhance their learning efficacy in the cultivation of CT skills at E-game more effectively than the higher scoring students. According to the results of pre-test and post-test by the higher and lower scoring students in collaborative learning (through cognitive apprenticeship), the learning efficacy of the higher scoring students showed no difference while that of the lower scoring students showed significant differences. In collaborative learning (through cognitive apprenticeship), the lower scoring students could enhance their learning efficacy in the cultivation of CT skills at E-game more effectively than the higher scoring students.

Effect of CT in E-game between two lower-scoring student groups

For the two lower scoring student groups in E-game, those in collaborative learning (through cognitive apprenticeship) had more prominent learning efficacy than those in individual learning. There was no significant difference in the post-test scores of the higher scoring students between collaborative learning (through cognitive apprenticeship) and individual learning. One of the possible reasons is the higher scoring students may frequently manipulate any tool for learning. But we need more evidence in the future. However, the post-test scores of the lower scoring students in these two groups (collaborative learning and individual learning) showed significant differences, suggesting that for the lower scoring students in E-game, collaborative learning (through cognitive apprenticeship) was a better learning method for them than individual learning. It was possible that when taking programming courses, the lower scoring students needed more scaffolding assistance for the construction of CT logical concepts. Although individual learning could also enhance learning efficacy, the lower scoring students' Chinese skills influenced their reading and thinking. They could not understand the Chinese words in the student learning manual and could not think effectively. In collaborative learning (through cognitive apprenticeship), the lower scoring students could interact with their peers under group discussions orally or non-orally in languages they were more familiar with. They could resonate more with learning through conversation and practice and further enhanced their learning efficacy.

This paper presented a case study on how students learned CT skills through a coding game in collaborative learning (through cognitive apprenticeship) or individual learning. Cognitive apprenticeship involves two people working side by side on one computer and collaborating closely to create a program. Compared with students who programmed independently, paired students were more likely to submit higher-quality solutions for their programming tasks (McDowell et al., 2002). The findings of Iskrenovic-Momcilovic's (2019) research showed that beginners in pair programming performed better than students who programed alone, and Leow & Huang (2021) found that if a lower-scoring student was paired with another lower-scoring student, the progress in the learning achievement for both students was not as great as the learning achievement of a pair that consisted of a higher-scoring student and a lower-scoring student. This study planned to create effective pairs (a high-scoring student paired with a low-scoring student). When pairs possess different levels of experience of programming skills, both students will benefit (Leow & Huang, 2021). This observation reminded us that when pairing up students to work together, good communication between the partners could help facilitate the collaborative process (Sullivan & Wilson, 2015). Chang et al. (2017) brought out a situational game-based learning system in their research, introducing the cognitive apprenticeship approach into fostering the students' CT skills of learning sorting algorithms. It not only helps to develop students' CT skills, but also helps to elevate students' learning motivation without causing extra cognitive burden on students. The characteristics of the learning platform, such as interactivity and reward systems (the E-game platform in our case), may encourage learners' participation and motivation (Buckley & Doyle, 2016; O'Rourke et al., 2016). This study showed that E-game, a graphical block-based coding game, makes the process of learning CT more interesting and challenging for students. The programming learning platform integrated with role-play games motivates students to learn. Both collaborative learning (through cognitive apprenticeship) and individual learning effectively enhance the learning effects of students' development for CT skills through E-game coding education. Furthermore, we've compared the learning performance between two lower-scoring groups in this research, who needed more scaffolding to learn. Make the students who applied collaborative learning through cognitive apprenticeship to work and discuss with the higher scored students can truly cut down their cognitive burden, and achieve better performance. It can be inferred from this finding that in collaborative learning, good communication and interaction together with active inquiry and guidance can make learning more effective. Overall, our findings suggest that students with lower scores who programmed with a higher-scoring partner learned more than when they programmed alone.

The results of the current study should be interpreted in light of certain limitations. The sample was restricted to primary students in Taiwan. This study focuses on the fifth graders' learning through collaborative learning (through cognitive apprenticeship) and individual learning in E-game, and examines the learning efficacy of programming in the cultivation of CT skills. It is suggested that future studies can extend to the third and fourth graders or compare the difference in the cultivation of CT skills between different programming environments, such as Scratch and E-game.

When thinking about learning programming, we can consider its connection with other academic subjects. The Dako Magic Village Level in Dako Island unit incorporates programming activities that involve lines and angles. When designing courses, we can try to combine it with the mathematical courses that teach lines and angles for further studies on interdisciplinary learning.

References

Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. *Journal of Educational Technology & Society, 19*(3), 47-57.

Australian Curriculum, Assessment and Reporting Authority(ACARA) (2013). *Draft Australian curriculum: technologies foundation to year 10.* http://docs.acara.edu.au/resources/Draft_Australian_Curriculum_Technologies_-_Consultation_Report_-_August_2013.pdf

Bers, M. U. (2019). Coding as another language: A pedagogical approach for teaching computer science in early childhood. *Journal of Computers in Education 6*(4), 499–528. https://doi.org/10.1007/s40692-019-00147-3

Bishop-Clark, C., Courte, J., & Howard, E. V. (2006). Programming in pairs with Alice to improve confidence, enjoyment, and achievement. *Journal of Educational Computing Research, 34*(2), 213-228.

Brown, J. S., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32-42.

Brown, J. A. S. (2017). *The use of reflective questioning as a peer coaching strategy in an asynchronous online cognitive apprenticeship* (Doctoral thesis, Old Dominion University). https://digitalcommons.odu.edu/cgi/viewcontent.cgi?article=1027& context =stemps_etds

Brusilovsky, P., & Spring, M. (2004). Adaptive, engaging, and explanatory visualization in a C programming course. In L. Cantoni & C. McLoughlin (Eds.), *Proceedings of World Conference on Educational Media, Hypermedia, and Telecommunications 2004* (pp. 1264-1271). Chesapeake: VA: AACE.

Buckley, P., & Doyle, E. (2016). Gamification and student motivation. *Interactive Learning Environments, 24*(6), 1162–1175.

Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research,* 0034654317710096.

Burton, E. P., Rich, P., Cleary, T., Burton, S., Kitsantas, A., Egan, G., & Ellsworth, J. (2020). Thinking big: Using computational thinking for data practices in high school science. *The Science Teacher, 87*(6), 30-36.

Chang, S. C., Hsu, T. C., & Su, P. C. (2017). The effectiveness of the students using a situational game based on cognitive apprenticeship for learning sorting computational thinking. *Journal of Education Research, 2017*(282), 43-58.

Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education, 109*, 162-175. https://doi.org/10.1016/j.compedu.2017.03.001

Computer Science Teachers Association (CSTA) (2017). *CSTA K–12 computer science standards*. https://www.csteachers.org/page/standards

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Erlbaum.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453-494). Lawrence Erlbaum Associates, Inc.

Dağ, F. (2019). Prepare pre-service teachers to teach computer programming skills at K-12 level: experiences in a course. *Journal of Computers in Education 6*(2), 277–313. https://doi.org/10.1007/s40692-019-00137-5

Dasgupta, S., Hale, W., Monroy-Hernández, A., & Hill, B. M. (2016, February). *Remixing as a pathway to computational thinking*. Paper presented at the Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing, San Francisco, CA, USA.

Ebel, R. L., & Frisbie, D. A. (1991). *Essentials of educational measurement* (5th ed.). Prentice Hall.

European Commission. (2016). *JRC science for policy report: Developing computational thinking in compulsory education*. https://publications.jrc.ec.europa.eu/ repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf

Felleisen, M., Findler, R. B., Flatt, M., & Krishnamurthi, S. (2004). The teachscheme! Project: Computing and programming for every student. *Computer Science Education, 14*(1), 55-77.

Figueiredo, J. A. Q. (2017). How to improve computational thinking: A case study. *Education in the Knowledge Society, 18*(4), 35-51.

Garcia-Cabrero, B., Hoover, M., Lajoie, S., Andrade-Santoyo, N., Quevedo-Rodríguez, L., & Wong, J. (2018). Design of a learning-centered online environment: a cognitive apprenticeship approach. *Educational Technology Research and Development, 66*(3), 813-835.

Giordano, D., & Maiorana, F. (2014, April). *Use of cutting edge educational tools for an initial programming course*. In Global Engineering Education Conference (EDUCON), 2014 IEEE (pp. 556-563). IEEE.

GOV.UK (Department for Education) (2013). *National curriculum in England: Computing programmes of study*. https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study.

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43.

Gupta, N., Tejovanth, N., & Murthy, P. (2012, January). *Learning by creating: Interactive programming for Indian high schools.* In Technology Enhanced Education (ICTEE), 2012 IEEE International Conference on (pp. 1-3). IEEE.

Iskrenovic-Momcilovic, O. (2019). Pair programming with scratch. *Education and Information Technologies, 24*(5), 2943–2952.

Jenkins, T. (2002). *On the difficulty of learning to program*. Paper presented at the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, Loughborough University, United Kingdom.

Kafai, Y. B. (2016). Education from computational thinking to computational participation in K-12 education seeking to reframe computational thinking as computational participation. *Communications of the ACM, 59*(8), 26-27.

Lai, C. S., & Lai, M. H. (2012, June). *Using computer programming to enhance science learning for 5th graders in Taipei*. In Computer, Consumer and Control (IS3C), 2012 International Symposium on (pp. 146-148). IEEE.

Larkins, D., Moore, J., Rubbo, L., & Covington, L. (2013). *Application of the cognitive apprenticeship framework to a middle school robotics camp*. Proceedings of the 44th ACM technical symposium on computer science education, 89-94. https://doi.org/10.1145/2445196.2445226

Leow, W. M. F., & Huang, W. (2021). Exploring the effectiveness of pair programming in developing students' computational thinking skills through scratch. In Looi, C.K., Wadhwa, B., Dagiené, V., Seow, P., Kee, Y.H., & Wu, L.K. (Eds.), *Proceedings of the 5th APSCE International Computational Thinking and STEM in Education Conference 2021* (pp. 2-7). Singapore: National Institute of Education.

Lewis, C. M. (2010, March). *How programming environment shapes perception, learning and goals: logo vs. scratch*. In Proceedings of the 41st ACM technical symposium on Computer science education (pp. 346-350). ACM.

Liu, M.-C. (2017). Maker education, computational thinking and programming. *Taiwan Educational Review Monthly, 6*(1), 138-140.

Lye, S. Y., & 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.

Mannila, L., Peltomaki, M., & Salakoski, T. (2006). What about a simple language? Analyzing the difficulties in learning to program. *Computer science education, 16*(3), 211-227.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education, 10*(4), 16:1-15.
McDowell, C., Werner, L., Bullock, H., & Fernald, J. (2002). The effects of pair programming on performance in an introductory programming course. *Proceedings of the Thirty-Third Technical Symposium on Computer Science Education*, (pp. 38-42). ACM Press.
National Research Council. (2010). *Committee for the workshops on computational thinking: Report of a workshop on the scope and nature of computational thinking*. National Academy Press. https://doi.org/10.17226/12840
O'Rourke, E., Peach, E., Dweck, C. S., & Popovic, Z. (2016). *Brain points: A deeper look at a growth mindset incentive structure for an educational game*. https://dl.acm.org/doi/pdf/10.1145/2876034.2876040
Parmar, D., Babu, S. V., Lin, L., Jörg, S., D'Souza, N., Leonard, A. E., & Daily, S. B. (2016, August). Can embodied interaction and virtual peer customization in a virtual programming environment enhance computational thinking? In *Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), 2016* (pp. 1-2). IEEE.
Preston, D. (2005). Pair programming as a model of collaborative learning: A review of the research. *Journal of Computing Sciences in colleges, 20*(4), 39-45.
Rich, K. M., Binkowski, T. A., Strickland, C., & Franklin, Diana. (2018). *Decomposition: A K-8 computational thinking learning trajectory*. Proceedings of the 2018 ACM Conference on international computing education research, 124-132. https://doi.org/10.1145/3230977.3230979
Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education, 17*(1), 77-92.
Selby, C. (2014). *How can the teaching of programming be used to enhance computational thinking skills?* (Doctoral dissertation, University of Southampton, Southampton, UK). https://www.researchgate.net/profile/Cynthia-Selby/publication/ 299464834_How_can_the_teaching _of_programming_be_used_to_enhance_computational_thinking_skills/links/5bb9bca74585159e8d87c69a/How-can-the-teaching-of-programming-be-used-to-enhance-computational-thinking-skills.pdf
Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22*, 142–158.
Standl, B. (2016). *A case study on cooperative problem solving processes in small 9th grade student groups*. Paper presented at the IEEE Global Engineering Education Conference (EDUCON), 2016.
Sullivan, F. R., & Wilson, N. C. (2015). Playful talk: Negotiating opportunities to learn in collaborative groups. *The Journal of the Learning Sciences, 24*(1), 5–52.
U.S. Department of Education (2016). *Every Student Succeeds Act*. http://www.ed.gov/essa?src=policy
Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies, 20*(4), 715-728.
Wang, H. Y., Huang, I., & Hwang, G. J. (2014, August). *Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses*. In Advanced Applied Informatics (IIAIAAI), 2014 IIAI 3rd International Conference on (pp. 382-387). IEEE.
Werner, L., & Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education, 42*(1), 29-49.
White House (2016). *Computer Science for All*. https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A, 366*, 3717–3725.
Wu, C.-S., & Lin, T.-Y. (2002). Cognitive apprenticeship. *Journal of Education Research, 99*, 148.
Ying, L. (2016, October). *Teaching programming based on Computational Thinking*. Paper presented at the 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA.