Education and Information Technologies
https://doi.org/10.1007/s10639-022-10915-x

The effectiveness of unplugged activities and programming exercises in computational thinking education: A Meta-analysis

Feng Li[1] · Xi Wang[1] · Xiaona He[2] · Liang Cheng[1] · Yiyu Wang[1]

Received: 22 August 2021 / Accepted: 23 January 2022

Abstract

This study adopted a meta-analysis to explore the effectiveness of unplugged activities (UA) and programming exercises (PE) teaching approaches on computational thinking (CT) education. Through a two-stage literature collection and selection process, 29 articles were included in the meta-analysis, 31 independent effect sizes (16 of UA and 15 of PE) from these articles were used, and a total of 2,764 participants were involved in these studies. CMA software version 3.3 was used to analyze the collected data. The result of the meta-analysis showed that both the UA and PE teaching approaches are useful in cultivating students' CT. Besides, the effect of the PE teaching approach is better than the UA teaching approach in CT education. Moreover, we analyzed the effect of moderator variables (grade level, interdisciplinary course, and experiments duration) on the relationship between UA or PE and CT education. The results showed that the effects of UA teaching approach in CT education was stronger (a) for primary school students than for secondary school students, (b) in interdisciplinary courses than in computer science courses, (c) with long duration teaching experiments than with medium and short duration teaching experiments. However, these effects are not significant. The effects of the PE teaching approach in CT education were stronger (a) for secondary school students than for primary school students, (b) in interdisciplinary courses than in computer science courses, (c) with short duration teaching experiments than with long and medium duration teaching experiments. These effects are not significant either. Therefore, we suggest that (1) the UA teaching approach should be used more for primary school students, while the PE teaching approach should be used more for secondary students; (2) CT education should be integrated into other subjects through UA and PE teaching approaches, and (3) the UA teaching approach requires more teaching time than the PE teaching approach does in CT education.

Extended author information available on the last page of the article

Keywords Computational thinking · Teaching approaches · Unplugged activities · Programming exercises · Meta-analysis

1 Introduction

With the development of computer and network technology, the Internet of Things, mobile terminals, and artificial intelligence became deeply integrated with various aspects of society and created a brand-new digital environment. The digital environment has changed the way things are done today by extending the power of human thought with computer and other digital tools (Barr et al., 2011). Whether in e-learning, online offices, or internet shopping, to efficiently apply these new tools, people need to use computational thinking (CT) to analyze requirements, design steps, process data, and optimize the process. CT has become important for people's everyday lives in the digital age (Gretter & Yadav, 2016; Grover & Pea, 2018).

In 2006, Wing introduced CT as a fundamental skill which involved problem solving, system design, and understanding human behavior by drawing on the concepts fundamental to computer science, emphasizing that it was just as important for every child as reading, writing, and arithmetic (Wing, 2006). This argument caught the attention of a broad academic community, and more detailed and operational CT definitions have since been proposed. The Computer Science Teachers Association and the International Society for Technology in Education have developed an operational definition that listed CT practice operations, which include formulating problems, logically organizing and analyzing data, representing data, automating solutions through algorithms, and so on (CSTA & ISTE, 2009). Computing at School (CAS) noted that CT, which involves conceptual and operational aspects, was the process of identifying calculations in our surrounding world, as well as the process of applying computing tools and technical understanding, inferring natural and artificial systems (CAS, 2015). Although there is no consensus on a definition of CT, a set of key skills of CT, such as abstraction, algorithmic thinking, decomposition, automation and generalization have been identified.

When CT was seen as a form of empowerment, some countries and regions adopted its use in schools (Bocconi et al., 2016; Moreno-León et al., 2017). In 2014, England implemented the computing curriculum in primary and secondary schools in hopes that students would use CT and creativity to understand and change the world as a result of the high-quality computing education (GOV.UK, 2014). In Australia, CT education was integrated into the technologies curriculum, aiming to develop students' competencies in using computational thinking and the key concepts of abstraction, data collection, representation and interpretation, specification, algorithms and implementation to create digital solutions (Falkner et al., 2014). The International Society for Technology in Education revised the "Technical Standards for Students", in 2016 highlighting the importance of CT in school education (ISTE, 2016). With the extensive development of CT education in schools, CT teaching approaches have become a hot research topic (Kuo & Hsu, 2020; Delal & Oner, 2020). In the current study, we investigate and discuss two kinds of mainstream CT teaching approaches:

unplugged activities (UA) and programming exercises (PE), then we compare their teaching effect through a meta-analysis.

2 Literature Review

2.1 Computational thinking teaching approaches

In teaching implementation, there are two main teaching approaches for CT education in school: unplugged activities (UA) and programming exercises (PE) (Brackmann et al., 2017; Olmo-Muoz et al., 2020). Different researchers have different views on the two teaching approaches. Some researchers have suggested that CT should be developed through logic games, physical activities, and card calculation without a computer programming environment (Curzon et al., 2014; Sentance & Csizmadia, 2015; Kuo & Hsu, 2020). Some researchers think that CT needs to be developed better through computer programming practice (Brennan & Resnick, 2012; Berland & Wilensky, 2015; Rose et al., 2017). In addition, some scholars consider the mixed approach combining UA and PE to be more conducive to the development of students' CT (Olmo-Muoz et al., 2020). The current study, therefore, aimed to identify the effectiveness of the teaching approaches in CT education and analyze the influences of the moderator variables on the relationship between the teaching approaches and students' CT development.

2.1.1 The UA teaching approach

Unplugged activities are commonly described as "learning computer science without a computer" (Bell et al., 1998). In the past 20 years, some educators have regrouped according to different cultures, learning environments, and learner characteristics (such as age and special needs), and they have also designed new activities (Huang & Looi, 2021). At present, "unplugged" refers to any activity or teaching strategy that has some characteristics in common with the original unplugged activities which is carried out through indoor or courtyard games, mechanical toys, riddles, cards, pen and paper, etc. (Brackmann et al., 2017; Caeli & Yadav 2020; Miguel, 2019). Bell et al. (2012) described the implementation of a well-known UA "classification network": helping students to understand the algorithm in the game and promoting "observation, questioning, critical thinking, and reflection." Kim et al. (2013) adopted the paper-and-pencil writing strategy to carry out teaching. Non-computer students improved the understanding and use of CT by expressing an idea logically through some representations created with pen and paper. Threekunprapa and Yasri (2020) designed the UA teaching approach by using flow blocks. Students must connect the provided process blocks, including a set of ready-made grammars (start, end, and repetition times). The goal of this card-based game is to deliver computer science concepts and train players to solve problems. The UA teaching approach emphasizes the basic algorithms and process design of problem-solving, which are the foundation of CT skills. This approach allows CT education to be carried out in most areas, especially in schools lacking basic technology infrastructure (Faber et al., 2017; Manabe et al., 2011; Unnikrishnan et al., 2016). However, some researchers inferred that CT education which only covered basic computer science concepts and skills may lack the potential to develop more complex CT skills (Bell & Vahrenhold, 2018; Threekunprapa & Yasri, 2020).

2.1.2 The PE teaching approach

There is a great amount of teaching work surrounding CT that is focused on the application of digital technology, especially in programming environments (Yadav et al., 2017). The choice of programming language is based on the principle of "low floor, high ceiling" (Grover & Pea, 2013), and teaching tools should have the characteristics of good portability, support for fairness, a strong system, and sustainability (Repenning et al., 2010). The PE teaching approach can be divided into four categories: graphical/modular programming languages (logo, Scratch, Alice, ToonTalk, App Inventor), web-based simulation creation tools (Agentsheets, Agentcubes, Caspio), open-source hardware devices (Arduino, GoGo Board, Little Bits), and high-level programming language (Python, Ruby, Java) (Fagerlund et al., 2021; Lye & Koh, 2014; Hsu et al., 2018). Kahn et al. (2011) used ToonTalk language to construct calculations models and procedures to allow students to explore topics in mathematics and science. Witherspoon et al., (2017) adopted the online robotics curriculum, developed by Carnegie Mellon University. The curricular materials capitalize on the engaging aspects of robotics competitions while emphasizing the practice of specific programming skills. Leonard et al. (2020) had developed a program that integrates choreography, computer programming, and virtual environments to cultivate CT, which broadens learning for more diverse students. This approach enables students to participate in the construction of multimedia digital products to express their ideas and cultivate students' CT in the technological world (Hague & Payton, 2011; Sung et al., 2017). The PE teaching approach focuses more on the comprehension of abstract concepts such as variables, conditions, and cycles, which supports students verifying their idea in an automatic way (Ince & Koc, 2021). However, sometimes it tends to emphasize students' acquisition of programming knowledge and skills while ignoring the development of mental thinking skills (Hsu et al., 2018; Li et al., 2020).

2.1.3 Problems and controversies of the two teaching approaches

Which approach—UA or PE—is more effective at cultivating students' CT? Some researchers have carried out experimental investigations. The results obtained are inconsistent, sometimes even showing opposite outcomes. Some studies have proven that, compared with PE, UA can stimulate learners' enthusiasm, enhance learners' learning motivation, facilitate multiple types of learning feedback during teaching, and improve students' CT (Brackmann et al., 2017; Curzon et al., 2014; Tsarava et al., 2018). Another part of the research results, however, showed that activities using high-tech computing devices and programming were thought to generate more chances to help students practice CT skills than low-tech and unplugged activities, and thus, PE would be more effective at developing students' CT (Black et al., 2013; Kalelioglu et al., 2016; Taub et al., 2012). In summary, the results emphasized the need for more research to provide systematic pieces of evidence to compare the effectiveness of UA and PE in CT education.

2.2 Moderator variables

Different variables may have different influences on the development of CT. We considered the different influences in grade level, interdisciplinary, and experiment duration.

2.2.1 Grade level

Wing (2006) argued that CT was a fundamental skill for everyone, not just for computer scientists, and that CT contained important learning content for K-12 students. In particular, many studies have emphasized the importance of starting CT education at an early age (Nouri et al., 2020; Nardelli, 2019; Kalelioglu & Gülbahar, 2014). Does this mean that teaching students CT in younger grades will lead to a greater improvement in CT? Some studies have explored the influence of grade level on the cultivation of CT. Atmatzidou & Demetriadis (2016) discovered that students' behavior in the various specific dimensions of CT has age-relevant differences. Conde et al. (2017) designed a CT educational experiment involving primary and secondary school students. This experiment proved the effectiveness of the CT teaching approach and reflected the difference among grade levels (g of primary students is 0.097; g of secondary students is 0.043).

2.2.2 Interdisciplinary

In addition to computer science classes, CT education is carried out in interdisciplinary courses, such as in physics classes, mathematics classes, etc. Can different course contexts have an impact on the improvement of students' CT? By analyzing previous studies, some course contexts were found to be relevant to science courses (Conde et al., 2017; Witherspoon et al., 2017; Hooshyar et al., 2021b), and the effect size of CT enhancement was not as high as that of interdisciplinary courses (Basu et al., 2017; Leonard et al., 2020; Ince & Koc, 2021). Therefore, we assumed that it is better to cultivate CT in interdisciplinary courses.

2.2.3 Duration

The duration of the teaching experiments on cultivating students' CT varies from one class (2–3 h) to one academic year (48 h). Does this have an impact on the results of CT education? In a meta-analysis article about Scratch and Arduino's cultivation of CT, it was found that the duration of experiments had a positive regulating effect on students' CT skills (Fidai et al., 2020). The next question is whether the development of CT through UA is also affected by the duration of the experiment. Similarly, evidence was found in several CT articles. Threekunprapa and Yasri's (2020) experiment lasted for 3 h, and its effect size ($g = 0.25$) was smaller than that of Jun et al. (2017) experiment ($g = 0.66$), which lasted for 15 h.

2.3 Purposes of the study

The purpose of the study is to focus on the effectiveness of teaching approaches in CT education based on the existing experiment and quasi-experiment studies. Specifically, the purpose is as follows:

(1) To examine which teaching approaches are more effective in cultivating CT between UA and PE teaching approaches.

(2) To explore the influences of moderator variables on the relationship between UA or PE teaching approaches and CT education.

3 Method

Meta-analysis is a statistical technique that integrates the results of multiple independent studies to obtain consolidated findings, which systematically considers the influence of participants, publication bias and other factors on the results of the analysis (Crombie & Davies, 2009). As a supplement to qualitative methods, some academic journals and research institutes encouraged the researchers to use meta-analysis to further explore the research questions on the basis of previous research data. UA and PE are used as the main teaching approaches for the cultivation of students' CT, while the previous studies provided research results. However, some results are inconsistent and even produced opposite results. To achieve the above study purposes, this study utilized a meta-analysis to examine the effectiveness of the UA and PE teaching approaches and explore the influences of moderator variables.

3.1 Literature search

To locate studies on UA and PE teaching approaches in CT education, we systematically searched for relevant literature using electronic databases such as Web of Science, EBSCO, Taylor & Francis, ScienceDirect, and Springer. Then, we collected all the relevant literature that has been published in journals or conferences with the help of Google Scholar and our university library databases. The range of time of publication for the collected literature was from 2006, when CT became a topic of interest, to 2021. Indexed keywords used in the literature search process included "programming," "coding," "unplugged," and "computational thinking," Exclusion was used to filter the collected studies written in English.

3.2 Literature exclusion criteria

Through a two-stage literature selection process, we determined the final articles that would be included in the subsequent meta-analysis. During stage 1, we screened the titles and abstracts to exclude duplicate articles and articles with inconsistent themes. During stage 2, we screened the full articles and included articles that fit the following criteria: (a) the article analyzes the relation between UA or PE and students' CT development; (b) the article provides quantitative data (e.g., standard deviation, mean value, and sample size) the experiment and quasi-experiment that can be calculated into the effect size; In this study, the distinction between experiment and quasi-experiment literature was based on whether the participants were randomly arranged into an experiment group or a control group (Rogers & Revesz, 2020). If the participants were randomly arranged into two groups, it was classified as experimental literature. Otherwise, it was classified as quasi-experimental literature; (c) the article measures students' CT. (see Fig. 1).

3.3 Coding study

First, two researchers selected available articles separately and had acceptable inter-rater reliability (Cronbach's $\alpha = 0.93$), and they consensually settled divergences via repeated discussion from both sides. Finally, we eliminated 369 articles that met the exclusion criteria and selected 29 available articles from 2011 to 2021. Among these articles, there were five experimental studies and twenty-four quasi-experimental studies.

We then coded the 29 articles according to their features and recorded the following data in each article: author(s) and publication year, grade level, experimental duration, whether the experiment was performed in an interdisciplinary course, and the number of participants. We finished the coding process according to the following criteria: (a) effect sizes were recorded for each independent sample within a study; (b) if a study included two or more experiments, we calculated a single experiment as one effect size; (c) if a study reported the correlation between UA or PE and multiple components of CT (e.g., sequencing, repeats, conditionals and debugging), we used their integrated effect size rather than separate effect sizes. After the coding, we determined 31 independent effect sizes from the selected 29 articles.

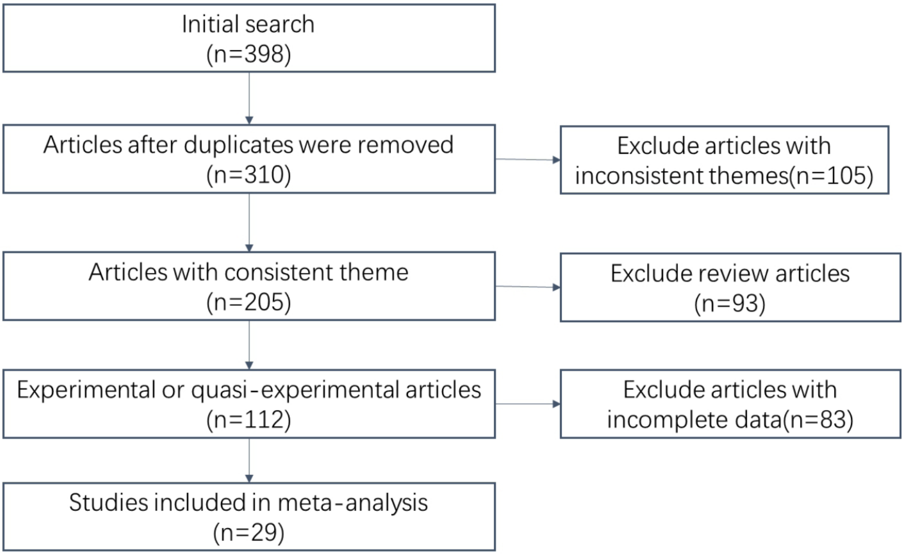

Fig. 1 Flow chart of the study selection

After completing the coding process under the guidance of meta-analysis principles, we calculated effect sizes between UA or PE and CT for each sample (Lipsey & Wilson, 2001). We tested whether the associations between UA or PE and CT were affected by the following moderator variables: (a) grade level; (b) whether the experiment was performed in an interdisciplinary course; (c) experimental duration. Grade level was coded as *primary education* (K-grade 5), *secondary education* (grade 6–12), or *college education* (university and higher vocational college). Interdisciplinary was coded as "yes" or "no" ("yes" means CT education integrated into physics, geography, language, and another course; "no" means purely computer science). The duration was coded as *long* (eleven weeks or more), *middle* (between four weeks and ten weeks), or *short* (three weeks or less). For coding results, see Table 1.

3.4 Data analysis

Comprehensive Meta-Analysis 3.3 (CMA 3.3) was used for the meta-analysis. The homogeneity test was calculated via Cochrane's Q and I^2, the formula of Q is: (Hedges & Olkin, 1985)

$$Q = \sum w_i (T_i - \bar{T})^2 \tag{1}$$

In this formula, w_i refers to the weighting factor for the i^{th} study, $\bar{T}$ calculated by Eq. (2). And T_i is the i^{th} effect estimator of k studies (i = 1,2…, k); in this paper, T_i stands for g effect index.

$$\bar{T} = \frac{\sum_i w_i T_i}{\sum_i w_i} \tag{2}$$

The formula of I^2 is (k is the number of effect size): (Higgins & Thompson, 2002)
I^2=

$$I^2 \begin{cases} \frac{Q-(k-1)}{Q}, & Q > (k-1) \\ 0, & Q \le (k-1) \end{cases} \tag{3}$$

According to Q and I^2, we decide whether we should use a random-effects model or a fixed-effects model. The corresponding p value of Q is <0.05, indicating the existence of heterogeneity. The larger I^2 is, the greater the heterogeneity is. If heterogeneity exists, the random-effects model should be chosen. The mean effect size is calculated using the average weight (within- and between- inverse variance weights) correlation coefficients of independent samples. Besides, we tested the effects of the following moderator variables: grade level, interdisciplinary, and duration.

There are multiple classifications of effect size tests in the literature (Cohen, 1992; Lipsey & Wilson, 2001; Thalheimer & Cook, 2002). In this study, we used

Table 1 Studies included in the meta-analysis

Name(year)	g	N	Grade level	Duration	Approach	Interdisci-plinary
Aslan, 2020	0.55	121	secondary	short	PE	yes
Atmatzidou, 2014	0.86	35	secondary	middle	PE	no
Atmatzidou, 2016	0.20	89	secondary	long	PE	no
Atmatzidou, 2016	0.76	75	college	long	PE	no
Grover et al., 2015	0.62	52	secondary	middle	PE	no
Hutchins et al., 2020	0.43	68	secondary	long	PE	yes
Ince, 2021	0.57	32	secondary	middle	PE	yes
Kim et al., 2013	0.56	110	college	long	PE	yes
Kwon et al., 2011	0.37	89	college	short	PE	no
Leonard et al., 2020	0.93	116	primary	long	PE	yes
Noh, 2020	0.37	155	primary	long	PE	no
Rodríguez-Martínez et al., 2020	0.01	47	primary	middle	PE	yes
Witherspoon et al., 2017	0.19	364	primary	middle	PE	no
Yin et al., 2019	1.36	32	secondary	short	PE	yes
Zha et al., 2020	1.24	64	secondary	middle	PE	yes
Basu et al., 2016	0.61	15	secondary	short	UA	yes
Basu et al., 2017	0.61	98	secondary	short	UA	yes
Brackmann et al., 2017	0.55	73	primary	middle	UA	no
Choi et al., 2017	0.35	82	primary	long	UA	no
Conde, 2017	0.10	84	primary	short	UA	no
Conde, 2017	0.04	24	secondary	short	UA	no
Hooshyar, 2021a	0.28	78	primary	short	UA	no
Hooshyar, 2021b	0.53	79	primary	short	UA	no
Jun et al., (2017	0.66	87	primary	long	UA	no
Kuo, 2020	0.34	52	secondary	middle	UA	yes
Miller et al., 2013	0.63	103	college	long	UA	no
Olmo-Muoz et al., 2020	0.73	84	primary	middle	UA	no
Pugnali et al., 2017	0.52	27	primary	short	UA	no
Shell et al., 2014	0.48	155	college	long	UA	yes
Threekunprapa, 2020	0.25	160	secondary	short	UA	no
Tonbuloglu, 2019	0.34	114	secondary	middle	UA	no

Note: “g” is a kind of effect size indicator in meta-analysis which represents the effect of teaching approaches (PE and AU) on CT.

“N” means the number of participants in each study.

“yes” means interdisciplinary course; “no” means computer science course.

Thalheimer and Cook’s (2002) classification, which is more frequently used in the literature and more elaborate than other classifications. Here are the ranges of this classification:

$-0.15 <$ effect size < 0.15 at an unimportant level.
$0.15 <$ effect size < 0.40 at a low level.
$0.40 <$ effect size < 0.75 at a medium level.
$0.75 <$ effect size < 1.10 at a high level.
$10 <$ effect size < 45 at a very important level

4 Meta-analysis results

After filtering the literature, in total, 31 independent effect sizes (16 effect sizes of UA and 15 effect sizes of PE) were calculated from 29 articles. In reviewed studies, a total of 2,764 participants were involved, and the sample size of each study ranges from 15 to 364.

4.1 The effectiveness of UA and PE in CT education

We calculated sample sizes (k), weighted effect sizes (g), and 95% confidence intervals (see Table 2). The effect size of UA was 0.392 with a 95% confidence interval from 0.308 to 0.475 ($p<0.001$). The effect size of PE was 0.576 with a 95% confidence interval from 0.408 to 0.734 ($p<0.001$). It can be seen that UA has a low positive effect on CT education and PE has a medium positive effect on CT education.

The test of homogeneity of UA was statistically significant (Q=14.883, df=15, $p<0.001$), which suggested that the data in the included studies were heterogeneous. However, I^2 =0.0% indicates that there was no heterogeneity in UA. According to Huedo-Medina, Sánchez-Meca, Marín-Martínez, and Botella (2006), we tend to give priority to I^2. To identify the influence of moderator variables on students' CT development of UA, we conducted moderator analysis.

The test of homogeneity of PE was statistically significant (Q=86.138, df=14, $p<0.001$), which suggested that the data in the included studies were heterogeneous. $I^2 \geq 50\%$(I^2=83.747%) also indicates the existence of heterogeneity in PE. The heterogeneity indicates the necessity of moderator analysis. The random-effects model should be used to eliminate the heterogeneity.

4.2 The effect of moderator variables in the UA teaching approach

To further explore whether students' CT is affected by moderator variables in the UA teaching approach, the meta-analysis of variance was used to examine the moderated influence of grade level, interdisciplinary and experiment duration. After conducting the homogeneity test across 16 (UA) effect sizes, the moderator analysis results are shown in Table 3.

4.2.1 Grade level

As indicated in Table 3, for experiment and semi-experiment studies using the UA teaching approach to cultivate students' CT, the effect size of the UA teach-

Table 2 Effect size and homogeneity test results of UA and PE

	k	g	95% CI	Homogeneity test			Tau-squared			Test of null (two tailed)	
				Q	p	I^2	Tau^2	SE	Tau	Z-value	p
UA	16	0.392	[0.308,0.475]	14.883	0.000	0.000	0.000	0.011	0.000	9.228	0.000
PE	15	0.576	[0.408,0.743]	86.138	0.000	83.747	0.080	0.048	0.283	6.731	0.000

Table 3 Results of moderator variables analysis of UA to students' CT development

	Between-group Difference	k	g (95% CI)	Homogeneity test
UA teaching approach				
Grade level	Q=3.382			
College		2	0.542[0.287,0.797]***	Q=0.299, I^2=0.000
Secondary		6	0.332[0.223,0.441]***	Q=5.116, I^2=2.265
Primary		8	0.467[0.313,0.620]***	Q=6.273, I^2=0.000
Interdisciplinary	Q=2.035			
No		12	0.367[0.275,0.459]***	Q=12.118, I^2=9.222
Yes		4	0.531[0.326,0.735]***	Q=0.916, I^2=0.000
Duration	Q=2.759			
Long		4	0.524[0.328,0.719]***	Q=1.293, I^2=0.000
Middle		4	0.413[0.258,0.568]***	Q=2.971, I^2=0.000
Short		8	0.335[0.221,0.449]***	Q=7.859, I^2=10.934

*$p<0.05$,**$p<0.01$,***$p<0.001$

ing approach was the largest among college students (g=0.542), followed by primary school students (g=0.467), and finally secondary school students (g=0.332). According to the classifications of effect size, it can be inferred that UA has medium positive effects on college students and primary school students' CT development, and low positive effects on secondary school students' CT development. However, Table 3 showed that the differences in the UA teaching approaches effect among the grade levels were not significant (Q=3.382, df=14, $p>0.05$).

4.2.2 Interdisciplinary courses

The results of the between-group analysis show that the effect size of the UA teaching approach in the interdisciplinary courses (g=0.531) is larger than that in the computer science course (g=0.367) regarding students' CT development. According to the classifications of effect size, it can be inferred that there are medium positive effects of the UA teaching approach in interdisciplinary courses and low positive effects of UA teaching approach in computer science course. Therefore, cultivating students' CT in interdisciplinary courses is more beneficial than in computer science course. However, Table 3 shows that there are no significant differences between interdisciplinary and single computer science courses in the UA teaching approach (Q=2.035, df=15, $p>0.05$).

4.2.3 Experiment duration

After analyzing the influence of the experiment duration on the relationship between the UA and the CT education, the results show that the effect size of short duration (g = 0.335) was smaller than that of the middle (g = 0.413) and long durations (g = 0.524). According to the classifications of effect size, the medium positive effects of the UA teaching approach during middle and long duration, and low positive effects of the UA teaching approach during the short duration. It can be inferred that the UA teaching approach is suitable for middle and long period teaching. However, Table 3 showed there are no significant differences among the experiment durations influence on the relationship between the UA and the CT education (Q = 2.759, df = 14, $p > 0.05$).

4.3 The effect of moderator variables in PE teaching approach

To examine whether the moderator variables influence the relationship between the PE teaching approach and students' CT development, we also utilized a meta-analysis of variance to examine the moderated influence of grade level, interdisciplinary courses, and experiment duration. After conducting the homogeneity test across 15 (PE) effect sizes, the moderator analysis results are shown in Table 4.

4.3.1 Grade level

As indicated in Table 4, for experiment and semi-experiment studies using the PE teaching approach to cultivate students' CT, the results showed the largest improvements in secondary school students (g = 0.687), followed by college students (g = 0.561), and finally primary school students (g=0.409). According to the classifications of effect size, it can be inferred that the PE teaching approach had medium

Table 4 Results of moderator variables analysis of PE to students' CT development

	Between-group Difference	k	g (95% CI)	Homogeneity test
PE teaching approach				
Grade	Q = 1.768			
College		3	0.561[0.183,0.940]**	Q = 5.534, I^2 = 63.860
Secondary		8	0.687[0.435,0.939]***	Q = 31.121***, I^2 = 77.507
Primary		4	0.409[0.084,0.734]*	Q = 37.100***, I^2 = 91.914
Interdisciplinary	Q=3.353			
No		7	0.441[0.245,0.636]***	Q = 28.435***, I^2 = 78.900
Yes		8	0.708[0.499,0.917]***	Q = 25.704**, I^2 = 72.767
Duration	Q = 0.400			
Long		6	0.544[0.265,0.822]***	Q = 29.376***, I^2 = 82.979
Middle		6	0.556[0.251,0.862]***	Q=28.421***, I^2=82.979
Short		3	0.693[0.297,1.089]**	Q = 14.152**, I^2 = 85.868

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$

positive effects for secondary school, college, and primary school students' CT. Table 4 also shows that the differences in the influence among the grade levels were not significant (Q = 3.382, df = 13, $p > 0.05$).

4.3.2 Interdisciplinary courses

The results of the between-group analysis showed the effect size of the PE teaching approach in the interdisciplinary courses (g = 0.708) is larger than that in the single computer science course (g = 0.441) for students' CT development. Therefore, adopting the cultivation of students' CT in interdisciplinary courses is more beneficial than in computer science course. According to the classifications of effect size, it can be inferred that the PE teaching approach has medium positive effects on interdisciplinary courses and computer science course. Similarly, Table 4 shows the differences of influence between the interdisciplinary courses and single computer science course in the PE teaching approach were not significant (Q = 3.353, df = 14, $p > 0.05$).

4.3.3 Experiment duration

After analyzing the influence of the experiment duration in the PE teaching approach, the results show that the effect of short duration experiments (g = 0.693) was larger than that of the middle (g = 0.556) and long duration experiments (g = 0.544). It can be inferred that the PE teaching approach can improve students' CT skills in a short period of teaching. According to the classifications of effect size, the PE teaching approach has medium positive effects on students' CT development in short, middle, and long duration experiments. However, Table 4 shows there are no significant differences among the experiment durations (Q = 0.400, df = 14, $p > 0.05$).

4.4 The comparison of UA and PE teaching approaches in moderator variables

We have integrated the results of the moderator variables analysis of the two teaching approaches affected by three moderator variables, as shown in Table 5. According to the results, the PE teaching approach has a medium positive effect on CT develop-

Table 5 Results of comparison of UA and PE teaching approaches in moderator variables

w	UA(g)	PE(g)
Grade		
College	0.542	0.561
Secondary	0.332	0.687
Primary	0.467	0.409
Interdisciplinary		
Yes	0.531	0.708
No	0.367	0.441
Duration		
Long	0.524	0.544
Middle	0.413	0.556
Short	0.335	0.693

ment for college,secondary and primary students, while the UA teaching approach has a medium positive effect for college and primary students,and a low level effect for secondary students. Both the UA and PE teaching approaches have a medium positive effect on students' CT development in interdisciplinary teaching, and PE especially has a better effect in interdisciplinary teaching. In addition, the PE teaching approach could have a medium positive effect on students' CT development in a short period of teaching, while the UA teaching approach needs a middle or longer period of teaching to get a medium positive effect.

5 Publication bias

To fully consider the effects of publication bias on research results and ensure the reliability of meta-analysis, a funnel plot and fail-safe number were used to evaluate publication bias. The funnel plot can be seen in Fig. 2. The effect sizes of these studies were distributed symmetrically on both sides of the average effect size, indicating that the data of the 31 studies had high reliability and could be used for meta-analysis. The fail-safe number was calculated to be 2,282. According to the formula *5k+10* (Rosenthal, 1979), the result is 165. When the fail-safe number is less than *5k+10*, publication bias should be vigilant (Rothstein, Sutton, & Borenstein, 2006). In our study, the fail-safe number of 2,282 is much larger than the result of 165, so the possibility of bias is small. Thus, the above result suggests that any publication bias would be minimal and negligible.

6 Discussion

The results of the meta-analysis indicate that both UA and PE teaching approaches can play a positive role in cultivating students' CT. Although the moderator variables

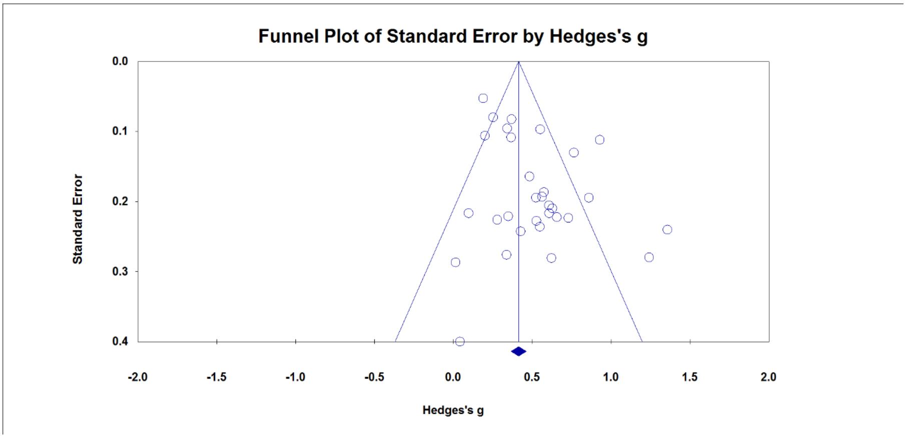

Fig. 2 Funnel plot

have no significant influence on UA or PE and students' CT development, the moderator variables do have some influences. Based on these results, this study attempts to compare the effectiveness of these two teaching approaches on CT education and discuss the effects of the moderator variables.

According to the meta-analysis result on the relationship between UA or PE and students' CT development, the effect size of PE on students' CT development is 0.576, and the effect size of UA on students' CT development is 0.392. It can be inferred that the effectiveness of PE to develop students' CT is better than that of UA. This result is consistent with Grover and Pea's suggestion (2013) of developing students' CT through programming in the digital environment. Therefore, to better develop students' CT, it is necessary to create a programming learning environment for students and encourage them to verify their ideas through programming exercises. In addition, there are some difficulties that students may encounter in the process of learning programming, for example, learning many programming concepts, remembering programming syntaxes, and so on (Hutchins et al., 2020; Witherspoon et al., 2017). Therefore, to overcome the problems, it is also necessary to combine UA and PE teaching approaches, to stimulate students' interest in developing CT by UA, and to strengthen students' deep learning CT skills by PE (Kuo & Hsu, 2020; Threekunprapa & Yasri, 2020; Hutchins et al., 2020; Atmatzidou & Demetriadis, 2016).

This study explored the influence of the moderator variables (grade level, interdisciplinary, and duration) on UA and PE teaching approaches. The results showed there was no significant effect of these moderators on UA or PE teaching approaches and students' CT development. According to the classifications of effect size, we found:

The effectiveness of the UA teaching approach on primary school students is better than that of secondary school students. This finding supports the conclusion of Olmo-Muoz et al. (2020), which demonstrated the feasibility of developing CT in students through UA at a younger age. The effects of the UA and PE approaches on CT education may be different between grade levels because of cognitive transfer in terms of age. Primary school students are in the concrete operational stage of Piaget's cognitive development theory. Their thinking tends to be concrete, so they are more likely to accept the UA teaching approach. Secondary school students are in the formal operations stage. Their thinking becomes sophisticated and advanced, and they have the cognitive ability to understand the abstract concepts of programming. Thus, the PE teaching approach is suitable for secondary school students. In addition, there are fewer effect sizes of UA and PE (2 and 3) in the college stage, so the influence of UA and PE on college students needs to be further verified in future research.

In terms of the types of courses adopted, the effectiveness of the UA and PE methods in interdisciplinary courses is better than that in computer science course. These results support the view that integrating CT into the other subjects not only can help students learn basic CT skills, but also improve their ability to apply CT skills to solve problems (Barr & Stephenson, 2011; Swaid, 2015; Yadav, Gretter, Good, & McLean, 2017). One possible explanation for this result is that interdisciplinary courses help students make sense of CT within a complex learning context, which not only aided them in learning CT skills but also in coping with other disciplinary issues through their CT skills and the development of their problem-solving ability. This encourages educators to develop CT education in interdisciplinary courses, as well as in STEAM and Maker courses (Qualls & Sherrell, 2010; Conde et al., 2019).

Regarding experiment duration, PE can achieve better effectiveness in developing students' CT in a short time, while UA needs a long time to achieve the effectiveness of CT education. The reason may be related to the students' learning experience during the process of the PE and UA. During the process of the PE, there are relatively fixed steps that include abstraction analysis, module decomposition, programming for automation, and debugging the programming. Therefore, students can understand and master CT skills in a short time. During the process of UA, there are diverse methods that are needed for students to conclude their problem-solving thought. As a result, students spend more time understanding and practicing CT skills. The results remind educators to implement longer teaching times for cultivating students' CT with the UA teaching approach than the teaching times they use for the PE teaching approach.

7 Conclusion and recommendation

This study examined the effectiveness of UA and PE in CT education. The results show that both the UA and PE teaching approaches were useful in cultivating students' CT. Besides, the effect of PE teaching approaches was better than the UA teaching approach in improving students' CT, which indicated that it is necessary for educators to provide programming environment in CT education. In addition, according to the result of moderator variables analysis, the grade level, interdisciplinary courses and experiment duration had different effects on the relationship between UA or PE and students' CT development. Especially in interdisciplinary courses, the effectiveness of UA or PE to students CT development was better than that in a single computer science course. Therefore, in CT education, we suggest (1) the UA teaching approach should be used more for primary school students, while the PE teaching approach should be used more for secondary students; (2) CT education should be integrated into other subjects through UA and PE. The result of moderator variables analysis indicated it was necessary to integrate CT education into other subjects. (3) The UA teaching approach requires longer teaching time than the PE teaching approach does within CT education.

8 Limitations

For this study, some limitations need to be noted. First, in the meta-analysis, only three moderator variables (grade level, interdisciplinary, experiment duration) were explored. However, some variables such as gender and culture could affect the relationship between the teaching approaches and CT education. Due to the lack of data on gender and culture in the collected studies, the corresponding moderator variables analysis was not carried out. Further research needs to focus on more moderator variables. Additionally, due to the limitation of sample size, for example, there are only 2 and 3 effect sizes of UA and PE in college CT education. The influences of the moderator variables need to be further explored with more studies collected in future research.

Acknowledgements All authors read and approved the manuscript.

Funding: This research was supported by the National Social Science Foundation for Education of China (BCA210081).

Disclosure

Conflicts of Competing Interest All authors have no relevant financial or non-financial competing interests.

References

Aslan, U., LaGrassa, N., Horn, M., & Wilensky, U. (2020). Putting the Taxonomy into Practice: Investigating Students' Learning of Chemistry with Integrated Computational Thinking Activities. In *American Education Research Association (AERA) 2020 Annual Meeting*

Atmatzidou, S., & Demetriadis, S. (2014, July). How to support students' computational thinking skills in educational robotics activities. In *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education* (pp. 43–50)

Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661–670

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: a digital age skill for everyone. *Learning & Leading with Technology*, 38(6), 20–23

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54

Basu, S., Biswas, G., & Kinnebrew, J. S. (2017). Learner modeling for adaptive scaffolding in a computational thinking-based science learning environment. *User Modeling and User-Adapted Interaction*, 27(1), 5–53

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and practice in technology enhanced learning*, 11(1), 1–35

Bell, T., Rosamond, F., & Casey, N. (2012). Computer Science Unplugged and Related Projects in Math and Computer Science Popularization. In H. L. Bodlaender, R. Downey, F. V. Fomin, & D. Marx (Eds.), *The Multivariate Algorithmic Revolution and Beyond* (7370 vol.). Berlin, Heidelberg: Springer. Lecture Notes in Computer Sciencehttps://doi.org/10.1007/978-3-642-30891-8_18

Bell, T., Witten, I. H., & Fellows, M. (1998). Computer science unplugged: off-line activities and games for all ages. *Great Ideas in Computer Science*, 43(1 Supplement), S21–S22

Bell, T., & Vahrenhold, J. (2018). CS Unplugged—How Is It Used, and Does It Work?. In H. J. Böckenhauer, D. Komm, & W. Unger (Eds.), *Adventures Between Lower Bounds and Higher Altitudes* (11011 vol.). Cham: Springer. Lecture Notes in Computer Sciencehttps://doi.org/10.1007/978-3-319-98355-4_29

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education & Technology*, 24(5), 628–647

Black, J., Brodie, J., Curzon, P., Myketiak, C., McOwan, P. W., & Meagher, L. R. (2013, July). Making computing interesting to school students: teachers' perspectives. In Proceedings of the 18th ACM conference on Innovation and technology in computer science education (ITiCSE '13). Association for Computing Machinery, New York, NY, USA, 255–260. https://doi.org/10.1145/2462476.2466519

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. *European Commission, JRC Science for Policy Report*, 68.https://doi.org/10.2791/792158

Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., & Barone, D. (2017, November). Development of computational thinking skills through unplugged activities in primary school. In Proceedings of the 12th workshop on primary and secondary computing education (WiPSCE '17). Association for Computing Machinery, New York, NY, USA, 65–72. https://doi.org/10.1145/3137065.3137069

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25)

Caeli, E. N., & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, 64(1), 29–36

CAS (2015). Computational thinking:A guide for teachers [EB/OL].[2021-10-1]. https://communityComputingatschool.org.uk/resources/2324/single

Choi, J., Lee, Y., & Lee, E. (2017). Puzzle based algorithm learning for cultivating computational thinking. *Wireless Personal Communications*, 93(1), 131–145

Cohen, J. (1992). Statistical power analysis. *Current directions in psychological science*, 1(3), 98–101

Conde, M., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V., & García-Peñalvo, F. J. (2017, October). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. *In Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2017)*. Association for Computing Machinery, New York, NY, USA, Article 7, 1–6. https://doi.org/10.1145/3144826.3145355

Conde, M., Fernández, C., Alves, J., Ramos, M. J., Celis-Tena, S., Gonçalves, J. … Peñalvo, F. J. G. (2019, October). RoboSTEAM-A Challenge Based Learning Approach for integrating STEAM and develop Computational Thinking. *In Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'19)*. Association for Computing Machinery, New York, NY, USA, 24–30. https://doi.org/10.1145/3362789.3362893

Crombie, I. K., & Davies, H. T. (2009). What is meta-analysis. What is,1–8

CSTA,&ISTE (2009). Operational definition of computational thinking for K-12 education[EB/OL].[2021-10-1]. http://csta.acm.org/Curriculum/sub/CompThinking.html

Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014, November). Introducing teachers to computational thinking using unplugged storytelling. In *Proceedings of the 9th workshop in primary and secondary computing education (WiPSCE '14)*. Association for Computing Machinery, New York, NY, USA, 89–92. https://doi.org/10.1145/2670757.2670767

Delal, H., & Oner, D. (2020). Developing middle school students' computational thinking skills using unplugged computing activities. *Informatics in Education*, 19(1), 1–13

Faber, H., Wierdsma, M., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network*, 12, 13–24. https://etenjournal.com/2020/02/07/teaching-computational-thinking-to-primary-school-students-via-unplugged-programming-lessons/

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28

Falkner, K., Vivian, R., & Falkner, N. (2014, January). The Australian digital technologies curriculum: challenge and opportunity. In *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148* (pp. 3–12)

Fidai, A., Capraro, M. M., & Capraro, R. M. (2020). "Scratch"-ing computational thinking with Arduino: A meta-analysis. *Thinking Skills and Creativity*, 38, 100726

GOV.UK. (2014). National curriculum in England: framework for key stages 1 to 4 in England [EB/OL].https://www.gov.uk/government/publications/national-curriculum-in-england-framework-for-key-stages-1-to-4/the-national-curriculum-in-england-framework-for-key-stages-1-to-4

Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: an integrated approach to teaching twenty-first century skills. *TechTrends*, 60(5), 510–516

Grover, S., & Pea, R. (2013). Computational Thinking in K–12 A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43

Grover, S., & Pea, R. (2018). Computational thinking: A competency whose time has come. *Computer science education: Perspectives on teaching and learning in school*, 19

Grover, S., Pea, R., & Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2), 199–237

Hague, C., & Payton, S. (2011). Digital literacy across the curriculum.Curriculum Leadership, 9(10)

Hedges, L. V., & Olkin, I. (1985). S–or meta-analysis. Orlando, FL:Academic Press.p123

Higgins, J. P. T., & Thompson, S. G. (2002). Quantifying heterogeneity in a meta-analysis. *Statistics in Medicine*, 21, 1539–1558

Hooshyar, D., Malva, L., Yang, Y., Pedaste, M., Wang, M., & Lim, H. (2021a). An adaptive educational computer game: Effects on students' knowledge and learning attitude in computational thinking. *Computers in Human Behavior*, 114, 106575

Hooshyar, D., Pedaste, M., Yang, Y., Malva, L., Hwang, G. J., Wang, M. … Delev, D. (2021b). From gaming to computational thinking: An adaptive educational computer game-based learning approach. *Journal of Educational Computing Research*, 59(3), 383–409

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296–310

Huang, W., & Looi, C. K. (2021). A critical review of literature on "unplugged" pedagogies in K-12 computer science and computational thinking education. *Computer Science Education*, 31(1), 83–111. https://doi.org/10.1080/08993408.2020.1789411

Huedo-Medina, T. B., Sánchez-Meca, J., Marín-Martínez, F., & Botella, J. (2006). Assessing heterogeneity in meta-analysis: Q statistic or I² index? *Psychological methods*, 11(2), 193

Hutchins, N. M., Biswas, G., Maróti, M., Lédeczi, Á., Grover, S., Wolf, R. … McElhaney, K. (2020). C2STEM: A system for synergistic learning of physics and computational thinking. *Journal of Science Education and Technology*, 29(1), 83–100

Ince, E. Y., & Koc, M. (2021). The consequences of robotics programming education on computational thinking skills: An intervention of the Young Engineer's Workshop (YEW). *Computer Applications in Engineering Education*, 29(1), 191–208. https://doi.org/10.1002/cae.22321

International Society for Technology in Education. (2016). ISTE standards for students [EB/OL]. http://www.iste.org/standards/standards/for-students. (Accessed 1 September 2021)

Jun, S., Han, S., & Kim, S. (2017). Effect of design-based learning on improving computational thinking. *Behaviour & Information Technology*, 36(1), 43–53. https://doi.org/10.1080/0144929X.2016.1188415

Kahn, K., Sendova, E., Sacristán, A., & Noss, R. (2011). Young Students Exploring Cardinality by Constructing Infinite Processes. *Technology Knowledge & Learning*, 16(1), 3–34

Kalelioglu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33–50. https://doi.org/10.15388/infedu.2014.03

Kalelioglu, F., Gülbahar, Y., & Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. *Baltic Journal of Modern Computing*, 4(3), 583–596. https://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf

Kim, B., Kim, T., & Kim, J. (2013). Paper-and-pencil programming strategy toward computational thinking for non-majors: Design your solution. *Journal of Educational Computing Research*, 49(4), 437–459

Kuo, W. C., & Hsu, T. C. (2020). Learning computational thinking without a computer: How computational participation happens in a computational thinking board game. *The Asia-Pacific Education Researcher*, 29(1), 67–83

Kwon, D. Y., Yoon, I. K., & Lee, W. G. (2011). Design of programming learning process using hybrid programming environment for computing education. *KSII Transactions on Internet and Information Systems*, 5(10), 1799–1813

Leonard, A. E., Daily, S. B., Jrg, S., & Babu, S. V. (2020). Coding moves: Design and research of teaching computational thinking through dance choreography and virtual interactions. *Journal of Research on Technology in Education* (2), 1–19

Li, Y., Schoenfeld, A. H., Graesser, A. C., Benson, L. C., English, L. D., & Duschl, R. A. (2020). Computational Thinking Is More about Thinking than Computing. *Journal for STEM Education Research*, 1–18

Lipsey, M. W., & Wilson, D. B. (2001). *Practical meta-analysis*. SAGE publications, Inc

Lye, S. Y., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41(dec.), 51–61

Manabe, H., Kanemune, S., Namiki, M., & Nakano, Y. (2011). CS Unplugged Assisted by Digital Materials for Handicapped People at Schools. In I. Kalaš, R. T. Mittermeir (Eds.), *Informatics in Schools. Contributing to 21st Century Education*. ISSEP 2011. Lecture Notes in Computer Science, vol 7013. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24722-4_8

Miguel, Z. R. (2019). Pensamiento computacional desenchufado. *Education in the Knowledge Society* (EKS)(20), 18. http://hdl.handle.net/10366/143339

Miller, L. D., Soh, L. K., Chiriacescu, V., Ingraham, E., Shell, D. F., Ramsay, S., & Hazley, M. P. (2013, October). Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses. In *2013 ieee frontiers in education conference (fie)* (pp. 1426–1432). IEEE. https://doi.org/10.1109/FIE.2013.6685067

Moreno-León, J., Román-González, M., Harteveld, C., & Robles, G. (2017, May). On the automatic assessment of computational thinking skills: A comparison with human experts. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2788–2795). https://doi.org/10.1145/3027063.3053216

Nardelli, E. (2019). Do we really need computational thinking? *Communications of the ACM*, 62(2), 32–35

Noh, J., & Lee, J. (2020). Effects of robotics programming on the computational thinking and creativity of elementary school students. *Educational Technology Research and Development*, 68(1), 463–484. https://doi.org/10.1007/s10956-019-09794-8. https://link.springer.com/article/

Nouri, J., Zhang, L., Mannila, L., & Norén, E. (2020). Development of computational thinking, digital competence and 21st century skills when learning programming in K-9. *Education Inquiry*, 11(1), 1–17

Olmo-Muoz, J. D., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832

Pugnali, A., Sullivan, A., & Bers, M. U. (2017). The Impact of User Interface on Young Children's Computational Thinking. *Journal of Information Technology Education: Innovations in Practice*, 16, 171–193

Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, 25(5), 66–71

Repenning, A., Webb, D., & Ioannidou, A. (2010, March). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 265–269). https://doi.org/10.1145/1734263.1734357

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: an experiment with sixth-grade students. *Interactive Learning Environments*, 28(3), 316–327. https://doi.org/10.1080/10494820.2019.1612448

Rogers, J., & Révész, A. (2020). Experimental and quasi-experimental designs. *The Routledge handbook of research methods in applied linguistics* (pp. 133–143). Routledge

Rose, S. P., Habgood, M., & Jay, T. (2017). An exploration of the role of visual programming tools in the development of young children's computational thinking. *Electronic Journal of e-Learning*, 15(4), 297–309. http://www.ejel.org/volume15/issue4/p297

Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, 86(3), 638

Rothstein, H. R., Sutton, A. J., & Borenstein, M. (Eds.). (2006). *Publication bias in meta-analysis: Prevention, assessment and adjustments* (pp. 1–7). John Wiley & Sons

Sentance, S., & Csizmadia, A. (2015). Teachers' perspectives on successful strategies for teaching Computing in school. In IFIP TC3 Working Conference 2015: A New Culture of Learning: Computing and Next Generations. http://www.ifip2015.mii.vu.lt/proceedings#.WFpvmlzgmqQ

Shell, D. F., Hazley, M. P., Soh, L. K., Miller, L. D., Chiriacescu, V., & Ingraham, E. (2014, October). Improving learning of computational thinking using computational creativity exercises in a college CSI computer science course for engineers. In *2014 ieee frontiers in education conference (fie) proceedings*, pp. 1–7. https://doi.org/10.1109/FIE.2014.7044489

Sung, W., Ahn, J., & Black, J. B. (2017). Introducing computational thinking to young learners: Practicing computational perspectives through embodiment in mathematics education. *Technology, Knowledge and Learning*, 22(3), 443–463

Swaid, S. I. (2015). Bringing computational thinking to STEM education. *Procedia Manufacturing*, 3, 3657–3662. https://doi.org/10.1016/j.promfg.2015.07.761

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12(2), 1–29

Thalheimer, W., & Cook, S. (2002, August). How to calculate effect sizes from published research articles: A simplified methodology. Retrieved September 29, 2020 from http://work-learning.com/effect_sizes.htm

Threekunprapa, A., & Yasri, P. (2020). Unplugged Coding Using Flowblocks for Promoting Computational Thinking and Programming among Secondary School Students. *International Journal of Instruction*, 13(3), 207–222
Tonbuloglu, B., & Tonbuloglu, I. (2019). The effect of unplugged coding activities on computational thinking skills of middle school students. *Informatics in Education*, 18(2), 403–426
Tsarava, K., Moeller, K., & Ninaus, M. (2018). Training computational thinking through board games: The case of Crabs & Turtles. *International Journal of Serious Games*, 5(2), 25–44
Unnikrishnan, R., Amrita, N., Muir, A., & Rao, B. (2016, June). Of elephants and nested loops: How to introduce computing to youth in rural india. In *Proceedings of the The 15th International Conference on Interaction Design and Children* (pp. 137–146)
Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35
Witherspoon, E. B., Higashi, R. M., Schunn, C. D., Baehr, E. C., & Shoop, R. (2017). Developing computational thinking through a virtual robotics programming curriculum. *ACM Transactions on Computing Education (TOCE)*, 18(1), 1–20
Yadav, A., Gretter, S., Good, J., & McLean, T. (2017). Computational thinking in teacher education. *Emerging research, practice, and policy on computational thinking* (pp. 205–220). Cham: Springer
Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the Acm*, 60(4), 55–62
Yin, Y., Hadad, R., Tang, X., & Lin, Q. (2020). Improving and Assessing Computational Thinking in Maker Activities: the Integration with Physics and Engineering Learning. *Journal of Science Education and Technology*, 29(2), 189–214. https://doi.org/10.1007/s10956-019-09794-8
Zha, S., Morrow, D. A. L., Curtis, J., & Mitchell, S. (2021). Learning Culture and Computational Thinking in a Spanish Course: A Development Model. *Journal of Educational Computing Research*, 59(5), 844–869. https://doi.org/10.1177/0735633120978530

Authors and Affiliations

Feng Li[1] · Xi Wang[1] · Xiaona He[2] · Liang Cheng[1] · Yiyu Wang[1]

✉ Feng Li
lifengys2001@163.com

Xi Wang
51194108010@stu.ecnu.edu.cn

Xiaona He
hexiaona@bnu.edu.cn

Liang Cheng
51204108036@stu.ecnu.edu.cn

Yiyu Wang
51204108023@stu.ecnu.edu.cn

1 Department of Education Information Technology, Faculty of Education, East China Normal University, Shanghai, People's Republic of China

2 Department of Research and Development, Education Training Center, Beijing Normal University, Beijing, People's Republic of China