DOI: https://doi.org/10.31686/ijier.vol10.iss4.3700

Characteristics and problems of unplugged computer science curriculum for young children: comparative and practical research based on the curriculum in four countries

Bao Ran
postgraduate student
South China Normal University,1st Huangpu Road, Guangzhou, Guangdong, China,
E-mail: 384057908@qq.com

Chen Jianyong
postgraduate student
South China Normal University,1st Huangpu Road, Guangzhou, Guangdong, China,
E-mail: 592116849@qq.com

Abstract

With the progress of computer science education in recent years, more and more educators have brought attention to computer science education among young children. Among all these strategies, the unplugged form has been shown to be more effective in teaching. However, recent studies have focused more on the impact of unplugged computer science courses on young children and less on whether these courses are appropriate for the developmental stage of young children. Therefore, this research summarized the curriculum characteristics by comparing different series of unplugged courses for young children from four nations. Then, in a 7-day workshop conducted in China's urban areas, we explored the issues that arise in the implementation of these courses. This research reveals that, although the existing courses cater to a young age range, four issues can still be found, including difficulty, ability difference, too much cooperation, and emphasis on abstraction. Some of these issues may be handled by instructors, while others need consideration of the connection between curriculum design and the physical and cognitive development levels of young children. Furthermore, this research explored the acceptance of unplugged computer science among Chinese young children as well as its impact on their computational thinking level, achieving positive results.

Keywords: unplugged; computer science; young children; content analysis; computational thinking

1 Introduction

Information technology (IT) has been increasingly important in people's lives since the turn of the century. According to statistics from 2021, 76% of U.S. states will have computer science curriculum standards from kindergarten to high school (code.org, 2021). Furthermore, countries such as the United Kingdom, France, Spain, Denmark, Finland, the Czech Republic, and Australia have implemented computer science policies and frameworks from kindergarten to high school (Balanskat & Engelhardt, 2015; European Schoolnet, 2014; Livingstone, 2012; Uzunboylu et al., 2017). It can be found that the curriculum of various countries pays attention to young children. Computer Science Unplugged (CSU), as a method of not utilizing text and screen, has received a lot of attention in recent years, especially when it comes to young children's physiological and cognitive development.

CSU is a method of learning computer ideas, programming logic, and computational thinking that does not need the use of computers. Instead, games and activities are used to teach students computer concepts, programming logic, and computational thinking (Bell et al., 1998). It was formally introduced into the "K-12 Computer Science Model Curriculum" by the Association for Computing Machinery (ACM) K-12 Working Group in 2003.

In the past five years, many scholars have carried out CSU research for young children. Faber and his mates have adapted the CSUnplugged.org curriculum to the learning environment and content of Dutch children (Hylke H. Faber, et al., 2017). Sermin Metin presented research on teaching CSU courses in kindergartens in a moderately developed region of Turkey, demonstrating the benefits of the unplugged approach to programming skills (Metin S., 2020). Sina Wetzel offers CSU research in Germany that uses a combination of indoor and outdoor learning methods to expand the space for CS education (Wetzel et al., 2020) . Olmo-Munoz and his colleagues presented research on CSU for children aged 5-7 years old in Spain, which indicated that when children were taught in an unplugged environment before being taught in a plugged one, their performance and attention improved dramatically (Del Javier Olmo-Mu-noz et al., 2020). Brackmann demonstrated a study of CSU courses conducted in disadvantaged parts of Brazil, which revealed that the unplugged technique may significantly increase students' problem-solving abilities in these impoverished settings (Brackmann et al., 2019). In addition, many researchers from different countries have also paid attention to the effect of unplugged computer science curriculum on children, and their research has achieved similar positive results (Conde et al., 2017; DELAL & ONER, 2020; Hsu & Liang, 2021; Lee & Junoh, 2019; Saxena et al., 2020; Sysło & Kwiatkowska, 2014; Un-Jou K et al., 2020; Wohl et al., 2015; Yuliana I et al., 2021).

However, current researchers have focused more on the impact of CSU courses on young children's computational thinking, and pay less attention to these CSU courses. As these curricula are often developed based on a particular computational thinking framework, some researchers have pointed out that these frameworks used in children's education do not take into account the uniqueness of children's development and the level of development of their cognitive and social communication abilities at various ages (Relkin et al., 2020). Moreover, some researchers believe that preschool children are more vulnerable to the harm of traditional programming education and the nature of CSU is more suitable for children's physiological development. Therefore, it is strange that there is little research on curriculum in preschool education (Battal et al., 2021). Therefore, the purpose of this research is to summarize their characteristics and explore the problems in practice by comparing the international computer science unplugged curriculum contents for young children. On the other hand, considering that there is little research on the impact of CSU courses on Chinese children, this paper will also discuss this issue. Therefore, the research questions of this paper can be stated as follows:

1. What are the characteristics of unplugged computer science courses for young children?

2. What are the problems in the implementation of these unplugged computer science courses for young children?

3. What is the effect of these unplugged computer science courses on Chinese children?

In response to these questions, the next section of this paper will describe the research methods used in the research, the process of course analysis and practice, the results of the research will be presented in section 3 and further discussion of the findings will be developed in section 4.

2 Methods and Procedure

Content analysis and pre-experimental design were employed in this study (DELAL & ONER, 2020). In order to understand their characteristics, courses were transcribed, coded, and examined for content analysis. Then they were assessed using the CT framework of adaptive development. Based on the evaluation results, the research summarized the characteristics of the courses and reorganized seven new courses. The impact of these courses was investigated in a one-group pre-test and post-test pre-experimental study with Chinese young children in a workshop.

2.1 Course Analysis

Four databases, Web of Sciences, Google Scholar, Education Resources Information Centre, and Association for Computing Machinery (ACM) Digital Library were included in the basic search using the keywords "Unplug", "CS" and "Programming". Several criteria were developed to screen the articles: the research must be available, and empirical, and it has to be conducted with students in K-6. Finally, 31 studies were selected, and the curriculum mentioned in these studies are collected. Among the available curricula that explicitly include CSU were mainly CSUnplugged.org in New Zealand, Code.org in the United States, CodeKinderen.nl in the Netherlands, Brazil's National Foundation Curriculum, Finland's National Curriculum, and the Nuri project in South Kore. Unplugged courses in New Zealand, the United States, the Netherlands, and Brazil were chosen after analyzing the integrity of the curriculum, as indicated in Table 1.

After excluding the curriculum for advanced age groups, the research uses the third version of the curriculum from The New Zealand curriculum website (CSUnplugged.org) and maintains 11 of the courses (coded as CS-xx). Meanwhile, the research employs its latest CS Fundamentals 2020-21 unplugged courses in the US curriculum (Code.org). The first four for lower age stages were chosen from the six stages of the series. After the elimination of items with similar course content, ten courses (coded as Co-xx) are retained. The research omitted language-based courses and incorrect content from the Dutch curriculum (CodeKinderen.nl) and chose 10 courses (coded as NL-xx) from a total of 12. The Course of Brazil (Computacional.com.br) selected 14 (coded as BR-xx) from 24 courses in its database that could recognize the course content and teaching method, removed the purely manual activity materials and courses based on Internet software, and removed the purely manual activity materials and courses based on Internet software.

Country	New Zealand	The U.S.	Netherland	Brazil
N	11	10	10	14
Coding	CS-xx	Co-xx	NL-xx	BR-xx
Website	CSUnplugged.org	Code.org	Codekinderen.nl	Computacional.com.br

Table 1 Course selection results and information

Since all of the curriculums are published in English (or Dutch), we translated them into Chinese to make it easier for Chinese educators to analyze and evaluate. Because each of the four curriculum series has a distinct theoretical basis for CT and considering the developmental stage of children, Professor Bers' developmentally appropriate framework for CT was selected as the basis for analysis and evaluation., as shown in Table 2 (Bers, 2021). Two in-service teachers were invited to conduct an independent evaluation with the authors, who both have a background in computer science. The items with contradictory evaluations were then discussed in a two-hour video conference, and the evaluation outcomes are detailed in Appendix A.

Indicator layer I	Indicator layer II	Indicator layer III	Contain?
Children's developmentally appropriate CT framework	Algorithm	Sequencing/order	N/Y
		Logical organization	N/Y
	Modularity	Breaking up a large job into smaller steps	N/Y
		Grouping a list of instructions into a given category or module to complete a larger project or integration	N/Y
	Control Structures	Recognizing patterns and repetition	N/Y
		Causes and effect	
	Representation	Symbolic representation	N/Y
		Models and migration	N/Y
	Hardware/Software	Understanding that objects don’t work by magic but by human-engineered	N/Y
	Design Process	The design process of "ask questions, imagine, plan, create, test, improve and share"	N/Y
	Debugging	Identifying problems	N/Y
		Problem-solving	N/Y

Table 2 Curriculum evaluation framework

2.2 Workshop

One of the participating teachers was invited to conduct a 7-day CSU workshop in China after completing the course analysis. The workshop was held for one hour per day in a school in Zhaoqing City, Guangdong Province, China. We had advertised our volunteer recruiting on the Internet to reach a larger spectrum of social groupings. The recruitment requirements of volunteers are children aged 6-8 years. A total of 15 children (10 males and 5 girls) took part in the experiment. The children varied in age from 4.8 to 7 years old (mean=5.59, SD=0.648) and were in grades ranging from middle kindergarten to first grade, as shown in Table 3. Meanwhile, none of them had any prior programming experience and, could be considered to have the same computer science background. On the first day of the workshop, all participants' parents signed a consent form acknowledging that this was fully voluntary educational research that can be withdrawn at any moment.

Items	N		
Participant	15		
Male	10	Female	5
	Min	Mode	Max
Age	4.8（7%）	6（47%）	7（7%）
Grade	Middle K（47%）	Middle K（47%）	First Grade（13%）
Programming learning experience	0	0	0
Parental education	Lower secondary（7%）	Bachelor（47%）	Master and above（27%）
Household income/month	Under $350（7%）	$850 and above（80%）	$850 and above（80%）

Table 3 Information of participates

The researchers rearranged and rewrote the four series of curricula described above, resulting in seven courses, based on the workshop's space-time features and the situation of volunteers. Table 4 contains pertinent information on the courses used in the workshop. At the end of each course, the participant completed a "self-evaluation card," which is displayed in Appendix B, to indicate his or her emotional efficacy (code.org). A brief measure of computational thinking was provided at the end of the last course, using challenges taken from BebrasUnplugged, which is shown in Appendix C (Bebras, 2016).

After the development of the CT measurement, 4 children who enrolled in the course but were unable to participate, with a mean age of 5.5 (SD=0.77) and no programming experience, were invited to complete a comprehension test. This was done to ensure that Chinese children above the age of five could understand and complete the questions after the teacher had explained to the children.

The teacher and researchers were required to give a reflective report based on their observations after each day of the workshop, and at the end of the workshop, the researcher interviewed the parents about their opinions. The materials reviewed for this workshop included a video recording of the workshop, learning resources, teacher reflections, CT assessment, and parent interviews.

	Name	Description
Lesson-1	Magic of Parity check (CSUnplugged.org)	Briefly introduce the overall course content, and promote the understanding among participants through a game of designing personal avatars. Then, let them feel the wonders of computer science through information transfer games and parity magic games, and increase their interest in the course.
Lesson-2	What’s in a phone(Codekinderen.nl; Computacional.com.br)	The participants will draw pictures of mobile phones in their daily lives and share them with their peers, then they observe and replicate the internal structure of real mobile phones with Lego blocks so that they can understand the components of smart devices, and finish the paper-cutting game "Assemble your own computer" freely before the end.
Lesson-3	Pixel coloring (CSUnplugged.org)	Participants will understand the principles of screen displays through a square paper coloring game.
Lesson-4	Sorting Networks(CSUnplugged.org; CSUnplugged.org)	By completing the dichotomy sorting activity in the form of body games on the floor, participants can understand the wonders of the sorting algorithm, with sortable objects include their age, height, and name strokes of them.
Lesson-5	Map Game(Basic)(code.org; code.org)	Complete the map walking game through the instruction card, so that participants can understand the basic principle and logic of programming and why the robot can "move".
Lesson-6	Map Game(Advanced)(Codekinderen.nl; CSUnplugged.org)	Expand the content of the map game (basic) to the floor, and complete the map game under body activities, which is more fun.
Lesson-7	Application Design(Computacional.com.br)	Understand what an application should contain, and design their application to solve practical problems.

Table 4 Workshop Course Description

3 Result

3.1 Course evaluation

According to the evaluation results, as shown in Table 5 and detailed in Appendix A, the most involved

CT modules in the 45 courses are a logical organization (75.6%, which means that the module is present in 75.6% of the courses), decomposition (73.3%), integration (66.7%), and symbolic representation (64.4%), and less involved in software and hardware recognition (24.4%) and modeling (15.6%). It means that logical structure, decomposition, and integration, as well as symbolic representation, are significant characteristics that any course pays attention to. On the other hand, the level of development of children, such as software and hardware recognition, is not taken into consideration in the CT base of these course series, therefore the issue is not addressed in the course, resulting in a low fraction of this item. In addition, modeling and migration are not easily represented in the course scheme provided in the form of text, thus there is only the lowest score.

The New Zealand course has the most content in sequencing (29.4 %), conditional structure (33.3 %), and modeling (42.9 %), the Dutch course has the most content in decomposition (30.3 %), integration (33.3 %), and design process (40.0 %), and the Brazilian course has the most content in logical organization (35.3 %), integration (33.3 %), and modeling (42.9 %).

Indicator layer II	Indicator layer III（Proportion of all courses）	Proportion of corresponding courses			
		CS-xx	Co-xx	NL-xx	BR-xx
Algorithm	Sequencing/order（46.7%）	38.10%	9.50%	23.80%	28.60%
	Logical organization（75.6%）	29.40%	14.70%	20.60%	35.30%
Modularity	Decomposition（73.3%）	24.20%	18.20%	30.30%	27.30%
	integrating（66.7%）	16.70%	16.70%	33.30%	33.30%
Control Structures	Pattern recognition（33.3%）	26.70%	13.30%	13.30%	46.70%
	Condition（40.0%）	33.30%	27.80%	16.70%	22.20%
Representation	Symbolic representation（64.4%）	27.60%	17.20%	24.10%	31.00%
	Models and migration（15.6%）	42.90%	0.00%	14.30%	42.90%
Hardware/Software	Recognizing（24.4%）	27.30%	18.20%	18.20%	36.40%
Design Process	Revision and editing（44.4%）	15.00%	25.00%	40.00%	20.00%
Debug	Identify problems（35.6%）	25.00%	25.00%	25.00%	25.00%
	Solve problems（33.3%）	26.70%	26.70%	26.70%	20.00%

Table 5 Evaluation results of courses

To represent the complexity of each series of courses and set of courses, all assessment items were totaled together without weight. As shown in Table 6, each series has at least one course with a difficulty of 0.8, and the primary difficulty is focused in the range of 0.17-0.75, indicating that the majority of courses include at least two CT modules. The average difficulty of the Dutch series is the highest, followed by New Zealand and Brazil.

		CS-xx	CO-xx	NL-xx	BR-xx
N		11	10	10	14
Mean		0.5236	0.383	0.528	0.4464
SD		0.21416	0.26571	0.20176	0.22809
Minimum		0.25	0.08	0.25	0.08
Maximum		0.83	0.83	0.83	0.83
Percentiles	25	0.33	0.17	0.3775	0.31
	75	0.75	0.6225	0.705	0.69

Table 6 Difficulty of different curricula series

3.2 Emotional efficacy

Participants gave back their emotional efficacy to the course by filling out a self-assessment at the end of each day's workshop. The results are shown in Table 7. 105 reports were gathered during the workshop, with 60 (57.14 %) indicating that the mood of participants was excellent, 30 (28.57 %) indicating very good, and 12 (11.43 %) indicating good. Only one (0.95 %) emotion self-assessment indicated unhappiness, which came on the second day of the course. Another two (1.90 %) reported doubt. In general, the majority of the workshop participants' emotions were positive, with just a few individuals reporting negative emotions in a course.

Emotion	Excellent	Very good	Good	Not good	Unhappy	Angry	Boring	Doubt
N	60	30	12	0	1	0	0	2
Percentiles	57.14%	28.57%	11.43%	0.00%	0.95%	0.00%	0.00%	1.90%

Table 7 Participants' self-assessment emotional efficacy

3.3 CT performance

At the end of the workshop, participants took the CT test. To guarantee that the exam is understandable by children of that age, four children who signed up but were unable to attend were chosen to take the test before the workshop. It was verified that the participants in the course could comprehend all of the test's questions. Because the participants were not pre-tested at the start of the course, the comprehension test results were utilized as a baseline in the quantitative analysis.

The final score was determined by tallying how many questions they successfully answered (6 points in total). The findings of the baseline and participants' tests were subjected to an independent sample Mann Whitney U test. Table 8 shows the findings. The mean baseline value was 3.5 (SD=0.577), whereas the mean post-test value was 4.7333 (SD=0.883). The exact sig = 0.027＜0.05, suggesting that participants' CT level had considerably improved compared to youngsters who did not attend the program. The effect size of the two tests was 1.4716, showing that the CSU courses in the workshop made a significant difference in CT improvement.

	Descriptive statistics			Independent sample Mann-Whitney U test				Effect Size
	N	Mean	SD	N	Test Statistic	Stand Error	Exact Sig.	Cohen's d
Baseline	4	3.5	0.57735	19	52	9.361	0.027	1.4716
Participants	15	4.7333	0.88372					

Table 8 CT test performance of baseline and participant

Figure 1 shows a comparison of each test dimension. Participants' abilities of algorithm thinking, decomposition, pattern recognition, and modeling improved significantly after the CSU program. The proportion of correct algorithmic thinking grew from 25% to 60%, and decomposition had risen from 50% to 80%. The accurate percentage of pattern recognition has grown from 50% to 80%, and modeling increased from 0% to 80%. The other two items (Symbolic representation and Integration) did not seem to make a significant difference.

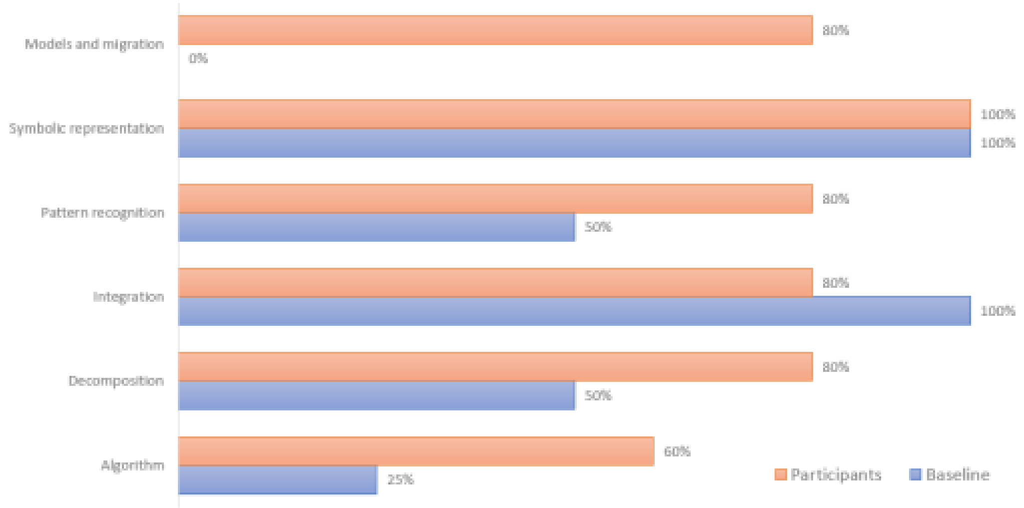

Fig. 1 Comparison of each dimension of the test

3.4 Parents Interviews

During the interview, all of the parents showed support for the workshop. Some parents reported that their children expressed a strong interest in the workshop's content to their parents at home. In an interview, the parent of a child who took part in the activity noted：

He (the child) kept inquiring when he was supposed to go to the workshop. Do you have any plans to provide related courses in the future? ... I hope you will continue to engage in similar activities. He is now tremendously interested in computers. Learn more, and when he's old enough, I'll teach him how to program! (Translated from Chinese)

Another mother of a child said in the interview：

He still enjoys this activity, although he admits that drawing is tough for him. He has no idea what to draw... But he enjoys playing with construction blocks... I hope your course will be able to continue. This thorough course has a lot of importance for kids. (Translated from Chinese)

4 Discussion

In this research, we compared and analyzed four unplugged computer science curriculum series across the world, defined the features of the CSU curriculum for young children, and reorganized a new unplugged computer science curriculum based on these series. This new series contains seven courses, which were used in an unplugged programming workshop for young children in Chinese urban areas. Through these efforts, this research analyzes three issues: the first is what the characteristics of these courses are, the second is what obstacles exist in the execution of these courses, and the third is how these courses affect Chinese children. This part of the article will go into the research process and findings in response to these questions.

4.1 Characteristic of courses in four countries

At present, there are few qualitative studies in CSU research that concentrate on the curricular content of young children. Faber, for example, highlighted the curriculum development and implementation process in primary schools in 2017 (Hylke H. Faber, et al., 2017), while AGYEI shared the content of CSU activities in the Finnish children's after-school interest group in 201 9 (Agyei et al., 2019). More researches concentrate on the impact of CSU, and often directly refers to the content of existing courses at younger ages, such as Leifheit's work for students in grades 3-4 (Leifheit et al., 2018). Comparing the characteristics and commonalities of these series of courses, we believe that they can provide useful help for the development of new courses for young children. The research summarizes the characteristics of these four series of courses as follows:

CSunplugged.org in New Zealand was the first CSU course, and its curriculum is similar to conventional programming courses, with a focus on the growth of algorithm thinking and control structure. Furthermore, the courses are standardized so that in-service instructors can easily understand and apply them.

The courses offered by code.org in the United States are modular. They are not only simple to dismantle and use independently, but they also feature a lot of activity content and are quite interesting.

The Codekinderen.nl course in the Netherlands is distinguished by its emphasis on linking practice and experience. It places a high emphasis on children's hands-on crafting and product development throughout the course. This notion pervades every part of their courses, making them one-of-a-kind.

The computational.com.br course in Brazil is playful. The material is filled with tales, chores, cartoon graphics, fictitious situations, music, and dance, and so on, all of which are appealing to young children.

To recap, the research divides the features of these courses into four categories, including the practicality of the teaching plan, the richness of activities, the interest of content, and the creativity of resources. This is essentially consistent with Bell's kinesthetic activity design guidelines proposed in 2008 (Bell et al., 2008).

4.2 Adaptation of courses

In the research, Rodriguez made some suggestions on the design of CSU courses (Rodriguez et al., 2016). However, while these discussions are not about the design for young groups, we also consider Bers' suggestions in her book on the design of programming courses for young children (Bers, 2021). The final course after reorganization and adaptation not only maintained the original characteristics of the course but also made the following important adjustments:1. Integrating comparable activities in numerous courses and picking more attractive activities that are adapted to children's cognitive development (e.g., Lesson-2, Lesson-5, and Lesson-6), 2. Increase the proportion of physical activities in the course (e.g., Lesson-1, Lesson-4, and Lesson-6), 3. Include Lego block activities to improve children's multidimensional understanding of course objectives (e.g., Lessons-2 and -3), and 4. Increase the relevance of course content to common information technology equipment (e.g., Lesson-2).

4.3 Challenges in implementation

Many researchers have illustrated the CSU curriculum implementation process in their research, but few have explored the challenges encountered throughout the process, such as the research of Olmo- Muoz J D in 2020. Through teacher reflection and video recording, this research describes the challenges encountered throughout the curriculum implementation process, as shown in Table 9. In the research, Conde et al. reported the age difference of CSU in CT development. Similarly, we believe that the age difference should be considered in the design of the CSU curriculum. Although young children's Computational Thinking skills have increased greatly as a result of the unplugged course, the information they can assimilate is generally simpler. Parity, for example, is listed on the CSunplugged.org website as appropriate for children ages 5-7, however, the children in this research were unable to comprehend the two-dimensional parity in that course. Besides, we discovered that variations in non-technical experiences, such as art instruction or sports training, had a substantial impact on participants' learning progress in unplugged activities. It is believed that designing more inclusive activities or giving certain shortcuts might significantly lessen the impact in this area. It should also be pointed out that although the current CSU courses cover children aged 4-7, these courses are often not developed for their cognitive development level. According to the relevant theories of developmental psychology, preschool children's cognitive development is in the pre-operation stage and they are developing the skills of cooperation with others (Feldman, 2006). This implies they are unable to do effective collaboration and abstract thinking like teens (Mioduser et al., 2009). This mismatch was reflected in the challenges of collaboration and abstraction in curriculum implementation.

Type	Examples
Difficulty	The participants struggled to comprehend the two-dimensional parity check, therefore they quickly converted it into a grid painting game in Lesson-1. When teachers introduced the one-dimensional parity check as adaptation, they could quickly grasp how it worked
Competence Differences	Some participants felt overwhelmed and could only follow after the instructor offered examples when they were asked to draw in Lesson-2. After questioning, those children who could start quickly had received art training.
Cooperation	Two participants shared one material dispute, and all believed they should have it alone at lesson-2. When the students needed to cooperate to complete a big project, they kept arguing at Lesson-3.
Abstraction	If participants were simply given a sheet of paper with a "monkey's head" printed on it to help them create moving instructions, they often utilized erroneous direction instructions (turning left or right) in Lesson-5. Their mistakes were dramatically reduced after utilizing a toy model (such as a plastic monkey) to replace the piece of paper. They were able to achieve more challenging tasks (even beyond the difficulty of curriculum design).

Table 9 Types of different problems that occurred in the workshop

4.4 Acceptance of Chinese young children

The finding of the participants' self-assessment of emotional efficiency shows that the majority of participants had a positive attitude toward the adapted curriculum. This is consistent with previous researchers' findings that such courses are often appealing to children (BUSUTTIL & FORMOSA, 2020; Rodriguez et al., 2016). The outliers in this research need to be discussed separately - the only unhappy evaluation. We went through the course with the support of a course video and teacher reflection. When sharing learning materials with others, a kid in this course got into an argument. He has been at a disadvantage because of his early age. Even though the conflict was resolved with the teacher's assistance, his mood did not seem to be better. This seems to be a minor incident in the workshop, but as previously said, it is the lack of consideration in curriculum design that leads to the possibility of dispute. This deficit in curriculum design may be compensated for if teachers anticipate potential difficulties and develop intervention measures in advance, but it imposes higher requirements on teachers using CSU (Huang & Looi, 2021). In addition to the young children, their parents strongly appreciate the curriculum implemented in the workshop and constantly inquire about the future curriculum plan during the interview. Based on this, the research concludes that CSU courses adapted from the four series have a high level of acceptability among Chinese urban children.

4.5 Impact of CT levels

Many research has been conducted to investigate the impact of CSU on children's computational thinking levels, and the majority of them have had positive results. Wohl's study, for example, demonstrates that the unplugged computer science teaching method may effectively impart rather complicated topics (such as computational thinking) (Wohl et al., 2015). Recent research by Yuliana and Saxena also indicates that CSU courses may help young children dramatically enhance their level of Computational Thinking (Saxena et al., 2020; Yuliana I et al., 2021). Chinese researchers have recently conducted various CSU courses to explore the enhancement of students' Computational Thinking (Sun, D. et al., 2021; Sun, L. et al., 2021). But they haven't paid attention to the situation of young children. This research's conclusion is consistent with the findings of other studies. When compared to children who did not participate in the program, the average score of those who did increase by 1.2 points. However, there are no substantial changes in Symbolic representation and Integration if we look at the changes of children in each item of computational thinking. Other studies have run across similar issues when attempting to assess children's computational thinking. In his most recent study, for example, Relkin discovered that there is a ceiling effect in the CT test of grade-2 pupils (Relkin et al., 2021). As early as 2011, Lee et al. discussed the absence of appropriate methods to test the computational thinking skill of young children (Lee et al., 2011). For young children, a few months of age difference may generate notable differences in their ability. As a consequence, a group of general test questions may not be appropriate for their varying levels of development. As a consequence, although many measurement methods have been employed to test children's computational thinking level following CSU courses, including this research, with positive findings, more adequate and persuasive Computational Thinking measurement tools are still needed.

In general, this research reveals their limitations via curricular implementation after describing and assessing the features of the unplugged computer science curriculum in the four nations. These issues may be utilized as a lesson for CSU curriculum implementation. Furthermore, we explored the adoption of these courses among Chinese youngsters as well as their influence on Computational Thinking, with favorable findings.

5 Limitation

We initially intended to conduct this research in kindergartens, but owing to the covid-19 epidemic, all kindergartens were firmly under closed management, and researchers were unable to get access. As a consequence, this research can only be conducted in the form of workshops, with a limited number of participants. Although we have attempted to identify the most prevalent problems with the present CSU curriculum in practice, there may be more flaws that need to be investigated if more students are involved.

The participants did not know each other since they were recruited via the internet, and the workshop setting was unfamiliar territory to them. Because the children were too apprehensive to complete the computational thinking exam before the start of the course, there was no stringent pre-test in this research. Unfamiliar surroundings, on the other hand, will have an impact on children's performance, therefore the study findings will not accurately represent their level (Crescenzi - Lanna, 2020).

Although the curriculum was well accepted by the children in this research, this does not guarantee that it would have the same impact in kindergarten. Three professional educators led 15 children in the program, allowing each kid to receive individual attention. However, there is only one professional teacher and three teaching assistants in China's kindergarten curriculum, and they must deal with classrooms of more than 30 students. More adaptations and revisions to the present CSU curriculum are required to handle such challenges.

The computational thinking measurement given by the Bebras organization is employed directly in this investigation. Despite their widespread usage across the world, their reliability and validity have yet to be properly validated (DAGIENĖ & STUPURIENĖ, 2016). Furthermore, we discovered that certain test questions have a ceiling effect, which means they may not accurately represent children's skill levels. As a result, the study's conclusion of enhancing children's computational thinking capacity may need a better CT test to validate.

6 Conclusion

People's everyday lives are becoming more reliant on information technology, and the age group for computer science education is broadening. Young children have been included in computer science education due to the attention of countries all over the world. Unplugged computer science, out of all of these methodologies, is better in tune with children's physiological and cognitive features. The positive impact of CSU on children's computational thinking has been explored by several researchers. Few people, however, wonder whether these courses are appropriate for young children in practice. This research summarizes the characteristics of four existing series of unplugged courses by comparing and analyzing their contents, including the practicality of the teaching plan, the abundance of activities, the interest of material, and the creativity of resources. Then, these courses were applied in practice after reorganization and integration. This research illustrates four types of issues in practice, including mismatch of content difficulty, failure to consider ability differences, excessive emphasis on cooperation, and requirements for abstract ability. Some of these issues need adjusting the curriculum content, while others necessitate teachers being flexible in the classroom. However, we argue that the computational thinking framework utilized in these curriculum designs fails to account for the cognitive and physiological development levels of young children, which is the fundamental cause of the issues. On the other hand, this research explored the acceptance of CSU courses and the influence on computational thinking in Chinese young kids, revealing positive results. It is conceivable that if these courses are more closely aligned with their developmental stage, children's learning impact of computer science will be more dazzling.

Statements on Open Data and Ethics

The dataset used in this article is available by contacting the corresponding author.

All the participants and their parents were told that they can quit the workshop at any time for any reason, and the parents also signed a consent form at the beginning of the workshop.

Declaration of Conflicting Interests

There are no potential conflicts of interest concerning the research, authorship, and/or publication of the

article.

Funding

The author received no financial support for the research, workshop, authorship, and/or publication of the article.

References

Agyei, E.E.Y., Laru, J., & Mäkitalo, K. (2019). Designing unplugged activities for learning computational thinking in the context K-2 pupils' afterschool coding club. In S.C., Andone, D., Biswas, G., Hoppe, H.U., Hsu, T.C., Huang, R.H., Kuo, B.C., Li, K.Y., Looi, C.K., Milrad, M., Sheldon, J., Shih (Ed.), *Proceedings of International Conference on Computational Thinking Education 2019* (pp. 366–369).

Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. Brussels: European Schoolnet.

Battal, A., Afacan Adanır, G., & Gülbahar, Y. (2021). Computer Science Unplugged: A Systematic Literature Review. *Journal of Educational Technology Systems*, *50*(1), 24–47. https://doi.org/10.1177/00472395211018801

Bebras. (2016). *Bebrasuplugged*. https://digitalcareers.csiro.au/en/Resources/Bebras-Unplugged

Bell, T., Witten, I. H., & Fellows, M. (1998). *Computer Science Unplugged…off-line activities and games for all ages* (1st ed.). http://csunplugged.org

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008). Computer science without computers:new outreach methods from old tricks. In *21st Annual Conference of the National dvisory Committee on Computing Qualifications (NACCQ 2008)cil*.

Bers, M. U. (2021). *Coding as a playground. Programming and computational thinking in the early childhood classroom.* (2nd ed.). New York NY: Routledge.

Brackmann, C. P., Barone D A C, Boucinha R M, & Reichert J (2019). Development of Computational Thinking in Brazilian Schools with Social and Economic Vulnerability. *International Journal for Innovation Education and Research*, *7*(4), 79–96. https://doi.org/10.31686/ijier.Vol7.Iss4.1390

BUSUTTIL, L., & FORMOSA, M. (2020). Teaching Computing without Computers: Unplugged Computing as a Pedagogical Strategy. *Informatics in Education*, *19*(4), 569–587. https://doi.org/10.15388/infedu.2020.25

code.org. *Feeling Faces - Emotion Image*. https://docs.google.com/document/d/1qdt7Y-wVkqFas_TkNO4iITz2K-LEmEVWDWLM50NNAOs/view

code.org. *Happy Loops*. https://curriculum.code.org/csf-20/coursea/7/

code.org. *Happy Maps*. https://curriculum.code.org/csf-20/coursea/3/

code.org. (2021). *2021 State of Computer Science Education*. https://advocacy.code.org/

Codekinderen.nl. *Cody Roby*. https://maken.wikiwijs.nl/100525/CodeKinderen#!page-6017161

Codekinderen.nl. *Knutsel je eigen computer*. https://maken.wikiwijs.nl/100525/CodeKinderen#!page-3963633.

Computacional.com.br. *Como é um computador por dentro?*

https://www.computacional.com.br/atividades/comoeumcomputadorpordentro.pdf.

Computacional.com.br. *Crie Um App (Canvas e Storyboard)*. https://docs.google.com/document/d/1kvrJ6nRYkIKO4SCyjMfqzwcz0cqUqjCQY4Kv36sGASo/edit?usp=sharing&usp=embed_facebook

Conde, M. Á., Fernández-Llamas, C., Rodríguez-Sedano, F. J., Guerrero-Higueras, Á. M., Matellán-Olivera, V., & García-Peñalvo, F. J. (2017). Promoting Computational Thinking in K-12 students by applying unplugged methods and robotics. In J. M. Dodero, M. S. I. Sáiz, & I. R. Rube (Eds.), *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality* (pp. 1–6). ACM. https://doi.org/10.1145/3144826.3145355

Crescenzi-Lanna, L. (2020). Multimodal Learning Analytics research with young children: A systematic review. *British Journal of Educational Technology*, *51*(5), 1485–1504. https://doi.org/10.1111/bjet.12959

CSUnplugged.org. *Colour by numbers*. https://csunplugged.org/en/topics/image-representation/unit-plan/colour-by-numbers/

CSUnplugged.org. *Investigating variations using the Sorting Network*. https://csunplugged.org/en/topics/sorting-networks/unit-plan/investigating-variations-using-the-sorting-network-junior/

CSUnplugged.org. *Parity magic*. https://csunplugged.org/en/topics/error-detection-and-correction/unit-plan/parity-magic-junior/.

CSUnplugged.org. *Reinforcing numeracy through a Sorting Network*. https://csunplugged.org/en/topics/sorting-networks/unit-plan/reinforcing-numeracy-through-a-sorting-network-junior/

CSUnplugged.org. *Rescue Mission*.]https://csunplugged.org/en/topics/kidbots/unit-plan/rescue-mission/

DAGIENĖ, V., & STUPURIENĖ, G. (2016). Bebras - a Sustainable Community Building Model for the Concept Based Learning of Informatics and Computational Thinking. *Informatics in Education*, *15*(1), 25–44. https://doi.org/10.15388/infedu.2016.02

Del Javier Olmo-Mu-noz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, *150*(1), 103832.1-19. https://doi.org/10.1016/j.compedu.2020.103832

DELAL, H., & ONER, D. (2020). Developing Middle School Students' Computational Thinking Skills Using Unplugged Computing Activities. *Informatics in Education*, *19*(1), 1–13. https://doi.org/10.15388/infedu.2020.01

European Schoolnet. (2014). *Computing our future: Computer programming and coding.* Belgium: European Commission.

Feldman, R. S. (2006). *Development across the life span, 4th ed.* Pearson Education.

Hsu, T.-C., & Liang, Y.-S. (2021). Simultaneously Improving Computational Thinking and Foreign Language Learning: Interdisciplinary Media With Plugged and Unplugged Approaches. *Journal of Educational Computing Research*, *59*(6), 1184–1207. https://doi.org/10.1177/0735633121992480

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

Hylke H. Faber, et al. (2017). Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network*, *12*, 13–24. http://jeten-online.org/index.php/jeten/article/view/131

Un-Jou K, Ki-Won N, & Lee J (2020). Exploring the effects of unplugged play for children aged 3, 4 and 5- Based on Bee-bot. *International Journal of Advanced Culture Technology*, *8*(2), 239–245. https://doi.org/10.17703/IJACT.2020.8.2.239

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37. https://doi.org/10.1145/1929887.1929902

Lee, J., & Junoh, J. (2019). Implementing Unplugged Coding Activities in Early Childhood Classrooms. *Early Childhood Education Journal*, *47*(6), 709–716. https://doi.org/10.1007/s10643-019-00967-z

Leifheit, L., Jabs, J., Ninaus, M., Moeller, K., & Ostermann, K. (2018). Programming unplugged: An evaluation of game-based methods for teaching computational thinking in primary school. In Melanie Ciussi (Chair), *ECGBL 2018 12th European Conference on Game-Based Learning.* Symposium conducted at the meeting of Academic Conferences and publishing limited.

Livingstone, I. (2012). *Teach children how to write computer programs.* The Guardian. Guardian News and Media. www.theguardian. com/commentisfree/2012/jan/11/teachchildrencomputerprogrammes

Metin S. (2020). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education*, *5*(3), 39. https://doi.org/10.1007/s10798-020-09616-8

Mioduser, D., Levy, S. T., & Talis, V. (2009). Episodes to scripts to rules: concrete-abstractions in kindergarten children's explanations of a robot's behavior. *International Journal of Technology and Design Education*, *19*(1), 15–36. https://doi.org/10.1007/s10798-007-9040-6

Relkin, E., R., L. de, & Bers, M. U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education. *Journal of Science Education and Technology*, *29*(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x

Relkin, E., Ruiter, L. E. de, & Bers, M. U. (2021). Learning to code and the acquisition of computational thinking by young children. *Computers & Education*, *169*, 104222. https://doi.org/10.1016/j.compedu.2021.104222

Rodriguez, B., Rader, C., & Camp, T. (2016). Using Student Performance to Assess CS Unplugged Activities in a Classroom Environment. In A. Clear, E. Cuadros-Vargas, J. Carter, & Y. Tupac (Eds.), *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 95–100). ACM. https://doi.org/10.1145/2899415.2899465

Saxena, A., Lo, C. K., Hew, K. F., & Wong, G. K. W. (2020). Designing Unplugged and Plugged Activities to Cultivate Computational Thinking: An Exploratory Study in Early Childhood Education. *The Asia-Pacific Education Researcher*, *29*(1), 55–66. https://doi.org/10.1007/s40299-019-00478-w

Sun, D., Ouyang, F., Li, Y., & Zhu, C. (2021). Comparing learners' knowledge, behaviors, and attitudes between two instructional modes of computer programming in secondary education. *International Journal of STEM Education*, *8*(1), 54. https://doi.org/10.1186/s40594-021-00311-1

Sun, L., Hu, L., & Zhou, D. (2021). Improving 7th-graders' computational thinking skills through unplugged programming activities: A study on the influence of multiple factors. *Thinking Skills and Creativity*, *42*, 100926. https://doi.org/10.1016/j.tsc.2021.100926

Sysło, M. M., & Kwiatkowska, A. B. (2014). Playing with computing at a children's university. In C. Schulte, M. E. Caspersen, & J. Gal-Ezer (Eds.), *Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE '14* (pp. 104–107). ACM Press. https://doi.org/10.1145/2670757.2670790

Uzunboylu, H., Kınık, E., & Kanbul, S. (2017). An analysis of countries which have integrated coding into their curricula and the content analysis of academic studies on coding training in Turkey., *6*(4), 783–791. https://doi.org/10.18421/TEM64-18

Wetzel, S., Milicic, G., & Ludwig, M. (2020). Experiencing Computational Thinking and the Concept of Loops in an Outdoor CS Unplugged Approach. In M. Ludwig (Ed.), *Research on Outdoor STEM Education in the digital Age. Proceedings of the ROSETA Online Conference in June 2020* (pp. 187–194). WTM-Verlag. https://doi.org/10.37626/ga9783959871440.0.23

Wohl, B., Porter, B., & Clinch, S. (2015). Teaching Computer Science to 5-7 year-olds : An initial study with Scratch, Cubelets and unplugged computing. In *ACM International Conference Proceeding Series* (09-11-November-2015, pp. 55–60). https://doi.org/10.1145/2818314.2818340

Yuliana I, Hermawan H D, Prayitno H J, Ratih K, Adhantoro M S, Hidayati H, & Ibrahim M H (2021). Computational Thinking Lesson in Improving Digital Literacy for Rural Area Children via CS Unplugged. *Journal of Physics: Conference Series*, *1720*, 012009.1-8. https://doi.org/10.1088/1742-6596/1720/1/012009

Appendix A Evaluation Results

Name	Algorithm		Modularity		Control Structures		Representation		Hardware /Software	Design Process	De
	Sequencing	Logical organization	Decomposition	Integration	Pattern recognition	Condition	Symbolic representation	Modeling	Recognition	Revision/Editing	Identi y proble ms
uessing game	✓	✓									
uessing game II	✓	✓	✓		✓			✓			
inary count		✓				✓	✓		✓		
Binary sort	✓					✓	✓				
Binary & Alphabetic	✓	✓	✓	✓		✓	✓	✓	✓		
oloring game		✓	✓				✓		✓	✓	

Parity magic		✓	✓		✓						
escue mission	✓	✓	✓	✓	✓		✓			✓	✓
tness exercise	✓	✓	✓	✓	✓	✓	✓			✓	✓
rting network	✓	✓	✓	✓		✓	✓				✓
rting network II	✓	✓	✓	✓		✓	✓	✓			✓
Happy map		✓	✓	✓						✓	✓
Happy loop		✓	✓	✓	✓					✓	✓
Big event!						✓	✓				
Move! Move!		✓	✓	✓		✓					
Come& Dance!			✓		✓		✓				
Right application									✓	✓	
y robot friend	✓	✓	✓	✓		✓	✓			✓	✓

(Continued)

Appendix A Evaluation Results (Continued)

Name	Algorithm		Modularity		Control Structures		Representation		Hardware /Software	Design Process	De
	Sequencing	Logical organization	Decomposition	Integration	Pattern recognition	Condition	Symbolic representation	Modeling	Recognition	Revision/Editing	Identi y proble ms
My robot friend	✓	✓	✓	✓		✓	✓			✓	✓
Binary Bracelet							✓		✓		
obot friend (loop)	✓	✓	✓	✓		✓	✓			✓	✓
onditional Poker						✓					
semble computer		✓	✓	✓					✓	✓	
earning machine language	✓	✓	✓	✓			✓			✓	✓
GO programming language	✓	✓	✓	✓			✓			✓	
ogramming each other	✓	✓	✓	✓	✓	✓	✓			✓	✓
mpile your name	✓	✓	✓	✓	✓		✓		✓	✓	
Cake making	✓	✓	✓	✓			✓	✓		✓	✓
aking sandwiches			✓	✓		✓	✓				✓
Design my app			✓	✓						✓	
esign my website			✓	✓						✓	
Cody Ruby		✓	✓	✓		✓	✓				

(Continued)

Appendix A Evaluation Results (Continued)

Name	Algorithm		Modularity		Control Structures		Representation		Hardware /Software	Design Process	De
	Sequencing	Logical organization	Decomposition	Integration	Pattern recognition	Condition	Symbolic representation	Modeling	Recognition	Revision/Editing	Identi y proble ms
What's in the computer		✓							✓		
Binary card							✓				
Drawing instruction	✓	✓	✓	✓	✓		✓				
ogical control		✓		✓		✓		✓			
Monica's Day	✓	✓	✓	✓			✓				
onica's Day II	✓	✓	✓	✓	✓		✓				✓
oloring game		✓					✓		✓		
emote control		✓					✓		✓	✓	
Algorithms in music		✓	✓	✓	✓	✓					
Cupcake programming	✓	✓	✓	✓	✓		✓	✓		✓	✓
Paper doll		✓	✓	✓	✓					✓	
My app			✓	✓					✓	✓	
gorithm parking lot	✓	✓	✓	✓	✓	✓	✓				✓
ungle escape	✓	✓	✓	✓	✓	✓	✓	✓			✓

Appendix B Self-evaluation Card

Appendix C The CT test