Research on Scratch Programming Teaching Design for the Cultivation of Computational Thinking

Xiao Wang* and Ya-lin Yin
Jianghan University, Wuhan, Hubei, China

*Corresponding author's e-mail:
1711335396@qq.com

Abstract With the rapid spread and development of information today, computational thinking has become an important skill in the current era of highly information and wisdom. It is an important part of the core literacy of high school information technology curriculum, and its cultivation is a long-term process. This study designed a teaching design model for Scratch programming in junior high school information technology aimed at cultivating computational thinking, and based on this model, specific teaching designs were carried out and applied to actual junior high school information technology classrooms. The article uses "storytelling" as a specific teaching case for detailed explanation. After a month of teaching practice, the research results obtained through the evaluation of students' pre and post computational thinking level show that under this teaching model, students' computational thinking level has been significantly improved. This provides some reference for the cultivation of students' computational thinking and the teaching practice of Scratch programming in the future.

Keywords Computational thinking; Scratch programming; Teaching design; Junior High School Information Technology

Cite This Article Wang X and Yin Y L. Research on Scratch Programming Teaching Design for the Cultivation of Computational Thinking. Advanced Journal of Engineering. 2023,3:30-36. https://doi.org/10.55571/aje.2023017

FOUR DIMENSIONS PUBLISHING GROUP

Received: July 3 2023 / Accepted: July 11, 2023 / Published: July 25, 2023

Introduction

In 2018, the Ministry of Education issued the Information Technology Curriculum Standards for Ordinary Senior High Schools (2017 Edition), which included computational thinking as the core quality of middle school information technology for the first time and proposed that the goal of high school information technology courses is to cultivate students' information literacy and computational thinking[1]. In addition, in the work Essentials of Education Informatization and Network Security in 2019 issued by the Ministry of Education in 2019, it is pointed out that artificial intelligence-related courses should be gradually set up in primary and secondary schools, and programming education should be gradually promoted[2]. Programming and one of the core aim of education is to cultivate students thinking, visible for computing thinking has a wide range of interest and attention, but how to calculate thinking integrated into high school information technology still faces many unsolved problem, if you can, and how to the cultivation of permeability calculation through programming teaching thinking remains to be seen in the corresponding teaching design and application. In view of this, it is of extensive theoretical and practical significance to train computational thinking and combine Scratch programming in teaching design and application.

Literature Review

Computational Thinking

Computational Thinking (CT) is also called Computational literacy and Computational thinking. The concept of computational thinking was first proposed by Seymour Paper, who believed that computational thinking is a way to show the relationship between programming and thinking skills. Students can promote the cultivation of learners' procedural thinking by programming with LOGO language [3]. At present, the research and practice of computational thinking has been paid attention to by many international organizations and groups, including the International Society of Educational Technology, the National Association of Computer Science and Technology Teachers, etc., and provide a large number of resources to cultivate computational thinking, committed to the K-12 stage of the curriculum, in order to cultivate students' computational thinking ability. Compared with foreign countries, the research on computational thinking in China is still more focused on the conceptual level. Professor Zhou Yizhen defined computational thinking in 2006, "Computational thinking refers to a series of thinking activities covering the breadth of computer science, such as problem solving, system design and human behavior understanding by applying basic concepts of computer science" [4]. Later, with the deepening of research, computational thinking has become an important part of the core literacy of information technology courses in senior high schools in China[5]. How to cultivate students' computational thinking in combination with specific disciplines has become a hot topic in the field of educational technology.

Scratch programming

Scratch is a set of programming software developed by MIT for children over 8 years old. Through Scratch, children can creatively design their own programs, including animation, games and interactive stories. Learning programming with it gets rid of the complicated syntax of program design and the boring format of program call. Students do not need to memorize the code, but only need to click and drag the corresponding code block to the script area to realize the programming [6]. Children can express themselves through Scratch, which helps them develop learning skills, develop creative thinking,

and learn to solve problems in their lives. At present, Scratch software is widely used abroad and in China. In programming institutions and schools across China, Scratch takes up a large proportion and there are many people learning it. However, since Scratch education started late in China, it still needs to be further improved and perfected in many aspects[7]. For example, what kind of teaching mode is used for Scratch education? How can Scratch education and computational thinking be better integrated? It also needs educators to promote the development of Scratch programming teaching on the basis of active practice and exploration.

Teaching Design Model

Scratch visual programming software is used to train students' computational thinking. Problem exploration is mainly taken as teaching driving force. Around a thematic task, tasks are decomposed step by step through problem driving to form problem solutions, and finally problems are solved through visual programming programs, namely, works are generated[8]. In the process of problem analysis and solution, students' computational thinking ability is cultivated. Therefore, the model elements of Scratch programming teaching design oriented to the cultivation of computational thinking include computational thinking, exploration activities and teaching behaviors of teachers and students. Therefore, this study takes computational thinking as the core, problem inquiry as the means, and combines teachers' guidance and inspiration with students' positive thinking to form a Scratch programming teaching design model oriented to the cultivation of computational thinking, as shown in Figure 1.This model takes the cultivation of computational thinking as the core, connects the teacher's behavior with the student's behavior, and penetrates the cultivation of computational thinking from the whole process of inquiry.

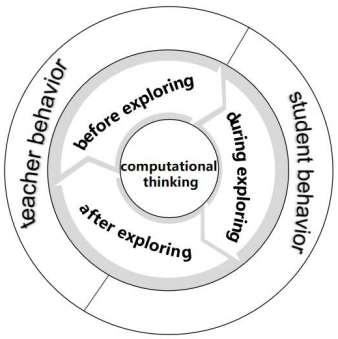

Figure 1. Teaching design model

First of all, before the exploration, the teacher should create a certain problem situation to arouse students' thirst for knowledge, and carry out Scratch programming teaching around this problem situation. For example, the problem situation of "farmer crossing the river" is used to guide students to think positively about how to make the farmer cross the river smoothly through design and programming. Secondly, in the exploration, the task is mainly solved by individual, supplemented by group assistance; Teachers should guide students to decompose problems and tasks, give priority to students' independent exploration, put forward their own solutions. Finally, after the inquiry, the teacher should guide the students to reflect on the whole process of inquiry, sort out the code used, and consolidate the knowledge learned.

Teaching design case

This study takes "storytelling" as a case to carry out specific teaching design, including front-end analysis, teaching process and computational thinking (as shown in Figure 2). Computational thinking penetrates into the teaching process, so it is introduced in the following two parts.

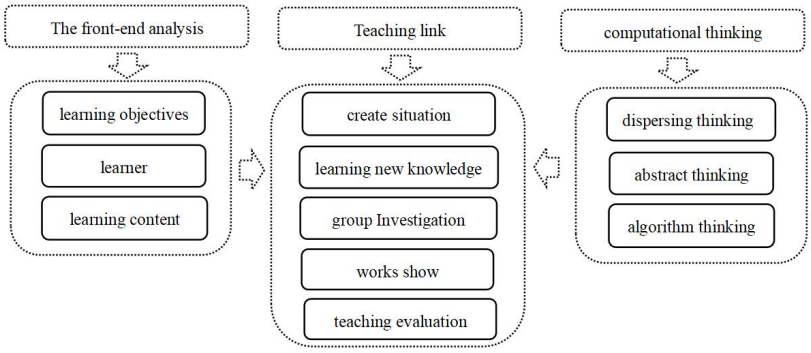

Figure 2. Instructional design

Front end analysis

Before class, teachers must do enough front-end analysis to better carry out the implementation of teaching links, mainly including learning objective analysis, learner analysis and learning content analysis.[9].

a: Learning objectives

Knowledge and skills: Grasp the concept of algorithms; Able to extract key information from problem situations; Learn to use Scratch programming software, including: learn to add roles and backgrounds; Use the startup statement of the program; Learn to switch between characters and backgrounds.

Process and method: Cultivate decomposition thinking in the process of writing storytelling programming, master the thinking method of decomposing complex problems into several simple ones; Develop divergent thinking and imagination in thinking about code combinations.

Emotional attitude and values： Experience the value of computational thinking, cultivate the consciousness of using computers to solve problems, mobilize the enthusiasm of learning programming.

b: Learner analysis

For junior middle school students, they are unfamiliar with relevant knowledge of algorithm and lack of cognition in this aspect. They do not understand calculation thinking, and their information consciousness is unclear and their cognition is not clear. However, most of the middle school students like to listen to stories and have strong interest in how to let the characters in the video tell stories. Moreover, most of the students are in touch with programming software for the first time and have a certain curiosity about programming learning. They have rich imagination and have certain programming learning ability.

c: Learning content

As for the knowledge of algorithm, the conceptual content is relatively abstract, which requires information technology teachers to integrate their own knowledge and think and design methods to improve students' computational thinking in the teaching process.

Teaching focus: understand the concept of algorithm, train students' decomposition thinking, calculation thinking; Through comprehensive analysis and serious thinking of the problem situation, the students can design and arrange the program, and finally solve the problem task.

Teaching difficulties: guide students to learn how to use flow chart to describe algorithms, and cultivate students' love of programming.

Teaching link

a: Create situation

Teacher: In our life, we often listen to our parents and friends tell us stories. Today, let's create interesting stories for them, ok? Let's watch a story about penguins and ducks first. (Playing Scratch video), would you like to make a story telling program like this? So let's learn about this amazing programming software called Scratch.

b: Learning new knowledge

Teacher introduces some functional areas of Scratch and shows students a Scratch teaching video to guide students to think about what steps are needed to write a "story". Develop students' decentralized thinking and abstract thinking. Break the problem down and have everyone write their own "story" as a group.

c: Group Investigation

Students discuss tasks and design schemes in groups, and then write programs and arrange and combine them. Teachers should walk down the platform to guide students, guide students to develop their imagination to write their own "story telling" procedures, the development of students' algorithm thinking and imagination.

d: Works show

After debugging the program and completing the creation, students can submit their works. A representative of each group will introduce their own video of "telling stories" program. Teachers and other groups will evaluate and revise their works.

Finally, the teacher made a brief summary at the end to deepen students' knowledge of writing and sorting Scratch programming, strengthen their knowledge and skills, cultivate their computational thinking ability, and learn how to apply computational thinking in practical situations.

e: Teaching evaluation

Teaching evaluation through the whole teaching process, teachers should instruct students when others were on display from the multiple dimensions of formative assessment, as well as to guide the students in the classroom last class performance on their work and summative evaluation, and to our team and other team work rate, and then improve their work. In this way, students can not only feel a sense of accomplishment, discover their advantages and enhance their confidence, but also see their shortcomings in the process of evaluation.

The Research Study

In order to evaluate the effectiveness of the model, the author developed a quasi experimental research design to test the changes in students' computational thinking level. This study used the International Computational Thinking Test (CTt) to measure students' computational thinking level before and after teaching practice, and conducted statistical analysis on the pre test and post test data.

Research objects and tools

The research subjects of this experiment are 30 eighth grade students from a certain high school, including 14 girls and 16 boys. The research tool is the CTt developed by Roma á n Gonza á lez, which has been verified to have good reliability and validity. Before and after the experiment, 30 copies of the computational thinking test questions were distributed to the research subjects, and the questionnaire

recovery rate was 100%.

Research results

Import the collected data into SPSS26.0 for descriptive statistics, and it can be found that the difference between the before and after data is normally distributed, and then conduct paired sample T test. The results are shown in Table 1.

Paired-Samples T Test

	M	t	P
Pretest-Posttest	-4.500	-.955	.347

Table 1. Comparison of test scores before and after CTt

After analysis, it can be seen that the average score of the post test of the computational thinking test is higher than that of the pre test, and there is a significant difference ($P < 0.05$) between the pre and post tests. After a month of Scratch programming learning, students' computational thinking level has significantly improved. In combination with students' participation in class, work completion, group communication, etc., it can also be seen that students' creativity, critical thinking, collaborative thinking, algorithmic thinking and other aspects of computing thinking have been improved to a certain extent.

Conclusion

This study combines the connotation of computational thinking to construct a Scratch programming instructional design model for cultivating computational thinking, and applies it to practical teaching practices in information technology classrooms. The teaching case of "storytelling" provides a detailed description of all teaching stages, and the cultivation of computational thinking permeates every teaching stage. The research results indicate that the teaching model has a significant effect on cultivating students' computational thinking. In short, relying on Scratch programming can transform abstract and complex programming code into simple code blocks, reducing students' programming burden, stimulating their learning interest, promoting the cultivation of their computational thinking, and making a modest contribution to the development of programming education in China [10].

References

1. Ministry of Education of the People's Republic of China. Information Technology Curriculum Standards for Ordinary High Schools (2017). http: //www. Moe. gov. cn /srcsite /A26 /s8001 /201801 /t20180115_324647. html, 2019-03-18.
2. Ministry of Education of the People's Republic of China. Key Points for Education Informatization and Network Security Work in 2019. http: / /www． moe． edu. cn /srcsite /A16 /s3342 /201903 /t20190312_373147. html, 2019-3-18.
3. 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(9), 296–310. https://doi.org/10.1016/j.compedu.2018.07.004
4. Yu Xiao-hua, Xiao Min. Research on the Training mode of computational Thinking based on Visual Programming—Also on the training of computational Thinking in Information Technology Classroom. Journal of Distance Education,2017,35(06):12-20.
5. Fan Wenxiang, Zhang Yichun, Li Yi. A review of computational thinking at home and abroad. Journal of distance education,2018,36(02):3-17.

6. Zhu Libin. Research on Scratch Programming Teaching Practice: Based on the Perspective of Experiential Learning Circle. Modern educational technology,2013,23(07):30-33.
7. Li Yuan. Study on Teaching Strategies of Scratch Programming in Elementary Schools based on Deep Learning Theory.Modern educational technology, Proceedings of 5th International Workshop on Education Reform and Social Science;s (ERSS 2022).
8. Zhang Mailing. Research on Cultivating Primary School Students' Computational Thinking Ability Based on Scratch Programming. China Education Technology Equipment,2023,30(04):1-4.
9. Dong mei. The construction framework of instructional design mode. Research of audio-visual education,2015,36(07):20-27.
10. Fu Qian, Xie Bochao, Zhang Yafeng. An empirical study on the Development of Junior Middle School Students' Computational Thinking in Programming Teaching Based on Graphic Tools. Electronic education research,2019,40(04):122-128