Cultivating Students' Computational Thinking Based on Scratch Programming Teaching

Sixun Liu, Yundi Xu

Jinjiang District Foreign Language Primary School in Chengdu, Chengdu 610011, China.

Abstract

Computational thinking is one of the core competencies in the new curriculum standards for information technology in primary and secondary schools. Scratch programming is an effective tool for cultivating students' computational thinking. Through the combination of classroom and new technologies generated by technological development, teaching is carried out. The use of visual tools such as mind maps is more in line with the physical and mental characteristics of primary school students, can effectively stimulate students' interest, cultivate their ability to analyze and solve problems, and thus enhance their computational thinking ability.

Keywords

Computational Thinking; Scratch Programming; Information Technology.

1. Research Purpose

The Scratch programming software developed by the Massachusetts Institute of Technology Media Laboratory is a graphical programming software for teenagers. In Scratch programming software, code blocks are composed of different shapes of building blocks. Students only need to drag and drop different code blocks with the mouse to build them like building blocks. Based on the application level of Scratch software, using the Scratch programming language as a tool to cultivate students' programming abilities in primary school curriculum content has high operability. And for students with insufficient computer learning abilities, they can still master the Scratch programming language. The learning of programming languages is beneficial for expanding and improving students' thinking skills. Primary school students can use the Scratch programming software to expand their thinking, quickly grasp content, and lay a solid foundation for subsequent learning of other knowledge. The primary school stage is a gradual transition from concrete thinking to abstract logical thinking. It is feasible to use the Scratch programming language to cultivate students' computational thinking. This is because the Scratch programming language can use the form of building blocks, simplifying complex programming languages into building blocks.

In the new curriculum standards for information technology courses in primary and secondary schools, the main goal of the current information technology classroom is not only to cultivate students' memorization knowledge and hands-on skills, but more importantly, to cultivate their computational thinking ability. The teaching of programming for primary school students is not aimed at cultivating programming experts, but rather at hoping that students can learn and experience the logic of programming through building blocks, improve their analytical and problem-solving abilities, and enable teachers to help students think, analyze, and solve problems at the thinking level. The intelligence and thinking of students in primary school are becoming increasingly mature. By conducting programming courses, students' computational thinking can be improved, and it can also lay a foundation for their subsequent learning of other knowledge content.

2. Research Content

Teaching through programming is one of the main ways to cultivate primary school students' computational thinking. But most programming software is relatively professional, such as the Java language, where the coding is not entirely suitable for elementary school students to operate. The interface design stage of Scratch software is interesting and visual, which can stimulate students' interest in learning. Students can learn logical thinking through project analysis and functional implementation. When using Scratch programming in the information technology classroom of elementary school, students can analyze project construction by introducing real-life situational problems. Students can actively think and propose solutions, and have their own solutions verified. This allows students to communicate with each other, expand their thinking, and explore different solutions. Mind maps and flow charts can be used throughout the entire teaching process to assist in teaching. The use of Scratch software for a series of programming situational problems is cultivating students' computational thinking. Next, we will analyze how to cultivate students' computational thinking ability in the classroom by taking "Autonomous Cars" as an example.

2.1. Analysis of Teaching Content

The content of "Autonomous Cars" is selected from a real case in the current technological development process, which can stimulate students' interest in learning. This project mainly teaches students the comprehensive application of commonly used instructions such as adding, looping, judging, and detecting character backgrounds. Through the project 'Unmanned Vehicles', students can proficiently master the analysis process and solutions for writing a project. The project 'Unmanned Vehicles' mainly includes a role that allows students to learn various ways to control the movement of cars, enable autonomous vehicles to automatically avoid obstacles, and record the number of obstacles avoided, learning variables, and other knowledge.

2.2. Analysis of Teaching Objects

The teaching object of the project "Autonomous Cars" is fourth grade students. By the third grade, students had already learned how to build building blocks and were able to apply conditional, judgmental, and loop sentence pairs. When learning, task driven approach can be used to encourage students to group and explore independently, stimulating their ability to actively explore. During class, teachers can present the final effects that autonomous vehicles need to achieve, and then have students analyze and sort out the functions that need to be achieved.

2.3. Teaching Objectives

Information awareness: Able to further understand and apply knowledge of sequence, loops, conditions, and variables. Computational thinking: Able to master the method of transforming abstract problems into practical problems. Reasonably select methods such as mind maps or flow charts to visualize and decompose problems. Digital learning and innovation: Under the guidance of teachers, it is possible to creatively increase the functionality of autonomous vehicles after completing specified tasks. Information society responsibility: programming projects using socially recognized behavioral norms and complying with relevant laws and regulations.

2.4. Key and Difficult Points in Teaching

Key point: Able to draw a mind map or flowchart based on the functions to be achieved by autonomous vehicles, and able to build projects.

Difficulty: Being able to decompose and understand the program execution process and draw a mind map or flowchart.

2.5. Teaching Process

(1) Scenario introduction. Stimulate students' interest and introduce new courses through videos of autonomous vehicles. Show the game project "Unmanned Vehicle" created using Scratch software, and invite students to come and try this game to stimulate their interest in Scratch program design learning, attract their attention, and stimulate their enthusiasm.

(2) New lesson and new teaching. After the students have tried playing the game, ask them questions: "What background and characters are needed to complete the construction of the car in this game, how to move the car forward, how to avoid obstacles, etc.". Students observe the game and discuss the scenarios, functions, etc. that the game needs to implement. They also need to leave space for students to freely expand, and use visual tools such as mind maps to display and summarize, as shown in Figure 1.

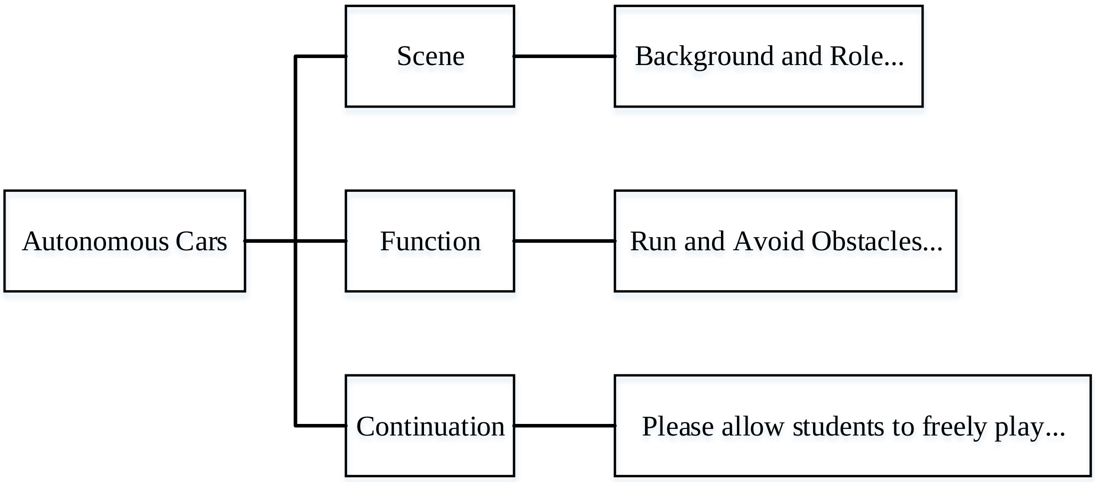
Autonomous Cars
Scene
Background and Role...
Function
Run and Avoid Obstacles...
Continuation
Please allow students to freely play...

Fig.1 Building a Mind Map for Game Analysis

Subsequently, a new Scratch project was created, and road backgrounds and characters were added to the scene according to the game design. From the teacher's prepared material folder, car characters were selected and added to the stage, and the character's shape was edited. Then, appropriate backgrounds were selected from the background library and added to the stage. Adding backgrounds, characters, and character shapes in Scratch is the fundamental knowledge of Scratch programming. This teaching process aims to enable students to proficiently master the methods of adding background characters, and also enable them to draw characters and backgrounds themselves, improving their hands-on abilities.

(3) Draw a flowchart. Based on the already drawn mind map mentioned above, after setting up the scene, students need to implement the functions of the car. Guide students to think about how the functions of these roles can be implemented using program instructions based on the effects and functions presented by small and medium-sized cars in the mind map. Before students use Scratch to add program instructions to characters, they first need to think about how the designed game is played, how the functions of the car are implemented, and how to use code blocks to implement the functions of these characters. Using a flowchart, students can transition from natural language to programming language, and using a flowchart, students can transform abstract problems into programming problems. Taking the functional module of car avoidance as an example, a program flowchart is drawn to demonstrate the implementation process. In program design, using the flowchart can intuitively and effectively describe complex algorithms and programming problems.

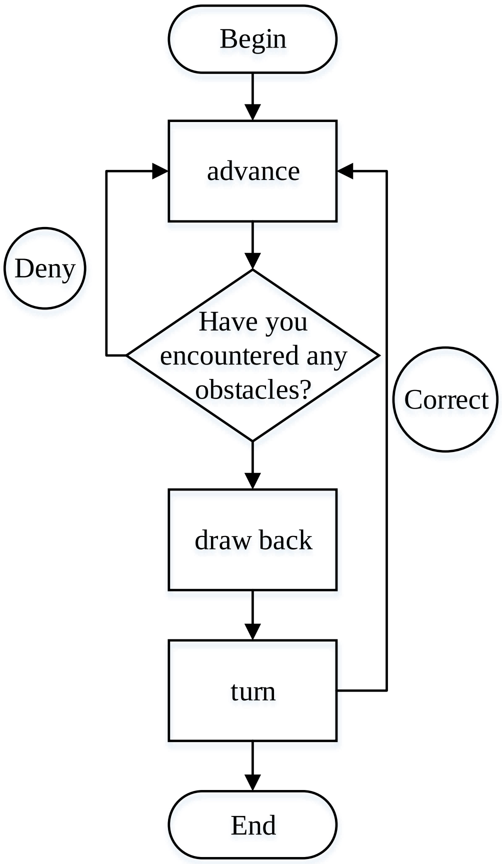

Fig. 2 Flow Chart of Avoiding Obstacles

(4) Build a program. Students draw a program flowchart of the car's functional effects based on the previous process, select program modules and program instructions that can achieve the role's functional effects in Scratch, and build the program to complete the operation of the program.

(5) Summary of knowledge. In the teaching process of knowledge summary, students can be invited to summarize and summarize the gains of this lesson, guiding them to summarize and review which commands are used to implement the various functions of the characters in the scenario of autonomous vehicles. Please show the mind map and program flowchart drawn in this lesson to help students improve their computational thinking. Assign tasks after class to let students continue to improve the game, make the game more interesting, explore more knowledge, let students know that there is still much room for improvement in the game, such as adding the number of obstacles to avoid for the game, creating game projects in Scratch, stimulating students' motivation for independent exploration and active learning, and improving students' ability to actively think, find and solve problems.

3. Conclusion

In the above case, the cultivation of technology and computational thinking brought about by the current real technological development is integrated into Scratch teaching. By using Scratch programming software to do new technologies that occur during the technological development process, students' learning interest can be greatly stimulated. The integration of visualization tools in practical teaching can transform abstract concepts and functions that need to be implemented into practical problems that can be implemented using Scratch, cultivate students' computational thinking, mind maps, and flow charts, which can help students internalize and understand these programming problems. In the following teaching work, it is necessary to propose multiple solutions for achieving the same function, conduct programming tests, and further improve students' computational thinking.

References

[1] Nan Dong, Guangwei Xu: Research on the Teaching Model for Cultivating Innovative Thinking in Primary School Students in the Scratch Environment [J]. Information Technology Education in Primary and Secondary Schools, 2022 (09): 36-38.

[2] Jie Cai:A Practical Study on Scratch Classroom Teaching Model Based on Computational Thinking [J]. China Education Informatization, 2022-28 (06): 111-116.

[3] Yanhua Wu: A Practical Study on the Application of Mind Mapping in Scratch Curriculum to Cultivate Computational Thinking among Junior High School Students [D]. Huazhong Normal University, 2022.