CCF Transactions on Pervasive Computing and Interaction (2023) 5:396–410
https://doi.org/10.1007/s42486-023-00135-8

REGULAR PAPER

CoAR-Maze: empowering children's collaborative tangible programming in augmented reality

Mingyu Zhang[1,2] · Jiaxiang Li[1,2] · Yiyan Lin[1] · Qiao Jin[3] · Danli Wang[1,2]

Received: 20 June 2023 / Accepted: 25 July 2023 / Published online: 1 September 2023

Abstract
As an effective way to develop children's computational thinking, programming education has been intensively studied by scholars. Among them, tangible programming is more in line with children's cognitive development and inherently supports collaborative learning. However, most of the tangible programming systems fall short in effectively stimulating active collaboration among children effectively. In this paper, we present CoAR-Maze, an augmented reality (AR) based tangible programming system that incorporates a collaborative mechanism designed for children's programming. The system assists children in learning programming concepts such as sequences, loops, conditional statements, and task decomposition, while emphasizing the stimulation of active collaboration and enhancing immersion in children's programming learning. User studies validate the effectiveness, usability, and support for collaboration among child users provided by the system.

Keywords Children · Tangible programming · Collaborative learning · Augmented reality · User study

1 Introduction

The previous research indicated that tangible programming not only helps children develop computational thinking, but also improves logical thinking, cognitive, creative, collaborative and social skills, and so on (Bers and Horn 2009; Burke and Kafai 2010). However, traditional text-based and symbol-based programming languages contain complex syntax and instructions are difficult for children to understand and learn (Kelleher and Pausch 2005). Researchers have carried out in-depth research on children's programming education by simplifying and visualizing the complex and abstract traditional text programming language so as to make it easy and available for children. Many computational tools for children have been designed, and a review (Yu and Roque 2018) divided them into two main categories: tangible programming tools and graphical programming tools. Graphical programming tools transform programming concepts into graphics displayed on electronic screens, like Scratch (John, et al. 2010). Tangible programming is the combination of tangible interaction and programming learning (Horn et al. 2009), like T-maze (Wang et al. 2011) and Quetzal (Horn and Jacob 2023). Compared with graphical tools, tangible tools are easier to use and learn for younger children because tangible interaction is better aligned with children's cognitive development ability (Horn et al. 2009).

In recent years, many scholars have been focusing on collaborative learning and its relevance to education. They have discussed the importance of fostering children's collaborative awareness (Slavin 1980; Etel and Slaughter 2019; Gennari et al. 2017) as well as the effectiveness of collaborative programming (Rahman et al. 2020; Farrokhnia et al. 2019; Thieme et al. 2017; Yashiro et al. 2017), highlighting the complementary relationship between collaborative learning and tangible programming. However, there are limited existing tangible programming systems that support collaborative

✉ Danli Wang
danli.wang@ia.ac.cn

Mingyu Zhang
zhangmingyu2022@ia.ac.cn

Yiyan Lin
linyiyan2003@qq.com

Qiao Jin
jin00122@umu.edu

1 Institute of Automation, Chinese Academy of Sciences, Beijing, China

2 School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

3 Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA

learning, and there is also a lack of research on children's collaborative learning in user evaluations.

Based on this background, we have designed a tangible programming system called CoAR-Maze for children aged 8–10. This system consists of nine categories of tangible programming blocks that facilitate and encourage collaboration among users. It allows children to control the actions of virtual characters within the software by using the programming blocks.

In the user study, this study employed single-player mode and co-player mode to complete programming tasks within the CoAR-Maze system. We invited 16 children aged 8–10 to participate, with each pair of participants forming a group. Each group's members first completed a co-player mode programming task, followed by independently completing a single-player mode programming task.

Our work has the following contributions:

- We have designed a new tangible programming system called "CoAR-Maze", which is based on the theory of collaborative learning to stimulate active collaboration among children. The programming environment is designed using augmented reality technology to enhance children's interest and programming experience.
- We have developed a set of collaborative mechanisms specifically tailored for this system, dividing the user's workspace into two sections, left and right, and introducing collaborative elements within the system, requiring the collaboration of two users to complete tasks. By establishing a set of instructions, we facilitate the collaborative process and enhance users' efficiency in collaboration.
- We conducted a comprehensive evaluation of users by analyzing quantitative data and integrating multidimensional data such as experimental observations, questionnaire and interviews. CoAR-Maze is user-friendly for children, as it is based on tangible interaction. Its collaborative mechanisms stimulate active collaboration among children during the learning process, providing an excellent interactive platform. We studied the forms of interaction happened during the collaborative programming. Compared to graphical programming and traditional programming languages, CoAR-Maze has gained high acceptance among child users, especially in its co-player mode.

2 Related works

2.1 Tangible programming

Tangible interaction technology advocates for interacting with real physical interfaces to provide digital information and control functionalities, manipulating target objects or interfaces (Shaer and Hornecker 2010). Tangible programming tools enable children to surpass the limitations of computer devices in learning programming. They can arrange, assemble, and combine physical objects in an open physical space according to certain rules to accomplish programming tasks. The input component of a tangible programming system hides complex programming details by assigning syntax rules and programming semantics to tangible programming blocks or tiles. Through the use of computer vision, sensors, tangible perception, and other technologies, the information encoded in the physical programming sequence is transmitted to the feedback module as the output of programming effects. Depending on the controlled programming objects, tangible programming tools can be classified into pure tangible programming, mixed reality programming, and VR/AR-based tangible programming tools. The following sections will introduce the study of them.

Pure tangible programming refers to a programming approach where both the input and output are composed of touchable physical components. Existing pure tangible programming tools, such as MicroBlock (Cabrera et al. 2019), are real-time programming environments. They consist of a 32-bit ARM microcontroller, sensors, buttons, an accelerometer, and a 5×5 LED grid. These tools support collaborative programming with blocks to control the changing rules of the LED grid. Matatalab (Papadakis 2020), a commercial tangible programming tool that includes a target robot, a control tower, a tangible programming board, and a map component. The camera on the control tower recognizes the programming sequences on the programming board and sends signals to the robot, enabling it to navigate different paths on the map.

Mixed reality programming refers to an approach where the input is composed of physical components, while the output is displayed on electronic device screens. It combines the advantages of tangible interaction and graphical interfaces, providing natural input and diverse graphical feedback. TLogic (Deng et al. 2018) consists of physical programming blocks with special patterns and a virtual environment on the PC. Before programming with TLogic, children need to manage the mutual exclusion relationships between virtual characters and design simple algorithmic logic. They then arrange different programming sequences to control multiple target characters. Lighters (Wang et al. 2023) is composed of a mobile application based on Android and a series of tangible programming blocks. Lighters provides a more diverse creative design environment and a wide range of modules, encouraging children to learn programming together. It is beneficial for fostering children's computational thinking, collaboration skills, and communication abilities.

VR/AR-based tangible programming tools are developed by integrating AR or VR technologies into the mixed reality approach. This allows for the utilization of the advantages of tangible interaction while enhancing the benefits of graphical interaction. The combination of tangible programming tools and VR/AR technologies enhances user interest in learning programming through novel forms of programming feedback. Furthermore, it enhances user immersion and improves the learning experience by providing rich, three-dimensional virtual environments and real-time visual feedback. AR-Maze (Jin et al. 2018) consists of a game application and a set of programming blocks, which can be classified into three types: start/end blocks, motion blocks, and loop blocks. Players can use tangible programming blocks to plan paths for virtual characters within the game, enabling the virtual characters to complete level objectives. AR–C&P (Zhao et al. 2022) supports the creation of three-dimensional scenes based on AR and provides various real-time feedback. By combining virtual systems with low-cost physical materials, it can enhance children's cognitive development and programming knowledge.

2.2 Collaborative learning and programming

The concept of collaboration initially referred to a pattern in which a group of individuals work together and assist one another to collectively accomplish a task (Dillenbourg 1999). The difference between collaborative learning and cooperative learning is that cooperative learning divides the partners within a small group into different parts and each member completes their respective tasks separately (Stahl et al. 2006). Each person acts independently, and the final results are then consolidated. Research has shown that pair collaboration tends to increases learners' enjoyment, engagement, and motivation, especially when children can engage in face-to-face communication and physical devices facilitate cooperation (Inkpen et al. 1999; Scott et al. 2003). The study (Xing et al. 2020) seem to indicate that collaboration can promote children’s ability of decision-making and communication by comparing the single-player mode. Collaborative playing is important for learning programming, as it can help reduce programming errors and players’ anxiety towards programming (Melcer and Isbister 2018). Particularly, pair programming refers to students’ collaboration in programming, in which two players collaborate to complete the programming tasks by respectively play a role (Begel and Nagappan 2008). Compared with single-player programming, pair programming can better increase students' interest in programming, stimulate their motivation for learning. And it could make students more concentrated, so that their programming ability is improved (Dai 2020; Thapaliya 2020).

CoProStory (Deng et al. 2019) consists of a PC-based animation program and physical programming blocks. It enables two children to create individual target characters and use the tangible programming blocks to program their respective characters. The programming allows them to control the characters' cooperation in completing a task within the virtual environment on the PC. Torino (Thieme et al. 2017) consists of several beads with varying appearances and tactile features, as well as different components. The beads are used as inputs, and audio feedback is provided as outputs. Children, regardless of visual impairments, can engage in cooperative learning using Torino by collaboratively connecting the programming beads in a specific sequence to generate different musical outputs.

3 Summary

The above work demonstrates outstanding achievements in the field of tangible programming. However, current systems face certain challenges. Firstly, users lack sufficient feedback during the completion of tasks, which can result in a high cognitive load, especially for younger children. Secondly, we have observed a lack of mobile-based physical programming systems that support collaborative learning among children. It has been confirmed that unrestricted collaboration can lead to confusion and inefficiency (Dillenbourg and Over-Scripting 2002). Establishing a set of instructions, such as guidelines on how group members should collaborate, can make the process more effective (Dillenbourg 1999).

To address the issues of insufficient user feedback and incomplete collaboration in existing tangible programming systems, this study introduces a new tangible programming system called CoAR-Maze. On one hand, it allows children to manipulate virtual characters in the system using tangible programming blocks. The system provides real-time feedback to users and offers prompts at appropriate times to assist them in completing tasks. On the other hand, drawing from the concept of pair programming (Thapaliya 2020) and user studies on mobile storytelling (Fails et al. 2010), which explore the effects of different collaboration modes (including content division and spatial sharing) on children's use of mobile stories, we designed both single-player mode and co-player mode in the CoAR-Maze system. In the co-player mode, children are required to collaborate with a partner using tangible programming blocks to accomplish complex tasks within a certain timeframe. We further analyzed the different effects of the two mode on children's programming learning outcomes and experiences. Our results indicate that the system reduces cognitive load for children during usage, facilitating their programming learning. Furthermore, compared to the single-player mode, children showed a greater willingness to engage in learning activities using the co-player mode.

4 System description

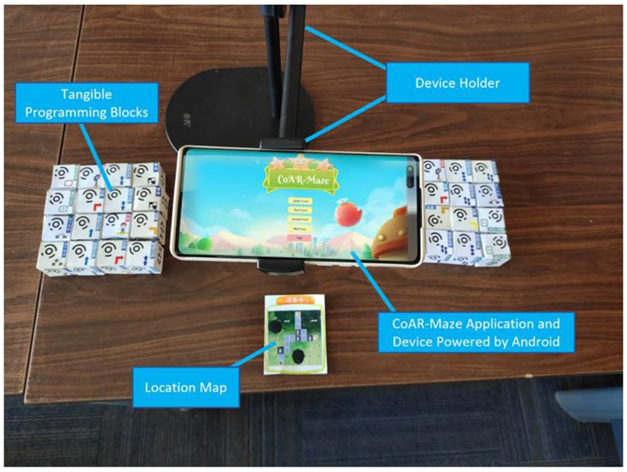
4.1 Overview

CoAR-Maze is a tangible programming tool designed for children aged 8–10. It consists of tangible programming blocks, location map, a mobile Android device, a device holder, and an AR-enabled app, as shown in Fig. 1. The programming environment consists of two 3D digital characters and a 3D maze map. To complete the programming task successfully, users are required to work together and utilize tangible programming blocks to control their respective characters, ensuring that both characters navigate through the maze and reach the endpoint. Our goal is to provide children with a more engaging and collaborative tangible programming experience through an AR environment.

4.2 Tangible programming blocks

The tangible programming blocks are made by 3 cm cubic wooden blocks. The top, bottom, front, and back faces of the blocks are affixed with TopCode (TopCode: Tangible Object Placement Codes) stickers, while the left and right faces serve as connection surfaces. The connection surfaces are divided into positive and negative poles, each equipped with a corresponding magnetic circular magnet. This type of connection design serves two purposes. On one hand, it assists children in ensuring the correctness and consistency of program sequences during programming. On the other hand, it facilitates the rotation of the top, bottom, front, and back faces around the magnet, thereby reducing the number of programming blocks required. In this system, the programming blocks are classified into nine types based on their semantics and functionalities, as shown in Fig. 2.

- Start block and end block: each program sequence should begin with a start block and end with an end block.
- Jump block: controls the character's jumping behaviour, divided into two types: up and down. These two types of blocks are placed on two different faces of the same regular hexahedron.
- Turn block: controls the character's in-place turning. It is further divided into two types: left turn and right turn, which are respectively placed on two opposite faces of the same regular hexahedron.
- Forward block: responsible for the forward movement of the character. Based on the length of the feasible path, the forward block is further divided into four types: forward 1–4 steps.


Tangible Programming Blocks
Device Holder
CoAR-Maze
CoAR-Maze Application and Device Powered by Android
Location Map
Fig. 1 Overview of CoAR-Maze system

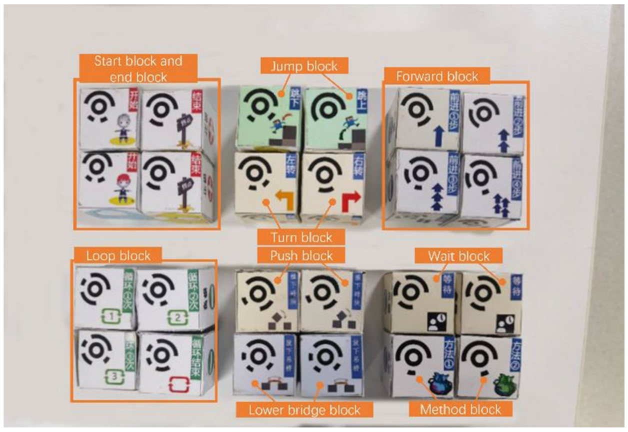


Fig. 2 Tangible programming blocks

- Loop block: divided into loop start blocks and loop end blocks, they are used together to form a complete loop structure.
- Push block: it can push the extra blocks ahead on the path, filling in the gaps in front, ensuring smooth progress for oneself and teammates.
- Lower bridge block: used at the location with a drawbridge button. It can assist teammates in lowering the drawbridge along the path, ensuring smooth progress for the other party.
- Wait block: coordinates the sequence of actions between two characters, allowing the character to wait for 1 to 4 s.
- Method block: helping the characters solve their respective problems.

To facilitate the collaborative operations between the two digital characters, this system establishes rules regarding their displacement and timing.

Displacement rule: only forward block can change the character's location. Action blocks such as turn left, turn right, lower bridge, push block, and the wait block are responsible for specific actions (e.g., altering the character's orientation), while keeping their position coordinates unchanged.

Timing rule: regardless of whether the character is moving or not, each action block consumes a fixed amount of time (one second). For example, moving forward by one step requires one second, and actions such as turning left, turning right, lowering the bridge, and pushing blocks also take one second.

4.3 Location map

Location map is a custom card, designed to identify the position of the virtual map. Within the CoAR-Maze game, the location map enables the detection and tracking of existing 3D models. As the position and orientation of the location map change, the associated virtual map dynamically adjusts accordingly.

4.4 CoAR-Maze application

The CoAR-Maze application was developed by Unity 3D, integrating the Qualcomm Vuforia platform to achieve AR effects. Location map serves as AR markers, allowing the application to scan and overlay 3D maze task. Users can select specific levels to generate different maze. The application consists of three stages: initialization stage, programming stage, and execution stage.

In the initialization stage, users select a level to start the game. Once a specific level is entered, the system recognizes the location map within the visible range and generates the 3D maze environment with digital characters.

In the programming stage, two users collaborate and share a set of tangible programming blocks, programming simultaneously on the left and right sides of the recognized area. As users create correct programming sequence, the system superimposes the feedback arrows above the 3D maze. To differentiate the feedback for each character, the system utilizes two distinct colors of arrows, representing the programming preview effects of the respective characters. If there are programming errors or the correct collaboration is not triggered in the corresponding programming sequence, the system will provide text and visual prompts (Text feedback refers to using text prompts on the screen to guide users on what types of tangible programming blocks to add. Visual feedback refers to using visual elements, such as arrows or crosses, near the corresponding characters to provide suggestive cues shown in Fig. 3).

In the execution stage, users can witness the collaborative progress of the two target characters as they navigate the maze based on the program they created during the programming stage (Fig. 4).

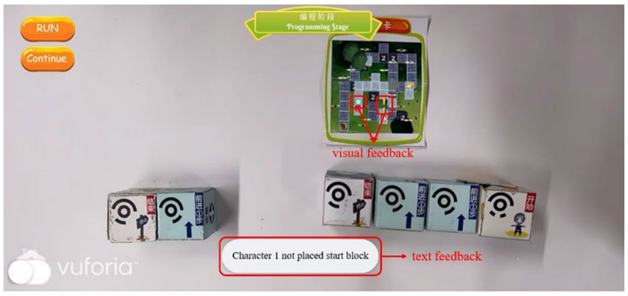


Fig. 3 The feedback in CoAR-Maze

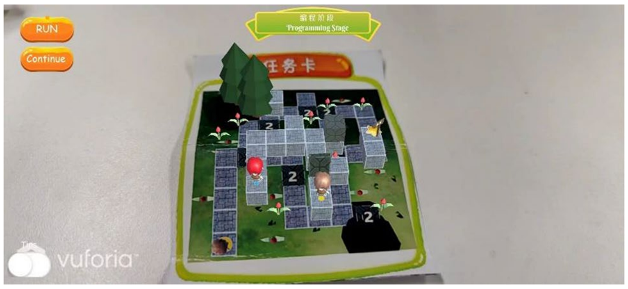


Fig. 4 The execution stage

4.5 Design of collaborative mechanism

The CoAR-Maze application was developed by Unity 3D, integrating the Qualcomm Vuforia platform to achieve AR effects. Location map serves as AR markers, allowing the application to scan and overlay 3D maze task. Users can select specific levels to generate different maze. The application consists of three stages: initialization stage, programming stage, and execution stage.

In the initialization stage, users select a level to start the game. Once a specific level is entered, the system recognizes the location map within the visible range and generates the 3D maze environment with digital characters.

In the programming stage, two users collaborate and share a set of tangible programming blocks, programming simultaneously on the left and right sides of the recognized area. As users create correct programmings.

CoAR-Maze introduces two game characters in the co-player mode and incorporates specific collaborative elements, as shown in Fig. 5. During the programming stage, both users collaborate and utilize a shared set of tangible programming blocks. However, they need to input separate code segments to control their individual game characters, promoting interactive collaboration throughout the programming process. Computer vision technology is leveraged in

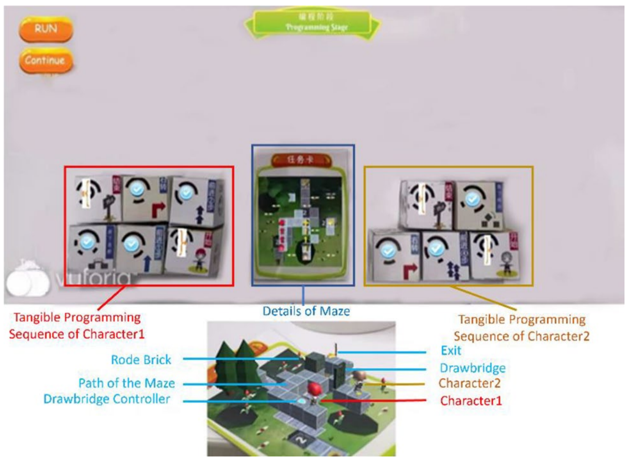


Fig. 5 Collaborative task of CoAR-Maze system

this system to divide the recognition area into distinct left and right halves. This division enables the simultaneous input of two programs. The program displayed in the left area governs the actions of the first character (Character 1), whereas the program displayed in the right area governs the actions of the second character (Character 2).

Each user controls one game character and they start from different starting points to reach the same destination. Users not only need to set the forward path for their own controlled character but also assist their teammate in overcoming obstacles along their teammate's path to ensure that both characters can reach the destination simultaneously. The programming input process adopts a left–right partition recognition strategy, and the design of programming blocks includes the addition of a wait block. The specific design rules are as follows (Note: "Character 1" and "Character 2" mentioned below can be interchangeable):

- Drawbridge and Button: during the progression of Character 1, there may be encounters with drawbridges that need to be lowered in order to proceed. However, the button to lower the drawbridge is placed along the forward path of Character 2. Therefore, it is necessary for Character 2 to step on the drawbridge button in order for Character 1 to continue moving forward.
- Bricks and Trenches: during the progression of Character 1, there may be encounters with trenches that can be filled by pushing bricks. Once the trench is filled, Character 1 can continue moving forward. However, the bricks are positioned along the forward path of Character 2. Therefore, it is only when Character 2 pushes the bricks that Character 1 can proceed further.
- Wait Block: if Character 1 has reached the location of a bridge or trench, but Character 2 has not yet pressed the bridge button or pushed the bricks, Character 1 needs to use the wait block in the programming sequence. This allows Character 1 to wait for assistance from Character 2 in overcoming the obstacles before proceeding.

5 User study

5.1 Study overview

The experiment invited 16 children aged 8–10, with each pair consisting of two individuals. In order to investigate the differences between individual learning and collaborative learning, CoAR-Maze included both co-player mode and single-player mode. In the CoAR-Maze system, co-player mode allows two participants each control a separate avatar and collaborate to complete maze tasks together. On the other hand, single-player mode only enables a single participant to control an avatar to complete the maze tasks individually. After mastering the use of the system, the participants were required to collaboratively complete two co-player programming tasks before independently completing a single-player programming task. During the data analysis phase, a multidimensional set of experimental data, including questionnaires, video analysis and interviews, was employed to explore the following research questions (RQ):

- RQ1: What are the programming learning effectiveness, usability and appeal of CoAR-Maze for children?
- RQ2: What types of interaction form happened during the collaboration?
- RQ3: How does the collaborative programming compared with individual programming in CoAR-Maze?

5.2 Experimental environment

The experiment was conducted on March 5th and 6th, 2022, in the conference room on the third floor of the Building at the Institute of Automation, Chinese Academy of Sciences. In addition to the necessary tables, chairs, and two researchers, the room was equipped with two video cameras, a mobile phone with the CoAR-Maze system and its device holder, a location map, and several physical programming blocks, as shown in Fig. 6. One researcher served as the experimenter and was responsible for programming instruction, experimental documentation, questionnaire interviews, and other experimental procedures. The other researcher assisted in preparing the experimental equipment and operating the video recording devices. The two video cameras recorded the participants' body movements, facial expressions, and collaborative interactions from the front and left-front angles, respectively, during the experimental phases.

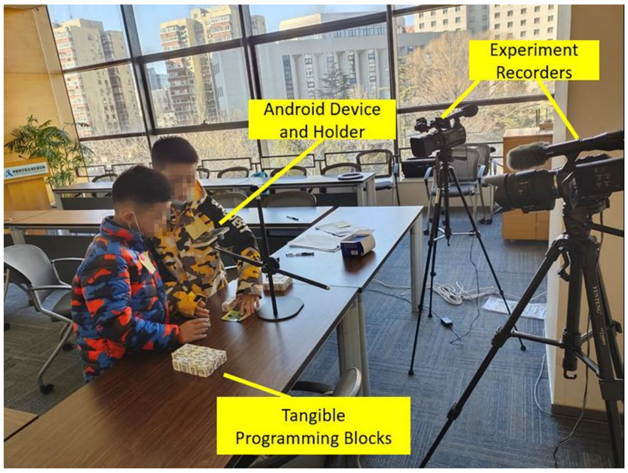


Fig. 6 Experimental environment

5.3 Participants

A total of 16 participants (5 boys and 11 girls) with an average age of 9.1 years took part in this experiment. The children were paired into 8 groups, with 2 children in each group. Among the participants, 12 had prior experience with graphic programming (such as Scratch) or traditional programming languages (such as Python, C++), while 4 had no programming experience. The majority of the children had collaborating experience, except for one child. Demographics information was collected through online shared documents before the study, participants are unfamiliar with each other and select their team members through free team formation.

Prior to the experiment, the parents of the participants were given sufficient time to read and sign the informed consent form and the COVID-19 prevention commitment letter. The experimental process strictly protected the safety and privacy rights of the participants by assigning each child a unique experimental code instead of using their real names.

5.4 Procedure and task design

The participants are unfamiliar with each other and select their team members through free team formation. The experimental process took approximately 60 min, and the specific procedure was as follows:

(1) Experimental instruction (7 min): researchers guided the two participants in learning how to use CoAR-Maze. They introduced the functions of each programming block, the various functional buttons of the user interface, the cooperative mechanism of the dual-player programming tasks, and the operation methods for the single-player programming tasks.

(2) Practice tasks (8 min): researchers guided the two participants in completing a programming task as practice. During this period, they were allowed to ask the researchers any questions to ensure a thorough understanding of how to use CoAR-Maze.

(3) Pre-test questionnaire (6 min): each participant was required to complete a questionnaire. The questionnaire also included two programming ability test questions, which presented two maze maps and their corresponding programming sequences. The participants were asked to determine whether "the target character can reach the endpoint from the starting point according to the given programming sequence." If not possible, they needed to identify the errors in the program. After completing the pre-test questionnaire, the participants did not receive the correct answers or any additional instruction. The post-test questionnaire included the same test questions to validate the effectiveness of CoAR-Maze in children's programming learning through a before-and-after comparison.

(4) Formal task (25 min): two participants collaborated to complete two programming tasks, followed by each participant completing an individual task. During this process, two video cameras captured the children's body movements, facial expressions, and eye interactions.

(5) Post-test questionnaire (7 min): each participant was required to complete a questionnaire to collect their subjective opinions on CoAR-Maze and their experience with collaborative programming. Additionally, their programming abilities were reassessed by the same questionnaire.

(6) Interview (7 min): researchers communicated with the participants to collect further insights about their collaborative programming experience compared with individual programming, their preferences for graphical programming, traditional programming, and tangible programming, among other aspects.

5.5 Measures and data collection

In order to answer our RQs, this study used a mixed methods approach. Both qualitative and quantitative data were collected for analysis.

5.5.1 Questionnaires

We used questionnaires to measure the programming learning effectiveness, usability and appeal of CoAR-Maze for children (RQ1, which are divided into a pre-test questionnaire and a post-test questionnaire. The pre-test questionnaires consists of two parts. The first part consists of pre-test questions which assess the participants' existing experiences

Table 1 Pre-test questionnaire

Question number	Questions
Pre-test Q1	Have you collaborated with children in learning or playing before?
Pre-test Q2	Do you enjoy playing with children?
Pre-test Q3	Have you played similar games or learned children's programming before?
Pre-test Q4	Have you mastered the programming content and methods taught during the instructional phase?
Pre-test Q5	Do you think you can complete the programming tasks? Do you have confidence in yourself?

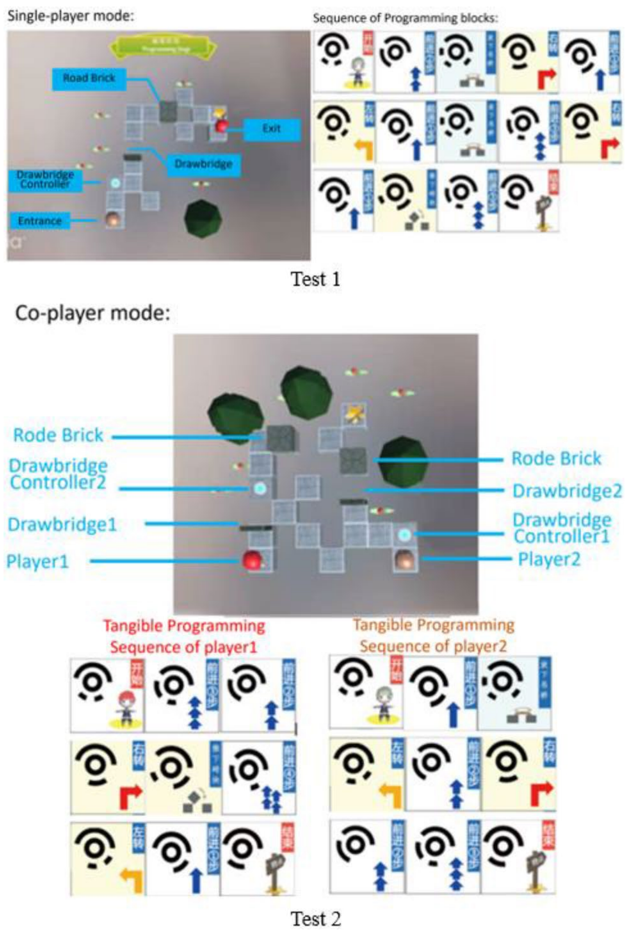


Fig. 7 Programming ability test

and subjective attitudes towards collaborative learning and programming learning, as shown in Table 1. The second part is the programming ability test (Fig. 7). The first question provided a map of a single-player mode and the corresponding programming sequence, while the second question provided a map for co-player mode. Users were asked to determine whether the target character could reach the endpoint based on the given programming sequence. If not, they were required to identify the errors in the program. Users did not receive correct answers or additional instruction after completing the pre-test questionnaire.

The post-test questionnaire is completed by the participants after completing the formal experimental tasks. It includes two parts: The first part consists of post-test questions which collect participants' experiences during the experiment, their evaluations of the programming system, and their attitudes towards cooperative programming and individual programming, as shown in Table 2. The second part is a programming ability test. The same test questions were included in the post-test questionnaire to examine whether there was an improvement in programming abilities after using CoAR-Maze.

5.5.2 Interview

The interview includes five questions as shown in Table 3. The researchers selectively ask these questions based on the users' pre-test and post-test questionnaires, as well as their performance during the experiment, in order to triangulate the quantitative data for RQ1 and compare the collaborative

Table 2 Post-test questionnaire

Question number	Questions
Post-test Q1	Are you satisfied with your partner?
Post-test Q2	Can using CoAR-Maze improve your cooperation skills and willingness to cooperate?
Post-test Q3	Can CoAR-Maze help you learn programming knowledge and increase your motivation to learn programming?
Post-test Q4	Is using CoAR-Maze for programming easy?
Post-test Q5	Are the prompts provided by the system helpful for programming? Can they assist in learning programming?
Post-test Q6	Is the real-time feedback from the system useful? Does it help you better understand programming?
Post-test Q7	Do the feedback and prompts during the programming process help with understanding programming and reducing difficulties?

Table 3 Interviews

Question number	Questions
Interview Q1	What do you think is the best aspect and the worst aspect of this game?
Interview Q2	In comparison to the single-player version, what are the advantages and disadvantages of the co-player mode? Do you prefer co-player or single-player mode? Why?
Interview Q3	What aspects of the co-player mode of the game do you think need improvement?
Interview Q4	Compared to the programming you learned previously, which one do you prefer? Why?
Interview Q5	What knowledge do you think you can gain from using CoAR-Maze? (Answer yes for post-test Q4) Why do you believe it doesn't help with learning programming? (Answer no for post-test Q4)
Interview Q6	Inquire about any exceptional performances observed during the children's experiments (optional)

and individual programming (RQ2). For example, if a user has previous experience with other programming languages, the interview may inquire about the type of programming they have learned and which type they prefer compared to CoAR-Maze. The answers to the questionnaire and interview questions are summarized and quantified, and further processing is performed on any outlier data, laying the foundation for result analysis.

5.5.3 Video analysis

In order to investigate what types of interaction form happened during the collaboration (RQ2) and compare the difference between the collaborative and individual programming (RQ3), we reviewed and qualitatively coded videos from three pairs, including collaborative session and individual session. Two researchers performed open coding to analyze and identify various forms of interaction. To further refine and organize the codes, a collaborative effort between two researchers was employed using a constant comparative grounded theory-inspired method (Muller 2014) to cluster the open codes. As a result, several significant categories emerged, forming the basis for our initial codebook of forms of interaction during the collaborative and individual programming. Based on the initial codebook, two researchers conducted the coding process for all videos. Each video was coded based on the presence of various forms of interaction (see Sect. 5.2). The inter-rater agreement (kappa coefficient) between the two experimenters was determined to be 0.85.

6 Results

6.1 What are the programming learning effectiveness, usability and appeal of CoAR-Maze for children?

By analyzing the post-test questionnaire (Q3), interview responses (Q5), and programming ability test results of the children, the effectiveness of CoAR-Maze in programming learning can be validated. Q3 in the post-test questionnaire asks whether CoAR-Maze can help in learning programming, while interview question Q5 explores what programming knowledge the children acquired or why CoAR-Maze did not offer assistance. Analyzing the responses to Q3 in the post-test questionnaire, it was found that 14 participants believed that CoAR-Maze could improve their programming skills, while 2 participants did not answer the question. Based on the responses to interview question Q5, children who believed that CoAR-Maze could enhance their programming skills mentioned learning concepts such as programming sequences, program completeness, loop control, parameter variables (e.g., step count, loop iterations), and conditional branching. The two participants who did not answer Q3 stated that they lacked prior programming learning experience, so they were unfamiliar with certain concepts and terminology. However, they expressed that the game sparked their interest in programming. In the pre-test and post-test, two programming ability test questions were given (as shown in Fig. 7), and each question contained two errors in the given program. The children's responses to the test questions are shown in Fig. 8. For the first question, which was a single-player level, the average number of errors found by the children before the test was 0.6875, and after

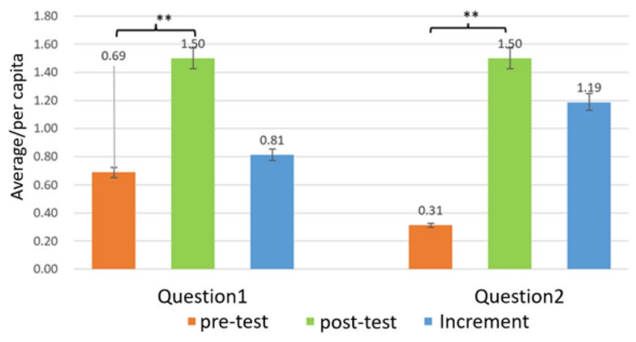


Fig. 8 Results of programming learning test

the test, they found an average of 1.5 errors, indicating an improvement of 0.8125. In the second question, which was a two-player level, the average number of errors found by the children before the test was 0.3125, and after the test, they found an average of 1.5 errors, indicating an improvement of 1.1875. Using the pre-test and post-test as variables, a Wilcoxon test was conducted on the ability test data, revealing a significant difference in the children's test results before and after the experiment (P1 = 0.004, P2 = 0.002). This suggests that the CoAR-Maze system can effectively facilitate children's learning of programming knowledge.

The usability of CoAR-Maze was evaluated through the analysis of participant responses to questions related to system usability, as shown in Fig. 9. After the programming instruction, the children were asked about their mastery of the taught content (pre-test Q4). Two participants indicated partial mastery, while 14 participants reported complete mastery. Based on experimental observations and the responses to pre-test Q5, it was observed that all participants successfully completed the instructional levels and expressed confidence in independently completing programming tasks. These findings suggest that CoAR-Maze has a low learning curve for children, as they were able to quickly grasp the learning methods through simple instruction and displayed a high level of self-efficacy.

The post-test questionnaire (Q4) asked the child participants about the ease of use of the system. 13 participants selected "yes", indicating that the system was easy to use, while 3 participants chose "no". In response to post-test questionnaire Q5, which asked about the helpfulness of system prompts in programming learning, the majority of participants (N = 15) selected "yes", with only one participant choosing "no". Furthermore, all participants unanimously agreed that the real-time feedback provided by the system (post-test Q6) was useful. In their responses to post-test questionnaire Q7, all participants acknowledged that the feedback and prompts during the programming process

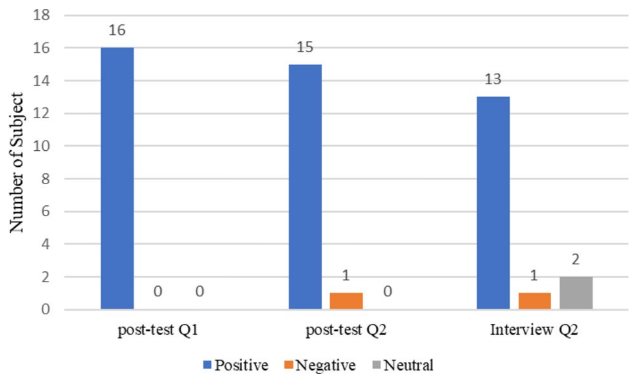


Fig. 9 Results of usability

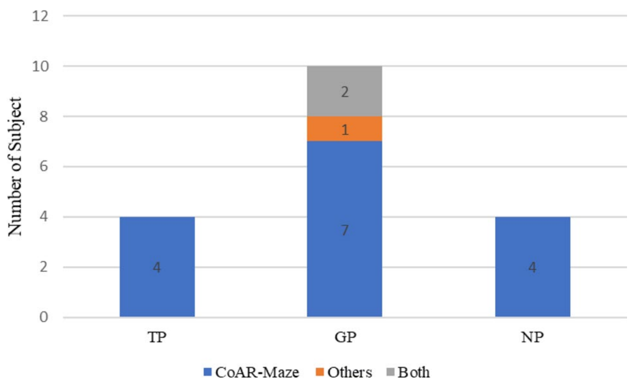


Fig. 10 Results of user preference. TP represents participants who have learned traditional programming, GP represents participants who have learned graphical programming, and NP represents participants who have not learned programming

helped them understand programming and reduced programming difficulties.

In interview question Q4 (as shown in Fig. 10), all children were asked about their preferences for different programming tools. The results are as follows:

(1) All 4 children who had experience with traditional programming expressed a preference for CoAR-Maze compared to traditional programming languages. They provided the following reasons: the tangible components were more engaging, easier to operate, and more enjoyable; there was no need to constantly stare at a screen, which was less straining on the eyes; they didn't have to use a mouse or keyboard; and the programming blocks could be easily dragged and their functions were easy to understand. They mentioned that when they learned programming previously, they found using a keyboard difficult, and many of the codes were unreadable and difficult to comprehend. However, when using the programming blocks of CoAR-Maze, they quickly understood concepts such as loops and how to use them.

(2) In of the 10 children who had experience with graphical programming, 7 expressed a preference for CoAR-Maze, 1 preferred graphical programming, and 2 liked both. The children who preferred CoAR-Maze mentioned that it allowed them to develop their hands-on skills and imagination, they could collaborate with their friends, the combination of paper and AR technology was "magical" and had a futuristic feel to it, and the tangible programming experience without using a mouse or keyboard yielded better results. They also highlighted that it was more eye-friendly. The child who preferred graphical programming mentioned that Scratch's program recognition on the computer was

more accurate, while CoAR-Maze sometimes had errors due to occlusion. The two children who liked both systems recognized the advantages and disadvantages of each. For example, graphical programming had faster and more stable program recognition with fewer errors, but it didn't support playing with another friend. On the other hand, CoAR-Maze allowed them to play with other children, fostering their collaborative abilities and awareness. They also found it interesting to use the recognition card and observe characters and mazes from various angles.

(3) All 4 children with no prior programming experience expressed a strong liking for CoAR-Maze. They stated that the system was easy to learn, even without prior programming experience, and they could understand all the programming operations and concepts. They mentioned that this learning experience sparked a great interest in programming for them. They appreciated the textual and visual prompts during programming, as they guided them to identify their own mistakes. One child even proactively asked if they could download the game from an app store to continue playing at home. Another child expressed a desire for this type of course to be offered at school because they felt there was still much more to learn from this system, and they found the AR effects fascinating.

Based on the data analysis, it can be observed that the group of children showed a high acceptance of tangible programming tools, especially among those who had experience with traditional programming languages or had no prior programming experience. Among all the programming tools, CoAR-Maze was the most preferred choice among the participants (15 children), while one child expressed a preference for graphical programming due to its faster program recognition speed. Through the interviews, the main advantages of CoAR-Maze were identified as being easy to understand, user-friendly, highly engaging with high user retention, supportive of collaboration, and to some extent, protective of visual health.

6.2 What types of interaction form happened during the collaboration?

The main behavior categories included linguistic interaction, eye contact, body language, ask for help, offer help and dispute.

- Linguistic communication: refers to the exchange of spoken words between two participants to negotiate cooperative abilities and action plans.
- Eye contact: refers to the cooperative action between two participants in the collaborative programming process when they make eye contact without verbal communication.
- Body language: refers to the cooperative behavior between two participants through actions such as handing or passing programming blocks, or using physical gestures.
- Ask for help: asking the partner for assistance or seeking advice.
- Offer help: offering assistance to the partner, such as correcting programming errors, answering their questions, or handing programming blocks.
- Dispute: refers to negative interactive behaviors such as arguing, having differences of opinion, or becoming angry.

Frequency of interaction which were quantified by using experimental records and video recordings data about the children's collaborative programming process is shown in Fig. 11. Among the 8 groups of participants, on average each group engaged in 11.25 instances of verbal communication, 1.63 instances of eye contact, 1.38 instances of body language communication, sought help from their partner 3.88 times, provided help to their partner 6.75 times, and experienced disputes 0.88 times. Experimental records revealed that there was a favourable atmosphere of collaboration among the 6 groups, and members communicated actively with each other. Particularly in the second collaborative programming task, the groups had a certain collaboration foundation and collaborated very well. However, conflicts arose between subjects in the second and eighth groups during collaboration. The fourth subject in the second group was very enthusiastic about collaborating but provided incorrect guidance, while the 2 subjects in the eighth group were the youngest children and did not have sufficient understanding of the system, resulting in differences and complaints

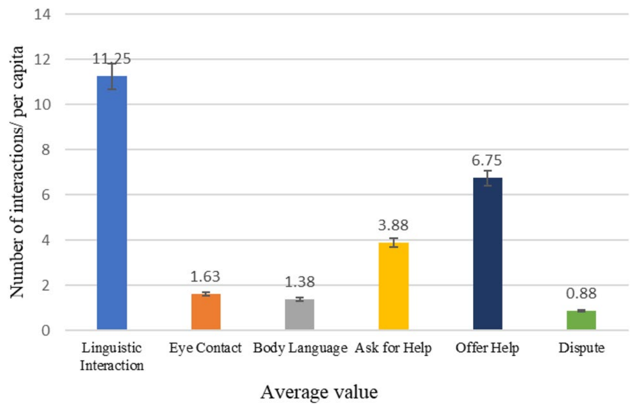


Fig. 11 The comparison of single task and collaborative task

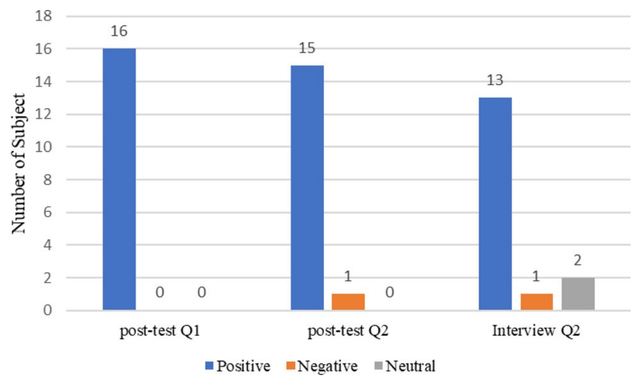


Fig. 12 Results of collaborative learning

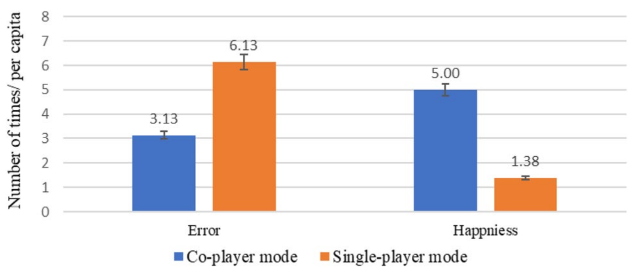


Fig. 13 The comparison of Co-player mode and Single-player mode

during precise coordination. Despite conflicts arising during collaboration, they all expressed that they still enjoyed collaborating because it allowed them to learn more knowledge, therefore, they were willing to continue collaborating.

6.3 How does the collaborative programming compared with independent programming?

The responses of children regarding collaborative learning-related questions are shown in Fig. 12. After completing all the programming tasks, the children were asked about their satisfaction with their partners (post-test Q1). Although some children expressed areas for improvement in their teammates, all participants (N = 16) indicated satisfaction with their partners. When asked whether CoAR-Maze improved their collaboration skills and willingness to collaborate (post-test Q2), the majority of children (N = 15) believed that it did, with only one child expressing the opposite opinion. Analyzing the responses to interview question Q2, it was found that the participant who did not believe in the system's ability to improve collaboration skills was the oldest fifth-grade child (Participant 14). They felt that the dual-player tasks were too simple and did not require collaboration. When asked about their preference between the two versions, 13 participants expressed a preference for the co-player mode, one participant preferred the single-player mode, and two participants liked both mode. The child who preferred the single-player mode was also the one who believed that the system did not enhance collaboration skills (Participant 14). Based on the above data, it can be inferred that CoAR-Maze supports and benefits children's collaborative learning, and the design of the collaborative mechanism is also favored by child users.

By comparing children's programming performance and emotional responses in collaborative programming assignments with those in single programming tasks (Fig. 13), we can explore the distinctions between cooperative and individual learning. Participants made an average of 3.13 errors per person in two collaborative programming tasks and 6.13 errors per person in one individual programming task, suggesting a greater incidence of errors during individual programming. A Wilcoxon test using collaborative task/solo task as a variable demonstrate a significant difference in programming performance between the two groups (P = 0.017), with children exhibiting better performance in collaborative programming. By quantifying the instances of happiness during programming, it was found that the average number of happy moments per person in two collaborative programming tasks was 5.00, while the average number of happy moments per person in one individual programming task was 1.38, indicating that children tend to experience more positive emotions when working collaboratively on programming tasks. A Wilcoxon test with collaborative task/single task as a variable reveal a significant difference in emotional states between the two groups (P = 0.028), with children remaining better emotional states during collaborative programming.

In the interview phase, researchers asked participants about their preferences and reasons for choosing either co-player mode or single-player mode. Among the 16 participants, 13 expressed a preference for co-player mode, 1 preferred single-player mode, and 2 said they liked both methods. Children who favored co-player mode stated that it could improve their collaboration skills and make tasks easier, help them make friends during the learning process, allow them to receive assistance in pointing out mistakes, remind them of forgotten content, and encourage patience while waiting for their teammates. The child who preferred solo programming was the oldest participant and felt that the task was simple enough to complete alone, and that working with a partner would involve waiting for the teammate.

7 Discussions and limitations

7.1 Tangible and collaborative programming are effective for helping children learn programming

Previous literature suggest tangible programming is more appropriate for children's cognitive development, as it can successfully integrate distributed cognition theory and ease cognitive demands on children (Horn et al. 2009; Sapounidis et al. 2019). Referring to Pair Programming (Thapaliya 2020) and collaboration frame of Mobile Stories (Fails et al. 2010), We propose a tangible programming system which contain co-player mode and single-player mode allowing children to control characters in game by arranging tangible programming blocks. Our research shows that both co-player mode and single-player mode are useful for helping children learn programming. After comparing the two modes, most children favoured the co-player mode, this mode can make tasks simpler and improve collaboration skills.

7.2 Limitation

There are some limitations in the system design as well. Firstly, since the system collects information through the camera, there can be instances where the camera's view is obstructed during children's interactions, which affects the interaction between children and the system. For example, when a user places a new programming block into the programming queue, the camera may capture the information slowly due to obstruction caused by the arm or other reasons. Future work should explore how to better support the information interaction between children and the system. Secondly, there are limited collaboration methods among users. Future work should explore how to better support collaboration among children. Perhaps involving children in the design process and providing more functional programming blocks could be beneficial. Thirdly, the rigor of experimental design needs improvement, especially in questionnaire design. In the post-test questionnaire, the placement of questions and options should be changed to avoid better test scores due to less time spent reading the questions. Further research is needed to investigate how children use CoAR-Maze in more relaxed or open-ended environments. The research should delve into the entire learning process and validate the effectiveness of collaborative learning. During the formal experiment, the subjects all completed the collaborative task before completing the single task, the source of users' programming ability enhancement was unclear, the stage ability determination of the collaborative task and the single task was missing, and the experimental process of completing the single task before completing the collaborative task was not set.

8 Conclusion and feature work

We propose a tangible programming system, CoAR-Maze, of which collaboration mechanism is designed based on collaborative learning theory, focusing on inspiring children's proactive collaboration. Through analyzing quantified video data and combining experimental observations, questionnaire interviews, ability tests, and other multidimensional data in the CoAR-Maze user experiment, it is found that the CoAR-Maze system, based on physical interaction and intention understanding, is easy to use for child users, and AR technology makes the system more interesting and appealing. At the same time, the system significantly assists children's programming learning and collaborative learning. The system's collaboration mechanism can stimulate children's active collaboration during the learning process by providing a good interactive platform. Compared to graphical programming and traditional programming, the CoAR-Maze system is highly recognized among child users, particularly the version that supports two-person collaboration.

In the future, we will improve the CoAR-Maze system based on current work. Our next step involves two aspects: software interaction design and hardware system design. Regarding software design, we aim to further optimize the interaction mechanism, enhance the stability of AR technology, reduce the chance of users mistakenly blocking recognition cards during operation, and improve the naturalness, stability, and fun factor of human–computer interaction. On the hardware design front, we plan to add function blocks, conditional branching blocks, and algorithm blocks to support more programming logic and more complex collaboration mechanisms. We intend to enrich the AR effects in the programming environment, making the maze environment more vivid and interesting, and increasing users' interest in programming and immersion during the learning process. In the experimental setup, a contrastive experiment was added to the formal experimental phase. Different experimental groups were divided into Group A and Group B. Group A completed the collaborative task first and then the individual task, while Group B completed the individual task first and then the collaborative task. A programming proficiency assessment module was added in the transition between the collaborative and individual tasks, enhancing the persuasiveness of the conclusions regarding the effectiveness of the system.

Funding This research is supported by the National Natural Science Foundation of China under Grant No. 61872363 and 61672507, the Natural Science Foundation of Beijing & Key project of Science and Technology Plan of Beijing Municipal Education Commission under Grant No. 21JD0044, the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA27000000.

Data availability All data generated or analysed during this study are included in the manuscript.

Declarations

Conflict of interest There is no conflict of interest among the authors.

References

Begel, A., Nagappan, N.: Pair programming: what's in it for me? In Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement, pp. 120–128 (2008)

Bers, M., Horn, M.: Tangible Programming in Early Childhood: Revisiting Developmental Assumptions through New Technologies (2009)

Burke, Q., Kafai, Y.B.: Programming & storytelling: opportunities for learning about coding & composition[C]//International Conference on Interaction Design & Children. ACM, 348–351 (2010)

Cabrera, L., Maloney, J. H., Weintrop, D.: Programs in the Palm of your Hand: How Live Programming Shapes Children's Interactions with Physical Computing Devices[C]// the Interaction Design and Children. 227–236 (2019)

Dai, X.: Group collaborative learning to improve students' Scratch programming ability [J]. Navigation in arts and Sciences (late), p 97 (2020)

Deng, X., Wang, D., Qiao, J.: TLogic: A Tangible Programming Tool to Help Children Solve Problems, pp. 255–262. Springer, Switzerland (2018)

Deng, X., Wang, D., Jin, Q.: CoProStory: A Tangible Programming Tool for Children's Collaboration[C]// International Conference on Computer Supported Collaborative Learning (CSCL2019). 25–31 (2019)

Dillenbourg, P.: What do you mean by collaborative learning? pp. 1–19 (1999)

Dillenbourg, P., Over-Scripting, C. S. C. L.: The risks of blending collaborative learning with instructional design, Kirschner, PA (ed.), Three worlds of CSCL. Can we support CSCL, 61–91 (2002)

Etel, E., Slaughter, V.: Theory of mind and peer cooperation in two play contexts [J]. J. Appl. Dev. Psychol. **60**, 87–95 (2019)

Fails, J. A., Druin, A., Guha, M. L.: Mobile collaboration: collaboratively reading and creating children's stories on mobile devices. In Proceedings of the 9th International Conference on Interaction Design and Children, pp. 20–29 (2010)

Farrokhnia, M., Pijeira-Díaz, H.J., Noroozi, O., et al.: Computer-supported collaborative concept mapping: The effects of different instructional designs on conceptual understanding and knowledge co-construction [J]. Comput. Educ. **142**, 30–45 (2019)

Gennari, R., Melonio, A., Torello, S.: Gamified probes for cooperative learning: a case study [J]. Multimed. Tools Appl. **76**(4), 4925–4949 (2017)

Horn, M.S., Jacob, R.J.K.: Tangible programming in the classroom: a practical approach[C]//In CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06). Association for Computing Machinery, pp. 869–874. New York, NY, USA (2023). https://doi.org/10.1145/1125451.1125621

Horn, M., Solovey, E., Crouser, R., et al.: Comparing the use of tangible and graphical programming languages for informal science education[C]//Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4–9. 975–984 (2009)

Inkpen, K., Ho-Ching, W. L., Kuederle, O., Scott, S. D., and Shoemaker, G. B.: This is fun! We're all best friends and we're all playing!: Supporting Children's Synchronous Collaboration (1999)

Jin Q, Wang D, Deng X, et al.: AR-Maze: a tangible programming tool for children based on AR technology[C]//Proceedings of the 17th ACM Conference on Interaction Design and Children, 611–616 (2018)

John, M., Mitchel, et al.: The scratch programming language and environment [J]. ACM Transact. Comput. Educ. **10**(4), 1–15 (2010)

Kelleher, C., Pausch, R.: Lowering the barriers to programming: a survey of programming environments and languages for novice programmers [J]. ACM Comput. Surv. **37**(2), 83–137 (2005)

Melcer, E. F., Isbister, K.: Bots & (Main) frames: exploring the impact of tangible blocks and collaborative play in an educational programming game. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 1–14 (2018)

Muller, M.: Curiosity, creativity, and surprise as analytic tools: Grounded theory method. In: Ways of Knowing in HCI, pp. 25–48. Springer, New York (2014)

Papadakis, S.: Robots and robotics kits for early childhood and first school age [J]. Int. J. Interact. Mobile Technol. (iJIM) **14**(18), 34–56 (2020)

Rahman, M. M., Sharker, M., Paudel, R.: Impact of Infusing Interactive and Collaborative Learning in Teaching Introductory Programming in a Dynamic Class[C]// SIGCSE '20: The 51st ACM Technical Symposium on Computer Science Education. ACM, 1315–1315 (2020)

Sapounidis, T., Demetriadis, S., Papadopoulos, P.M., Stamovlasis, D.: Tangible and graphical programming with experienced children: a mixed methods analysis. Int. J.Child Computer Interact. **19**, 67–78 (2019)

Scott, S.D., Mandryk, R.L., Inkpen, K.M.: Understanding children's collaborative interactions in shared environments. J. Comput. Assist. Learn. **19**(2), 220–228 (2003)

Shaer, O., Hornecker, E.: Tangible User Interfaces[J]. Now Publishers Inc.PUB4850Hanover, MA, USA, 1–137 (2010)

Slavin, R.E.: Cooperative learning [J]. Int. Encycl. Educ. **50**(2), 315–342 (1980)

Stahl, G., Koschmann, T., Suthers, D.: Computer-supported collaborative learning: An historical perspective [Electronic Version]. http://lilt.ics.hawaii.edu/lilt/papers/2006/CSCL_American_English. Accessed 07 June 2007. pp. 409–426 (2006)

Thapaliya, A.: Evaluation of brain activity while pair programming. In 2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). IEEE, pp. 104–106 (2020)

Thieme, A., Morrison, C., Villar, N., et al.: Enabling collaboration in learning computer programing inclusive of children with vision impairments [C]// the 2017 Conference. 739-752 (2017)

TopCode: Tangible Object Placement Codes. http://users.eecs.northwestern.edu/~mhorn/topcodes/

Wang, D., Zhang, C., Wang, H.: T-Maze: a tangible programming tool for children [C]. Proceedings of the 10th International Conference on Interaction Design and Children, pp. 127–135 (2011)

Wang, X., Xing, Q., Jin, Q., et al.: "Be a lighting programmer": supporting children collaborative learning through tangible programming system [J]. Int. J. Hum. Comput. Int. 1–19 (2023). https://doi.org/10.1080/10447318.2022.2163783

Xing, Q., Wang, D., Zhao, Y., and Wang, X.: Clas-Maze: An Edutainment Tool Combining Tangible Programming and Living Knowledge. In International Conference on Entertainment Computing. Springer, Cham, pp. 353–368 (2020)

Yashiro, T., Harada, Y., Mukaiyama, K.: Plugramming: A tangible programming tool for children's collaborative learning [C]// International Conference on Human-computer Interaction. Springer, Cham, 398–409 (2017)

Yu, J., Roque, R.: A survey of computational kits for young children. In Proceedings of the 17th ACM conference on interaction design and children. pp. 289–299 (2018)

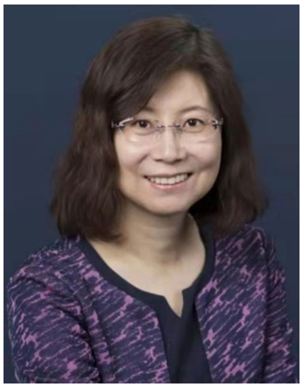
Zhao Y, Feng S, Wang D.: AR-C&P: A Tangible Programming for Children Based Augmented Reality. In The Tenth International Symposium of Chinese CHI (Chinese CHI 2022), October 22–23, 2022, Guangzhou, China and Online, China. ACM, New York, NY, USA, p 18 (2022)

Mingyu Zhang is a master student in artificial intelligence at University of Chinese Academy of Science, China. His research interests include human-computer interaction and machine learning.

Jiaxiang Li received the B.S. degree in Automation from Xi'an University of Technology, Xi'an, China, in 2019. He then worked as a software engineer at the Tenth Institute of Telecommunication, China Information Communication Technologies Group Corporation. He is currently working towards a master's degree in computer science at the Institute of Automation, Chinese Academy of Sciences. His research interests include multi-agent systems and multimodal machine learning.

Yiyan Lin is an undergraduate student in digital media art at Beijing Normal University. Her research interests include visual communication design and animated short film creation.

Qiao (Georgie) Jin is a PhD candidate from Grouplens Research Center at the University of Minnesota - Twin Cities. Her research focuses on using AR/VR/MR to support remote education, collaboration and social connection.



Danli Wang is a researcher and professor at the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. She received her Ph.D. degree in Beihang University and carried out postdoctoral research in Institute of Automation of Chinese Academy of Sciences. Her research interests include human-computer interaction technology, multimodal user interface, user experience artificial intelligence and psychosomatic computation.