AR-Maze: A Tangible Programming Tool for Children Based on AR Technology

Qiao Jin[1, 2]
jinqiao2018@ia.ac.cn

Danli Wang[1]
danli.wang@ia.ac.cn

Xiaozhou Deng[1,2]
dengxiaozhou_rush@163.com

Nan Zheng[1]
nan.zheng@ia.ac.cn

Steve Chiu[3]
chiustev@isu.edu

[1]The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

[2]School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China

[3]Electrical Engineering, Idaho State University Pocatello, ID, U.S.A.

IDC '18, June 19–22, 2018, Trondheim, Norway

ACM ISBN 978-1-4503-5152-2/18/06…$15.00
https://doi.org/10.1145/3202185.3210784

Abstract
Programming is an effective way to foster children's computational thinking. We present AR-Maze, which is a novel tangible programming tool using Augmented Reality (AR) technology for young children. AR-Maze superposes constant feedback on the physical world and maintains a positive, low-cost learning environment. Using this system, children could create their own programs by arranging programming blocks and debug or execute the code with a mobile device. In addition, they will be able to learn fundamental programming concepts, such as parameters, loop logic, debug, etc. We design and implement this system, as well as conduct a preliminary user study and analyze the results, which can guide a better design of AR-Maze. With this work, we intend to help children programming in an interesting and intuitive way.

Author Keywords
Augmented Reality; Programming Languages; Tangible Programming; Children; Edutainment.

ACM Classification Keywords
H.5.1 Artificial, augmented and virtual realities, H.5.2. User interfaces.

Introduction

Programming learning has been proved to be beneficial for children to acquire deep knowledge of computer working principle. Meanwhile its positive impact for children on the development of computational thinking, problem-solving and design strategies is considerable [8]. In particular, computational thinking has been described by Wing as a fundamental skill for everyone, not just for computer scientists [14]. Therefore, an appropriate programming learning tool is essential for children.

Most current programming languages are text-based, and difficult for children to understand without sufficient context. Comparing to static images and texts on a computer screen, Tangible User Interface (TUI) is better at accommodating habits of young children, who prefer to hold toys in hands to operate [5]. Tangible languages have the potential to make it easier for children to be involved in programming. There have been broad researches on tangible programming languages designed for children such as T-Maze [12], TanProRobot 2.0 [13] and Tern [4], which aim at ameliorating the user experience in programming learning. But few of them attempts to apply the mixture of virtual and physical objects on the same position. This paper explores the method to enrich the real-time feedback which can integrate the physical world in an inexpensive and effective way with existing technologies.

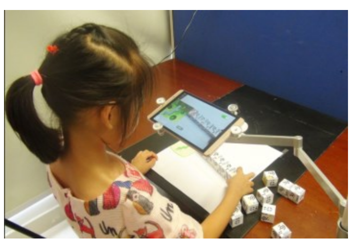

Figure 1: A child was playing with AR-Maze.

Augmented Reality (AR) technology provides new opportunities for education by combining physical language and virtual environment that enables constant feedback. It has been shown that a real object with augmented interaction is superior to an abstraction of system with screen only in many aspects, such as providing a better intuitive interface and more efficient real-time feedback system through tangible interface [6].

In this paper, we propose a new tangible programming tool – AR-Maze (Figure 1). It combines virtual system with physical programming blocks, which helps children program in a physical form and executes the generated code in virtual environments with a coherent view of the operating space. This system makes the programming learning more interesting and intuitive for children. Moreover, AR-Maze is able to provide sorts of real-time operation feedback in both programming and running stages.

Related Work

There are already several attempts to apply AR technology in programming education. LightUp [2] which has an AR-based informational mobile view enables learners to observe the inner process of their program. Tiles that Talk [1] uses different tiles to represent computer syntax, software libraries and hardware, which can visualize the system architecture of project through the camera. Thymio II [7] is an educational robot integrating AR. Students could learn computer concepts with a tablet that displays the event executed on the robot and its visual programming environment. AR Scratch [10] is an AR authoring environment designed for children based on the Scratch, a programming platform allows children to create programs that mix real and virtual spaces. Code Bits [3] is a tangible computational thinking toolkit. Students create programs by using the tangible paper bits and use the AR-based mobile game to process the code. Most of the tools mentioned above assign the

Figure 2: AR-Maze is composed of a) AR-Maze Game, b) Location Map, c) Programming Blocks.

augmented reality effect at the period of program running. However, they fail to provide real-time interactive feedback or physical-based tips during coding period.

We are particularly concerned with providing a tool which is potable, inexpensive and fully feedback engaged to facilitate children's programming learning experiences by AR technology. It can also convey programming concepts and computational thinking to younger children.

Design and Implementation

AR-Maze is a tangible programming tool with AR technology for children. It contains three parts (Figure 2): 1) AR-Maze Game on mobile devices which provides image, textual and audio feedback superimposed on the real scene; 2) Location Map tracking the position of the virtual map during the running stage; 3) Three kinds of Programming Blocks linking physical programming blocks and internal logic of AR-Maze game.

Each level in this system includes programming stage and running stage, which are equivalent to programming and code executions in traditional programming language process. In the programming stage, the player must learn how to program to complete the task, which is constructing a path with programming blocks and navigating this virtual character with this compiled path to reach the destination. There will be a visualized maze map on the right side of the screen to assist task execution. In the running stage, the virtual maze traces the location map shown on the screen. Players could observe the characters' actions and movements as they coded in the programming stage.

AR-Maze Game

This game is implemented in Unity 3D and Qualcomm Vuforia platform [9] that can use a designated image (Location Map) as a marker and display a 3D scene on top of it. Players control the AR scene to provide the image, textual and audio feedback superimposed on the real scene. The debugging process is feasible via the screen as well (Figure 3 and Figure 4). In the process of programming stage, an arrow will appear to indicate the intention (changing orientation or location) of the current program along with putting a new programming block. If the user wants to check whether their code has violated the rule of programming, the tips button will help check the code and render error message on the corresponding programming block. Musical tones and a textual prompt will also be triggered to remind the player to modify the program. In the running stage, the image tips will show the location of the current programming block when the program was executed. Audio and textual feedback will also notify children which programming block is running currently.

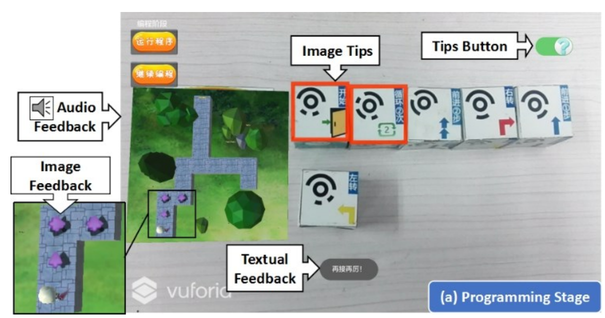

Figure 3: Feedback in the programming stage

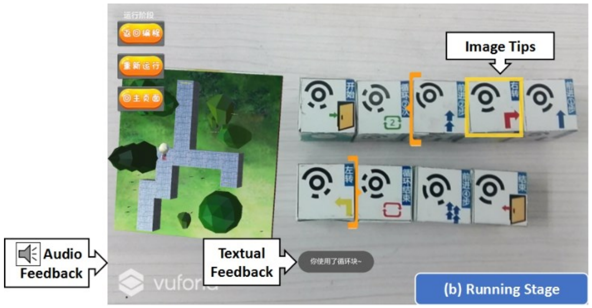

Figure 4: Feedback in the running stage

Figure 5: Using Location Map in AR-Maze.

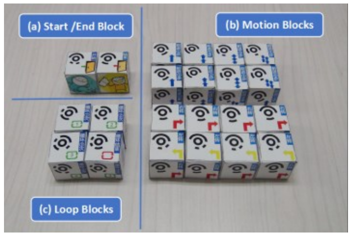

Figure 6: Programming Blocks consist of a) Start /End Block, b) Motion Blocks, c) Loop Blocks.

Location Map

Location Map is a self-made picture card, whose role is to locate the position of the virtual map (Figure 5). AR-Maze game allows for the detection and tracking of location map based on existing 3D models. As the user adjusts the position and angle of the location map, the attached virtual map will be updated accordingly.

Programming Blocks

Programming blocks are made by 3cm wooden brick cube with surface texture. TopCode [11] was applied to build the connection between physical programming blocks and AR-Maze game logic layer. Magnets are arranged onto two opposite sides of programming blocks to help children keep track of correct building direction (the anode must be connected to the cathode). Also, such a small circular installed magnet allows the programming blocks to rotate easily. In order to offer better clarification of the program metaphor, we added pictures and text to indicate the functionality of the block. We make fully use of all the four faces in each block to express different programming semantics, which could optimize the number of required programming blocks and lower experiment budget.

According to the semantics and functions, we divide the blocks into three types: Start/End Block, Motion Blocks, and Loop Blocks (Figure 6). Start/End Block is used to mark the beginning or end of the program. Participants should place it at the beginning or the end of the sequence, where the program content is modeled by blocks in the middle part. Motion Blocks are divided into Turn Blocks and Forward Blocks, which are responsible for controlling the movement of the game character. Loop Blocks represents “for” in traditional programming and make code blocks execute repeatedly.

Others

Taking into consideration that children’s hands are relatively small and they will feel uncomfortable to hold the device during the game, we use a bracket to support the mobile device and provide them an immersive AR programming experiment.

Preliminary User Study

A preliminary user study was carried out to study children’s behaviors in such a game with mixed physical and virtual worlds, the type of most attractive interactions to them, and any major usability faults. 8 children (5 male and 3 female) without previous programming experience aged 5-9 were participated into the study.

During the experiment, they could create their programs with programming blocks. Communications regarding their feelings and any questions are encouraged. Furthermore, a video-recorder was used to keep track of the whole user study process and helped us find usability faults.

The children were required to control their characters to move towards the destination in three levels. All of them completed these tasks with approximately 20 minutes on average. They could even apply loop logic, identify and fix bugs correctly without instructor's assistance. After the initial game introduction, the children appeared to be enjoyable and engaged, showing strong curiosity to this experiment game system. When we introduced the programming blocks, one child began to guess how to use them. For example, he said "there are magnets in the block so they should be placed this way". And another child asked the role's movement principle during this session. When they finished programming, the children indicated that these tools are conducive for them to understand several computer concepts, such as loop logic, parameter, etc. They also claimed that the AR-Maze improved their interests in learning programming.

We also found out that children are more likely to ignore the feedback from plain text. One of the reason is that children are too young to be quite sensitive to plain text and even don't understand their meanings. Another possibility is the visual competition. We also provide the feedback from the pictures. With this kind of feedback arises in the AR map and programming blocks, it will attract more attention.

Specifically, one uncovered issue is the physical arrangement of the technology, like bracket and pad. The height of the bracket need to be adjusted manually according to each child's height in advance. Otherwise, the child will hit the bracket or change the angle of the pad camera easily, and inappropriate poses caused target loss and the child had to pause the game to readjust the components. Another issue is the spatial limitation. The programming blocks and location maps must be arranged within the capture range of the camera. This issue will be magnified exponentially once the complexity of program hits specific threshold that we need bigger physical space and larger interface to deploy the experiment. The possible solution is using AR projection in glasses or making code segment reusable within the whole context.

From these observations, we concluded that the game was useful to support the learning of computer programming concepts for children in a fun and enjoyable way. But spatial limitation as well as physical arrangements should be considered in system design and use cases.

Conclusion and Future Work

In this paper, we describe a programming tool designed for children with AR technology – AR-Maze. Children program with a tangible language, and the program is running in a virtual world. They gained confidence and interest as they learned computer concepts easily from a maze game with AR real-time feedback. The tool is low cost, intuitive and portable. These features allow young children to learn programming concepts easily in normal environments such as home or school.

In the future, we will continue to improve our system. More programming concepts, such as functions and branches will be added. Also, we will provide more accurate and richer AR effect, such as using different location maps to achieve different operating results according to gender differences. Furthermore, the physical arrangement and tracking effect of location maps needs further improvement, and more interaction will be extended in the running stage with AR technology. Finally, physical programming is very suitable for cooperation. In our future research, cooperation elements will be included in the system.

Acknowledgements
This research is supported by the National Key Research and Development Program under Grant No. 2016YFB0401202, and the National Natural Science Foundation of China under Grant No. 61672507, 61272325, 61501463 and 61562063.

References
1. David Bouchard and Steve Daniels. 2015. Tiles that Talk: Tangible Templates for Networked Objects. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '15). ACM, 197-200.
2. Joshua Chan, Tarun Pondicherry, and Paulo Blikstein. 2013. LightUp: an augmented, learning platform for electronics. In *Proceedings of the 12th International Conference on Interaction Design and Children* (IDC '13). ACM, 491-494.
3. Sidhant Goyal, Rohan S. Vijay, Charu Monga, and Pratul Kalita. 2016. Code Bits: An Inexpensive Tangible Computational Thinking Toolkit For K-12 Curriculum. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '16). ACM, 441-447.
4. Michael S. Horn and Robert J. K. Jacob. 2007. Designing tangible programming languages for classroom use. In *Proceedings of the 1st international conference on Tangible and embedded interaction* (TEI '07). ACM, 159-162.
5. Heller S. The meaning of children in culture becomes a focal point for scholars. *The Chronicle of Higher Education*, 1998: A14 -A16.
6. Valentin Heun, Shunichi Kasahara, and Pattie Maes. 2013. Smarter objects: using AR technology to program physical objects and their interactions. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '13). ACM, 2817-2818.
7. Stéphane Magnenat, Morderchai Ben-Ari, Severin Klinger, and Robert W. Sumner. 2015. Enhancing Robot Programming with Visual Feedback and Augmented Reality. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education* (ITiCSE '15). ACM, 153-158.
8. Timothy S. McNerney. 2004. From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal Ubiquitous Comput*. 8, 5 (September 2004), 326-337.
9. Qualcomm Vuforia, http://www.qualcomm.com/Vuforia
10. Iulian Radu and Blair MacIntyre. 2009. Augmented-reality scratch: a children's authoring environment for augmented-reality experiences. In *Proceedings of the 8th International Conference on Interaction Design and Children* (IDC '09). ACM, 210-213.
11. TopCode: Tangible Object Placement Codes. DOI: http://hci.cs.tufts.edu/topcodes/
12. Danli Wang, Cheng Zhang, and Hongan Wang. 2011. T-Maze: a tangible programming tool for children. In *Proceedings of the 10th International Conference on Interaction Design and Children* (IDC '11). ACM, 127-135.
13. Danli Wang, Lan Zhang, Chao Xu, Haichen Hu, and Yunfeng Qi. 2016. A Tangible Embedded Programming System to Convey Event-Handling Concept. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '16). ACM, 133-140.
14. Wing, Jeannette M. "Computational thinking." *Communications of the ACM* 49.3 (2006): 33-35.