ARCat: A Tangible Programming Tool for DFS Algorithm Teaching

Xiaozhou Deng
The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, School of Computer and Control Engineering, University of Chinese Academy of Sciences,
Beijing, China
dengxiaozhou2018@ia.ac.cn

Danli Wang
The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences,
Beijing, China
danli.wang@ia.ac.cn

Qiao Jin
The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, School of Computer and Control Engineering, University of Chinese Academy of Sciences,
Beijing, China
jinqiao2018@ia.ac.cn

Fang Sun
School of Computer and Information Technology, Liaoning Normal University,
Dalian, China
sf8552@126.com

ABSTRACT

In this paper we present ARCat, a tangible programming tool designed to help children learn Depth First Search (DFS) algorithm with augmented reality (AR) technology. With this tool, children could use tangible programming cards to control a search process, rather than control virtual characters directly. With the special design of card semantics and real-time feedback, the

---

IDC '19, June 12–15, 2019, Boise, ID, USA

ACM ISBN 978-1-4503-6690-8/19/06..$15.00
https://doi.org/10.1145/3311927.3325308

KEYWORDS
Tangible User Interfaces; Augmented Reality; Algorithm education; Computational thinking

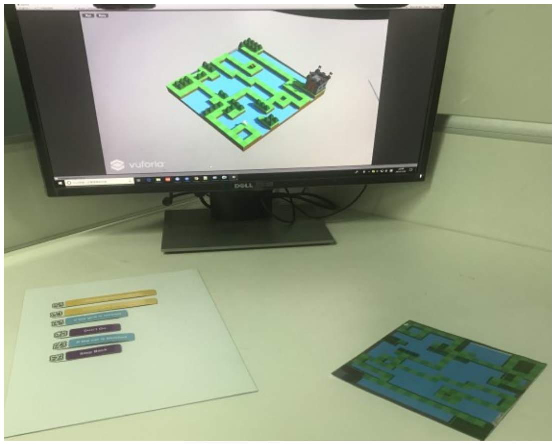

Figure 1: Overview of ARCat.

cognitive load of the learning process had been proved to be affordable to children (ages 8-9) with the result of our preliminary evaluation, which shows the possibility of basic algorithm education for young children with tangible interface.

INTRODUCTION

As programming education shows great efficiency in cultivating computational thinking, which has been described as a fundamental skill for everyone [5], the scope of programming education has been expanded to young children. Meanwhile, algorithm education has a positive effect on logical thinking, general problem formulation and solving skills. With that consideration, it's worth trying to combine programming education with basic algorithm for young children.

However, programming appears to be quite a challenge for young children. With the traditional way of programming, children have difficulties in some basic function implementation. As for algorithm, it seems impossible for most young children to learn easily with rigid syntax and complex data structure. As a method, Tangible User Interface (TUI) has been used to reduce the cognitive load of programming. With tangible cards, children can write program by assembling physical objects without keystroke [1], which is much easier to involve children in programming.

In this paper we present ARCat (see Fig. 1), a tangible programming tool for DFS algorithm teaching. The tool decomposes a DFS problem into search process and control rules to help children learn. Finally, a preliminary user study was conducted.

RELATED WORK

There are many excellent tangible programming tools designed for children, which inspire our work a lot.

Many tangible programming tools use building blocks to program, such as Electronic Block [10] and KIBO [8]. Electronic Block embedded each block with a processor and electronic parts and provide three types of building blocks: sensor blocks, logic blocks and behavior blocks. Since using physical electronic modules usually results in high cost, many tangible tools use computer vision technology as solution. KIBO is a robotics kit, with which children can construct robots with motors, sensors, and craft materials, and program the robots' actions with some wooden blocks. Compared to Electronic Block, KIBO uses computer vision technology to capture the block sequence. With computer vision technology, many programming tools chose to use tiles rather blocks, such as Strawbies. Strawbies [2] is a real-time tangible programming game designed for children age 5-10. It is an iPad app. Children could play with it by constructing physical programs out of wooden tiles in front of an iPad. Recently, many tangible programming tools apply AR technology to provide new visual environment for children. AR Scratch [6] 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. AR-Maze [3] and Code Notes [7] attempt to promote children's computational thinking with low cost materials, enabling children program in a physical way with AR outcome.

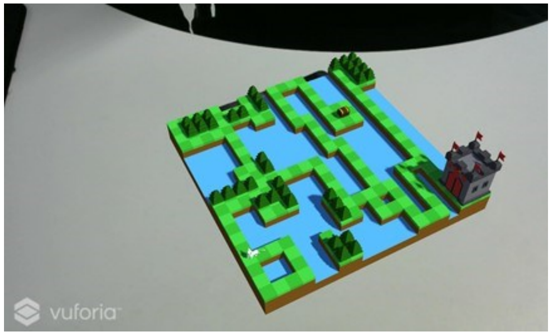

Figure2: Game in AR environment.

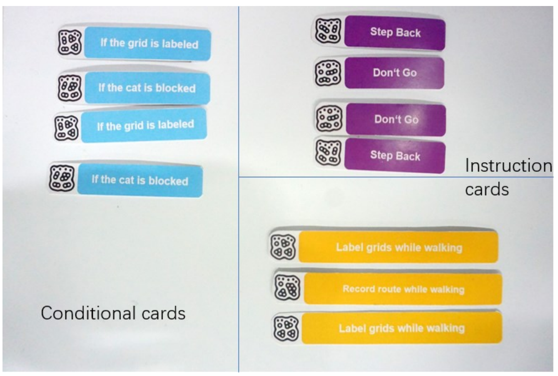

Figure 3: Programming cards (In our user study, we used native language. We provided duplicate cards due to the reason that children may try their own rule combinations, such as *If the grid is labelled -Don't go* and *If the cat is blocked-Don't go* to solve the problem)

Majority of algorithm tools are based on visualization with traditional programming language, so it's worth to try combining tangible interface with algorithm education and find the possibility of teaching algorithm to young children.

Combined many virtues of these works, we created ARCat and explored the efficient way of teaching algorithm.

DESIGN AND IMPLEMENTATION

The implementation of ARCat consists of two parts: a visual game task shown in AR environment and tangible cards. Since our goal is to teach DFS algorithm, we implant a classic search problem to the game task, which could be described as: A cat is searching for a chest in a maze. Since the cat isn't as smart as people, she needs children guiding her to find the treasure chest (see Fig. 2). The visual game is presented in AR environment, which make the visual scene more convenient for observation with the movement of image target. We use Vuforia SDK for implementation [9].

As mentioned in the Introduction, it's too hard for young children to program DFS algorithm without any support. Besides, with no guidance, many children would choose to control the virtual cat mechanically rather than to think out an automatically search algorithm. To guide children to solve this problem with DFS algorithm and make the programming process easier to get start with, we decompose the DFS algorithm and provide children with a basic search pattern, which is presented with animation. The virtual cat with basic search pattern would go to the available adjacent grids with a simple search priority: forward>left>right>back, which is just used to make the movement of the cat seemly rational. Children need to add rules to this basic pattern to help the cat find the chest.

Tangible programming cards are made by paper-based cards and magnetic disks, which help children assemble rules in an iron plate easily. Inspired by related works, we provide different kinds of instruction cards and conditional cards, which are painted with different colors (see Fig. 3). The cards painted in yellow are instructions which could be used directly, while cards in purple need to be used with conditional codes, which are painted in blue. And considered the cognitive load of learning progress, we make the semantics of tangible cards intuitive. The cards are corresponding to the operations in DFS. For instance, the code *record route while walking* is corresponding to the pushing operation of stack. And the *If the grid is labelled* card could avoid repeated process combined with the *Don't Go* instruction card. As for the popping operation of stack in DFS, we use the combination of *If the cat is blocked* condition card and *Step Back* instruction card to substitute. We use intuitive expressions to represent the rules DFS algorithm uses to reduce the cognitive load. In our tool we use ReacTIVision framework [4] for real-time recognition of programs.

During programming, errors and bugs are unavoidable, which requires clear real-time feedback to help children complete programming (see Fig. 4 and Fig. 5). In our case, since the search process is dynamic, the effect of instructions could be shown in real-time but the result of children's program should be delayed. To make compensation to this situation, we use a trick in map design: using cycles near start grid. Under this map design, incorrect rules result in immediate failure.

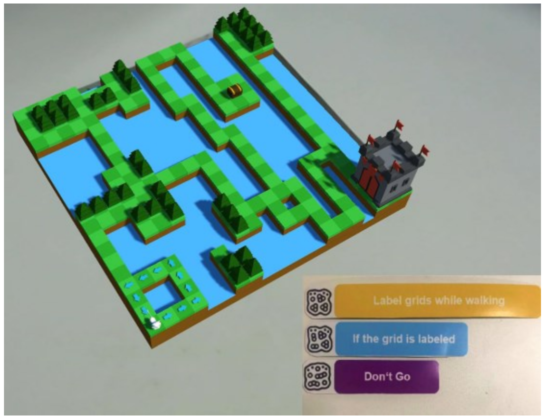

Figure 4: Real-time feedback: the cat would immediately stop searching with the rule: *If the grid is labelled -Don't go*.

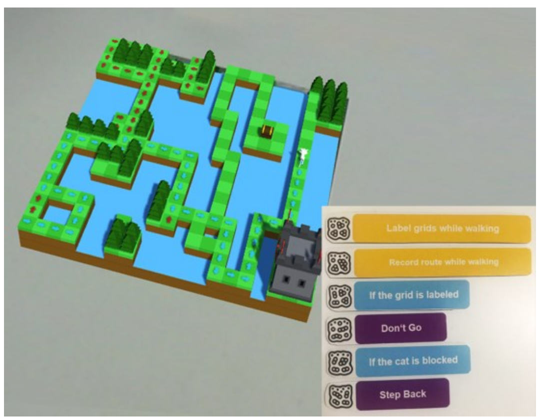

Figure 5: Real-time feedback: the blue arrows represent the route and the red represents blocked area. With the two rules (If the grid is labelled -Don't go. If the cat is blocked-Step Back) programmed, the cat will search the whole maze.

PRELIMINARY USER STUDY

We conducted a preliminary user study with 6 children ages 8-9 (average 8.33; 3 males and 3 females; two of them got previous programming experience) to find whether children could get a general understanding of DFS algorithm with our tool.

Before the evaluations, we briefly introduced the tangible cards and the game task. During programming, we recorded the time children needed and the errors children made. After programming, children needed to answer several questions to show whether they could understand how DFS algorithm works.

Result shows that except one child, the others successfully completed the game task in average 14.2 minutes (most in 22minutes) by themselves after several attempts. The reason the child failed to program independently is that she still wanted to control the cat mechanically with direction instructions after the introduction of basic search pattern. Researchers had to help her with the observation of obvious reluctance.

During programming, the most common error that children made was using instruction cards in purple without conditional cards. We noticed this error usually occurred when the virtual cat went to some critical state. For example, while the virtual cat was going to a labelled grid, children would use *Don't Go* card without *If the grid is labelled* card. Children often wanted to use instruction cards to solve the intuitive critical state rather than consider the whole process. But with the real-time feedback, after few times of attempts, children would find the correct rules.

After programming, we tested children with three prepared questions (contents in Fig. 6):

- Q1: Which grid will the character go to firstly?
- Q2: Draw the trace of the character according to the rules you programmed.
- Q3: If the map changes, will the cat find the chest successfully according to the rules you programmed?

The results of this test are cheerful: children all selected the right gird in Q1 and five children drawn the trace of the character correctly, which indicates children could get a general understanding of this search algorithm. As for Q3, five children thought the cat would find the chest in a different map while one child mentioned the disconnected situation and gave a negative answer. Hence, we infer that children could get a preliminary cognition of automation.

CONCLUSION AND FUTURE WORK

In this paper, we present ARCat, a tangible programming tool for DFS algorithm teaching. The tool decomposes DFS algorithm to guide and help children program to solve a search problem.

In order to guide children to program an automatically searching process rather than use direction instructions mechanically, we provide children with a basic search pattern, which also reduce the cognitive load of learning progress. And with the combination of visual environment presented in AR environment and tangible cards, children could program by assemble physical objects and get real-time feedback from the virtual scene.

From the preliminary user study, our results show children could have a preliminary understanding regarding automation, and be able to notes the certain detail of DFS algorithm. Our

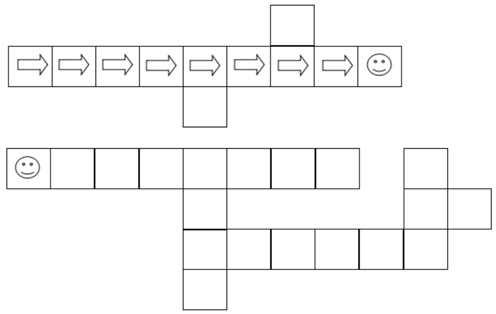

Figure 6: Contents of the test: the first part is corresponding to Q1 and the second is for Q2.

ACKNOWLEDGMENTS

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. 61872363, 61672507, 61272325, 61501463 and 61562063.

findings indicate the possibility of teaching basic computer algorithm such as DFS to young children.

In the future, this study could be improved in several ways. As shown in the preliminary user study, the real-time feedback could be improved to further reduce the cognitive load and avoid reluctant situation. Furthermore, more basic computer algorithms could be implanted into our tool and the visual presentations of these algorithms are needed to be examine.

REFERENCES

[1]. Michael S. Horn, R. Jordan Crouser, and Marina U. Bers. 2012. Tangible interaction and learning: the case for a hybrid approach. Personal Ubiquitous Comput. 16, 4 (April 2012), 379-389. DOI: https://doi.org/10.1007/s00779-011-0404-2

[2]. Felix Hu, Ariel Zekelman, Michael Horn, and Frances Judd. 2015. Strawbies: explorations in tangible programming. In *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. ACM, New York, NY, USA, 410-413. DOI: http://dx.doi.org/10.1145/2771839.2771866

[3]. Qiao Jin, Danli Wang, Xiaozhou Deng, Nan Zheng, and Steve Chiu. 2018. AR-maze: a tangible programming tool for children based on AR technology. In *Proceedings of the 17th ACM Conference on Interaction Design and Children (IDC '18)*. ACM, New York, NY, USA, 611-616. DOI: https://doi.org/10.1145/3202185.3210784

[4]. Martin Kaltenbrunner and Ross Bencina. 2007. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Proceedings of the 1st international conference on Tangible and embedded interaction (TEI '07)*. ACM, New York, NY, USA, 69-74. DOI: http://dx.doi.org/10.1145/1226969.1226983

[5]. Jeannette M. Wing. 2006. Computational thinking. Commun. ACM 49, 3 (March 2006), 33-35. DOI: https://doi.org/10.1145/1118178.1118215

[6]. 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, New York, NY, USA, 210-213. DOI: http://dx.doi.org/10.1145/1551788.1551831

[7]. Alpay Sabuncuoğlu, Merve Erkaya, Oğuz Turan Buruk, and Tilbe Göksun. 2018. Code notes: designing a low-cost tangible coding tool for/with children. In *Proceedings of the 17th ACM Conference on Interaction Design and Children (IDC '18)*. ACM, New York, NY, USA, 644-649. DOI: https://doi.org/10.1145/3202185.3210791

[8]. Amanda Sullivan, Mollie Elkin, and Marina Umaschi Bers. 2015. KIBO robot demo: engaging young children in programming and engineering. *In Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. ACM, New York, NY, USA, 418-421. DOI: http://dx.doi.org/10.1145/2771839.2771868

[9]. Qualcomm Vuforia, http://www.qualcomm.com/Vuforia

[10]. Peta Wyeth and Helen C. Purchase. 2000. Programming without a Computer: A New Interface for Children under Eight. In *Proceedings of the First Australasian User Interface Conference (AUIC '00)*. IEEE Computer Society, Washington, DC, USA, 141-148