E-Block: A Tangible Programming Tool for Children

Danli Wang[1], Yang Zhang[1], Tianyuan Gu[1], Liang He[1,2], Hongan Wang[1]
1. Institute of Software 2. Graduate University
Chinese Academy of Sciences, Beijing, China
{danliwang2009, bennybeihang, tianyuan.gu, edigahe, hongan.w}@gmail.com

ABSTRACT
E-Block is a tangible programming tool for children aged 5 to 9 which gives children a preliminary understanding of programming. Children can write programs to play a maze game by placing the programming blocks in E-Block. The two stages in a general programming process: programming and running are all embodied in E-Block. We realized E-Block by wireless and infrared technology and gave it feedbacks on both screen and programming blocks. The result of a preliminary user study proved that E-Block is attractive to children and easy to learn and use.

Author Keywords
Tangible Programming; Children; Tangible User Interface; Programming Languages; Education; Maze; Sensors

ACM Classification
K.3.1 [Computers and Education]: Computer Uses in Education; H.5.2 [Information Interfaces and Presentation]: User Interfaces

General terms
Design; Experimentation; Human Factors

INTRODUCTION
Works by Papert [3] and by Resnick [4] demonstrated that learning how to program may result in changes to the ways people think. Early studies with Logo also showed that programming can help children improve visual memories and basic numbers senses, as well as develop problem-solving techniques and language skills [1]. However, most of the existing programming languages are designed for professionals and are based on texts and symbols which are difficult for children to understand [5]. Hence, many research are conducted aiming at lowering the barrier of programming for children.

One of the earliest tangible programming projects is AlgoBlock [6]. Children could write their own programs to play a marine game by connecting the objects. Tern [2] is another tangible programming language. Children use program to control virtual roles or real walking robot. Though most of the previous works have offered children a

UIST '12, October 7–10, 2012, Cambridge, Massachusetts, USA.
ACM 978-1-4503-1582-1/12/10.

Figure 1: E-Block

feasible tool to write programs in tangible way, there are still some improvements could be done. Some of them have space limitation such as wires connecting blocks. Others offer little help on real-time debugging since children have to move the complete block sequence under the camera to compile. Therefore, this paper proposes a tangible programming tool which has less space limitations and real-time feedbacks. E-Block which uses the programming blocks and the sensors as input enables children to write programs to lead the character out of the maze as Figure 1 shows. It is more feasible and intuitive to children.

DESIGN AND IMPLEMENTATION
Our goal of design and implementation are as follows: a) E-Block is easy to learn for children. b) Children can program with no space limitation. c) The feedbacks are real-time for the convenience of debug. E-Block is composed of three parts: the virtual maze, the programming blocks and the sensors. We will introduce each kind of part next.

Virtual Maze
The virtual maze is composed of four kinds of cells: start cell, end cell, normal cell and sensor cell. There is a face on the top left corner of screen to show the real-time feedback.

Programming Blocks
Programming blocks send computer their physical information which is then translated into the program semantics. In E-Blocks, there are four kinds of programming blocks: start block, end block, direction block and sensor block. Each block has a single-chip microcomputer (SCM), an infrared transmitter and receiver

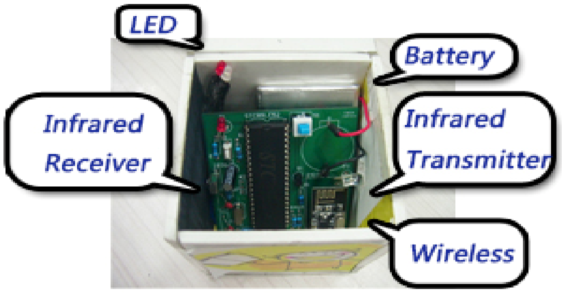

Figure 2: SCM in E-Block

module, a cell battery module, a wireless module and two LEDs. When a new block is added to the array, the former block's infrared signal will activate the new added one. The new added block will send its identity code to wireless box which is connected to PC through USB port. At the same time, it will open its own infrared transmitter. Computer (include wireless box and PC) will get the physical information from the content and the order of the identity codes received and translate it into program semantic. Then it will give new added block a respond after analyzing whether the block is correct. The respond controls the smiling face and LEDs' light to give children real-time feedbacks.

Sensor Input Devices
Sensor input devices have two functions. Firstly, when children finish programming, they need to press the start button on the sensor input devices to change the programming stage into the running stage. Secondly, in the running stage, when the character hits the sensor cell, children need to trigger the relative sensor, otherwise the character will be stopped. There are three kinds of sensors: temperature sensor, light sensor and tangible button sensor.

SCENARIO
In the maze game, user needs first to program a path for the character to escape the maze by placing blocks. Then, run the program and trigger sensors when necessary. In this part, a simple use case will be described.

Programming
Children first place the start block as the start of the block sequence. Then they need to add proper direction blocks into the block sequence which indicates the direction they want the path to go. Relative sensor blocks need to be placed to go on the path if the path hits any sensor cells. The smiling face on the screen will turn sad and the LED in the problematic block will turn red if children place wrong programming block into the sequence.

Running
When children finish the task in the programming stage, they need to press the start button on the sensor input devices to enter the running stage. And the character will start walking according to the path programmed before. When the character encounters a sensor cell, it will be stopped until user triggers the relative sensor.

PRELIMINARY USER STUDY
We conducted a preliminary user study for the E-Block's prototype. 6 children (3 boys and 3 girls) aged 5 to 9 took part in our experiment. All children finished the task within the stipulated time. In the questionnaire, when asked how much they like the game (very like, like, normal, dislike, very dislike), 4 children said they liked it very much and 2 selected 'like'. When asked about feedbacks, 5 children said that the feedbacks on block and on screen helped them to find the wrong block much more quickly and accurately.

CONCLUSION
E-Block's contribution could be summarized into the following points: Firstly, it presents a programming tool which enables children to write programs by placing wooden blocks with less space limitations. Secondly, it provides a new map method between the physical blocks and the program semantics. Thirdly, the system offers help on debug by giving real-time feedbacks on both screen and the programming blocks, which effectively help children to learn programming. In the future, we will design more programming blocks and add more scenarios into this tool.

ACKNOWLEDGMENTS
This research is supported by the National Natural Science Foundation of China under Grant No.60970090, the Frontier Project of the Knowledge Innovation of Chinese Academy of Sciences under Grant No. ISCAS 2009-QY03 and the Strategic Cooperation Project of Guangdong Province and Chinese Academy of Sciences under Grant No. 2011B090300086.

REFERENCES
1. Clements, D. The future of educational computing research: the case of computer programming. ITCE 1999, AACE Press (1999). 147-179.
2. Horn M, Jacob RJ K. Tangible Programming in the Classroom with Tern. In *Proc. CHI 2007,* ACM Press (2007), 1965-1970.
3. Papert, S. Mindstorms: Children, computers, and powerful ideas. New York: Basic Books. 1980
4. Resnick, M., Martin, F., Sargent, R., and Silverman, B. Programmable Bricks: Toys to Think With. IBM Systems Journal (1996), 443-452.
5. Revelle, G., Zuckerman, O., Druin, A., Bolas, M. Tangible User Interfaces for Children. In *Proc. CHI EA 2005*, ACM Press (2005), 2051-2052.
6. Suzuki, H. and Kato,H. Interaction-Level Support for Collaborative Learning: AlgoBlock—An Open Programming Language. In *Proc. SCSL 1995*, ACM Press (1995), 349-355.