Exploring the Child-Robot Interaction with the Programming in Mind

Bridging Physical and Virtual Programming for Young Children

Yuhan Lin
University of Maryland
jimmylin@umd.edu

ABSTRACT

My research is focused on the VEX 123, a robot and accompanying set of programming tools designed for children (ages 5-8). VEX 123 supports programming via three distinct methods designed to support young learners at various points along the conceptual and developmental programming trajectory. The goal of my work is to design a programming curriculum continuum from physical blocks to virtual blocks, to Switch mode (writing Python inside a block), and finally to text-based programming. My planned study employs a mixed method comparison to examine students' attitudes and learning outcomes across the various programming approaches. Through my research, I seek to better understand ways to support children learning programming with an easy to use robotics.

CCS CONCEPTS

• **Human-centered computing** → Human computer interaction (HCI); Interactive systems and tools Additional Keywords and Phrases: Computer Science Education, Block-based Programming, Tangible Programming, Child-computer interaction.

ACM Reference Format:

Yuhan Lin. 2022. Exploring the Child-Robot Interaction with the Programming in Mind: Bridging Physical and Virtual Programming for Young Children. In *Interaction Design and Children (IDC '22), June 27–30, 2022, Braga, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3501712.3538834

1 INTRODUCTION

Programming is becoming an increasingly important skill for learners to develop. While environments designed to teach programming have historically focused on older learners, there are a growing number of tools and environments designed to teach younger learners to program [1–3, 9]. Environments designed for younger learners are employing a growing array of approaches, including allowing kids to define behaviors by directly interacting with a robot (e.g., [1]), designing tangible programming tools that do not require a computer (e.g., [6]), leveraging crafts or motion as a means to define behaviors (e.g., [5]), and virtual programming environments that use symbols or animation to allow pre-literate users to define instructions [4, 8].

As the number of tools and environments, and the forms of programming they support, continue to expand, questions emerge as to how these tools fit together. My work focuses on understand how and when children move from one tool/environment to the next. Related to this focus is understand whether the skills, knowledge, or interest developed by using one environment carry over to new environments. This is particularly relevant for tools designed for younger children as many of these tools have extremely low floors (i.e., they are very easy to get started), but also have relatively low ceilings (i.e., they are limited in their capabilities and the sophistication of the programs that can be authored). Thus, an important question for designers interested in helping support youth as they progress along a programming learning trajectory is: How can we support young learners in moving from the extremely low-floor but also low-ceiling introductory environments to more powerful computational environments? It is this open design question that I seek to address with my work.

2 PRIOR WORK

In my previous work, I conducted a thorough analysis of the current landscape of block-based programming [7], interviewed teachers on the transition of block-based programming to text-based programming, and ran a child-robot study to evaluate the impact of physical blocks in learning programming. In the child-robot study, I introduced VEX123 (Figure 1a), a new programmable robot designed for young learners that supports multiple programming approaches to scaffold learners as their programming knowledge and skills grow. The VEX 123 can be programmed in three distinct ways: 1) direct manipulation, where programs are defined by pressing buttons directly on the robot, 2) a tangible programming approach that utilizes a novel, unplugged Coder and Coder Cards (Figure 1b) where commands written on plastics strips can be sequentially arranged and then sent to the robot, and 3) a block-based programming environment (Figure 1c).

The study found that the multiple programming methods support students in learning programming. In particular, the Coder and Coder Cards retain key aspects of tangible programming (i.e., the organization of physical objects to control the robot) but also includes key features that make it like virtual programming (i.e., organizing textual/graphical commands in a sequence to be run).

My research takes an interdisciplinary approach to support children learning programming. I am currently designing and developing the new approach to supporting children moving from block-based programming to text-based programming called Switch Mode. Switch Mode allows students to author text-based programs (such

IDC '22, June 27–30, 2022, Braga, Portugal

ACM ISBN 978-1-4503-9197-9/22/06.
https://doi.org/10.1145/3501712.3538834

Figure 1: (a) The VEX 123 Robot. (b) Coder and Coder Cards, and (c) virtual programming environment. ©VEX Robotics

Figure 2: Switch Mode Block Example - The Switch mode block (green) and the regular block (blue) will give the same outcome

as Python) inside a block within a block-based program (See Figure 2). The first step of transition is just to convert a single block from block mode to Switch Mode enabling students to manipulate and change the text-based version of that block while still inside a block-based program. Thus, students will be able to gradually transition their programming from BBP to TBP and they will not lose their comfort zone inside the BBP (See Figure 1).

3 METHODS

Using the design of Switch Mode I mentioned as a starting point, I plan to draft a series of 6-8 lessons based on the curriculum in block and text. Lessons will aim to teach students with basic programming on VEX virtual robotics using block-based programming first, then transition to text-based programming directly or transition to Switch mode and finally to text-based programming. The study will conduct in 4 different classrooms. Two classes will start with block-based programming and then transition to text-based programming directly. Two other classes will start with block-based programming and then go into the middle field of Switch Mode, then all eventually transition to text-based programming. I will conduct pre-, mid- and post-test and altitude surveys to understand students' programming ability and I will conduct post-interviews to understand students' perception on this transition method. See Figure 3 for study design.

3.1 Section 1: Qualitative Study

Total of 65 to 80 students should come from the same grade. Ideally it is coming from 2 or 4 classes, which would result in over 30 students per condition. Classes will be separated into control or experimental groups and the study should have the same length (either a semester or a year).

For the control group, the class will (1) start with block-based programming and (2) transition to text-based programming directly towards the second half of the class. For the experimental group, the class will (1) start with block-based programming, then (2) transition to switch mode, and (3) at the end transition to text-based programming completely.

At the beginning of the study, the beginning of switch mode, the end of switch mode and the end of the study, for a total of 4 times, all students regardless of control or experimental group will take an assessment to evaluate their knowledge and fill out an altitude and engagement survey. The assessment will help us to understand

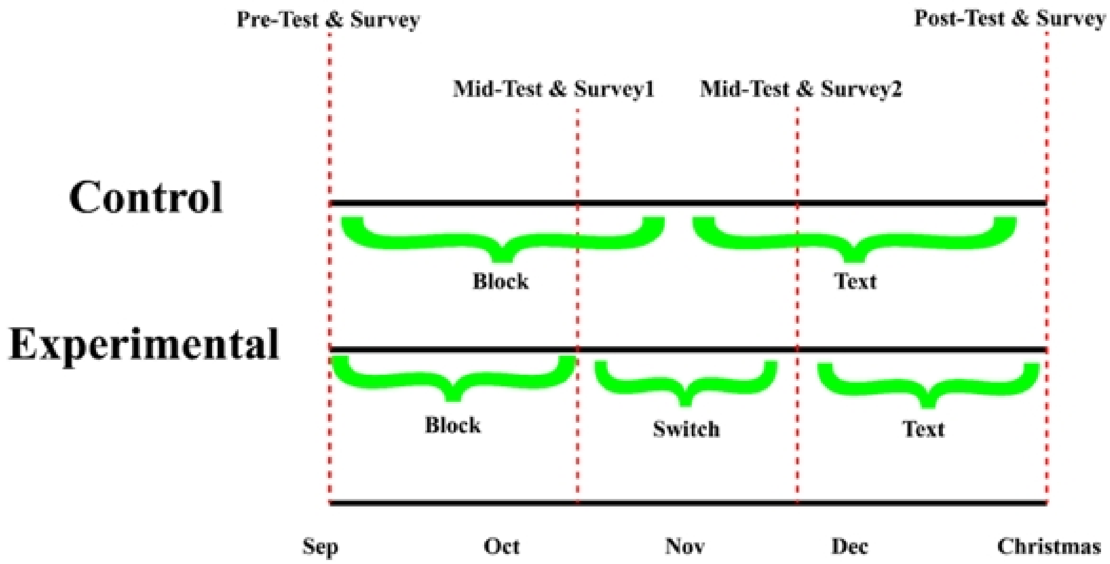

Figure 3: Study Design

if students have understood the concepts. The altitude and engagement survey will help us to understand students' perception of the Switch mode.

Selected students (probably one group per class) will also have four interviews during the study. Selected students will also be recorded for their group discussion. Each teacher will also have four interviews during the whole study.

All students will use VEX VR for the whole time. All instruction time will be video recorded.

3.2 Section 2: Quantitative Study

A total of 6-10 additional classes will be separated into the control or experimental group. These classes will only be collected for the 4 times assessment and survey via the teacher or the online assessment tool. The students in these classes will be system log recorded for all their actions in VEX VR.

4 RESEARCH QUESTIONS

The research questions I am pursing with this work are::

- Do embedded text-based blocks inside a block-based programming environment support students' transitioning from BBP to TBP? If so, how?
- How does programming modality (Physical Block, Virtual Block, Switch Mode, Text) impacts students' perception of programming with respect to motivation, engagement, authenticity, and attitudes towards programming?
- How does programming modality (Physical Block, Virtual Block, Switch Mode, Text) impact students' conceptual understanding, and attitude?

5 FUTURE WORK

Moving forward, I will use iterative design on Switch mode and conduct a mixed-method comparison study on children using various modality on programming. More specifically, I will examine students' attitudes and learning outcomes using the newly developed Switch mode as the tool and transition from block-based programming to Switch Mode and finally to text-based programming. This implementation will allow me to evaluate the impact of having physical blocks and having Switch mode. I believe using the Switch mode could have additional scaffolds for children when programming. With the physical blocks, it provides young children a hands-on tangible feeling and the teacher or facilitator could reduce the blocks they gave to children to reduce the noise and distraction from more advance blocks. With the exact same robot across modalities, children will not be introduced to new device every time they switch a modality.

REFERENCES

[1] Marina Umaschi Bers. 2018. Coding, playgrounds and literacy in early childhood education: The development of KIBO robotics and ScratchJr. In 2018 IEEE Global Engineering Education Conference (EDUCON), 2094–2102. DOI:https://doi.org/10.1109/EDUCON.2018.8363498
[2] Marina Umaschi Bers, Louise Flannery, Elizabeth R. Kazakoff, and Amanda Sullivan. 2014. Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. Computers & Education 72, (March 2014), 145–157. DOI:https://doi.org/10.1016/j.compedu.2013.10.020
[3] Caitlin Duncan, Tim Bell, and Steve Tanimoto. 2014. Should your 8-year-old learn coding? In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WiPSCE '14), Association for Computing Machinery, New York, NY, USA, 60–69. DOI:https://doi.org/10.1145/2670757.2670774
[4] Louise P. Flannery, Brian Silverman, Elizabeth R. Kazakoff, Marina Umaschi Bers, Paula Bontá, and Mitchel Resnick. 2013. Designing ScratchJr: support for early childhood learning through computer programming. In Proceedings of the 12th International Conference on Interaction Design and Children (IDC '13), Association for Computing Machinery, New York, NY, USA, 1–10. DOI:https://doi.org/10.1145/2485760.2485785
[5] Rostislav Fojtik. The Ozobot and education of programming | New Trends and Issues Proceedings on Humanities and Social Sciences. Retrieved October 21, 2020 from https://un-pub.eu/ojs/index.php/pntsbs/article/view/2666

[6] Michael S. Horn, Erin Treacy Solovey, R. Jordan Crouser, and Robert J.K. Jacob. 2009. Comparing the use of tangible and graphical programming languages for informal science education. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 975–984. Retrieved July 3, 2021 from http://doi.org/10.1145/1518701.1518851
[7] Yuhan Lin and David Weintrop. 2021. The landscape of Block-based programming: Characteristics of block-based environments and how they support the transition to text-based programming. Journal of Computer Languages (November 2021), 101075. DOI:https://doi.org/10.1016/j.cola.2021.101075
[8] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, and Yasmin Kafai. 2009. Scratch: programming for all. Commun. ACM 52, 11 (November 2009), 60–67. DOI:https://doi.org/10.1145/1592761.1592779
[9] Junnan Yu and Ricarose Roque. 2018. A survey of computational kits for young children. In Proceedings of the 17th ACM Conference on Interaction Design and Children (IDC '18), Association for Computing Machinery, New York, NY, USA, 289–299. DOI:https://doi.org/10.1145/3202185.3202738