Project Bloks: Embodied and Collaborative Learning with Tangible Interfaces for Young Children

Veronica Lin, Stanford Graduate School of Education, vronlin@stanford.edu
Paulo Blikstein, Teachers College, Columbia University, paulob@tc.columbia.edu

Abstract: This interactive demo presents Project Bloks, a development platform for tangible programming to enable young children to learn programming. The modularity of the platform demonstrates technical and design advances that can benefit not only designers, makers, and educators who can create activities for this platform, but also the children we strive to engage. With this demo, we hope to facilitate new thinking around about embodied and collaborative learning with tangible interfaces for young children.

Keywords: children, tangible interfaces, computational thinking, programming, collaborative problem solving

Introduction

Recent years have seen a steady increase in efforts to teach computer programming in formal and informal learning environments. Organizations and governments, domestic and foreign, have committed in various ways to making computer science education accessible to all students, and new research reveals the promise and challenges of teaching programming (Blikstein, 2018). The idea of computer science education first began in the 1960s with Seymour Papert and Logo (Papert, 1980), and in K-12 education in particular, this movement has since exploded in the number of computer and mobile applications (e.g. Scratch, Kodable, Tynker), building sets (e.g. littleBits, LightUp, MakeyMakey), programmable robots (e.g. Dash & Dot, KIBO, Cozmo), and others (e.g. Robot Turtles, Osmo).

The design space for apps, robots, and other technologies for bringing computational thinking and computer science to children has been in intense transformation during the last five years because of three factors. First, high-profile private and public initiatives and campaigns around coding education drew the attention of design firms, engineers, research labs, and creative educators. Second, the popularization of crowdsourcing platforms gave individuals and groups a viable way to fund and commercialize their ideas for tools to teach computational thinking. Finally, the development of new technologies such as low-power wireless communication, low-cost rapid prototyping, and new types of microcontrollers and microprocessors expanded the design space in unprecedented ways. It became possible to offer higher levels of abstraction using more sophisticated hardware and software, and therefore a more creative mix of digital and tangible interfaces.

However, this rapid design explosion brought about some shortcomings, such as the emergence of many single purpose, proprietary and often expensive designs that were incompatible with each other. Because many of these products were designed for consumption by individuals, they were not designed for schools and formal education, which inherently diminished the designers' focus on collaboration and generating low-cost designs. Therefore, despite the growth of creativity and new products, many design opportunities still remain within the tangible programming space, especially in regards to collaborative classroom activities around coding and particularly for young children.

The development of this platform aimed to address the problems of incompatible designs and increasingly complex hardware and software, while utilizing the learning affordances of an interface that is both digital and tangible. Prior research on tangible interfaces suggests many benefits for learning, including greater accessibility for young children, increased engagement and reflection, and better support for collaborative learning (Marshall, 2007). We drew inspiration from other tangible platforms, such as the Programmable Bricks (Resnick et al., 1998), Tern (Horn & Jacob, 2007), Topobo (Raffle et al., 2004), and Robo-Blocks (Sipitakiat & Nusen, 2012).

Introducing Project Bloks

Just as Google's Blockly is a platform for the creation of on-screen, block-based languages, the goal for Project Bloks was to be a platform for the creation of tangible programming languages. The design was inspired by Papert, Resnick, and Silverman's idea of low thresholds, high ceilings, and wide walls; to make it easy for novices to get started, but also possible for experts to build complex and diverse creations (Resnick & Silverman, 2005). We also wanted to allow designers, educators, and makers a wide variety of form factors, materials, and feedback channels (haptic, visual, and auditory), rather than having to deal with the technical aspects of developing technologies for learning. Ultimately, Project Bloks is for children as well as for developers; the components are designed to make sophisticated programming ideas such as conditionals, functions, and recursion accessible to young minds.

One of the key design features of the project is the separation of the (more expensive) hardware blocks and the (inexpensive) "pucks," which fit on top of the blocks and can be easily customized by educators, developers, and designers. Therefore, the same set of hardware blocks can support the design of multiple sets of special kits of "pucks" (e.g., one for robotics, another one for music, yet another one for cartesian motion). Another key feature is the ability for the system to learn new "functions", which can be assigned to blank pucks that children can write or draw on to indicate what they do. While existing tangible programming platforms can convey loops and conditionals, Project Bloks is the first to also support functions. Finally, another important feature is the facilitation of classroom collaboration: the system was designed to be used by several children at the same time in a classroom environment. For example, using the "function" feature, an entire classroom might collaboratively generate a solution at the center of the room, with individual groups of children working on their own procedures for this central classroom-wide program.

Hardware components

The system consists of the following (see Figure 1):

- A central brick, which contains the main source for power and communication. It includes wireless connectivity to support communications with activity software, hardware, and other blocks. It also contains a large green button, which users press to execute their programmed sequences.
- A set of blank bricks, including basic bricks and special bricks (e.g., repeat bricks, if and if-else bricks, function bricks), which can be used in various configurations. Each brick becomes a specific, individual instruction that corresponds with the puck placed on top. These instructions can be programmed by designers to express a wide range of actions, such as "move in a certain direction", "get flower", or "turn on or off". The special bricks can be used to express conditions and functions.
- A set of pucks, which can be customized for different activities. Each puck is an individual tile that represents an instruction. Pucks can be static ("get flower") or dynamic ("move in a certain direction"); the latter features rotary dials, buttons, and sliders for controlling a setting within an action.

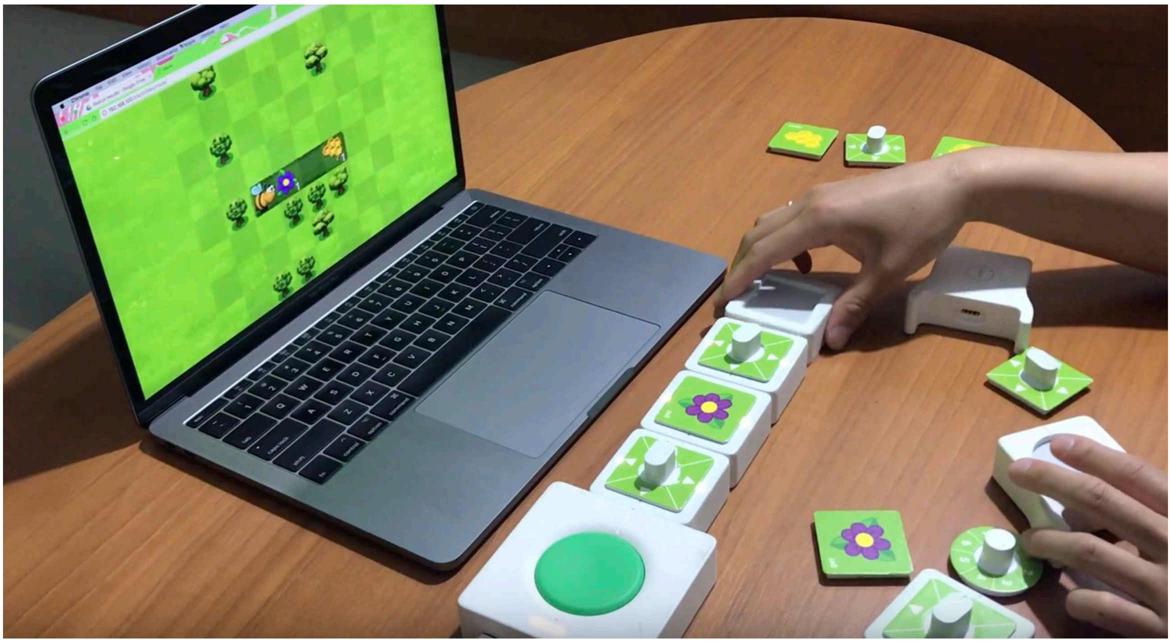

Figure 1. The Project Bloks system, shown with the code.org activity set.

Supporting hardware

The kit uses a mediator tablet so that an adult facilitator can specify the coding activity and block settings. For activities that require a presentation screen (such as the code.org activity in Figure 1), a web app can be launched on a separate laptop or tablet, which connects to the central brick via Wi-Fi.

Activities

Three different activities help demonstrate the diversity of applications that Project Bloks currently supports:

- **Code.org:** Children use the blocks and pucks to solve a series of maze puzzles, where the objective for each puzzle is to guide a bee through a maze using code.
- **Mirobot:** In using Project Bloks to control the Mirobot, an inexpensive drawing robot, children are able to create art ranging from simple line drawings to complex patterns.
- **LEGO Education WeDo:** After building a LEGO creation enhanced with a motor, distance sensor, tilt sensor, or LED light, children can use the blocks to control what they built.

Preliminary studies

Following the development of Project Bloks, we conducted task-based studies with over 40 first- and second-grade children in the U.S. and Brazil. Our research included studies with individuals as well as pairs, as prior work suggests that a key benefit of tangible interfaces is promoting productive collaborative interaction (Marshall, 2007). In general, we observed that the physical cues enabled children to learn how to use the system rather quickly, and also that the tangibility of the tool supported embodied learning in particular. In addition, children were able to engage with complex computational thinking ideas such as loops. Whereas with digital programming languages, children may infinitely duplicate series of instructions, Project Bloks has a limited-block design, such that the practical limitation of having physical blocks created new opportunities for children to learn a key concept (Lin & Blikstein, 2018).

Expected outcomes and contributions

We hope that demonstrating Project Bloks at CSCL will facilitate a larger conversation about embodied and collaborative learning with tangible interfaces for young children. In particular, one of the main design motivations was to allow better collaboration during programming activities in classrooms. On-screen programming, both on computers and tablets, does not always optimally facilitate collaboration; in order to be engaged, children need to be within reach of and facing the screen, which is especially difficult with small screens. For young children, we believe that Project Bloks can more effectively support collaborative programming. The physicality of the interface not only favors exploratory behaviors because tangibles are natural and intuitive to use (Marshall et al., 2003), but also establishes joint attention between multiple learners and creates a shared space for concurrent interaction (Fernaeus & Tholander, 2005; Suzuki & Kato, 1995). In our studies of pairs of children using Project Bloks, we observe that children engage in various strategies to negotiate shared control. Furthermore, while the physicality of the tool increases visibility of the other learners' activity and thinking, both learners also begin to develop a common language for planning and communication.

With the interactive demo at CSCL, we wish to highlight the potential and challenges for tangible interfaces in collaborative learning activities for young children.

References

Blikstein, P. (2018). Pre-college Computer Science Education: A survey of the field. Mountain View, CA: Google LLC. Retrieved from https://goo.gl/gmS1Vm.

Fernaeus, Y. & Tholander, J. (2005) Looking at the computer but doing it on land: Children's interactions in a tangible programming space. In: *McEwan T., Gulliksen J., Benyon D. (eds) People and Computers XIX — The Bigger Picture*. Springer, London, 3-18.

Lin, V. & Blikstein, P. (2018). Learning loops: Affordances and challenges of Project Bloks. ICLS 2018 Proceedings, 1531-1532.

Horn, M. & Jacob, R. (2007). Tangible programming in the classroom with Tern. *CHI 2007 Proceedings*.

Marshall, P. (2007). Do tangible interfaces enhance learning? *TEI 2007 Proceedings*, 163-170.

Marshall, P., Price, S., & Rogers, Y. (2003). Conceptualising tangibles to support learning. *IDC 2003 Proceedings,* 101-109.

Papert, S. (1980). Mindstorms: Children, computers and powerful ideas. New York: Basic Books.

Raffle, H., Parkes, A., & Ishii, H. (2004). Topobo: a constructive assembly system with kinetic memory. *CHI 2004 Proceedings.*

Resnick, M., Martin, F., Berg, R., Borovoy, R., Colella, V., Kramer, K., & Silverman, B. (1998). Digital manipulatives: New toys to think with. *CHI 1998 Proceedings*, 281-287.

Resnick, M. & Silverman, B. (2005). Some reflections on designing construction kits for kids. *IDC 2005 Proceedings*, 117-122.

Sipitakiat, A. & Nusen, N. (2012). Robo-Blocks: Designing debugging abilities in a tangible programming system for early primary school children. *IDC 2012 Proceedings*, 98-105.

Suzuki, H. & Kato, H. (1995). Algoblocks: an open programming language. *CSCL Proceedings 1995*, 349-355.

Acknowledgements

The technology was designed in partnership between Google Research, Google Creative Lab, IDEO, and Paulo Blikstein, with contributions from Arnan Sipitakiat. Jayme Goldstein, João Wilbert, Maggie Johnson, Steve Vranakis, Zebedee Pedersen, and Will Carey. This work was supported by a Google Faculty Award.