Tangible Coding Board: A Stepping Stone to Computing and Fabrication for Children

Erika Root[1], Swamy Ananthanarayan[2], Wilko Heuten[1], Susanne Boll[2]

OFFIS - Institute for Information Technology, Oldenburg[1]
Media Informatics and Multimedia Systems, University of Oldenburg[2]

Abstract

Computational thinking is increasingly important in today's world and teaching children this skill is important for their future. Teaching children programming is one way to develop computational thinking skills. HCI researchers have already explored different construction kits for teaching programming to different age groups. But, there is a trade-off between functionality and easy of use for newcomers. In this paper, we present the design of a Tangible Coding Board for children between the ages of 10 to 12 years which could help children to get into fabrication and minimal programming. With our toolkit, children can potentially program sensors and actuators in a tangible way to make their own interactive designs and fabricate personally meaningful artifacts.

1 Introduction

It is becoming increasingly important to teach computational thinking in early childhood given the growing use of computing, algorithms and data in a variety of disciplines (Wing, 2008). Computer programming is one way to develop these computational thinking skills (Orr, 2009), and tangible programming has been shown as a good approach to teach programming since it facilitates intuitive interaction with young children. Interacting with various physical artifacts not only increases the playfulness of learning (Marshall et al., 2007), but also offers an opportunity for children to fabricate their own interactive designs that are personally meaningful (Wang et al., 2011). On one end of the spectrum, there are many microcontrollers and construction kits which are based on the Arduino platform, such as Arduino Uno[1] and Arduino LilyPad (Buechley and Eisenberg, 2008). These kits have already shown that they can attract underrepresented groups to STEM (Buechley and Hill, 2010) and help users to create self-expressive

[1] https://www.arduino.cc/en/Main/ArduinoBoardUno

Veröffentlicht durch die Gesellschaft für Informatik e. V. 2017 in
M. Burghardt, R. Wimmer, C. Wolff, C. Womser-Hacker (Hrsg.):
Mensch und Computer 2017 – Workshopband, 10.–13. September 2017, Regensburg.
 https://doi.org/10.18420/muc2017-ws08-0337

and personally meaningful computational designs (Kafai et al., 2014). They can be used in educational settings, but they require an understanding of circuits and manual skills like soldering and sewing (Resnick and Silverman, 2005). They are also more applicable for older children in high school (Kafai et al., 2014). On the other end of the spectrum, there are many construction kits like littleBits[2], Cublets[3], and KIBO[4], which combine electronic building blocks with approachable drag-and-drop blockly programming. However, there is a trade-off between functionality and easy to use for newcomers. Arduino gives users a great deal of flexibility but can be difficult to program for children. Plug-and-play systems like Cublets and littleBits are easy to use but have limited functionality and expressiveness.

To fill this gap, we present the design of a tangible toolkit which could help children playfully program and fabricate their own personally meaningful artifacts. Children can program sensors and actuators tangibly and fabricate the results into artifacts which are personally meaningful. In our toolkit, we aim to strike a balance between the Arduino platform and plug-and-play construction kits.

2 Related Work

Computational toolkits have been widely researched since the 1980s. The first influential toolkit emerged with the development of the LEGO/Logo platform by MIT's Media Lab and many generations of "programmable bricks". Lego/LOGO was a computer-based learning platform that combined LEGO construction with the Logo programming language. New bricks were designed specifically for the Logo platform, which included gears, motors and, sensors. Children built machines out of traditional LEGO pieces and programmed their constructions (Resnick, 1993). In the late 90s, Arduino was presented and extended the early platforms by including new types of sensors, actuators and, ways to interact with computers. They were especially designed for hobbyists and interaction designers[5]. In recent times many other toolkits have been designed with new form factors, architectures, and industrial designs. These toolkits are broadening the reach of physical computing to new audiences for new use cases. A notable Arduino based kit is LilyPad; the Arduino LilyPad hardware platform focuses on e-textiles and provides a new medium for engaging particularly female students in engineering and computer science (Buechley and Eisenberg, 2008). The LilyPad kit contains sewable electronic components, including a programmable microcontroller and an assortment of sewable sensors and sewable actuators (Buechley and Eisenberg, 2008). Programming however is still typically accomplished through an IDE. To facilitate intuitive and tangible programming, Cublets is a computational construction kit that encourages users to experiment and play with a collection of sensor, logic and, actuator blocks. These robotic blocks allows children to build simple robots by snapping together active blocks (Schweikardt and Gross, 2006). KIBO is another such robot kit which uses wooden blocks without electronics and a robot programming platform which can read and execute the optical codes on the wooden blocks without a computer (Sullivan et al., 2015).

[2] http://littlebits.cc/
[3] http://www.modrobotics.com/cubelets/
[4] http://kinderlabrobotics.com/
[5] https://www.arduino.cc

LittleBits is another kit consisting of electronic blocks that can be connected magnetically to assemble circuits. These blocks are color-coded to help children clearly identify inputs, outputs, logical operators and, power. The design ensure that only functional circuits can be assembled (Bdeir and Ullrich, 2011). Recently, Kazemitabaar et. al introduced MakerWear, a new wearable construction kit for children that uses a tangible, modular approach to wearable creation. This kit includes different modules with sensors, actions and, modifiers for building wearable artifacts (Kazemitabaar et al., 2017). Programming the modules however is limited and left for future work.

In our work, we aim to teach computation through existing children's craft culture. They can fabricate what is personally meaningful, such as a smart Halloween costume or an interactive plant for their classroom. Our goal is to fill the gap between complex high functionality platforms like Arduino and the pre-programmed construction kits where the children snap blocks together. Furthermore, we think that the idea of the Tangible Coding Board is a good fit for children between the ages of 10 to 12 because the Arduino programming environment might be too challenging, while the plug-and-play construction kits are less functional with limited expressiveness for personalized designs. In the following section, we present our idea of the Tangible Coding Board in detail.

3 Design Idea

We present our idea through a small scenario. Imagine a 10-year-old girl who wants her backpack to switch on a light when it is evening, or for the light to glow blue when it is cold outside. So in our scenario, the girl can take the Tangible Coding Board and start to program her interactive backpack. The Tangible Coding Board has two main areas, an input area for sensors on the left side, and an output area for actuators on the right side. Figure 1 shows a first sketch of our idea and Figure 2 shows a paper prototype to illustrate the tangibility. In our scenario, the girl can put the light sensor or the temperature sensor on the left side and it automatically indicates that the sensor can be set up. With knobs the girl can define the min or max values of the sensor, like hot or bright. When she saves the value for the sensor, the sensor light turns off and a display shows the saved value. Afterwards, an LED can be put on the right side of the board and the girl has the possibility to customize the behavior of the LED through knobs and sliders. All changes are synchronous and the girl sees the results in real-time. If the girl wants to change the value of a sensor or the parameters of an actuator, she can press the tangible object. It lights up and is changeable again. It is also possible to define more than one input-condition or output reaction, meaning that the girl can program two sensors in one row on the left side of the board or for example two LEDs in one row on the right side.

In our initial design we aim to support a light sensor, an accelerometer, a temperature sensor and an ultrasonic sensor because they are commonly available in the market. Behind all the sensors are pre-defined min and max values. However, we want to discover which sensors are really important and mostly used by children for creating meaningful artifacts. We also aim to support actuators like light, vibration, and sound. The possible actuators and sensors as well as their values are shown in Figure 2. Our design is modular because (Zuckerman et al., 2005) has

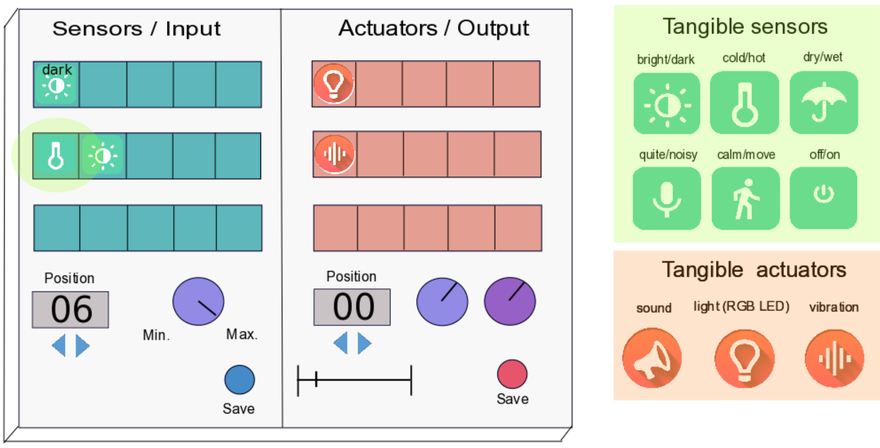

Figure 1: First sketch of the Tangible Coding Board

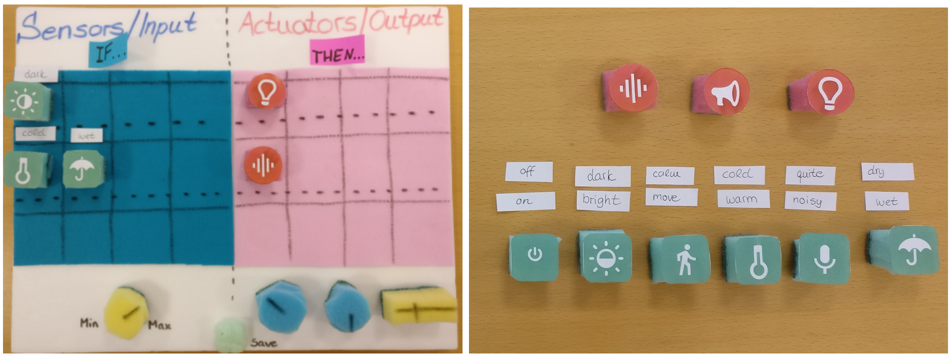

Figure 2: Paper prototype

showed modularity as a general design criteria for tangible programming. By this we mean, the modules are independent and can be combined in many ways. After the child has programmed the sensors and actuators, he/she can take them from the board and use them for creating their own meaningful artifact. After that, the integrated modules communicate with each other. That means when the programmed value of the sensor is measured, the sensor module sends this to the connected actuators for making the programmed reaction.

With the tangible approach, we can manipulate the sensor parameters with knobs and sliders. This will help children to first understand the basic concepts behind programming. With the Tangible Coding Board children can create own things without any other programming environment. We think it is important to enable children to be their own designer because children learn best when they are actively engaged in designing and creating things (Resnick and Silverman, 2005).

4 Research Questions

Given our design, we aim to explore the following research questions:

1. *How should the board and the tangibles be designed for children between the ages of 10 to 12 years?*
   There are many possibilities for size and form factor for both the board and the tangibles. We have to investigate, which size is manageable for this age group. We also have to find out if there is a need of differentiating the design of actuators and sensors. Additionally, the design and size of the whole toolkit should be portable.

2. *How can children program using tangible knobs and sliders?*
   Actuators have many parameters which can be changed for a pattern. But which are important and suitable for children's understanding and should be manipulable by the children? How can we abstract sensors and actuators for children? Also, it could be too difficult for the children to handle different input and output modalities on one board.

3. *How is the coding board used by children?*
   With the Tangible Coding Board we want to enable children to playfully design and program their first digital pattern. So, we have to investigate if children find it easy to use and what programming concepts they utilize, e.g., if statements. Furthermore, we want to investigate if the Tangible Coding Board motivates the children to go deeper into programming.

References

Bdeir, A. & Ullrich, T. (2011). Electronics as material: Littlebits. In Proceedings of the fifth international conference on tangible, embedded, and embodied interaction (pp. 341–344). TEI '11. Funchal, Portugal: ACM.

Buechley, L. & Eisenberg, M. (2008). The lilypad arduino: Toward wearable engineering for everyone. IEEE Pervasive Computing, 7(2), 12–15.

Buechley, L. & Hill, B. M. (2010). Lilypad in the wild: How hardware's long tail is supporting new engineering and design communities. In Proceedings of the 8th acm conference on designing interactive systems (pp. 199–207). DIS '10. Aarhus, Denmark: ACM.

Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. Trans. Comput. Educ. 14(1), 1:1–1:20.

Kazemitabaar, M., McPeak, J., Jiao, A., He, L., Outing, T., & Froehlich, J. E. (2017). Makerwear: A tangible approach to interactive wearable creation for children. In Proceedings of the 2017 chi conference on human factors in computing systems (pp. 133–145). CHI '17. Denver, Colorado, USA: ACM.

Marshall, P., Rogers, Y., & Hornecker, E. (2007). Are tangible interfaces really any better than other kinds of interfaces? In Chi '07 workshop on tangible user interfaces in context & theory. San Jose, California, USA: ACM.

Orr, G. (2009). Computational thinking through programming and algorithmic art. In Siggraph 2009: Talks (31:1–31:1). SIGGRAPH ’09. New Orleans, Louisiana: ACM.

Resnick, M. (1993). Behavior construction kits. Communications of the ACM, 36(7), 64–71.

Resnick, M. & Silverman, B. (2005). Some reflections on designing construction kits for kids. In Proceedings of the 2005 conference on interaction design and children (pp. 117–122). IDC ’05. Boulder, Colorado: ACM.

Schweikardt, E. & Gross, M. (2006). Roblocks: A robotic construction kit for mathematics and science education. In Proceedings acm international conference on multimedia interaction (icmi) (pp. 72–75). Banff, Canada: ACM.

Sullivan, A., Elkin, M., & Bers, M. U. (2015). Kibo robot demo: Engaging young children in programming and engineering. In Proceedings of the 14th international conference on interaction design and children (pp. 418–421). IDC ’15. Boston, Massachusetts: ACM.

Wang, D., Zhang, C., & Wang, H. (2011). T-maze: A tangible programming tool for children. In Proceedings of the 10th international conference on interaction design and children (pp. 127–135). IDC ’11. Ann Arbor, Michigan: ACM.

Wing, J. M. (2008). Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 366(1881), 3717–3725.

Zuckerman, O., Arida, S., & Resnick, M. (2005). Extending tangible interfaces for education: Digital montessori-inspired manipulatives. In Proceedings of the sigchi conference on human factors in computing systems (pp. 859–868). CHI ’05. Portland, Oregon, USA: ACM.