Toward Inclusive Learning: Designing and Evaluating Tangible Programming Blocks for Visually Impaired Students

Zhiyi Rong, Ngo Fung Chan, Taizhou Chen, and Kening Zhu(✉)

City University of Hong Kong, Kowloon, Hong Kong
zingaiyung@hotmail.com, hugochan525@gmail.com,
taizhou.chen@my.cityu.edu.hk, keninzhu@cityu.edu.hk

Abstract. Tangible programming toolkits are widely used to nurture computational literacy in the young generation. However, novice learners with visual impairment have been neglected as these toolkits are primarily designed for sighted students, and mostly rely on visual cues in the whole manipulation process. To fill this gap, we present CodeRhythm (Fig. 1), a tangible programming toolkit for engaging blind and visually impaired (BVI) students to learn basic programming concepts by creating simple melodies. In this paper, we describe the design features of CodeRhythm and discuss the feedback and future improvement based on the preliminary user study.

Keywords: Accessibility · Tangible user interfaces · Computer-science education · Inclusive design

1 Introduction

Over the past decade, learning programming is no longer a priority of only a small group of people. Block-base programming language is becoming incrementally pervasive in schools, for example, Scratch [18] and Alice [14], empowering students to think creatively and systematically. Meanwhile, tangible user interfaces (TUIs) [11] have demonstrated their benefits to design iteration [5, 7], digital fabrication [19], and digital entertainment [21, 22]. Especially with the application of TUIs for Education, some new possibilities and definitions were added to block-based programming language, which provided effective and inviting pathways for programming constructs learning [4, 8–10, 23].

To realize the educational aim of supporting engagement and diversity, more and more effort has been made to engage students with special educational needs to learn basic programming knowledge. However, the early kids-coding toolkits did not pay specific consideration to novice learners with visual disability, since existing programming toolkits are primarily targeted for sighted students, who rely on visual cues to manipulate the system and experience the visual outcome. While some researchers started to develop programming-learning systems that are inclusive for learners with visual disability [2, 13], these systems are developed for BVI users with prior programming skills, which may result in barriers for novice BVI learners.

M. Kurosu (Ed.): HCII 2020, LNCS 12183, pp. 326–338, 2020.
https://doi.org/10.1007/978-3-030-49065-2_23

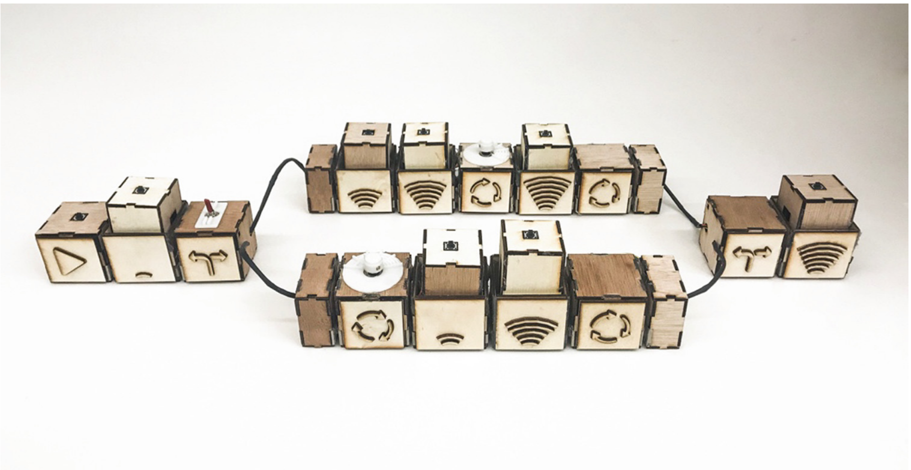

Fig. 1. CodeRhythm is a tangible programming toolkit for engaging blind and visually impaired (BVI) students to learn basic programming concepts by creating simple melodies.

To fill this gap, we present CodeRhythm, a tangible programming toolkit for engaging visually impaired students to learn fundamental programming concepts by creating a simple melody. We adopt the form factor of blocks as fundamental elements to represent codes. The whole toolkit contains a set of blocks comprising tangible syllables blocks - do, re, mi, fa, so, la, ti - and several distinctive function blocks, representing the programming concepts of execution, looping, conditional branching. In the rest of the paper, we will first discuss the background of tangible educational toolkits and accessible programming tools, describe the design features of CodeRhythm, and discuss the feedback and future improvement by the preliminary user study.

2 Background

2.1 Inclusive and Accessible Programming Tool

Researchers have started to explore different approached to introduce BVI students with programming. Bigham et al. [2] studied engaging students with personalized chatbots to study computer-programming language. Bonk enabled BVI users can create interactive and accessible audio games [14]. StructJumper [1] was created to help blind users quickly navigate the code with the assistant of audio cues and shortcuts. Another approach emphasizes on integrating digital fabrication, for example, 3D printing [13] into the instruction for BVI students by writing the Ruby program. These tools are beneficial to BVI users with programming skills and Braille literacy; however, they are too complicated and confusing for novice learners such as children.

2.2 Block-Based Graphical and Tangible Programming Tools

Scratch [18] is one of the most widely used block-based languages around the world, which aims at nurturing children with computational thinking through the Scratch Online Community [3]. Other educational tools such as Code.org [12] also provide a similar learning pathway using block-base language. However, these learning tools highly rely on visual properties and inaccessible for BVI users. Tangible tools have been paving an alternative and inviting pathway for learning computational constructs. Strawbies [10] utilized computer vision to capture the blocks assembled by children, and then control the game character in iPad. Tern [9] explored to control a moving robot by capturing the instruction of woodblocks. While BVI users can possibly distinguish the blocks by different shapes, these toolkits are still highly reliant on visual activities, for example, seeing feedback from the screen.

To address this issue, some effort has been made to address these tangible blocks to enable BVI children to learn basic programming literacy by connecting cables between blocks and creating audio feedback, such as Totino [17, 19]. While connecting cables and sockets may place challenges to young BVI users whose body and spatial awareness may be affected [6], CodeRhythm uses embedded magnets to connect the blocks rather than cables which make users confused about cable's direction and struggle to assemble. Our preliminary user study suggested that using magnets as the connection is helpful and preferred by BVI users. In addition, compared to existing musical programming blocks, which may only play the melody after full assembly, each syllable block of CodeRhythm can provide independent audio feedback as a complement to the tactile patterns, helping users distinguish syllable blocks. We also design the push-and-pull feature for adjusting the duration parameter of syllable, providing more variation on the sound-based programming. Lastly, CodeRhythm comprise more diverse functions, such as Switch function, which is not utilized in Torino, to provide a more inviting and diverse combination of tangible blocks. We address these features as complementary approaches to a similar challenge.

3 System Overview

CodeRhythm contains two categories of blocks: syllable blocks - do, re, mi, fa, so, la, ti, and distinctive function blocks – Start, Switch, Loop, Pause (Fig. 2). All of the blocks are connected by magnetic force. Each syllable block represents a specific note in the melody, and the user can directly control the parameter value of duration by pulling and pushing a protruding cube on the top of each syllable block (Fig. 3). The syllable block houses an Arduino Mini board and speaker module, which controls the block independently, enabling the user to experience the audio feedback immediately without connecting all sets of blocks. The Start block acts as an execution block, which locates on the left side of the overall sequence. The button on the top side allows users to initiate the program. Switch Block is designed to create the conditional branching function, which enables the user to choose different programming paths by controlling the knitter switch on the top of the block. The function of the Loop block is intended to repeat the syllable blocks attached to it. Pause block could be used with syllable blocks to set a duration of pause between them.

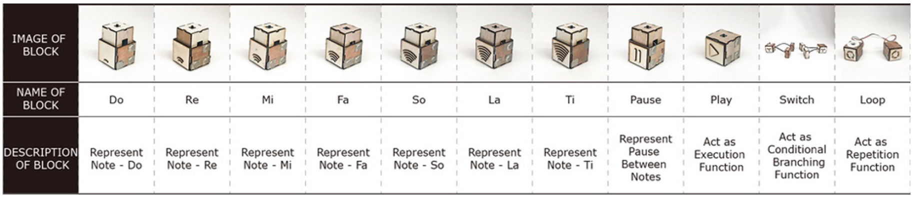

Fig. 2. CodeRhythm contains two categories of blocks: syllable blocks - do, re, mi, fa, so, la, ti, and distinctive function blocks – Start, Switch, Loop, Pause

Fig. 3. Each syllable block represents a specific note in the melody, and the user can directly control the parameter value of duration by pulling and pushing a protruding cube on the top of each syllable block

4 Inclusive Design Features

In our inclusive tangible learning system, each block represents one line of code. Since our toolkit is designed for BVI students, it is significant to lower the usability burden as much as possible. That is, make the identity of each block be distinguished easily and clearly, and make the assembly process as convenient as possible. In the following part, we will highlight and describe several inclusive design features of CodeRhythm.

4.1 Clear Embodiment of Block Character

We choose cubic block as a primary design metaphor because the flat interface of the cubic block can be combined quickly and adequately with the embed magnet. Also, this cubic metaphor can make the overall toolkit congruent in form factor. The Start block is a single cubic block, and the Syllable block has a protruding cube on the top, which can be pulled and push (Fig. 4). For the Switch Blocks, we use one block as an original block and two small blocks as end blocks, which connected with physical cables, to show the conditional branching action in a tangible way. For the Loop block, we use a cable to combine the start block and end block, indicating that the syllable blocks set between the two blocks would be manipulated to repeat. As a result, BVI users can distinguish the primary function of each type of block easily and quickly.

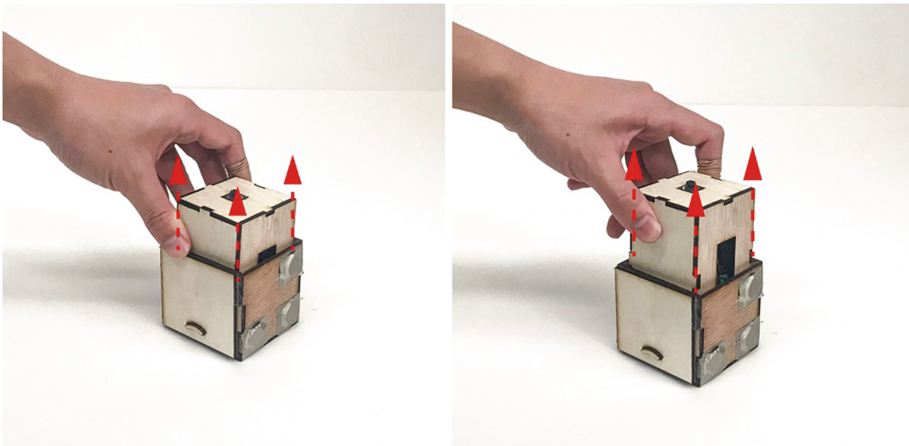

Fig. 4. Syllable block has a protruding cube on the top, which can be pulled and push.

4.2 Assistive Block Connection

Wire connection often causes BVI users to struggle to locate where they should plug the wire into sockets. It also disables them to distinguish if the direction of the wire is right. In contrast to using the wire as connection of blocks, each of the block in our system is connected by the embed magnet inside the interface. Regarding the signal transmission issue, we choose conductive tape as the medium. The conductive tape is attached to the block's surface, which performs well in transmitting the signal when we conduct experiments and user tests.

One of the essential advantages is that with the use of the magnet, we can decrease the difficulty of connection as much as possible (Fig. 5). In this way, the user can assemble two blocks if the direction of embedded magnets matches. The magnetic force also acts as an indication of the correct connection for BVI users. (Otherwise, two blocks with the wrong connection would repulse each other.)

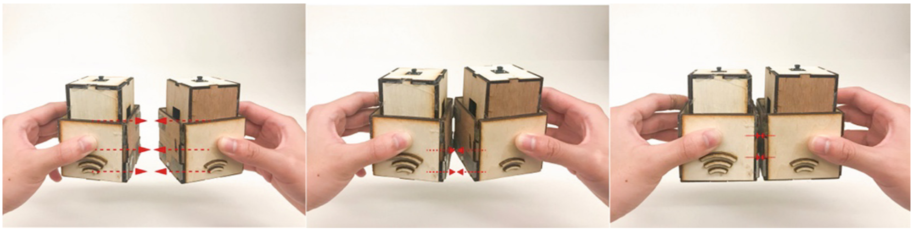

Fig. 5. A figure caption is always placed below the illustration. Short captions are centered, while long ones are justified. The macro button chooses the correct format automatically.

Another advantage is that using magnet and conductive tape as a connection method can shorten the connection length compared to wire connection. In this way, the connected blocks would form a continuous interface, enabling BVI users to follow and trace the tactile patterns rapidly.

4.3 Identify Block's Function by Tactile and Audio Feedback

Touching and recognizing tactile patterns is one of the essential ways for BVI users to distinguish blocks. Since not all children learn to read Braille, we design the easily distinguish symbols as the tactile patterns and attach them to the surface of corresponding blocks. For instance, we use a triangle as the symbol of the Start function, and we use two arrows pointing to different directions to represent the Switch function.

Besides the tactile patterns, to create a more inviting and efficient interaction mechanism for the syllable block, we install an Arduino Mini board and speaker module into the block, which controls the block independently, enabling the user to experience the audio feedback immediately without connecting all set of blocks (Fig. 6). For example, the BVI user can pick up a syllable block, touch the tactile pattern on it and recognize which note it represents, and press the bottom on the top side to listen to the sound of the note immediately. At the same time, the user can also push or pull the protruding box to experience the duration of the note immediately. With the combination of tactile and audio sensation, CodeRhythm creates a more exciting and intuitive interaction mechanism, which provides a complementary approach to the blocks with only tactile function.

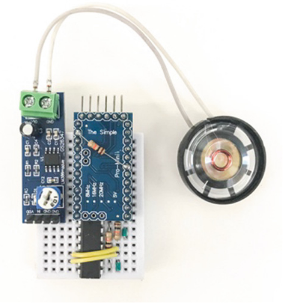

Fig. 6. We install an Arduino Mini board and speaker module into the block, which controls the block independently, enabling the user to experience the audio feedback immediately without connecting all set of blocks.

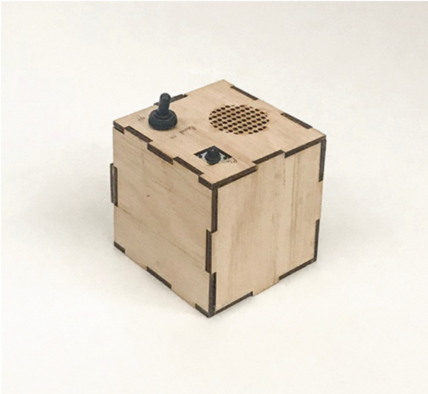

Fig. 7. We are developing and testing the recording block, which can record and replay sound that the user intent to customize. Engage Diverse Users for Collaboration

4.4 Possibility of Personalization and Customization

In addition to designing and developing syllable blocks whose duration can be adjustable, we also consider creating blocks that can provide a personalized and customized interaction experience. Currently, we are developing and testing the recording block (Fig. 7), which can record and replay sound that the user intent to customize. Mixing a recording block with syllable blocks can enable a more flexible interaction scenario and cultivate user's creativity.

Although CodeRhythm is designed for BVI students, we still want to make it as inclusive as possible, that is, engaging more diverse users in the learning process. Since the tactile symbols in our system are from ordinary and simple visual elements, they are also accessible for sighted students. Thus it provides an inclusive learning environment in which BVI students and sighted students can manipulate the toolkit collaboratively.

Fig. 8. The preliminary user study was conducted in a Special Education Needs Room of a University's library.

5 Preliminary User Feedback

5.1 Procedure

The initial prototype was demonstrated to a 30-year-old female with visual disability who was a BVI educator with a specialization in the music field and a female BVI student majoring in marketing. The preliminary user study was conducted in a Special Education Needs Room of a University's library (Fig. 8). The researchers first introduced the idea and goal of CodeRhythm and explained the function of each block. During this introduction session, researchers encouraged the participants to pick up each type of block, touch, and interact with it. Then the participants were told to connect Start Block and one of the syllable blocks, which enabled the users to create the first and basic program intuitively.

After the brief introduction, the researchers picked up one Start Block and two syllable blocks, connecting them with Switch and Loop Block, respectively. As a

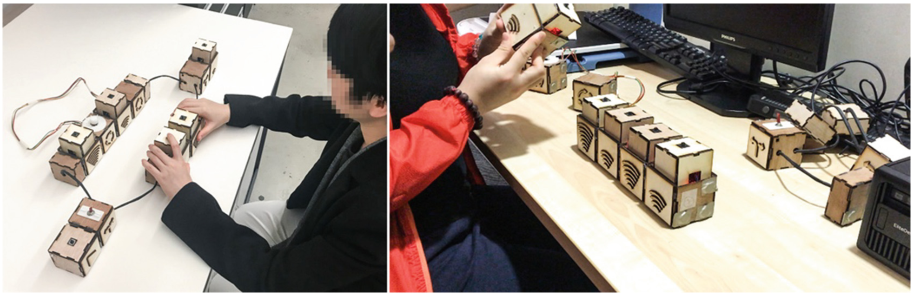

Fig. 9. The participants are using CodeRhythm to create melodies.

result, two different simple melodies were created, which introduced the participant how these two functions work in the program. In this process, the participants were told to touch and experience these two sequences of blocks.

In the next part, the researchers assemble a Start block, five syllable blocks, a Switch block, and a Loop block, to demonstrate a more diverse melody for participants. After experiencing and interacting with the overall effect of blocks, the participants were encouraged to create a personalized melody with assembling any types of blocks. In the end, we interviewed the participants about the usability challenge, the key advantages of the toolkit, and potential improvement in the future (Fig. 9).

5.2 Finding

In the experience process, the participants were able to assemble blocks to create a simple melody successfully. Also, the participants could connect blocks easily without much hesitation and exploration. When tracing the block sequence by touching, the participants were able to distinguish the different characteristics of each type of block clearly.

In the interview session, both of the participants commented that the experience was fun and inviting. Notably, they thought the audio feedback, which matched the tactile pattern on the syllable blocks, was surprising and helpful. Regarding the connection with magnet and conductive tape, the participants appraised that it was beneficial and eliminated the exploration time.

"The magnets help a lot. Now I can connect blocks without hesitation" - Educator Participant
"It is so cool to connect blocks quickly." – Student Participant

Also, the educator participant noted that we could also utilize the toolkit to help BVI students understand electric literacy since the connection was one of the significant parts of electric, and our method of using magnet and conductive tape performed well in decreasing the connection difficulties.

"Connecting blocks has the similar experience of connecting circuit. I think your toolkits could also be used for teaching BVI students the electric literacy."

Regarding the tactile pattern design, the student participant suggested that we can develop different versions of the tactile patterns for BVI students with distinctive cognition ability.

"I think you could use English alphabet patterns for students who have learned English and music literacy, and adopt different numbers of curves to represent specific notes in melody for students with young ages."

The student participant also said that our toolkits reminded her of the experience of her childhood, in which she was the only BVI student in class. Regarding this learning experience, she pointed out that CodeRhythm had the potential to enable sighted students and BVI students to learn together and play together, which could help BVI students to integrate with society.

"More and more BVI students have been getting access to education. As a BVI student, I think it is necessary to learn everything about the world and integrate with society. Thus the ideal learning product is the one that could enable sighted students and BVI students to learn together and play together. I can see the great potential of CodeRhythm."

The participants mentioned that mastering all of the concepts and functions and manipulating them in a short time was challenging and difficult, which required high-level cognitive ability. Regarding this challenge, the participants suggested that it was better to design a well-structured and detailed curriculum and applied it to a workshop of multiple sessions.

"Since there are various functions of blocks, I think it is challenging for young students to learn so much content in a short time. You know, it requires a high cognition ability. Maybe some divided class would seem more reasonable."- Educator Participant
"I think you can design a well-structured curriculum, such as teaching how to use Syllable Blocks in the first class, then teaching Switch function in the next class, etc." - Student Participant

For the improvement of expressiveness, the educator participant suggested that we can add some more diverse and personalized tunes to provide a more compelling and exciting experience.

"Maybe you can add more possibilities to the blocks. More tunes seem better. And you can even design a recording blocks to record and replay various sounds."

Regarding the educational goal of CodeRhythm, the educator participant noted that it was useful and inviting to learn the basic and simple concept, but not proper enough for learning the advanced and traditional computational literacy.

"The toolkit is so interesting, and I think it is an inviting learning toys for young children. But it seems unnecessary for the BVI people who want to learn formal computer science."

Regarding the same question, the student participants said that our toolkit was so attractive that she would like to play with it and listen to the melodies, even though she did not intend to learn programming literacy.

"You know, it is satisfying enough for BVI people to build some toy blocks line by line because we cannot see what the blocks look like. However, when I play with your toolkit, I feel surprised, because the audio feedbacks of blocks create a new interaction experience for BVI people to play with blocks. I want to combine different blocks to get various audio feedbacks."

6 Conclusion and Future Work

We present CodeRhythm, a tangible programming toolkit for engaging blind and visually impaired (BVI) students to learn basic computational concepts by creating simple melodies. Our preliminary user study shows that CodeRhythm is an inviting and inclusive toolkit to learn computational concepts, which decreases the usability burden as much as possible. In the future, we will design and incorporate a well-structured and detailed curriculum into workshops of multiple sessions to teach BVI students computational thinking with CodeRhythm systematically. We will also develop blocks with more various tunes and blocks with recording and replaying functions to expand the expressiveness of CodeRhythm. Moreover, we will try to apply CodeRhythm in the inclusive learning environment where sighted students and BVI students can learn together and play together. While our toolkit is designed for learning computational thinking, we are pleased to see the potential for incorporating CodeRhythm into electric literacy teaching for BVI students.

Acknowledgement. This research was supported by the Young Scientists Scheme of the National Natural Science Foundation of China (Project No. 61907037), and the Centre for Applied Computing and Interactive Media (ACIM) of School of Creative Media, City University of Hong Kong. We thank the participants who took part in the user evaluation. We are also grateful for the help of Steve Ching and Shanshan Jiang.

References

1. Baker, C.M., Milne, L.R., Ladner, R.E.: StructJumper: a tool to help blind programmers navigate and understand the structure of code. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, pp. 3043–3052 (2015). https://doi.org/10.1145/2702123.2702589
2. Bigham, J.P., Aller, M.B., Brudvik, J.T., Leung, J.O., Yazzolino, L.A., Ladner, R.E.: Inspiring blind high school students to pursue computer science with instant messaging chatbots. In: Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education, SIGCSE 2008, pp. 449–453 (2008). https://doi.org/10.1145/1352135.1352287
3. Brennan, K., Resnick, M.: New frameworks for studying and assessing the development of computational thinking. In: AERA 2012, pp. 1–25 (2012). http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf. Accessed 18 Sept 2017
4. Buechley, L., Eisenberg, M., Catchen, J., Crockett, A.: The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2009, pp. 423–432. ACM, New York (2009). https://doi.org/10.1145/1357054.1357123
5. Chen, T., Wu, Y.S., Zhu, K.: DupRobo: interactive robotic autocompletion of physical block-based repetitive structure. In: Lamas, D., Loizides, F., Nacke, L., Petrie, H., Winckler, M., Zaphiris, P. (eds.) INTERACT 2019. LNCS, vol. 11747, pp. 475–495. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29384-0_29
6. Coleman, J.M.: The use of music to promote purposeful movement in children with visual impairments. J. Vis. Impair. Blindness **111**(1), 73–77 (2017)

7. Dancu, A., et al.: Emergent interfaces: constructive assembly of identical units. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA 2015, pp. 451–460. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2702613.2732509
8. Horn, M.S., Jacob, R.J.K.: Tangible programming in the classroom with Tern. In: CHI 2007 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2007, pp. 1965–1970. ACM, New York (2007). https://doi.org/10.1145/1240866.1240933
9. Horn, M.S., Solovey, E.T., Crouser, R.J., Jacob, R.J.K.: 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, CHI 2009, pp. 975–984. ACM, New York (2009). https://doi.org/10.1145/1518701.1518851
10. Hu, F., Zekelman, A., Horn, M., Judd, F.: Strawbies: explorations in tangible programming. In: Proceedings of the 14th International Conference on Interaction Design and Children, IDC 2015, pp. 410–413. ACM, New York, (2015). http://dx.doi.org/10.1145/2771839.2771866
11. Ishii, H.: The tangible user interface and its evolution. Commun. ACM **51**(6), 32–36 (2008). https://doi.org/10.1145/1349026.1349034
12. Kalelioglu, F.: A new way of teaching programming skills to K-12 students: Code. org. Comput. Hum. Behav. **52**, 200–210 (2015)
13. Kane, S.K., Bigham, J.P.: Tracking @Stemxcomet: teaching programming to blind students via 3D printing, crisis management, and all Twitter. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education, SIGCSE 2014, pp. 247–252 (2014). https://doi.org/10.1145/2538862.2538975
14. Kane, S.K., Koushik, V., Muehlbradt, A.: Bonk: accessible programming for accessible audio games. In: Proceedings of the 17th ACM Conference on Interaction Design and Children, IDC 2018, pp. 132–142. ACM, New York (2018). https://doi.org/10.1145/3202185.3202754
15. Kelleher, C., Pausch, R., Kiesler, S.: Storytelling Alice motivates middle school girls to learn computer programming. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2007, pp. 1455–1464. ACM, New York (2007). https://doi.org/10.1145/1240624.1240844
16. Ladner, R.E.: Design for user empowerment. Interactions **22**(2), 24–29 (2015). https://doi.org/10.1145/2723869
17. Morrison, C., et al.: Torino: a tangible programming language inclusive of children with visual disabilities. Hum.-Comput. Interact. (2018). https://doi.org/10.1080/07370024.2018.1512413
18. Resnick, M., et al.: Scratch: programming for all. Commun. ACM **52**(11), 60–67 (2009). https://doi.org/10.1145/1592761.1592779
19. Thieme, A., Morrison, C., Villar, N., Grayson, M., Lindley, S.: Enabling collaboration in learning computer programming inclusive of children with vision impairments. In: Proceedings of the 2017 Conference on Designing Interactive Systems, DIS 2017, pp. 739–752. ACM, New York (2017). https://doi.org/10.1145/3064663.3064689
20. Zhu, K., Dancu, A., (Shen) Zhao, S.: FusePrint: a DIY 2.5D printing technique embracing everyday artifacts. In: Proceedings of the 2016 ACM Conference on Designing Interactive Systems, DIS 2016, pp. 146–157. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2901790.2901792

21. Zhu, K., Chen, T., Han, F., Wu, Y.-S.: HapTwist: creating interactive haptic proxies in virtual reality using low-cost twistable artefacts. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, pp. 1–13. Association for Computing Machinery, New York (2019). Paper 693. https://doi.org/10.1145/3290605.3300923
22. Zhu, K., Nii, H., Fernando, O.N.N., Cheok, A.D.: Selective inductive powering system for paper computing. In: Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology, ACE 2011, pp. 1–7. Association for Computing Machinery, New York (2011). Article 59. https://doi.org/10.1145/2071423.2071497
23. Zhu, K., Ma, X., Wong, G.K.W., Huen, J.M.H.: How different input and output modalities support coding as a problem-solving process for children. In: Proceedings of the The 15th International Conference on Interaction Design and Children, IDC 2016, pp. 238–245. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2930674.2930697