Shape of Music: AR-based Tangible Programming Tool for Music Visualization

Qiao Jin
jin00122@umn.edu
Department of Computer Science and Engineering, University of Minnesota
Minneapolis, MN, USA

Danli Wang
danli.wang@ia.ac.cn
Institute of Automation, Chinese Academy of Sciences
Beijing, China

Haoran Yun
haoran.yun@upc.edu
ViRVIG Research Center, Universitat Politècnica de Catalunya
Barcelona, Spain

Svetlana Yarosh
lana@umn.edu
Department of Computer Science and Engineering, University of Minnesota
Minneapolis, MN, USA

ABSTRACT

Integrating music into Computer Science (CS) education can stimulate children's creativity, change the stereotypical perspective of CS, and encourage women, ethnic or cultural minorities involved in the Computer Science area. In this paper, we use Augmented Reality (AR) technology to design a tangible programming system - AR-MPro for children, acting as a bridge between programming and music. It allows children to create customized AR effects to visualize music with low-cost materials by constructing tangible program sequences. AR-MPro is expected to broaden participation in computing, and be more intuitive, intriguing and instructional to enrich children's creating and programming experiences.

CCS CONCEPTS

• **Human-centered computing** → **Systems and tools for interaction design**; **Mixed / augmented reality**; • **Social and professional topics** → **Children**; • **Applied computing** → *Interactive learning environments.*

KEYWORDS

Augmented Reality, AR, music, tangible programming, creation, children, computational thinking.

ACM Reference Format:
Qiao Jin, Danli Wang, Haoran Yun, and Svetlana Yarosh. 2023. Shape of Music: AR-based Tangible Programming Tool for Music Visualization. In *Interaction Design and Children (IDC '23), June 19–23, 2023, Chicago, IL, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3585088.3593872

1 INTRODUCTION

Programming learning helps children acquire knowledge of computer science (CS), learn problem-solving and design strategies, and develop logical and creative abilities. Most importantly, it can cultivate computational thinking (CT) abilities. Graphical user interface (GUI) and tangible user interface (TUI) offer flexible interfaces, helping children handle the code easier. In particular, TUI is better for catering to the habits of young children preferring the operation by holding toys in hand [6]. As demonstrated by the survey of computational kits designed for young children [30], physical features strongly influence how children perceive and use a computational kit. Tangible languages offer a sensory engagement to children [8, 31], have the potential to simplify the understanding of complex syntax, enhance the learning effect [17], and improve children's involvement in programming [18].

In recent years, research has integrated programming learning with other disciplines. Some added music components and investigated advanced methods for computational systems to support musical expression and interaction [23]. Program and music share many similar identities [3]. For example, a program is simply a set of instructions given to a computer to carry out a certain task or task. And a song is also a set of musical notes, read as instructions, played with the instrument. The repeated note can be seen as the loop function in the program. Bridging music and CS education is considered a good way for stimulating children's expression and creativity. In addition, there are implicit stereotypes relating to people in the field of computer science [2, 19]. They are often portrayed as nerds and socially distant from other people. However, introducing music to CS learning experience could drastically change students' perspectives on CS learning [11]. Moreover, it can encourage women and ethnic or cultural minorities who are under-represented in CS to become involved in the CS field [23].

In this paper, we present a new tangible programming system named AR-MPro, which helps children learn programming by visualizing a part of a song. We designed this system to enable youth to create their own music visualization by programming. Our goal is to engage children to learn basic programming knowledge, such as the parameter, loop or sequence, change students' perceptions of computing, and provide a transformative programming learning experience.

IDC '23, June 19–23, 2023, Chicago, IL, USA

ACM ISBN 979-8-4007-0131-3/23/06.
https://doi.org/10.1145/3585088.3593872

2 RELATED WORK

2.1 Tangible Creation and Programming Tools for Children

Several tangible creations and programming tools [30] have been developed for children. TanProRobot 2.0 [26], Electronic Blocks [28], MakerWear [10], TanPro-Kit [25], and KIBO [24] which have tangible I/O system and no digital screen. Through them, children can run, verify and correct the program with the observation of electronic devices (such as machine components). However, Merely relying on the physical interface (such as sensors, lighting, etc.) or voice feedback during programming shrinks the visual forms of feedback.

Augmented Reality (AR) technology provides new opportunities to optimize TUI-based computational toolkit by compensating for the lack of visual feedback. A real object with augmented interaction is considered superior to an abstraction of a system with a screen only in many aspects. Moreover, the use of interactive AR in education is conducive to attracting students' interest in the study, thereby improving study efficiency [13]. Researchers have applied AR technology in tangible programming. In Tiles that Talk [1], different tiles represent computer syntax, software libraries and hardware, showing the system design of a project through the camera. Thymio II [16] is an educational robot integrating augmented reality and learning computer concepts by its visual programming environment. ARScratch [20] is an AR authoring environment based on the Scratch programming platform, allowing children to create programs that mix real and virtual spaces. ARMaze [9] and Code Notes [22] are tangible computational thinking toolkits, on which programs are created by physical materials (wooden blocks or paper cards) and codes are processed using the AR-based mobile game. By providing students with tangible objects to manipulate and interact with in AR environments, these tools enable a more hands-on and immersive learning experience. Our study explores the potential of AR-based tangible programming tools in the field of music visualization. We are specifically interested in investigating the use of AR scaffolding in loop learning because it is a fundamental structure (repetition) in music composition, and it can be challenging for children to grasp as a programming concept [14].

2.2 Programming with Music for Children

Development of interdisciplinary programming platforms has become increasingly popular in the education area [15]. Music is viewed as a promising domain for integrating computational thinking, as both involve patterns, repetitions and identification of structures [5, 12]. While several programming tools that support music creation and visualization exist, they primarily use graphic user interfaces, such as ScratchJr [4] and ARcadia [11]. These tools may present challenges for younger children due to their limited experiences in computer or related infrastructure operations, such as keyboard and mouse. There are a few works that combine music and pure tangible programming language. Sabuncuoglu's work [21] presents an affordable and accessible tangible music platform for visually impaired children that aims to teach programming concepts through music creation. TuneTable[29] is a computational musical tabletop exhibit for children, which teaches coding concepts by making music using several sound samples and tangible blocks.

With particular considerations combining both AR-based tangible programming and music, we implement AR-MPro, to introduce programming concepts to children under 9 years old through musical stimulus and intuitive and instructive AR scaffolding. Unlike prior research in this area, which mostly focuses on constructing a sound sequence to create a music, our approach links musical notes with corresponding AR visual effects, providing a new way to introduce programming concepts to young children.

3 DESIGN AND IMPLEMENTATION

Augmented Reality for Musical Programming, or AR-MPro for short, is a music visualization and programming tool with AR technology designed for children. Our goal is to create a bridge between music and program, allowing children to use this tool intuitively, and to understand how to construct a program correctly using AR feedback. To achieve this goal we created a system that consists of four parts (Figure 1): *AR-MPro Application* on mobile devices that display the musical notes and visual feedback; *Code Tokens* which are used to construct a program; *Loop Prompter* helps children optimize the program; *Show Cards* which are used to identify the current musical track and show visualization effects.

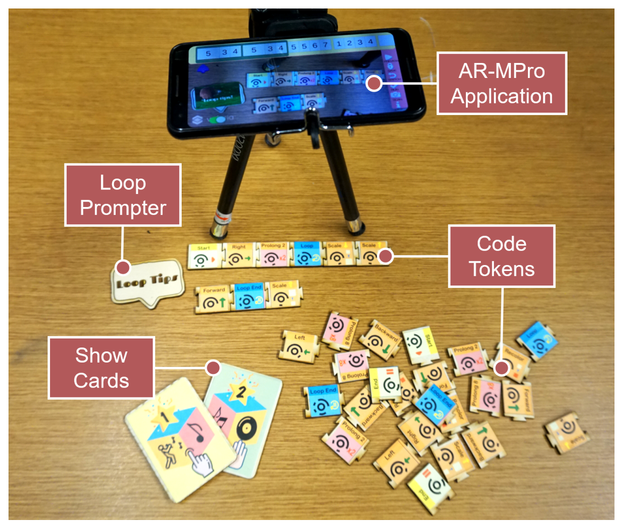

Figure 1: AR-MPro is composed of a series of code tokens, an AR-MPro application, a loop prompter and show cards.

3.1 AR-MPro Application

The AR-MPro application is implemented by Unity 3D and Qualcomm Vuforia platform which displays 3D scenes on the image targets (e.g., loop prompter and show cards). The sound material chosen for the musical aspect is piano, and the musical notes are presented through the Numbered Musical Notation (NMN) system [27], in which each note is denoted by a number representing its scale degree in a particular key. NMN simplifies the notation of traditional music, replacing complex characters and symbols,

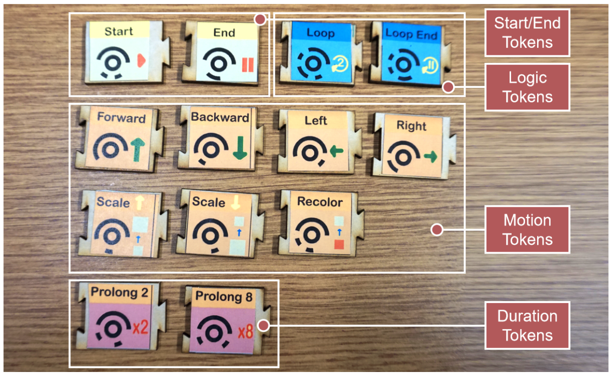

Figure 2: Four types of Code Tokens: Start/End, Logic, Motion, Duration.

and enabling non-musicians to learn and perform the music. The NMN representation will be displayed at the top of the screen, with varying lengths corresponding to different note durations. We highlighted the looped musical pieces in the musical track (Figure 3).

The application contains two stages, programming and running stage (Figure 4), corresponding to programming and code execution in traditional terms of computer programming. In the programming stage, after previewing the given notes of an episode, along with an arrow sign that marks the point to be programmed next, players are programming with Code Tokens. Each musical note must be linked to a motion token, with or without a duration token depending on the note length. Once the final program is created and tested, players are supposed to click the *switch* button or change the show card to switch to another soundtrack or save the current program and move to running stage. The running stage provides AR feedback on the program created by the player on the show card along with the related music. This visual scenery of a program can help players understand the execution process of their program. Users can also test their code after finishing a part of the program using the *scan* button in the system. This will allow the users to correct or debug their initial sequence using visual feedback (e.g., if logic tokens appear in pairs; if every note is linked to a motion token). The *tips* button gives the user one of the solution programs of a track in a graphic way. Users can refer to this to derive their own optimized solution.

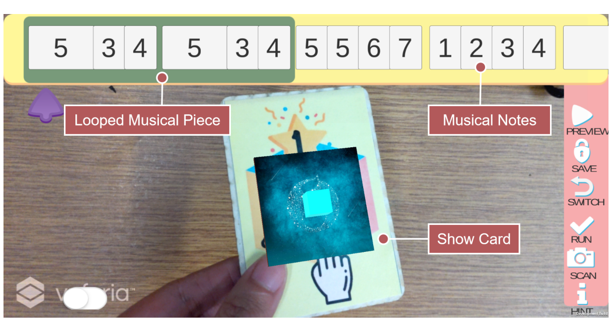

Figure 3: The virtual model shown on top of a Show Card.

3.2 Code Tokens

Code Tokens (Figure 2) are made using wood-based cards with TopCode [7] texture, which children can easily use to assemble a program on the table. To provide children with a more intuitive programming experience, we designed code tokens with different functionalities and colors. There are four types of code tokens: Start/End Tokens, Logic Tokens, Motion Tokens, and Duration Tokens. Each token is color-coded according to its category of functionality. Start/End tokens are used to indicate the start or end of the program. Logic tokens contain Loop tokens whose functionality is similar to the logic of "for" loop in a traditional programming language. Players should place them at the beginning or end of the token sequences, other tokens should be placed in the middle. Motion tokens are used to indicate the motion of the objects, which contain scale, color changing, moving, etc. Duration tokens control the duration of a note. Children are expected to recognize how long a note is played and use the prolong token accordingly.

3.3 Loop Prompter

Loop Prompter is a self-made wooden card used in the programming stage (Figure 5). This helps the user detect patterns or repetitive structures that can be optimized by using loop tokens. When loop prompter is recognized by the application, AR feedback will appear in each repeated and continuous program segments, letting users learn and use loop structure more intuitively.

3.4 Show Cards

Show Cards locates the place of the virtual AR model, which are also markers indicating different music track. Dual soundtrack can be triggered by two show cards in the running stage (Figure 6). AR-MPro application will detect and keep tracking the show card, then render the existing 3D models in the same location. AR 3D model will be updated and transform shape and position following the program accordingly which be coded before. During the execution, users are able to change the position or angle of the show card. It will not disrupt the flow of virtual model.

4 PILOT TESTING AND ONGOING WORK

To gather preliminary feedback and refine the system design, a pilot study was conducted with three university students who had no prior programming experience and musical knowledge. The pilot testing procedure involved introducing the participants to our AR-based tangible programming tool and asking them to use it to visualize two music tracks, both of which contained repeated musical pieces. We instructed them to construct their program on their own and then use the loop prompter to optimize their program. Throughout the 40-minute testing session, we observed their interactions with the AR-MPro and asked them to provide feedback on their experience (e.g., Were there any parts of the AR-MPro system that were difficult to understand or use? Did you find the loop prompter feature helpful in optimizing your program? Were there any features or functions you wished the AR-MPro had?).

Overall, we received positive feedback and summarized their feedback in the following. First, the one-to-one correspondence

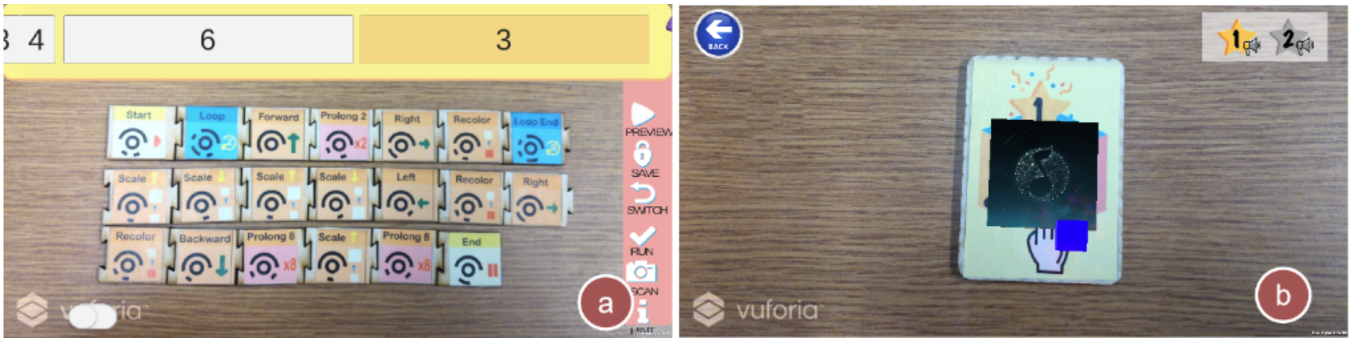

Figure 4: a) In the programming stage, users arranged Code Tokens which correspond to the musical notes. b) In the running stage, the visualized program is shown in the Show Card. When playing a musical note, which corresponds to a motion token (e.g., move forward, change color, scale up, etc.) with or without a duration token, the 3D model (in this case, a cube) displayed on the show card will perform the corresponding motion for the designated duration.

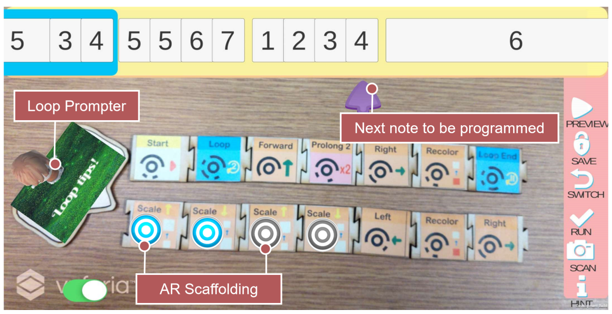

Figure 5: Loop Prompter could help users recognize repeated programming segments by showing AR Scaffolding.

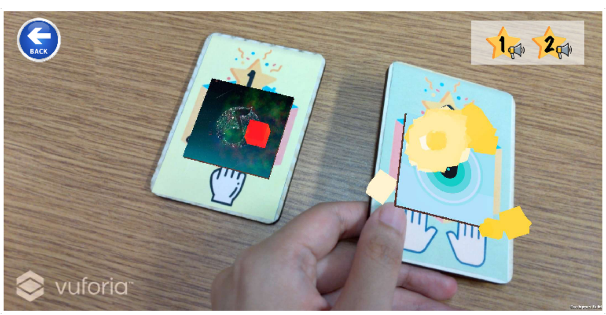

Figure 6: Dual soundtrack can be triggered by two show cards.

between the repeated musical piece and loop structure in programming is helpful for understanding the usage of the loop concept. Besides, the AR scaffolding and highlight UI elements are easy to understand. Finally, all of them think the AR and music visualization in the running stage makes the whole system attractive and enjoyable. They appreciated the free one-to-one mapping between code tokens and musical notes, which allowed them to create a unique AR visualization in the system. We also have captured some issues from the initial testing. For example, the duration of the note required fundamental musical knowledge, which might be hard to be understood by children. Then, the AR model and music are pre-defined and are not customized enough. Users may get tedious if keep working with the same materials. Participants suggested that future improvements could include customized visual effects, such as a wider variety of motion blocks and the ability to change AR models in show cards.

We plan to conduct a formal study with AR-MPro with a targeted group of children between the ages of 6 and 9. In the study, we will provide the participants with fundamental musical knowledge and include a detailed description of how the AR-MPro application functions. In addition to assessing task completion and ease of creation and programming, we will further explore the impact of music and AR scaffolding on the children's learning experience. Specifically, we will observe how children identify and create their programs, particularly for the loop structure construct, and how AR scaffolding helps them revise their programs. The results of this formal study will provide us with a deeper understanding of the technical opportunities for supporting more logical and complex musical visualization creation and programming experiences for children, as well as finding ways to develop their interest in computer science.

5 CONCLUSION

We present AR-MPro, a unique AR system designed to enable children to create their own music presentations using the tangible programming language. Our aim is to bridge music and programming, empowering children to explore the intersection of these two fields in an engaging and creative way. Based on our pilot testing, the system's AR scaffolding has the potential to make it easier for children to understand programming concepts while promoting creativity and self-expression. As its effectiveness with children has not yet been empirically validated, we plan to conduct a formal user study to further explore the user experience and the influence of AR scaffolding and music on children. Moreover, we aim to improve the system by integrating more customized elements such as tokens, display models (e.g., adding cartoon or animated figures), and AR effects.

ACKNOWLEDGMENTS

We express our sincere gratitude to all the participants who dedicated their time and efforts to participate in this study. We would also like to acknowledge Ye Yuan and Kathan Shah for their helpful support and feedback on this project.

REFERENCES

[1] David Bouchard and Steve Daniels. 2015. Tiles that Talk: Tangible Templates for Networked Objects. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 197–200.

[2] Sapna Cheryan, Victoria C Plaut, Paul G Davies, and Claude M Steele. 2009. Ambient belonging: how stereotypical cues impact gender participation in computer science. *Journal of personality and social psychology* 97, 6 (2009), 1045.

[3] C. Dobrian. 1988. Music Programming–An Introductory Essay. https://music.arts.uci.edu/dobrian/CD.MusicProgramming.htm.

[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*. 1–10.

[5] Jesse M Heines, Gena R Greher, and Sarah Kuhn. 2009. Music performamatics: Interdisciplinary interaction. In *Proceedings of the 40th ACM technical symposium on Computer science education*. 478–482.

[6] Scott Heller. 1998. The meaning of children in culture becomes focal point for scholars. *Chronicle of Higher Education* 44, 48 (1998), A14–A14.

[7] Michael Horn. 2012. TopCode: Tangible Object Placement Codes.

[8] Michael S Horn, Erin Treacy Solovey, R Jordan Crouser, and Robert JK 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*. ACM, 975–984.

[9] Qiao Jin, Danli Wang, Xiaozhou Deng, and Steve Chiu. 2018. AR-Maze: a tangible programming tool for children based on AR technology. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*. 611–616.

[10] Majeed Kazemitabaar, Jason McPeak, Alexander Jiao, Liang He, Thomas Outing, and Jon E Froehlich. 2017. Makerwear: A tangible approach to interactive wearable creation for children. In *Proceedings of the 2017 chi conference on human factors in computing systems*. ACM, 133–145.

[11] Annie Kelly, R. Benjamin Shapiro, Jonathan de Halleux, and Thomas Ball. 2018. ARcadia: A Rapid Prototyping Platform for Real-time Tangible Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, 409:1–409:8.

[12] Vishesh Kumar, Tuhina Dargan, Utkarsh Dwivedi, and Poorvi Vijay. 2015. Note Code: A Tangible Music Programming Puzzle Tool. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction* (Stanford, California, USA) *(TEI '15)*. Association for Computing Machinery, New York, NY, USA, 625–629. https://doi.org/10.1145/2677199.2688817

[13] Sang Hwa Lee, Junyeong Choi, and Jong-il Park. 2009. Interactive e-learning system using pattern recognition and augmented reality. *IEEE Transactions on Consumer Electronics* 55, 2 (2009), 883–890.

[14] Veronica Lin and Paulo Blikstein. 2018. Learning loops: Affordances and challenges of Project Bloks. International Society of the Learning Sciences, Inc.[ISLS].

[15] Yu-Tzu Lin, Ming-Tsan Wang, and Cheng-Chih Wu. 2019. Design and implementation of interdisciplinary STEM instruction: Teaching programming by computational physics. *The Asia-Pacific Education Researcher* 28, 1 (2019), 77–91.

[16] Stéphane Magnenat, Morderchai Ben-Ari, Severin Klinger, and Robert W Sumner. 2015. Enhancing robot programming with visual feedback and augmented reality. In *Proceedings of the 2015 ACM conference on innovation and technology in computer science education*. ACM, 153–158.

[17] Andrew Manches and Claire O'malley. 2012. Tangibles for learning: a representational analysis of physical manipulation. *Personal and Ubiquitous Computing* 16, 4 (2012), 405–419.

[18] Timothy S McNerney. 2004. From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal and Ubiquitous Computing* 8, 5 (2004), 326–337.

[19] Katarina Pantic, Jody Clarke-Midura, Frederick Poole, Jared Roller, and Vicki Allan. 2018. Drawing a computer scientist: stereotypical representations or lack of awareness? *Computer Science Education* 28, 3 (2018), 232–254.

[20] Iulian Radu and Blair MacIntyre. 2009. Augmented-reality scratch: a children's authoring environment for augmented-reality experiences. In *Proceedings of the 8th International Conference on Interaction Design and Children*. ACM, 210–213.

[21] Alpay Sabuncuoglu. 2020. Tangible Music Programming Blocks for Visually Impaired Children. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Sydney NSW, Australia) *(TEI '20)*. New York, NY, USA, 423–429.

[22] Alpay Sabuncuoğlu, Merve Erkaya, Oğuz Turan Buruk, and Tilbe Göksun. 2018. Code notes: designing a low-cost tangible coding tool for/with children. In *Proceedings of the 17th ACM Conference on Interaction Design and Children*. ACM, 644–649.

[23] R Benjamin Shapiro, Annie Kelly, Matthew Ahrens, Ben Johnson, Heather Politi, and Rebecca Fiebrink. 2017. Tangible distributed computer music for youth. *Computer Music Journal* 41, 2 (2017), 52–68.

[24] Amanda Sullivan, Mollie Elkin, and Marina Umaschi Bers. 2015. KIBO robot demo: engaging young children in programming and engineering. In *Proceedings of the 14th international conference on interaction design and children*. ACM, 418–421.

[25] Danli Wang, Yunfeng Qi, Yang Zhang, and Tingting Wang. 2013. TanPro-kit: a tangible programming tool for children. In *Proceedings of the 12th International Conference on Interaction Design and Children*. ACM, 344–347.

[26] Danli Wang, Lan Zhang, Chao Xu, Haichen Hu, and Yunfeng Qi. 2016. A tangible embedded programming system to convey event-handling concept. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 133–140.

[27] wikipedia. 2023. Numbered musical notation. Retrieved Mar 15, 2023 from https://en.wikipedia.org/wiki/Numbered_musical_notation

[28] Peta Wyeth and Gordon F Wyeth. 2001. Electronic blocks: Tangible programming elements for preschoolers. (2001).

[29] Anna Xambó, Brigid Drozda, Anna Weisling, Brian Magerko, Marc Huet, Travis Gasque, and Jason Freeman. 2017. Experience and Ownership with a Tangible Computational Music Installation for Informal Learning. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction* (Yokohama, Japan) *(TEI '17)*. Association for Computing Machinery, New York, NY, USA, 351–360. https://doi.org/10.1145/3024969.3024988

[30] 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*. ACM, 289–299.

[31] Oren Zuckerman, Saeed Arida, and Mitchel Resnick. 2005. Extending tangible interfaces for education: digital montessori-inspired manipulatives. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 859–868.