Young Children's Perceptions of Coding and Implications

Junnan Yu
Department of Information Science,
University of Colorado Boulder, Boulder, CO, USA;
School of Design, The Hong Kong Polytechnic University
Hong Kong, China
junnan.yu@colorado.edu

Ricarose Roque
Department of Information Science,
University of Colorado Boulder
Boulder, CO, USA
ricarose@colorado.edu

ABSTRACT

To inform the design of personally meaningful computational learning experiences for young people, it is necessary to investigate children's perceptions of coding, such that new learning experience designs can better leverage their funds of knowledge. We conducted focus groups with 20 young children in two coding workshops to learn their perceptions of coding. The young participants mainly form their perceptions of coding based on past coding- and computer-related experiences and typically associate coding with controlling the computer, creating projects, expressing ideas, playing video games, and the language for communicating with the computer. Importantly, young children might not know the right strategies to debug code but try practices like restarting the device and deleting the whole code. Based on the findings, we highlight the pedagogical implications to design productive computational learning experiences for young people.

CCS CONCEPTS

• **Social and professional topics → Computer science education**; **K-12 education**; **Computing literacy**; **Children**.

KEYWORDS

Young Children, Perceptions of Coding, Design Implications, Focus Groups

ACM Reference Format:
Junnan Yu and Ricarose Roque. 2022. Young Children's Perceptions of Coding and Implications. In *Interaction Design and Children (IDC '22), June 27–30, 2022, Braga, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3501712.3535285

1 INTRODUCTION

Many efforts have been focused on introducing coding and computational thinking to young people recently due to the widely acknowledged importance of computing [14], such as developing new computational kits [15, 16] and exploring various educational interventions [6]. However, learning to code has long been reported to be challenging for novices, and it is unclear what kind of coding interventions will generate more effective learning outcomes for what age groups of learners [11]. To inform the design of productive computational learning experiences for young people, it is necessary to investigate children's perception of coding so that educators can better leverage their positive conceptions, address their misconceptions, and finally help them form positive attitudes towards coding and computing. In this work-in-progress paper, we report young children's (ages 7-11) perceptions of coding from focus groups with 20 kids in a creative coding event. We examined what Computer Science (CS) and coding meant to the child participants, what they believed interesting about coding, and what was challenging about coding. Based on the findings, we highlight educational strategies to shape young learners' positive conception of computing and provide them with effective learning experiences.

2 RELATED WORK

Previous studies show that some children associate coding with computers [10] and children who have prior coding experiences recognize coding as giving commands to programmable devices like computers and toys [8, 12]. Among young children who have coding experiences, writing programs and *"understanding coding standards"* can be difficult [5]. Additionally, some young children (5-7 years old) connect *"code"* and *"coding"* to the pin codes for unlocking mobile devices or passwords for logging into a computer, while others associate *"program"* and *"programming"* with watching video programs (e.g., *"programming means that one watches some program"*) or reading manuals [8]. Some children also think of coding as manipulating media, e.g., considering collecting static and moving images and sound as programming [12]. On the other hand, many young people are unfamiliar with the term *"coding"* or *"computer programming"* and do not know what coding is [8, 13]. For example, around half of the children between 6-10 years old in [12] were unable to answer the question *"what are computer programs,"* nor did they know how computer programs were generated. Additionally, children rarely develop ideas about coding without previous involvement in programming experiences [8, 12]. In sum, not many studies have examined young children's perceptions of coding, and even fewer are up to date, i.e., Rucker and Pinkwart [10] in 2016 and Mertala et al. [8] in 2019 for the most recent ones, while other studies were conducted about fifteen to twenty years ago – Sheehan [12] in 2003, Li and Prasad [5] in 2005. As new computing technologies and CS pedagogies are consistently emerging, children's perceptions of coding may have changed significantly. As such, it is necessary to keep exploring how children perceive coding, especially for designing learning experiences that connect to young learners' *"funds of knowledge"* [9]. The current work continues this conversation and adds new understandings of how young children perceive coding.

IDC '22, June 27–30, 2022, Braga, Portugal
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9197-9/22/06.
https://doi.org/10.1145/3501712.3535285

Table 1: The participant groups in the two in-person workshops. We created code names (e.g., W1G1 for the participant Group One in Workshop One) to represent each group

Group name	Participants' sex and age	Group name	Participants' sex and age
W1G1 (Workshop 1 Group 1)	M (9), M (8), M (8)	W2G1 (Workshop 2 Group 1)	F (11), F (9)
W1G2	F (9), M (9), F (8)	W2G2	F (9), F (8), M (7)
W1G3	M (10), F (8)	W2G3	M (9), M (8), M (8)
W1G4	M (8), M (8)	W2G4	M (9), M (8)

3 METHOD

We report data from the focus groups with young children who participated in two in-person coding workshops where they were guided to create physical play projects using coding kits. The workshops were conducted in collaboration with a local museum in the Mountain West Region of the United States in February 2020. Each workshop involved ten children, 20 children in total for two workshops (7 females and 13 males). Participants were between 7 to 11 years old (average = 8.55) with the majority between 8-9 years old (N=17). They were recruited through the museum's weekly newsletters. All participants had coding experience from their schools, local public libraries, or homes using coding kits designed for children, such as Scratch, Sphero robot, and Hour of Code.

At the beginning of each workshop, we divided the children into small groups (Table 1, eight groups in total) and conducted focus groups to learn their perception of coding. The focus groups focused on how the participants perceived coding, e.g., what coding is, what is interesting about coding, and what is challenging about coding. We also asked what CS meant to them. The focus groups were audio-recorded (N=8), which were first transcribed, followed by thematic analysis [1] to reveal children's perceptions of coding by the first author. We identified themes such as *"unsure," "controlling," "creating," "language,"* and *"video games,"* which are further reviewed and refined by the second author, another experienced qualitative researcher. These themes are unfolded in the next section.

4 FINDINGS

4.1 What is CS

First, kids in five focus groups mentioned that they were ***"unsure"*** (W1G1, W1G2, W2G2) or ***"don't know"*** (W1G3, W2M3) what Computer Science (CS) is. Second, children in five focus groups **equaled CS to coding,** e.g., *"Computer Science is the coding"* (W1G2) and *"it [CS] is mainly programming"* (W2G2). Participants also mentioned other ideas of what CS meant to them, i.e., *"do[ing] science on the computer"* (W1G4), *"searching for things on a computer"* (W2G1), *"doing research on a computer"* (W2G2), *"figuring out how a computer works, and how the system is built"* (W2G2), and *"a computer degree you can get in a college"* (W1G3). In sum, young people might have no idea of CS, think of CS as coding, or interpret CS based on its literal meaning, such as doing science on computers or researching the computer system.

4.2 What is Coding

When asked about what coding is, kids in five focus groups mentioned ideas around **controlling** the computer or other objects, e.g., *"you give the computer some commands, then the computer will do it"* (W1G1) and *"programming something to do something"* (W1G1, W1G2). In these cases, coding functions as a language to **communicate** with the computer, i.e., *"[coding] is just like writing and talking, it's like you have to type a different language on computers"* (W1G4) and *"coding is like a computer language, you can tell a robot to do something with coding"* (W2G3). For those children, *"coding is instructions of telling the computer what to do"* (W2G3) and *"you are teaching the computer what to do, you are the teacher"* (W2G4). Five focus groups talked about coding as a way of **creating** digital media, such as creating video games (W1G1, W1G3, W1G4), websites (W1G3), and building animation projects (W2G2). Additionally, some kids in two groups were **unsure** what coding is and said, *"I don't know"* (W1G2, W2G3).

We further asked the kids the words that came to their minds when thinking of coding – seven groups mentioned programming languages and their associated concepts, e.g., JavaScript (W1G3), Scratch (W2G2), the loop concept (W2G4), numbers and letters (W1G2, W1G4, W2G1); five groups mentioned video games (W1G1, W1G2, W1G3, W1G4, W2G2); three groups mentioned robots (W1G1, W1G4, W2G1) and computer (W1G1, W1G3, W2G1); and one group mentioned science (W2G1) and art (W2G2), respectively. Notably, these concepts were mainly connected to their past coding-related experiences when asked why, e.g., *"creating video games on Scratch at school"* (W1G1) and *"my dad programmed a website"* (W1G3).

4.3 What is Interesting about Coding

First, kids in six focus groups shared the exciting part was that coding supported them to **create projects, express ideas,** and **see their coding results.** Some examples include *"creating something new"* (W2G2), the flexibility in creating (W1G2 – *"just adding stuff you want to"*; W1G3 – *"there are different tools you can use, like if-else, or stuff like that"*), as well as seeing and interacting with the final projects (W2G4 – *"when you are done coding, see what you are making your computer do, that's just really exciting"*). Second, four groups mentioned they enjoyed how coding supported them to **play video games,** primarily based on their past learning experiences with video games on coding platforms like Scratch and Hour of Code (W1G2, W1G3, W1G4, W2G2). Additionally, some kids mentioned **other ideas** of what they liked about coding, i.e., learning to code is *"like learning a new language... it's cool"* (W1G4), the rewarding feeling after solving coding challenges (W2G2), and *"coding a robot"* (W1G2).

The participants also shared ideas on what would be an interesting coding class for them. Five groups wanted more **gameful** coding experiences and supported the idea of **creating** video game projects, then **playing** them, e.g., *"making a game would make it [a coding class] more fun"* (W1G2) and *"to code a game and play once you code it"* (W1G3). Three groups talked about making coding classes more **active,** e.g., *"maybe like you are able to run around and exercise because the coding classes I have been to like you cannot do anything [active]"* (W1G2). Other kids expected to code **robots** (W2G3, W2G4) and wanted more **flexibility** in coding tasks, e.g., *"I wish you can choose whatever things you want to make... she[instructor] tells me exactly what to do, making the exact thing as someone else."* Overall, the participants enjoyed being able to create projects with coding and desired more playful learning experiences like playing video games and controlling robots.

4.4 What is Challenging about Coding

The most common challenge was centered around children's lack of coding knowledge and *"bugs."* Specifically, seven focus groups talked about the frustration when their codes were not working, e.g., *"[code] does not work"* (W1G1) and *"sometimes things go wrong... coding can be so frustrating"* (W1G2). When facing such challenges, some kids did not know appropriate **debugging strategies** and stated, *"you will have to restart the whole computer"* (W1G2), or *"you have to delete the code and do it again"* (W1G4). This challenge can be further ascribed to children's difficulties in figuring out coding tasks and understanding **coding commands,** e.g., *"when you are using like repeat, you can mess up"* (W1G3), *"I could not find the right [coding] puzzle pieces, so I just gave up"* (W2G2), and *"what's challenging for me is finding out how the [coding] pieces would go together"* (W2G3). **Other challenges** include coming up with project ideas (W2G1 – *"it's hard figuring out what we want to do, I have to think for a really long time"*), *"accidentally deleting something"* (W1G1), and *"getting them[robots] down to stairs"* (W2G1). In brief, understanding the meaning of coding syntax, employing them to achieve desired interaction outcomes, and troubleshooting bugs in code are the most significant challenges for young people.

5 DISCUSSION AND IMPLICATIONS

In this section, we discuss the implications of the above findings, the limitations, and our future work plans.

Some children's perceptions of coding identified in this paper align with those in previous studies, e.g., some young people regard coding as a way of controlling computers [8, 12], writing programs and understanding coding commands are reported to be challenging [5]. Our findings add that children with previous coding experiences may still feel unsure about coding. Additionally, we identify new conceptions of coding in young people – some children thought of coding as a way of creating projects and expressing ideas, as a language to communicate with the computer, or associated coding with numbers, letters, robots, and science. Such perceptions can be attributed to children's past computer and coding-related experiences [8, 12], e.g., programming a robot at home and creating video games in a coding class. These coding experiences shape how young people imagine what coding can be and how to learn to code. For example, our participants frequently mentioned video games when asked to think about fun ways to engage in coding. Therefore, it is crucial and necessary to design computational learning experiences that can cultivate young learners' positive perceptions and attitudes towards coding and CS, especially given that children as young as six years old may already form negative stereotypes about computing, e.g., boys are better at robotics and programming [7]. By designing coding experiences that are enjoyable and playful, it is promising to reshape how children think of coding, attract more children's interest in computing, and finally broaden their participation in computing. For example, some kids shared that they realized coding could be fun and combined with anything rather than just computers after participating in our physical-play-based coding workshops. However, what kind of coding experiences might be considered enjoyable and attractive for what groups of learners needs tailored exploration, e.g., e-textile-based computing can be interesting for girls from nondominant groups [2, 4].

A common challenge our child participants mentioned was that their code might often not work as expected, resulting in frustration during learning. Unfortunately, many young children did not know the effective strategies to figure out right solutions but tried practices like restarting the device and deleting the whole code. We all make mistakes when writing computer programs, especially for novice coders. As such, it is important to let young people know the commonness of *"bugs"* in programming and teach them to become patient with coding at the very early stage of exposure. Moreover, we should help young people become familiar with the concept of debugging and teach them effective debugging strategies, such that they are equipped with the tools to troubleshoot their code and figure out solutions even without adult scaffolding, rather than simply deleting the code or restarting the device.

Meanwhile, the kids shared how they enjoyed seeing their coding results, e.g., the movements of robots and animations on the screen. This kind of tangible coding experience may particularly benefit young learners as they can experience the outputs of coding visually or physically. Therefore, future coding kits and learning experiences designed for young children should prioritize perceivable and tangible interactions as well as immediate feedback. Finally, our findings also lend support to project-based learning in CS education for young people (e.g., [3]) because a fun coding class for many of our child participants would be creating projects like video games and coding robots. However, young people might not enjoy following strict coding guidance and working on the exact same projects as others, or do not know what to create. Accordingly, more coding-related learning experiences in the future can guide participants to create projects that allow space for children to express personal ideas and interests, while also providing appropriate scaffolding for those who are not sure what projects to pursue. Taken together, we hope our findings will inspire and contribute to the design of productive computational learning experiences for young people.

5.1 Limitations and Future Work

Our participants all had previous coding experiences. This recruiting result was not purposeful and may be attributed to the fact that most local elementary schools in the area offer coding courses to children, local informal learning spaces like museums often organize coding events for young people, and the recruitment happened through our community partner's email lists. Therefore, the findings may not speak for those who have no coding experience or whose coding experiences are different from our participants. Future research should recruit children without coding experiences, in other age ranges, and from different cultural backgrounds. Meanwhile, involving children with coding experiences can provide insights into the perceptions that may not emerge from young people without coding experiences, such as the challenges in coding and their debugging strategies. That being said, our findings are valuable for educators and researchers to inform computational learning experience designs grounded in young children's relevant experiences.

For the next steps, we plan to talk to young children who have no prior coding experiences and complement the findings reported in this paper. Specifically, instead of focus groups as reported in this paper, we plan to conduct individual interviews with children between 8-9 years old to narrow down the developmental range of the participants. In addition to coding, we also plan to investigate their perceptions of coding-related learning technologies like programmable robots.

ACKNOWLEDGMENTS

This project is supported by the CU Boulder Office for Outreach and Engagement. We thank the Museum of Boulder, especially the museum's education director Emily Zinn, for providing the space and helping recruitment for the workshop; the participants and their parents for supporting the workshop and our research practices; and the facilitators (Mariana Aki Tamashiro, Andrea DeVore, Joy Dale Weinberg, and Julisa Granados) for helping implement the workshops and collect data.

REFERENCES

[1] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[2] Deborah A Fields, Yasmin B Kafai, Tomoko Nakajima, and Joanna Goode. 2017. Teaching Practices for Making E-Textiles in High School Computing Classrooms. In *Proceedings of the 7th Annual Conference on Creativity and Fabrication in Education (FabLearn '17)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3141798.3141804

[3] Rita Francese, Carmine Gravino, Michele Risi, Giuseppe Scanniello, and Genoveffa Tortora. 2015. Using Project-Based-Learning in a mobile application development course-An experience report. *Journal of Visual Languages and Computing* 31 (2015), 196–205. https://doi.org/10.1016/j.jvlc.2015.10.019

[4] Gayithri Jayathirtha and Yasmin B Kafai. 2020. Interactive Stitch Sampler: A Synthesis of a Decade of Research on Using Electronic Textiles to Answer the Who, Where, How, and What for K–12 Computer Science Education. *ACM Trans. Comput. Educ.* 20, 4 (oct 2020). https://doi.org/10.1145/3418299

[5] Xiaosong Li and Christine Prasad. 2005. Effectively teaching coding standards in programming. In *Proceedings of the 6th conference on Information technology education (SIGITE '05)*. Association for Computing Machinery, New York, NY, USA, Newark, New Jersey, USA, 239–244. https://doi.org/10.1145/1095714.1095770

[6] James Lockwood and Aidan Mooney. 2018. Computational Thinking in Education: Where does it fit? *International Journal of Computer Science Education in Schools* 2, 1 (2018), 1–58.

[7] Caitlin K Martin, Nichole Pinkard, Sheena Erete, and Jim Sandherr. 2016. Connections at the family level: Supporting parents and caring adults to engage youth in learning about computers and technology. In *Moving Students of Color from Consumers to Producers of Technology*. IGI Global, 220–244. https://doi.org/0.4018/978-1-5225-2005-4.ch011

[8] Pekka Mertala. 2019. Young children's conceptions of computers, code, and the Internet. *International Journal of Child-Computer Interaction* 19 (2019), 56–66. https://doi.org/10.1016/j.ijcci.2018.11.003

[9] Luis C Moll, Cathy Amanti, Deborah Neff, and Norma Gonzalez. 1992. Funds of knowledge for teaching: Using a qualitative approach to connect homes and classrooms. *Theory Into Practice* 31, 2 (mar 1992), 132–141. https://doi.org/10.1080/00405849209543534

[10] Michael T. Rücker and Niels Pinkwart. 2016. Review and Discussion of Children's Conceptions of Computers. *Journal of Science Education and Technology* 25, 2 (2016), 274–283. https://doi.org/10.1007/s10956-015-9592-2

[11] Ronny Scherer, Fazilat Siddiq, and Bárbara Sánchez Viveros. 2020. A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in Human Behavior* 109, 0318 (2020). https://doi.org/10.1016/j.chb.2020.106349

[12] Robert Sheehan. 2003. Children's perception of computer programming as an aid to designing programming environments. In *Proceedings of the 2003 conference on Interaction design and children (IDC '03)*. Association for Computing Machinery, New York, NY, USA, Preston, UK, 75–83. https://doi.org/10.1145/953536.953548

[13] Mike Van Duuren, Barbara Dossett, and Dawn Robinson. 1998. Gauging Children's Understanding of Artificially Intelligent Objects: A Presentation of "Counterfactuals". *International Journal of Behavioral Development* 22, 4 (1998), 871–889. https://doi.org/10.1080/016502598384207

[14] Jeannette M. Wing. 2006. Computational Thinking. *Commun. ACM* 49, 3 (2006), 33–35.

[15] 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)*. ACM, Trondheim, Norway, 289–299. https://doi.org/10.1145/3202185.3202738

[16] Junnan Yu and Ricarose Roque. 2019. A review of computational toys and kits for young children. *International Journal of Child-Computer Interaction* 21 (2019), 17–36. https://doi.org/10.1016/j.ijcci.2019.04.001