

It's Challenging but Doable: Lessons Learned from a Remote Collaborative Coding Camp for Elementary Students

Yingbo Ma¹, Julianna Martinez Ruiz¹, Timothy D. Brown¹, Kiana-Alize Diaz¹, Adam M. Gaweda²,

Mehmet Celepkolu¹, Kristy Elizabeth Boyer¹, Collin F. Lynch², and Eric Wiebe²

yingbo.ma@ufl.edu, juliannamartinez@ufl.edu, timothy.brown@ufl.edu, kianaalizediaz@ufl.edu, agaweda@ncsu.edu

mckolu@ufl.edu, keboyer@ufl.edu, cflynch@ncsu.edu, wiebe@ncsu.edu

¹ University of Florida, Gainesville, Florida

² North Carolina State University, Raleigh, North Carolina

ABSTRACT

The COVID-19 pandemic shifted many U.S. schools from in-person to remote instruction. While collaborative CS activities had become increasingly common in classrooms prior to the pandemic, the sudden shift to remote learning presented challenges for both teachers and students in implementing and supporting collaborative learning. Though some research on remote collaborative CS learning has been conducted with adult learners, less has been done with younger learners such as elementary school students. This experience report describes lessons learned from a remote after-school camp with 24 elementary school students who participated in a series of individual and paired learning activities over three weeks. We describe the design of the learning activities, participant recruitment, group formation, and data collection process. We also provide practical implications for implementation such as how to guide facilitators, pair students, and calibrate task difficulty to foster collaboration. This experience report contributes to the understanding of remote CS learning practices, particularly for elementary school students, and we hope it will provoke methodological advancement in this important area.

CCS CONCEPTS

• Social and professional topics → K-12 education.

KEYWORDS

Elementary CS Education; Remote Learning; Collaborative Learning

ACM Reference Format:

Yingbo Ma¹, Julianna Martinez Ruiz¹, Timothy D. Brown¹, Kiana-Alize Diaz¹, Adam M. Gaweda², Mehmet Celepkolu¹, Kristy Elizabeth Boyer¹, Collin F. Lynch², and Eric Wiebe². 2022. It's Challenging but Doable: Lessons Learned from a Remote Collaborative Coding Camp for Elementary Students. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2022)*, March 3–5, 2022, Providence, RI, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3478431.3499327>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGCSE 2022, March 3–5, 2022, Providence, RI, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9070-5/22/03...\$15.00

<https://doi.org/10.1145/3478431.3499327>

1 INTRODUCTION

Providing CS learning opportunities to elementary school-aged students can help them build a strong foundation in CS and can increase their interest in computing and related fields [7, 25]. Collaborative computing has been emphasized as an important practice in the new K-12 Computer Science Framework [6]. For many younger learners, CS learning opportunities have been part of in-person classroom settings and after-school programs [9, 20], and have been structured as collaborative learning [27]. Collaborative learning has been shown to be beneficial for younger learners to improve learning performance [14, 23] and develop group-work skills [24, 27]. However, the COVID-19 global pandemic, and the sudden shift to online learning, have disrupted many of these learning opportunities [15, 22], presenting particularly acute challenges for collaborative CS learning activities for elementary school students.

Many challenges of remote learning are not specific to collaborative learning: A recent study evaluating elementary students' perception towards remote learning highlighted several challenges including communication difficulties due to slow and insufficient network bandwidth, and a lack of knowledge on how to set up and manage devices [1]. Moreover, in light of the isolation that students have experienced during the COVID-19 pandemic, the socialization provided by collaborative learning has become even more important, including stimulating higher-order thinking skills [10, 16] and enhancing social interaction skills [8, 24]. It is important for CS instructors to carefully plan and coordinate remote collaborative learning activities to prevent students from feeling isolated and maintain the benefits of collaboration.

In this experience report, we describe a three-week remote after-school coding camp implemented with 24 elementary school students participating in a series of CS learning activities. This report's contributions are two-fold. First, we describe the structure of our camp including recruitment processes, learning activities, and the facilitation model for others to replicate our efforts. Second, we present insights from four camp facilitators, including challenges they encountered, and recommendations for future changes. We conclude with a series of practical implications centered around how to guide facilitators, pair students, and calibrate task difficulty to foster collaboration. We hope that these insights are helpful for researchers and practitioners in the CS education community to understand the challenges in remote collaborative CS learning activities, as well as providing a strategic way to tackle them to improve learning and foster collaboration.

2 RELATED WORK

There have been many recent efforts to introduce CS concepts to elementary students. For example, Martínez et al. [17] implemented CS learning activities with a group of 135 elementary school students (ages 8-11), and found that students at this age could learn fundamental CS concepts such as conditionals, loops, and parameters. Nche [18] used a custom video game to teach 40 elementary students code tracing and the results of this study highlighted the potential of games to improve students' computational thinking skills. Jin [11] taught elementary students the concepts of loops with structured learning activities. Their results suggested that understanding and applying conditional loops still remained challenging to elementary students. Lee et al. [12] explored enhancing coding instruction in school libraries and media centers. Their results showed that these activities increased some students' intrinsic motivation significantly. The aforementioned studies were all conducted in the classroom. However, informal learning (e.g., coding camps) are under-investigated for elementary students, with most prior coding camps designed for middle and high school students. Bryant et al. [3] conducted a five-day coding camp for middle school students and found that the camp successfully increased students' self-efficacy and perceptions of computer science. Similarly, LePendu et al. [13] created a one-week summer coding camp for high-school students and found that the coding camp is an effective informal learning experience that facilitates students' self-efficacy and engagement.

There have also been efforts to introduce collaborative CS learning to elementary students in the form of pair programming. In pair programming, two students collaborate on coding tasks by taking two roles: *driver* and *navigator*. The *driver* is responsible for writing the code and the *navigator* helps in catching mistakes and providing feedback [4]. In a pair programming study, Shah et al. [21] conducted a summer enrichment program with 45 elementary school students who engaged in a series of coding tasks. They found that inequities emerged in the dyads as students positioned themselves and their partners as more or less competent in computer science. Tsan [26] analyzed collaborative dialogues of elementary students completing pair programming activities in classrooms, and identified a set of dialogue moves during collaborative learning, such as self-explanation, questions, uptake, praise, and antagonization. The current experience report builds on this body of work by reporting how elementary students experience remote collaborative CS learning and providing implications for future collaborative remote studies.

3 REMOTE CS LEARNING CAMP

This after-school camp was conducted as part of a broader multi-year project to investigate and support collaborative CS learning in upper elementary school (fourth and fifth grade) with virtual learning companions (animated characters that promote good collaboration). Our goals with the camp were to provide a high-quality learning experience to the students and collect data from their interactions with our system for subsequent analysis and system refinement.

3.1 Recruiting Participants and Scheduling Facilitators

We recruited students for the camp through an online flyer which listed the camp dates and included a link to an online registration form. The flyer stated that the camp was targeted at fourth and fifth grade students, was cost-free, and required no prior coding experience*. When signing up, we asked prospective parents to select from a range of potential camp dates and times. We then assigned the students with an eye to balancing the camp schedule and maintaining stable groups. We worked with the local school district and interested teachers to send the flyer to parents of appropriately-aged students. We also distributed the information through Facebook and emails to parent networks.

Our recruitment efforts began four weeks before the camp's start date. Initially, we received very little response, but then within two weeks we received over 129 applications. Determining our maximum capacity for an online camp was a nontrivial task. After we identified the most popular times and dates for our applicants, we polled our facilitators, who included one research scientist, one Ph.D. student, two full-time research staff, and one undergraduate student. Ultimately, we were able to accept 24 students (14 male, 9 female, and 1 who declined to report their gender). The mean age of the students was 9.8 (ranging from 9 to 11) and 75% of them had some prior coding experience (e.g., Code.org [5]).

To schedule the facilitators, we assigned each student pair a facilitator who supported their learning activities throughout the camp. However, our university's risk assessment and youth compliance policies required that there should be at least two adults on Zoom interactions with students. To comply with this requirement we assigned a rotating facilitator to periodically visit each group and provide assistance when needed.

3.2 Camp Structure and Setting

3.2.1 FLECKS Coding Environment. The learning activities for this study were conducted using the FLECKS environment[†], which extends Netsblox [19]. FLECKS (Figure 1) includes two pedagogical virtual agents designed to foster collaborative learning experiences by modeling good collaboration practices for children and helping them learn about resolving potential conflicts with their partners. The block-based coding environment provides a set of CS learning activities, and students work on the learning activities synchronously.

3.2.2 Zoom Remote Meeting Environment. We used Zoom [30] for students to participate in learning sessions remotely. Each after-school session began in the main Zoom room. After all the students in the session had logged into the main room, each pair of students was redirected to their respective breakout room for collaborative coding activities. Before the activities, students were familiarized with the Zoom interface and the facilitators were available with continuous support for navigating it. Each of the zoom meetings was recorded to the cloud to support data management and analysis, and to avoid the security risks associated with local storage. We enabled the "multiple participants can share simultaneously" setting

*<https://flecks.csc.ncsu.edu/school-programs>

†www.flecksproject.org/

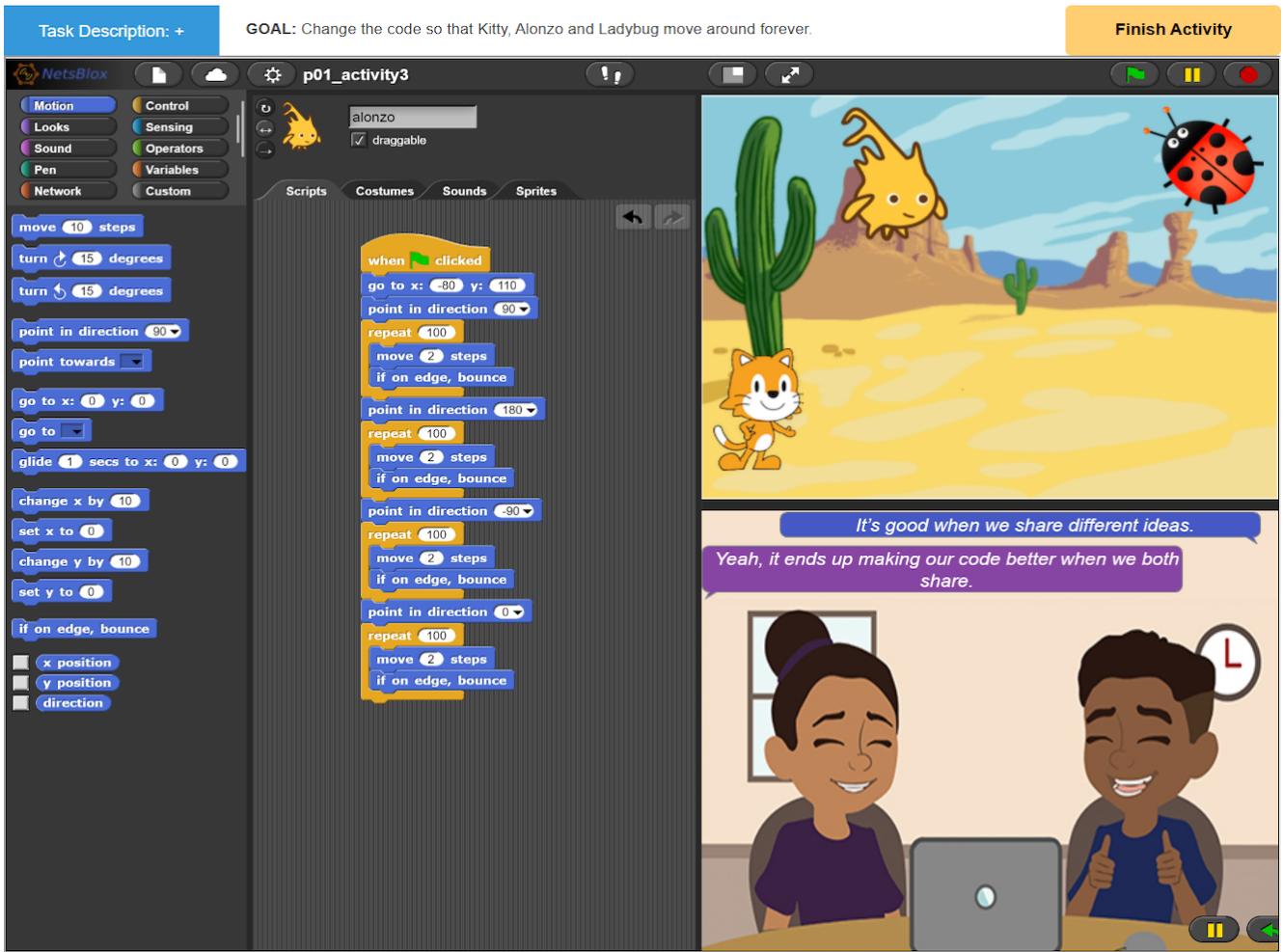


Figure 1: The interface of the FLECKS coding environment used for the CS learning activities during the camp.

on Zoom to allow both students to share their screens at the same time. To support future research, we also used Zoom to record the activity screens, students' face camera, and their conversations. All of these processes were fully approved by the university's Human Subjects Review Board (IRB) and we obtained parental consent and student assent for every participant.

3.3 Remote CS Learning Activities

The after-school CS learning experience was a fully virtual three-week after-school coding camp for 4th and 5th grade students, with the option of Monday/Wednesday sessions or Tuesday/Thursday sessions. Two different time slots were available for both sessions: 2-3 pm and 4-5 pm. Each student attended our camp twice a week for 3 weeks, or 6 sessions in total (Table 1). Each session was completed by a student pair supported by a facilitator (Figure 2). This section provides a detailed description of each remote learning session.

Day 1/Week 1: The day 1 activity started with welcoming students, taking the CS assessment test consisting of 10 block based programming questions [29], and completing a CS attitude survey

consisting of 11 questions [28]. The goal of the pre-data collection was to capture students' existing CS knowledge and CS attitudes before they participated in the learning activities. After the pre-data collection, we asked students a set of questions to understand how much they knew about computer science or coding. Some of the questions were: *Have you done any coding before? If yes, what? If not, do you have any coding knowledge?* While the majority of students claimed to have some experience with coding, those that

Table 1: After-School CS Camp Activities

DAY	CAMP ACTIVITIES
1	Introduction, Pre Test, & Blockly Game
2	Intro to the Block-Based Environment, & Zoom Practice
3	Pair Programming on Program Flow and Loops
4	Pair Programming on Loops and Conditionals
5	Pair Programming on Loops and Nested Conditionals
6	Post Test and Interviews

claimed not to have any experience were assured that they were going to learn about coding in the camp. Next, we asked about their experiences with Zoom and made sure that students knew how to share their screen and computer sound, since part of the camp experience involved synchronously watching cut-scenes of the pedagogical virtual agents talking. Next, we continued with ice-breaking Blockly games [2] to increase students' excitement, familiarize them with coding and also prepare them for the transition to block-based programming for future learning activities.

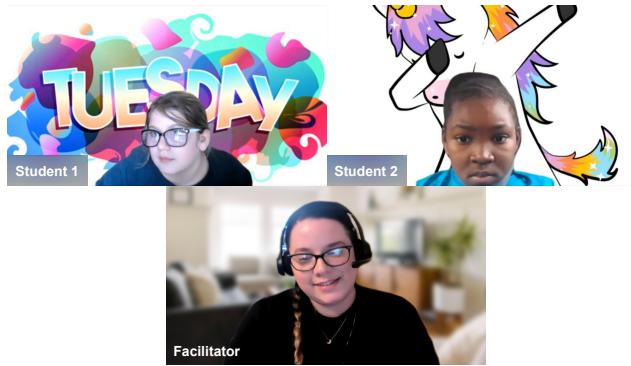


Figure 2: Student pair collaborating on a learning task, guided by a facilitator.

Day 2/Week 1: The day 2 activities focused on exploring the block-based coding environment, and learning about several programming concepts such as program flow and loops. The facilitator gave a demonstrative lesson while frequently checking students' screens to ensure they were following along. Both the students and facilitators started with an empty project, and built a sample code with basic coding blocks such as motion and loops. The students learned about sprites (costumes students needed to control with their code); they were also encouraged to download and import their favorite images (e.g., animal or car) into the coding environment's interface and make them move around.

Day 3/Week 2: The day 3 activity began with students collaborating on the code following the pair programming paradigm, which brought additional challenges. The facilitator informed students they would be working together and explained the roles of *driver* and *navigator*. The *driver* student controlled the activity and shared their screen and sound with the other student. During the activity, students watched a cut-scene of virtual agents that articulated the importance of *Asking Why Questions* while debugging a coding problem. Then, the system instructed students to debug broken code for a sprite, so that it moved in a square pattern. After fifteen minutes of coding, the facilitator turned their video back on, and asked students to switch *driver* and *navigator* roles. Once student pairs finished the activity, they were asked to complete a quiz (developed internally by our team) individually, which consisted of five coding questions related to the learning activity.

Day 4/Week 2: The day 4 activity began with a review of the "Sprites" section and taught students about loops and conditionals. In addition to "Sprites", students reviewed the "Point in Direction" coordinate, and the "Forever" block that would help them on the

day's activity. During the activity, students watched a cut-scene of the virtual agents that highlighted the importance of *Sharing Ideas*. Next, students were asked to use blocks that were previously reviewed to make three sprites move forever in different patterns (back and forth, in a square, and randomly). The question of what constitutes a "random movement" prevailed through many pairs. Students had a tendency to experiment with different kinds of movements until one seemed "random enough." The trial and error process made good use of the allotted time for the activity, though as with the previous day, some sessions went over time. Again, when student pairs finished the activity, they were asked to complete a quiz individually, which consisted of five coding questions related to the learning activity.

Day 5/ Week 3: The day 5 activity began with a review of the "Change size" and "If, else" blocks. Next, students watched a cut-scene of the virtual agents that highlighted the importance of *Listening to Each Other's Suggestions*. Next, students were asked to make three sprites talk when they came into contact with each other. This activity required the use of the "If, else", "say" and "Change size", blocks all covered previously. Similar to the last two learning sessions, student pairs were asked to complete a quiz individually, which consisted of five coding questions related to the learning activity.

Day 6/Week 3: The last day's activity began with taking the CS assessment test and completing a CS attitude survey to assess the change in students' coding knowledge gains and CS attitude after the camp. After the data collection, students were given a list of Snap! [‡] games to peruse in their own time, and were encouraged to pursue other CS-related programs to expand their knowledge.

3.4 Data

Students completed a demographics survey, a validated CS attitude survey [28] and a CS learning assessment test [29] prior to and after participating in the camp. Students were also asked to complete a learning assessment test after each collaborative coding activity (Days 3, 4, and 5).

Camp facilitators took field notes during students' collaborative learning activities and met after each session to debrief the team regarding the problems students faced during problem solution as well as other difficulties caused by remote collaboration. Finally, the facilitators independently completed an open-ended post survey after the camp ended, reporting their experiences of implementing the learning activities with students remotely. The facilitator post survey had the following open ended questions:

- (1) What did you think went well?
- (2) What did you find challenging? What would you not do again?
- (3) What kinds of CS concepts did students struggle with when they were doing the activities?
- (4) What are your suggestions for the improvement of future remote after-school coding camp?

[‡]<https://snap.berkeley.edu/>

4 REMOTE CS CAMP OUTCOMES

This section presents the quantitative analysis of students' responses to the CS attitude survey and CS learning assessments, and the qualitative analysis of the facilitator observations.

4.1 Quantitative Outcomes

We utilized a paired-samples *t*-test to analyze how children's CS knowledge scores changed before and after the remote CS learning camp. The average pre CS knowledge score was 6.32 out of 10 ($SD=1.78$) and the average post CS knowledge score was 6.27 ($SD=2.3$). This difference is not statistically significant based on a paired-samples *t*-test ($t(21)=.095$, $p=0.93$, $d=.024$).

Next, we compared how children's CS attitude survey scores changed before and after the remote CS learning camp. The average pre CS attitude survey score was 45.1 out of 11 ($SD= 4.87$) and the average post CS attitude survey score was 44.5 ($SD=7.26$), and this difference is not statistically significant based on a paired-samples *t*-test ($t(21)=.69$, $p=.5$, $d=.13$).

We also analyzed students' learning test scores after they participated in the collaborative coding activity with their partners (Days 3, 4, and 5). Students' average score for Day 3 was 3.7/5, for Day 4 was 3.4/5 and for Day 5 was 4/5.

4.2 Facilitator Observations

This section presents the results of the facilitator responses to the open-ended questions about their experiences during the camp.

4.2.1 What Went Well.

Successful Recruitment of Participants: The number of applicants for the camp far exceeded the team's expectations. The facilitators agreed on the series of practices used for recruiting participants that led to this success, including dispersing flyers in local school districts via email, posting information about the camp on social media, and creating a Google form for convenient online registration for potential applicants.

Well-organized Camp Structure: The facilitators appreciated the detailed schedule, updated daily, assigning them to camper pairs. They also mentioned that having a step-by-step guide for each activity helped them support the learning activities more easily.

Successful Collaborative Problem Solving: The facilitators observed that the students were open to communicating with each other even though they did not know each other before the camp. Not surprisingly, the students were also eager to engage in more social discussions. For example, one facilitator reported:

"They were both kind of eager to know about the other person, they wanted to have time not necessarily coding but just talking and learn [sic] about each other."

4.2.2 Observed Challenges.

Technical difficulties: The remote camp made it difficult to give immediate assistance when students experienced technical issues, such as issues due to slow internet connection or hardware problems. These issues sometimes introduced additional disruptions for the learning activities and sometimes forced learning sessions to run overtime.

Different Knowledge Levels within Student Pairs: Facilitators reported that the significant difference in students' previous CS knowledge levels often led to imbalanced learning opportunities. Students with less CS knowledge sometimes fell behind when their partners finished the coding tasks too quickly. One facilitator said:

"Student [sic] with much more coding experience tended to finish tasks quickly, while their partners became frustrated easily because they wanted clearer explanations."

Late Students: Some students were frequently late and there was no mechanism to make sure all students were available at the scheduled time. The facilitators sent emails to the parents when a student was late for more than five minutes; however, this situation created additional challenges for the students who were on time as they had to wait for their partners to start the learning activity. A facilitator reported:

"The worst day for me was waiting 37 minutes for a student while their partner just sat and waited ... If the child was late or didn't take the coding tasks seriously, it made the process difficult and unpleasant for the other child as well as the facilitator."

4.2.3 Suggestions for Future Remote Camps.

Additional Guidelines for Facilitators: All facilitators recommended additional guidelines for facilitators to handle unexpected situations, such as connectivity issues and students being late. One facilitator wrote:

"We need a protocol to prevent any technical issues and avoid students being late. We need to protect the attending students' experience if their partners are not showing up."

Greater Level of Difficulty of Coding Activities: Facilitators suggested to provide more difficult coding tasks for the students with prior programming experience. This could decrease boredom, increase engagement, and foster collaboration. One facilitator said:

"If students knocked these tasks out very quickly, they didn't have to collaborate. Collaboration happens when coding becomes difficult."

Extra Period of Time for Kids to Break the Ice: Facilitators suggested allocating a small period of time before each coding activity (up to 10 minutes) for students to warm up. Meanwhile, this period of time could be used for checking the possible technical issues. One facilitator wrote:

"It is important to do this (ice-breaking) because I observed my students being nervous or anxious. The existence of their partners was more like a competition instead of a collaboration."

5 DISCUSSION

The overarching goal of this experience report was to provide insights for CS researchers and practitioners to implement remote collaborative CS learning activities with elementary school students. Quantitative analysis of students' pre- and post-test results showed that neither their CS knowledge nor their CS attitudes changed significantly over the course of the camp. However, further analysis

of the data indicated that some students did not take the post-test seriously. While students took an average of 14.4 minutes to complete the 10 questions on the pre knowledge assessment, they spent significantly less time on the post knowledge assessment, an average of 9.4 minutes. For example, one student spent much less time (3.1 minutes) than average (9.4 minutes) finishing the post test and scored only 2/10 compared to the average score of 6.4/10. Another student scored 8/10 in the pre knowledge assessment test, but their score dropped to 2/10 in the post test score, even though this student got one of the highest scores in the learning assessment tests after each collaborative learning task. We believe the post-test should have been shortened to address this problem.

The facilitator reflections pointed out that the initial preparation and planning allowed the team to reach a high number of applicants and made the recruitment process much easier. In particular, receiving additional help from the local school district and sending flyers to parents through email lists made more people interested in the camp. Moreover, most parents probably wanted their children to participate in learning activities that differed from their daily course work, as the children did not have many extracurricular activity options during the pandemic. During the learning activities, facilitators noticed that most students were engaged in the collaborative activities and enjoyed interacting with other students.

On the other hand, there were various challenges that the facilitators faced during the planning of the remote learning activities. One of the main challenges was to coordinate the times and availability for student pairs and facilitators, because university policy required that there should be at least two adults available on Zoom interactions with students. Some of the facilitators had to lead more than one session on each day.

Another challenge for the facilitators was helping children when they struggled with computer tasks such as opening a browser or locating downloaded files. To mitigate these problems, the facilitators often discussed the common technical issues the students and the facilitators faced during the learning activities and shared experiences on how they solved these issues. In the end, the facilitators were able to create a step-by-step guideline for facilitators to assist students to tackle technical issues in the future.

Some facilitators reported difficulty with motivating students to stay focused during the collaborative learning tasks. In particular, when the students were holding the *navigator* role, they acted more passively and sometimes did not pay attention to their partner's actions on the coding interface. One partial solution was to ask both students to share their screens at the same time, which reduced off-task and multitasking behaviors by navigators. However, some students seemed bored when they were not actively coding and had to watch their partner who was controlling the activity.

We randomly assigned partners to teams based on their best available times to participate in the camp, and this situation led to a significant problem of having pairs of students with different knowledge levels. The facilitators reported that working with a partner with far more coding experience discouraged collaboration between some partners.

Finally, some students were often late to the activities, which led the team to send a reminder email to all the parents before each session. Although this step prevented some students from being too late, it increased the workload of the facilitators.

6 IMPLICATIONS

This section summarizes several implications for CS researchers and practitioners to implement remote collaborative CS learning activities with elementary school students:

- **Recruitment applications** should collect information about the prospective students that will help camp organizers admit diverse students. Note that random selection or first-come-first-served often perpetuates systems of marginalization of girls and Black, LatinX, and Native American learners.
- **Having a direct communication channel to parents** can save time for the facilitators, as students sometimes forget to join the learning sessions on time. Asking for parent phone numbers and permission to text message them may require special IRB approval, but it may significantly reduce no-shows or late students.
- **Preparing detailed camp guidelines** helps facilitators to easily support activities, handle common and unexpected issues with a quick procedure, and ensure smooth collaborative learning sessions.
- **Forming groups based on students' experience levels** can help students make progress at the same pace during collaborative coding activities, hence creating more balanced learning opportunities within the student pair.
- **Calibrate difficulty of coding activities** to help students with different experience levels find appropriate challenges, and to increase their engagement with each other.
- **Shorter survey and assessments** can keep students motivated and ensure they take these surveys and assessments seriously.

7 CONCLUSIONS AND FUTURE WORK

This experience report described a remote collaborative CS learning camp implemented with 24 elementary school students over three weeks. Despite the growing interest in CS education for younger learners, there is a lack of literature around implementing remote learning experiences for elementary students, and there are currently no practical guides or documentation on how to implement these learning activities. In this experience report, we presented a detailed description of our implementation of an after-school coding camp, including activity design and participation recruitment, as well as the preparation of each activity and students' data collection before, during, and after the camp. In addition, we described the challenges facilitators experienced and presented potential implications for future studies, which are centered around how to recruit students, guide facilitators, pair students for collaborative learning tasks, and calibrate task difficulty to foster collaboration. We hope this work can inform researchers and practitioners in the CS education community to conduct such remote learning experiences with young learners, and provoke methodological advancement for research.

8 ACKNOWLEDGMENTS

This research was supported by the National Science Foundation through grant DRL-1721160. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the authors, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

REFERENCES

[1] Ach Baidowi. 2020. The Evaluation of Online and Face to Face Physical Education Learning Policy at Elementary School Level. *Tegar: Journal of Teaching Physical Education in Elementary School* 4, 1 (2020), 35–44.

[2] Blockly. [n.d.]. <https://blockly.games/>.

[3] Caelin Bryant, Yesheng Chen, Zhen Chen, Jonathan Gilmour, Shyamala Gu-midyala, Beatriz Herce-Hagiwara, Annabella Kouris, Seoyeon Lee, James Msekela, Anh Thu Pham, et al. 2019. A middle-school camp emphasizing data science and computing for social good. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 358–364.

[4] Mehmet Celepkolu and Kristy Elizabeth Boyer. 2018. The importance of producing shared code through pair programming. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 765–770.

[5] Code.org. [n.d.]. <https://code.org/>.

[6] K-12 Computer Science Framework Steering Committee et al. 2016. *K-12 computer science framework*. ACM.

[7] Diana Franklin, Charlotte Hill, Hilary Dwyer, Ashley Iveland, Alexandria Killian, and Danielle Harlow. 2015. Getting started in teaching and researching computer science in the elementary classroom. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. 552–557.

[8] Simin Ghavifekr. 2020. Collaborative Learning: A Key to Enhance Students Social Interaction Skills. *MOJES: Malaysian Online Journal of Educational Sciences* 8, 4 (2020), 9–21.

[9] Cecily Heiner. 2018. A robotics experience for all the students in an elementary school. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 729–734.

[10] Norizwan Ideris, Siti Mastura Baharudin, and Norhasyimah Hamzah. 2019. The effectiveness of scratch in collaborative learning on higher-order thinking skills in programming subject among year-six students. In *Atlantis Press*. 421–425.

[11] Karen H Jin. 2018. A “Loopy” Encounter: Teaching Elementary Students the Concept of Loops. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. 1099–1099.

[12] Victor R Lee, Frederick Poole, Jody Clarke-Midura, Mimi Recker, and Melissa Rasmussen. 2020. Introducing Coding through Tabletop Board Games and Their Digital Instantiations across Elementary Classrooms and School Libraries. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 787–793.

[13] Paea LePendu, Cecilia Cheung, Mariam Salloum, Pamela Sheffler, and Kelly Downey. 2021. Summer Coding Camp: Curriculum, Experiences, and Evaluation. In *2021 ASEE Virtual Annual Conference Content Access*.

[14] Colleen M Lewis. 2011. Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education* 21, 2 (2011), 105–134.

[15] Alex Lishinski, Joshua Rosenberg, Michael Mann, Omiya Sultana, and Joshua Dunn. 2021. How CS1 Students Experienced COVID-19 In the Moment: Using An Experience Sampling Approach to Understand the Transition to Emergency Remote Instruction. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 1254–1254.

[16] Kaili Lu, Feng Pang, and Rustam Shadiev. 2021. Understanding the mediating effect of learning approach between learning factors and higher order thinking skills in collaborative inquiry-based learning. *Educational Technology Research and Development* (2021), 1–18.

[17] Cecilia Martinez, Marcos J Gomez, and Luciana Benotti. 2015. A comparison of preschool and elementary school children learning computer science concepts through a multilanguage robot programming platform. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*. 159–164.

[18] Olivia Nche. 2019. Engaging African American Elementary School Children in Code Understanding. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 1296–1296.

[19] Netsblox. [n.d.]. <https://netsblox.org/>.

[20] Alpaslan Sahin, Mehmet C Ayar, and Tufan Adiguzel. 2014. STEM Related After-School Program Activities and Associated Outcomes on Student Learning. *Educational Sciences: Theory and Practice* 14, 1 (2014), 309–322.

[21] Niral Shah, Colleen Lewis, and Roxane Caires. 2014. Analyzing equity in collaborative learning situations: A comparative case study in elementary computer science. In *Proceedings of the 11th International Conference of the Learning Sciences*. 495–502.

[22] G Shamir and I Levin. 2021. Fostering Computational Thinking in Remote Learning Settings. In *Proceedings of EDULEARN21 Conference*, Vol. 5. 7565–7573.

[23] Han-Yu Sung and Gwo-Jen Hwang. 2013. A collaborative game-based learning approach to improving students’ learning performance in science courses. *Computers & Education* 63 (2013), 43–51.

[24] Andrew Kenneth Tolmie, Keith J Topping, Donald Christie, Caroline Donaldson, Christine Howe, Emma Jessiman, Kay Livingston, and Allen Thurston. 2010. Social effects of collaborative learning in primary schools. *Learning and Instruction* 20, 3 (2010), 177–191.

[25] Yune Tran. 2018. Computer programming effects in elementary: Perceptions and career aspirations in STEM. *Technology, Knowledge and Learning* 23, 2 (2018), 273–299.

[26] Jennifer Tsan, Jessica Vandenberg, Zarifa Zakaria, Danielle C Boulden, Collin Lynch, Eric Wiebe, and Kristy Elizabeth Boyer. 2021. Collaborative Dialogue and Types of Conflict: An Analysis of Pair Programming Interactions between Upper Elementary Students. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 1184–1190.

[27] Jennifer Tsan, Jessica Vandenberg, Zarifa Zakaria, Joseph B Wiggins, Alexander R Webber, Amanda Bradbury, Collin Lynch, Eric Wiebe, and Kristy Elizabeth Boyer. 2020. A Comparison of Two Pair Programming Configurations for Upper Elementary Students. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*. 346–352.

[28] Jessica Vandenberg, Arif Rachmatullah, Collin Lynch, Kristy Elizabeth Boyer, and Eric Wiebe. 2021. Interaction effects of race and gender in elementary CS attitudes: A validation and cross-sectional study. *International Journal of Child-Computer Interaction* 29 (2021), 100293.

[29] Jessica Vandenberg, Arif Rachmatullah, Collin Lynch, Kristy Elizabeth Boyer, and Eric Wiebe. 2021. The Relationship of CS Attitudes, Perceptions of Collaboration, and Pair Programming Strategies on Upper Elementary Students’ CS Learning. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V*. 1. 46–52.

[30] Zoom. [n.d.]. <https://zoom.us/>.