

# Pair Programming in a Pandemic: Understanding Middle School Students' Remote Collaboration Experiences

Aisha Chung Galdo, Mehmet Celepkolu, Nicholas Lytle, Kristy Elizabeth Boyer

University of Florida, Gainesville, Florida  
aishagaldo@ufl.edu, mckolu@ufl.edu, nlytle@ufl.edu, keboyer@ufl.edu

## ABSTRACT

The COVID-19 pandemic has demonstrated that learning remotely is a crucial skill for K-12 students. However, remote instruction and collaboration bring a new set of challenges for these students, especially in the context of pair programming. An important goal for the CS education community is to understand these younger learners' experiences during remote programming activities. This experience report describes a three-day learning experience in which 18 middle school students engaged in remote pair programming activities by modeling scientific processes in a block-based programming language. After three remote pair programming sessions, we conducted individual interviews to understand middle school students' experiences during remote pair programming activities as well as comparing these new experiences to their previous co-located pair programming experiences. The results from these interviews suggest that the majority of the students (72%) enjoyed the remote activities despite many (55%) experiencing some form of technical difficulty. The interviews revealed important opportunities and challenges that being remote brought to pair programming within themes of changes in communication and focus, pair programming dynamics, and available resources. Students also identified issues with remote collaboration such as technical difficulties from software that impaired their ability to work and to communicate. These observations inform new efforts to adapt CS education to the increased demand for remote collaborative work and reveal patterns that may increase success in this new work style.

## CCS CONCEPTS

• **Social and professional topics** → **Computing education.**

## KEYWORDS

Remote pair programming, collaborative learning, middle school

### ACM Reference Format:

Aisha Chung Galdo, Mehmet Celepkolu, Nicholas Lytle, Kristy Elizabeth Boyer. 2022. Pair Programming in a Pandemic: Understanding Middle School Students' Remote Collaboration Experiences. 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/10.1145/3478431.3499324>

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](mailto: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/10.1145/3478431.3499324>

## 1 INTRODUCTION

The CS Education community has identified the ability to work collaboratively, especially remotely, as a crucial skill for students in an increasingly computationally dependent world [24]. As a collaborative coding paradigm, pair programming is widely used in CS education, which allows students to collaborate on the same coding task while holding different roles: the *driver* implements the solution, and the *navigator* provides the instructions and checks the code. Remote pair programming has been studied extensively in higher education [3, 4], where remote pairs have been shown to provide similar outcomes compared to co-located pairs [1, 23], sometimes even outperforming them [29]. However, there has been limited research on remote pair programming with younger learners, which means it is necessary to understand the ways in which these experiences differ for these students. A critical age group for examining these questions is middle school students, as it has been identified as a critical age for developing interest in STEM fields [13]. Although remote collaborative work has been studied for this age group in classes such as writing and physical science [20, 25, 27], there is still much that is unknown about their experience in remote pair programming. The CS education community faces many open questions around the opportunities and challenges that come with remote pair programming.

In this experience report, we present a learning experience where 18 middle school students participated in a three-day remote CS learning activity. Students had previous experience in co-located pair programming in their classroom and practiced remote pair programming for the first time over video-conferencing software (Zoom) during the COVID-19 pandemic. These students then participated in a set of semi-structured interviews, where our goal was to explore the experiences that remote pair programming presented for the younger learners. The contributions to the SIGCSE community are three-fold: first, we describe students' sentiment to understand students' overall perception toward these activities. Second, we report the way students responded to four important factors identified in the literature as important for remote collaboration: perceived success, other collaborative experiences, perceived impact as navigator, and technical difficulties. The goal of this description is to provide insight into the extent to which middle school students' responses reflected these factors. Finally, we present other opportunities and challenges that students identified throughout the interviews. The goal of this investigation is to shed light on students' current attitudes and understanding of remote pair programming, the opportunities and challenges that students may find, and how the CS education community can expand on opportunities and better support students through these challenges. The findings suggest that middle school students felt successful in this remote approach, previously had positive experiences with collaboration,

and often felt impactful in the navigator role. Also, students reported that working remotely made them have more autonomy in their learning environments, and they were able to navigate more efficiently compared to co-located pair programming. Students also identified several issues with remote collaboration such as technical difficulties from software that impaired their ability to work and communicate. We report on these findings in order to better understand students' perceptions of remote pair programming and support young learners in adapting to a new style of working.

## 2 RELATED WORK

While remote collaborative work is not prevalent in middle school education in general, many students have grown more accustomed to remote learning, often having positive experiences [17]. Even prior to the COVID-19 pandemic, younger learners have become more familiar with remote collaboration through the use of editors such as Google Docs [20, 27]. These studies have found that middle school students are receptive and benefit from the remote collaboration process. However, this research has not been extended to understanding how students experience remote work in a CS education context, in particular, through remote pair programming.

Before the COVID-19 pandemic, there were several co-located pair programming studies implemented with middle school students [7, 10]; yet, the previous research surrounding remote pair programming primarily focused on industry or higher education settings. These studies have concluded that remote pairs often are equally effective as co-located pairs [16, 23], were able to receive the same benefits as traditional pair programming [28], and often rated their communication with their partner as more satisfactory than co-located pairs [2]. However, there is a gap in the literature about remote pair programming with middle school students and to what extent these benefits are observed with younger learners.

Previous research reported challenges unique to remote collaboration. For example, Isenhour et al. [18] found that in virtual learning environments, the increase in number of computers leads to increased technical difficulties. These issues tend to be unavoidable and consume valuable time. They suggest making students aware of the steps they can take to avoid these issues. Stotts et al. [23] also identified issues such as the inability to point, the learning curve associated with online communication, and increased time spent on verbal communication which could have been solved with a visual diagram. Some of these challenges are also likely to happen amongst younger learners; thus, we focus on some of these challenges specifically in our analysis.

## 3 METHODS

### 3.1 Participants

The remote learning activities reported in this experience report were implemented with middle school students in the southeast United States. To plan the activities, we met with the students' teacher to discuss the best way to implement the remote activities. The teacher suggested the sign up tool "www.signupgenius.com", which the students had previously used for other remote activities for the class. To schedule the meeting times with students, we provided them with time slots to choose from which ranged from 9am to 6pm every weekday for 3 weeks. The teacher then sent the

students' parents the link to the sign up sheet along with an online consent form. A subset of the parents consented and selected the times where the students would be available for the activities. We randomly matched each child with another child based on their selected available times. Next, the teacher sent them a Google Doc link, which contained information about their partner, their meeting time and the Zoom meeting link. The students needed to click the link in the document to join the meetings.

26 students voluntarily signed up to participate in the studies, but only 18 students participated in all three days of the study, including the interview. We report the results from these 18 students in this study. Out of the 18 students, there were 10 girls (55.6%), 7 boys (38.9%) and 1 unspecified (5.5%). The set of students included those who identified as Asian/Pacific Islander (61.2%), White (16.7%), Hispanic/Latino (5.6%), or Other (16.5%). The mean age was 12.2 with ages ranging from 11 to 13, and 45% of students reported having some prior coding experience. These students were part of an ongoing pair-programming implementation planned for two semesters. In the first semester, which was in-person, they learned CS concepts including variables, conditionals, loops, and object-oriented programming in co-located pairs by using the Snap! block-based programming environment<sup>1</sup>. The students modeled scientific processes based on the lesson topics such as *Water Cycle* and *Light Waves*. The second semester of activities, the focus of this work, was originally scheduled to be done in person but shifted to remote due to the COVID-19 pandemic. We detail the experience shifting to remote pair programming and the data we captured from students below.

### 3.2 Procedure

The partner school switched to remote only instruction in response to the COVID-19 pandemic in the second semester. As such, we made changes to provide the activities in a remote setting. During these remote activities, the researchers met with each student pair three times. In session 1, our goals were to prepare students to do the activities remotely, make sure they were comfortable with the video conferencing application (Zoom), and introduce them to a coding concept in Snap!, cloning, which they needed for the next two activities. All students had previously attended some remote sessions with their classroom teacher and were familiar with Zoom. Additionally, we also showed them the basic functionalities that they would need to use during the coding activities (e.g., screen sharing, switching driver/navigator roles). As part of the pair programming paradigm, only one of the students (driver) was allowed to make changes on the Snap! programming interface during a given time while sharing their screen. The other child (navigator) provided feedback). To switch roles, the driver would save the progress and send the link to the navigator who would then open the link and continue from where their partner left off. This role-switching process took about 1 minute to complete.

In session 2, students completed a science modeling activity in which they coded an *Evolution* science model (Figure 1). Evolution was a concept they had recently covered in their science class. The goal of the activity was to build code that would create two different butterfly populations: one that got smaller with each reproduction

<sup>1</sup><https://snap.berkeley.edu/>

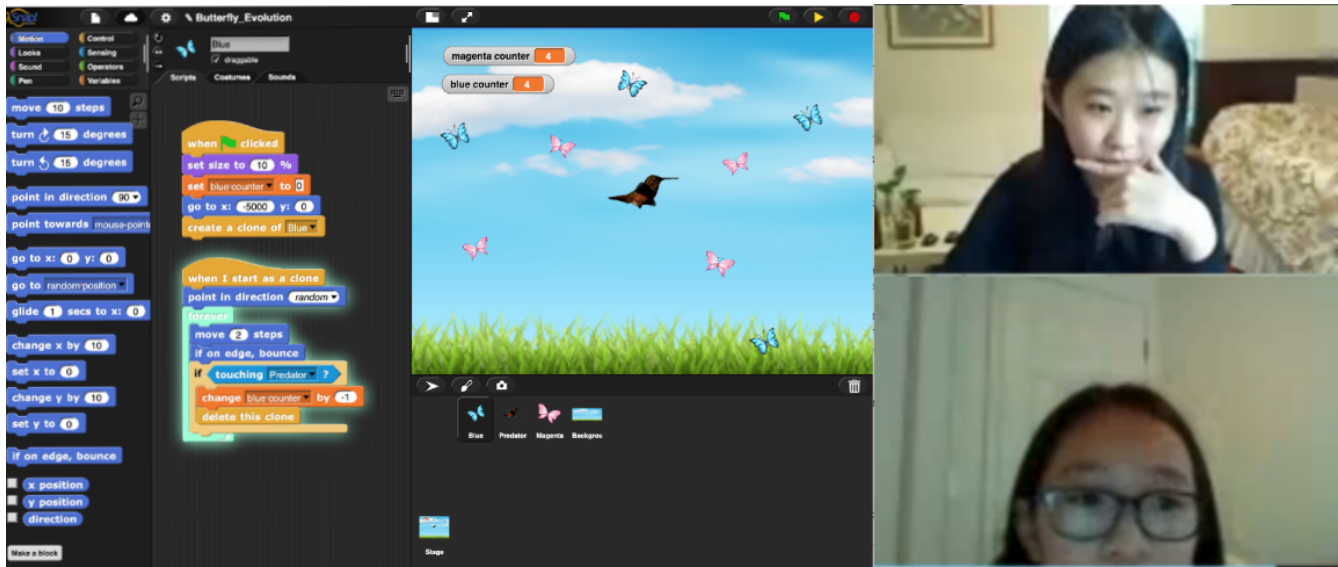


Figure 1: Students working remotely on the *evolution science modelling activity on Snap!*.

due to a genetic mutation and one population that stayed the same size. Reproduction was programmed through the cloning feature taught to the students in the first session. As a bird would move around the screen and eat butterflies it touched, the population of butterflies that got smaller had a higher likelihood of staying alive. This led to there being more of the small butterflies than the ones that remained the same size.

In session 3, students completed another science activity similar to the one conducted in session 2. Students coded a *Food Web* science model that represented the feeding relationship among a group of animals (hawks, birds, and butterflies). Once students finished the activity, or after 45 minutes had passed, they were asked to stay in order to conduct individual interviews, which lasted around 15 minutes. The interviews were semi-structured, with the researchers asking specific questions regarding students' experiences collaborating remotely.

At the beginning of each session, the researcher provided the written instructions for the activity and then reminded the students to follow the pair programming method. After the introduction, the researchers turned off their camera/microphone, but they were always available to answer students' questions or address possible technical difficulties. The researchers did not interrupt students working on the activity unless the students explicitly asked for help, as the goal was to allow students to have a natural interaction and collaborate on the problem together. While some pairs asked for help a few times, some pairs did not ask any questions throughout the entire session. The researcher also reminded students to switch driver/navigator roles every 20 minutes to ensure each child had equal time driving and navigating.

### 3.3 Analysis

We examined the student interviews from three different analysis approaches: (1) we analyzed the students' sentiment toward remote pair programming activities; (2) we examined the students'

experiences on targeted areas which the literature indicated would be important to these young learners; (3) we investigated what opportunities and challenges students identified in remote pair programming by answering open-ended questions.

**3.3.1 Sentiment Analysis.** Understanding student sentiment is important for remote pair programming, as young learners can form lasting impressions from the first experience with a new activity [14]. Previous research utilized sentiment analysis for both remote [22] and face-to-face [9] pair programming, but these studies were conducted with students in grades other than middle school.

To analyze the data, three researchers read the interview transcripts and manually rated on a five-point Likert scale how much each student liked or disliked the remote activities. This scale ranged from 1, completely negative to 5, completely positive.

After rating each student's sentiment, the researchers computed the inter-rater reliability score, which resulted in a Kappa score of 0.83, indicating substantial agreement [11].

**3.3.2 Directed Content Analysis.** To understand how middle school students responded to the factors of remote pair programming that the literature indicated to be important, we asked specific questions during the interview process. Two researchers first independently labeled the predetermined categories. Then they discussed any differences that arose in their labels and until they reached an agreement for each label. These questions were:

- (1) Did you experience technical difficulties? What kind?
- (2) Do you feel your collaboration was successful? Why?
- (3) What other remote activities do you enjoy now? Why?
- (4) Did you feel impactful as a driver/navigator?

The first category was experiencing technical difficulties. We expected this topic to arise because it has been a factor in both pair programming [12] and online collaboration in middle schools [18]. For this topic researchers identified if the student reported

a technical difficulty by choosing yes or no. Then the researchers labeled the technical difficulties the students listed.

The second category chosen for the directed content analysis was perceived success in remote pairs, which has been previously studied through an evaluative test-based lens [21], however, we also wanted to know what factors middle school students find important to their own views of success. This question was targeted during the interviews by asking students if they felt successful about their collaboration and why.

The third category was about prior experiences with remote collaboration as we wanted to understand if students were already accustomed to working remotely with others, and what parts of these remote collaborations they enjoy.

The fourth category was about students' perceived impact in the navigator role. In young programmers, it is important that they feel empowered during activities [19]. Since the navigator's role can sometimes become blurred [6, 8], we wanted to ensure that students still felt they were impacting the construction of code in a remote environment.

**3.3.3 Thematic Analysis.** To further investigate what other opportunities and challenges students may find in remote pair programming, we used a thematic analysis method for identifying, analyzing and reporting patterns (themes) within data [5]. This approach allowed us to investigate the most salient experiences middle school students perceived during remote pair programming. The thematic analysis was conducted on the following questions from the semi-structured interview:

- How do you like the remote Snap! activities so far? Do you notice any advantages or disadvantages to being remote?
- Were there any advantages or disadvantages to being a remote navigator? Were there any advantages or disadvantages to being a remote driver?

Using inductive content analysis [15], two researchers first independently labeled students' responses to interview questions. These labels were created from the detailed responses that students gave, simple yes/no answers to questions were not labeled. Then the researchers came together and iteratively merged highly similar labels together and finally grouped thematically similar labels from this revised set leading to four themes.

## 4 RESULTS

### 4.1 Sentiment Analysis Results

The student responses indicated that they overall enjoyed the activity, with an average of 3.95 on a 5 point Likert scale. 13/18 students (72%) reported that they liked the remote activity more than a three: 4 students were completely positive (5) and 9 students were positive (4). On the other hand, the remaining 5 students felt neutral (3) toward the activity. None of the students had negative sentiment (less than 3) toward the remote activities.

### 4.2 Directed Content Analysis Results

In this section, we present a summary of the findings within each category we investigated in the directed content analysis. Table 1 shows the category, the labels which fell under that category, and the frequency of that label.

**Table 1: Directed Content Analysis Results**

Category	Sample Labels	Freq.
<b>Technical Difficulties</b>	Unstable internet connection	8
	Snap! website crashing	3
<b>Perceived Success</b>	We could finish the activity	8
	We worked well together	6
	We learned	5
<b>Other Experiences with Remote Collaboration</b>	Fun to communicate with friends	9
	Fun to work together	3
	Enjoy having company	2
	Content is fun and accessible	2
<b>Perceived Impact As Navigator</b>	Partner listened to my suggestions	8
	I helped answered questions	6
	Partner implemented suggestions	4

The first category shows that the majority of students (10/18, 55%) experienced technical difficulties. Eight students experienced issues due to their internet connection such as sound cutting out or slow screen sharing. Three students had issues with the Snap! website crashing. One student had an issue with his mouse lagging more than it normally did.

The responses for the second category, perceived success, showed that all students felt they were successful. Students stated that they felt successful for various reasons like feeling like they had learned (5 students), feeling as if they worked well together with their partner (6 students), and feeling successful about completing the activity (8 students).

The third category showed that students have previously engaged in many different remote collaborative activities that they enjoy. The most popular activity was video games (7) followed by school sponsored activities like arts and crafts or cooking club (6), and FaceTime/phone calls (4). Some of the reasons they enjoyed these activities were because they find it fun to work together and communicate, enjoy having company, and the actual content of the activity is fun and accessible.

The fourth category about perceived impact as the navigator showed that all but one student felt they were impactful as the navigator. The students who said they felt impactful as the navigator pointed to their role in reading the instructions and helping answering questions as well as their partner listening to and implementing suggestions. The student did not feel as impactful in the navigator role:

*When I was navigator, I just kind of like, you know, like read the instructions and like, um, tried to help [...] but like a few times [my partner] kind of like cut me off, like when I was saying something.*

### 4.3 Thematic Analysis Results

This section presents the outcomes of the thematic analysis, which provides insights into the other opportunities and challenges middle school students identified during the interviews. Table 2 shows each theme, a sample label present in each theme, and the number of times the label was mentioned by a student.

**Table 2: Thematic Analysis Results**

Theme	Sample Labels	Freq.
<b>Changes in Communication</b>	Remote improves communication	3
	Remote impairs communication	14
<b>Changes in Pair Programming Dynamic</b>	Changes in driver role	3
	Changes in navigator role	7
<b>Changes in Focus</b>	Increased focus	2
	Decreased focus	7
<b>Resource Changes</b>	Ability to seek helpful resources	4
	Reliance on software	8

**4.3.1 Theme 1: Changes in Communication.** This theme encompasses the labels related to the reported changes in students' communication. The frequency of students who found remote pair programming impairing their communication was much higher than the students who found it improving their communication. The students stated that it was simply harder to communicate (10/18), it was better to communicate when you were in physical proximity (3/18), and found the quality of communication better when in person (1/18). The most common reason for remote work impairing communication was the inability to point:

*Like if you're trying to help them and like, they placed the wrong block or something like in person, you could like point.*

In contrast, some students found it easier to communicate with their partners but were unable to pinpoint the specific reasons.

**4.3.2 Theme 2: Changes in Pair Programming Dynamics.** This theme refers to students' perceived change in the pair programming roles. The majority of the changes occurred in the navigator position due to: being able to see the both their own screens and their partners (4/18), the annotation function(1/18), the navigation being easier (1/18), and having less control over their partners (1/18). Below is an excerpt from a student who felt positively affected by the ability to see both screens.

*It would be harder to show it to us if we were in class, because it would be hard to contact on separate computers.*

Students who felt that the driver role changed found that the driver did not have to adhere to typical pair programming rules (2/18), and the driver is more autonomous (1/18).

*You could just tell them and they [driver] would probably find it by themselves.*

**4.3.3 Theme 3: Changes in Focus.** Students found that being remote led to either increased or decreased focus. The majority of students found that it led to decreased focus because of several reasons: there was less structure (2/18), it was less efficient (2/18), they had a preference for being in-person (2/18), and there were more distractions (1/18).

*Even the online I feel it's not that bad, I guess, scheduling issues. I don't think of, because I'm online and I'm doing this myself and not having like a teacher planned out for me.*

Students who had increased focused cited: less distractions, and being more efficient.

*The environment is just quieter and more peaceful, so, and like usually in person it can get a little loud and sometimes that can be distracting.*

**4.3.4 Theme 4: Resource Changes.** The students found that the resources available to them changed due to being remote. Some found that they were able to use outside resources (Google) which was advantageous (1/18), and that the instructor was more available remotely (3/18).

*I kind of found it easier because a lot of the time, I wasn't sure what I was doing though with that. I would just, I was able to help her out by looking things up.*

Other students found that their reliance on software made remote pair programming more difficult. This was due to technical difficulties (6/18) and having to switch between windows (2/18).

*She couldn't be on zoom and construction at the same time, because it's hard to do things like that unless you had two desktops, which I do not. And so being, so she would, you have to switch back and forth and that's more difficult than having myself on the screen next to her and then hold up on the screen in front of her.*

## 5 DISCUSSION

We investigated the opportunities and challenges of remote pair programming activities for middle school students through three modes of analysis on student post-interview questions. The sentiment analysis results suggest the majority of students enjoyed the activities despite many experiencing some form of technical difficulty. This is important as students this age are likely to form lasting impressions based on their first experiences [14], so ensuring that remote collaboration is perceived as positive is important.

This positive perception of the activity was also found in the Directed Content Analysis, with all students reporting feeling successful and some mentioning they felt successful because of their collaboration with their partner. This result indicates promise for remote pair programming in middle school. The current body of research shows that middle school students are cognizant of the benefits that collaboration provides [20], and the results suggests that students consider it a marker of success. Many students felt successful because they were able to complete the activity. This helps reflect a view of learning which may be especially prevalent in formal education settings where grades may be at stake.

Most students were familiar with other forms of remote collaboration which could have contributed to their feeling of success, as

they had prior experiences. Students also identified social factors which contribute to the enjoyment of remote activities, with some specifying that it was their friends who made remote collaboration enjoyable. This aligns with the current research [26] about co-located pair programming in middle schools as students who partner with friends are become more confident throughout the learning process compared to non-friend partners.

A large majority of students felt that they were impactful in the role of the navigator. The current research regarding collegiate pair programmers tends to find that the role of the navigator is misunderstood [8, 21] leading to the perception that the navigator does not affect the code as much. However, for these students, the navigator appeared to have a significant role in the pair programming process. The thematic analysis confirmed this finding, with many students citing how their navigator partner was helpful during that role.

A majority (56%) of students experienced technical difficulties which was expected based on the existing literature [12, 18]. It is interesting that despite the prevalence of these issues, students still found themselves enjoying the activities. The studies with adults suggest the opposite: Coman [12] found that adults can get sidetracked by technical difficulties during remote collaboration and spend valuable time focusing on those. While the theme of change in focus does point to how students realized that remote learning caused additional difficulties, they found workarounds such as asking their partner to repeat themselves or refreshing.

## 6 IMPLICATIONS

This experience provides several implications for educators looking to implement remote pair programming activities in the future.

- Students may experience additional communication challenges during remote pair programming. To mitigate this problem, facilitators can pair students who they know will work well together. Otherwise, it may cause challenges for these students to collaborate in a new environment and with a new person.
- Technical issues do present a challenge for remote pair programming, though these findings suggest this challenge may not be as significant for young learners as it is for adults. Middle school educators should not let the possibility of technical difficulties stop them from implementing remote pair programming.
- Remote pair programming offers an opportunity for educators to help students feel more equal throughout the learning process, especially in the navigator role. It is important for educators to make sure the time spent in both roles is equal.
- Middle school educators should be aware that students may not adhere to the driver/navigator pair programming dynamic when placed in a remote setting. It is important that educators take an active role in preventing this from happening.
- Educators should be explicit in the ways being remote can be advantageous by showing students how to control the two windows or use search engines to solve issues.
- Since younger students may not be accustomed to making decisions about their education (e.g., time management, maintaining focus), educators may see benefit in guiding students in building environments conducive to learning.

## 7 CONCLUSION AND FUTURE WORK

This experience report explored how middle school students perceived remote pair programming. The results provide insight into how younger students approach factors of pair programming commonly found in the existing literature and themes which have yet to be explored in the literature. The directed content analysis yielded promising results in regards to students' overall view of pair programming, with many finding themselves successful and impactful as the navigator. While many reported technical difficulties, students were more adept at responding to these difficulties than expected. The findings suggest that students within this age group are capable of benefiting from remote pair programming. The themes emerging from 18 different students indicated that they perceived various elements of pair programming to change substantially when remote. The changes in communication due to transitioning to remote work was the most frequent theme, and it highlighted the impact of relatively small technical issues in shaping students' perspectives of remote programming. Communication issues affected young learners the same way as adult learners, and many still preferred to work in co-located pairs. However, there were several unanticipated positive results especially with regards to students' perceptions of changes in the pair programming dynamic and available resources. These results can better prepare educators and researchers for the challenges that middle school students may face and illuminate opportunities that could be expanded upon within remote pair programming.

Because remote pair programming in middle schools is not a common practice yet, there is much to explore within the patterns and themes discussed in this paper. First, there is a need to conduct long-term studies with these younger learners in order to determine if the perceived benefits and drawbacks of remote collaboration change with regular use. Additionally, due to the pandemic, attitudes towards remote collaboration are possibly affected since there is no other option. We need to discern whether these attitudes will change once social norms return. Also, this paper focused on the perceptions students had about remote pair programming; however, we did not evaluate students' code nor their ability to work together. There is a need to understand if remote pair programming affects the quality of code young learners present as well as the quality of their communication. Finally, due to the pandemic, we conducted the study with a relatively small number of students, who voluntarily attended these remote CS learning activities. It is important to conduct similar remote studies with a larger sample size over several semesters.

## 8 ACKNOWLEDGMENTS

This research was supported by the National Science Foundation through grant DRL-1640141. 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] Prashant Baheti. 2002. Assessing Distributed Pair Programming. In *Companion of the 17th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*. 50–51.
- [2] Prashant Baheti, Edward Gehringer, and David Stotts. 2002. Exploring the Efficacy of Distributed Pair Programming. In *Extreme Programming and Agile Methods – XP/Agile Universe 2002*. Springer Berlin Heidelberg, 208–220.
- [3] Maxwell Bigman, Ethan Roy, Jorge Garcia, Miroslav Suzara, Kaili Wang, and Chris Piech. 2021. PearProgram: A More Fruitful Approach to Pair Programming. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 900–906. <https://doi.org/10.1145/3408877.3432517>
- [4] Kristy Elizabeth Boyer, August A D Wight, R Taylor Fondren, Mladen A Vouk, and James C Lester. 2008. A development environment for distributed synchronous collaborative programming. In *Proceedings of the Conference on Integrating Technology into Computer Science Education, ITiCSE*. ACM, 158–162. <https://doi.org/10.1145/1384271.1384315>
- [5] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.
- [6] Sallyann Bryant, Pablo Romero, and Benedict du Boulay. 2008. Pair programming and the mysterious role of the navigator. *International Journal of Human-Computer Studies* 66, 7 (2008), 519–529. <https://doi.org/10.1016/j.ijhcs.2007.03.005>
- [7] Shannon Campe, Jill Denner, Emily Green, and David Torres. 2020. Pair programming in middle school: variations in interactions and behaviors. *Computer Science Education* 30, 1 (2020), 22–46.
- [8] Mehmet Celepkolu and Kristy Elizabeth Boyer. 2018. Predicting Student Performance Based on Eye Gaze During Collaborative Problem Solving. In *Proceedings of the Group Interaction Frontiers in Technology, GIFT 2018*. ACM, 1–8. <https://doi.org/10.1145/3279981.3279991>
- [9] Mehmet Celepkolu and Kristy Elizabeth Boyer. 2018. Thematic Analysis of Students’ Reflections on Pair Programming in CS1. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE ’18*. ACM, 771–776. <https://doi.org/10.1145/3159450.3159516>
- [10] Mehmet Celepkolu, Aisha Chung Galdo, Kristy Elizabeth Boyer, Eric N. Wiebe, Bradford Mott, and James C. Lester. 2020. Student Reflections on Pair Programming in Middle School: A Thematic Analysis. In *Proceedings of the 14th International Conference of the Learning Sciences (ICLS)*. Springer, 803–804.
- [11] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46.
- [12] Irina D Coman, Pierre N Robillard, Alberto Sillitti, and Giancarlo Succi. 2014. Cooperation, collaboration and pair-programming: Field studies on backup behavior. *Journal of Systems and Software* 91 (2014), 124–134. <https://doi.org/10.1016/j.jss.2013.12.037>
- [13] Katherine P Dabney, Robert H Tai, John T Almarode, Jaimie L Miller-Friedmann, Gerhard Sonnert, Philip M Sadler, and Zahra Hazari. 2012. Out-of-school time science activities and their association with career interest in STEM. *International Journal of Science Education, Part B* 2, 1 (2012), 63–79.
- [14] Gregory J DiGirolamo and Douglas L Hintzman. 1997. First impressions are lasting impressions: A primacy effect in memory for repetitions. *Psychonomic Bulletin and Review* 4, 1 (1997), 121–124. <https://doi.org/10.3758/BF03210784>
- [15] Jennifer Fereday and Eimear Muir-Cochrane. 2006. Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods* 5, 1 (2006), 80–92. <https://doi.org/10.1177/160940690600500107>
- [16] Brian Hanks. 2005. Student performance in CS1 with distributed pair programming. *ACM SIGCSE Bulletin* 37, 3 (2005), 316–320. <https://doi.org/10.1145/1151954.1067532>
- [17] Danna Harvey, Diana Greer, James Basham, and Bo Hu. 2014. From the Student Perspective: Experiences of Middle and High School Students in Online Learning. *American Journal of Distance Education* 28, 1 (2014), 14–26. <https://doi.org/10.1080/08923647.2014.868739>
- [18] Philip L Isenhour, John M Carroll, Dennis C Neale, Mary Beth Rosson, and Daniel R Dunlap. 2000. The virtual school: An integrated collaborative environment for the classroom. *Educational Technology and Society* 3, 3 (2000), 74–86.
- [19] Siu Cheung Kong, Ming Ming Chiu, and Ming Lai. 2018. A study of primary school students’ interest, collaboration attitude, and programming empowerment in computational thinking education. *Computers and Education* 127 (2018), 178–189. <https://doi.org/10.1016/j.compedu.2018.08.026>
- [20] Jenell Krishnan, Andrew Cusimano, Dakuo Wang, and Soobin Yim. 2018. Writing Together: Online Synchronous Collaboration in Middle School. *Journal of Adolescent and Adult Literacy* 62, 2 (2018), 163–173. <https://doi.org/10.1002/jaal.871>
- [21] Jeffrey S Saltz and Ivan Shamshurin. 2017. Does pair programming work in a data science context? An initial case study. In *Proceedings of 2017 IEEE International Conference on Big Data*. IEEE, 2348–2354. <https://doi.org/10.1109/BigData.2017.8258189>
- [22] Maya Satratzemi, Despina Tsompanoudi, Stelios Xinogalos, and Leonidas Karamitopoulos. 2019. Examining the compatibility of students in distributed pair programming. In *Proceedings of the European Conference on e-Learning*. Academic Conferences Limited, 510–518. <https://doi.org/10.34190/EEL.19.023>
- [23] David Stotts, Laurie Williams, Nachiappan Nagappan, Prashant Baheti, Dennis Jen, and Anne Jackson. 2003. Virtual teaming: Experiments and experiences with distributed pair programming. In *Conference on Extreme Programming and Agile Methods*. Springer, 129–141.
- [24] Allen B Tucker, Owen Astrachan, Kim Bruce, Robert Cupper, Peter Denning, Scot Drysdale, Tom Horton, Charles Kelemen, Cathy McGeoch, Yale Patt, Viera Proulx, Roy Rada, Richard Rasala, Eric Roberts, Steven Rudich, Lynn Stein, Allen Tucker, and Charles Van Loan. 1996. Strategic directions in computer science education. *Comput. Surveys* 28, 4 (1996), 836–845. <https://doi.org/10.1145/242223.246876>
- [25] Jillian L Wendt and Amanda J Rockinson-Szapkiw. 2015. The Effect of Online Collaboration on Adolescent Sense of Community in Eighth-Grade Physical Science. *Journal of Science Education and Technology* 24, 5 (2015), 671–683. <https://doi.org/10.1007/s10956-015-9556-6>
- [26] Linda Werner, Jill Denner, Shannon Campe, Eloy Ortiz, Dawn DeLay, Amy C Hartl, and Brett Laursen. 2013. Pair programming for middle school students. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. ACM, 421–426. <https://doi.org/10.1145/2445196.2445322>
- [27] Megan Woodrich and Yanan Fan. 2017. Google Docs as a tool for collaborative writing in the middle school classroom. *Journal of Information Technology Education: Research* 16 (2017), 391–410.
- [28] Nick Zacharis. 2009. Evaluating the Effects of Virtual Pair Programming on Students’ Achievement and Satisfaction. *International Journal Of Emerging Technologies In Learning (IJET)* 4, 3 (2009), 34–39.
- [29] Nick Z Zacharis. 2011. Measuring the effects of virtual pair programming in an introductory programming java course. *IEEE Transactions on Education* 54, 1 (2011), 168–170. <https://doi.org/10.1109/TE.2010.2048328>