Collaborative Talk Across Two Pair-Programming Configurations

Zarifa Zakaria, Danielle Boulden, Jessica Vandenberg, Jennifer Tsan, Collin F. Lynch and Eric N. Wiebe
zzakari@ncsu.edu, dmboulde@ncsu.edu, jvanden2@ncsu.edu, jtsan@ncsu.edu,
cflynch@ncsu.edu and wiebe@ncsu.edu
North Carolina State University

Kristy Elizabeth Boyer, University of Florida, keboyer@ufl.edu

Abstract: Given that pair programming has proved to be an effective pedagogical approach for teaching programming skills, it is now important to explore alternative collaborative configurations. One popular configuration is where dyads collaborate by sharing a single computer sitting side-by-side. However, prior research points to potential challenges for elementary students when sharing a single computer when collaborating. This prompted us to explore another configuration where dyads sit side by side but collaborate on a shared virtual platform with individual computers. We compared the discourse of students' collaboration under these two settings. Results show that although there are no significant differences in the amount of collaborative talk between the two configurations, there is qualitative evidence of how differing affordances of two configurations shape collaborative elementary students' practices.

Introduction

A plethora of studies suggest that collaboration is an effective pedagogical approach for teaching students programming skills (e.g., Hank et al. 2011; Williams et al., 2000) Additionally, the ability to collaborate effectively helps prepare students for the future workforce (National Research Council, 2013). However, there are many alternative configurations to implement collaborative programming in classrooms. For example, dyads can each use individual computers in a two-computer pair-programming configuration or a single computer to do one-computer pair programming (Figure 1). There can be differences in the capacity of collaboration under different physical configurations. In pair programming, a common configuration used in industry is sharing one computer (Hank et al. 2011). Here, one partner takes turns as a driver and the other as a navigator of the programming environment. Empirical studies exploring the benefits of such one-computer programming shows that students or employees working together on a single computer increases enjoyment (Muller, 2006), retention rate (Carver et al., 2007) and fast problem-solving (Xu & Rajlich, 2006).

While adapting industry practices for use in educational contexts makes sense, there is reason to believe that changes in context would need to be accounted for to optimize its benefits. First, studies have shown that collaboration can be improved in terms of enjoyment, engagement, and time-efficiency if there is equality of individual control over the computer (Infante, 2009; Scott, 2003). Secondly, the traditional one-computer configuration has mostly been studied either in industry or with college students (e.g., Salleh et al., 2011). For younger students there can be issues with equity (Shah et al., 2014; Tsan et al., 2018) which may hamper productive collaborative talk, raising questions as to the appropriateness of pedagogical strategies requiring a high degree of interpersonal negotiation like turn taking.

Thus, we aim to explore a two-computer configuration where partners collaborate in a virtual workspace sitting side-by-side at individual computers. There are studies with similar configurations: collaboration with multiple mice on a shared screen (Echeverría, 2012; Infante, 2009; Scott et al., 2003), multiple touch options on shared screens such as tablets (Hsiao et al., 2014), or individual screens (Cockburn, 2005; Dewan et al., 2009). However, little is known about collaboration with individual computers sharing a virtual space in close proximity (i.e., two-computer configuration). This configuration affords individual agency with system input but allows face-to-face dialogue with a partner. Our goal is to contrast the discourse level in such two-computer with one-computer collaboration in an emerging area of programming in K-12 academics.

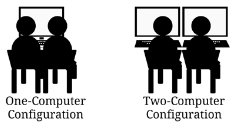

Figure 1. The One- and Two-computer Configurations in Pair Programming.

Theoretical framework

We focus our study on differentiating students' collaborative discourse throughout two different configurations of collaboration. When doing so, it is important to explore constructs that consider how the differences in these configurations might shape interactional levels of collective problem-solving. Mercer proposed the concept of Intermental Development Zone (IDZ) that focuses on the nature of interactive processes between teacher and student or peer-to-peer (Mercer, 2000). It draws on both Vygotsky's Zone of Proximal Development (ZPD) (Vygotsky, 1980) and Bruner's Scaffolding (Wood et al., 1976), to explain productive talk between peers. In a joint activity, three types of conversations can occur which Mercer termed 'Cumulative talk,' 'Disputational talk' and 'Exploratory talk.' In *Cumulative* talk, speakers build positively but uncritically on what the other has said, while *Disputational* talk is characterized by disagreement and individualized decision making. In *Exploratory* talk, participants engage critically, but productively, by challenging and offering alternative hypothesis to each other's ideas. Exploratory talk has been shown to expand the joint ZPD by enabling partners to achieve a better mutual understanding of the problem (Fernandez et al., 2015). Other research has shown that challenging viewpoints can foster conceptual change (Clark et al., 2003). As language is a prime medium of collaboration, even in computer-supported collaboration (Dillenbourg et al., 2009), it is appropriate to utilize Mercer's framework to explore language artifacts (i.e., spoken utterances) of dyadic collaboration.

Results from this study will explore how the two configurations might facilitate computational problem-solving during programming activities by quantitatively and qualitatively comparing these configurations. We sought to contrast the conversational attributes of these two configurations by documenting the three types of collaborative talk situated in the IDZ framework. We also describe a process of translating the framework into practice in the context of an elementary level collaborative programming environment.

Guided by Mercer's (2000) IDZ conceptual framework and prior empirical findings, we explore the following research questions: 1) What is the balance of different types of collaborative talk under different pair-programming configurations? 2) How are these types of talk different or similar across the two pair-programming configurations?

Method

Participants

This study is part of a design and development project investigating collaborative programming in elementary schools. Participants were from a fifth-grade classroom of a suburban school in the Southeastern United States. All the students were in an Academically and Intellectually Gifted program. We used data on 11 students (10-11 years), among whom 4 (36%) were girls and 7 (63%) were boys. In total, we had 7 dyads in one-computer and 6 dyads in two-computer configuration. Students were paired in alternative combinations. No set of two students were paired more than once. Except for one, all students participated in activities under both the configurations.

Procedure

Students participated in problem-solving activities with a 5-day programming curriculum developed by the authors. In these activities, students used the NetsBlox block-based programming environment to develop a program by dragging blocks and combining them. All the pairs solved problems of the same difficulty level. Dyads collaborated using the two different configurations on alternating days. In the one-computer configuration, students were informed that one partner would act as the driver who would control the input devices, while the other partner, the navigator, would talk to the driver about the problems and anticipate actions. The teacher used a timer to make sure each partner had equal opportunity to be a driver and a navigator. Dyads in the two-computer configuration shared a virtual platform with individual computers.

Data collection and analysis

The data source was video recordings of dyads collaborating on solving a programming problem. In the one-computer configuration, video recordings captured both the students' faces, voices, and the computer screen using the Open Broadcaster Software. For the two-computer configuration, the same software recorded each students' face, voice, and screens separately which were then merged using Adobe Premiere for analysis. Collaboration sessions lasted an average of 53 minutes. We utilized around 20 minutes of each dyad's collaboration. Criteria for selecting a section of videos were: audio-video clarity, comparable difficulty level of the activities, and if both the partners took turns as driver and navigator in a one-computer configuration.

We analyzed the data using a mixed method approach (Creswell & Creswell, 2017). First, we coded the videos to identify the distribution of three types of talk and analyzed these distributions using non-parametric tests. After the quantitative phase, we employed qualitative analysis to identify patterns and complexities. For the qualitative analysis, the video-recorded dialogue of all pairs and their respective schematic coding were qualitatively analyzed as a multiple case study design (Yin, 2014). To guide these analyses, we relied upon our theoretical propositions using Mercer's IDZ framework and prior literature on pair programming to aid in yielding descriptions of the contextual conditions of each thematic case. These descriptions enabled us to offer plausible explanations for the results from the quantitative phase.

Categorization and coding

Our qualitative analysis was initially inspired by a study conducted by T'sas (2018) that focused on the three types of talk from IDZ framework to classify conversation. Guided by T'sas' articulation of the talk types, a scheme aligned with Mercer's original framing was utilized. Further piloting refined T'sas' approach by changing a few problematic sub-categories. For example, 'counter-challenge' changed to 'counter-challenge followed by consensus' and placed in Exploratory talk, while 'counter-proposition' changed to 'counter proposition with no consensus' and put in Disputational talk. Also, utterances in a time interval which did not fall into either Exploratory or Disputational category were considered as Cumulative talk given that they were characterized without any disagreement or critical exchange and thus considered as assertive, yet uncritical. The finalized coding scheme consisted of three categories: Exploratory, Cumulative and Disputational (Table 1).

A professional transcription service transcribed the video recordings for coding. We utilized a time interval coding method (Bakeman, 2000) similar to that used in some other dialogue studies (e.g., Baines et al., 2009) to code not just an utterance but an aggregated series of responses in an interval, representing a particular type of talk. Piloting determined that 10-second intervals optimized capture of unique phrases with appropriate context. Thus, we divided a 20-minute recording into 120 ten second intervals and then labeled each interval exclusively to one category. Utilizing concurrent video with the transcripts allowed us to leverage facial expressions, body postures, and gestures to interpret the transcript. After training on the coding scheme, three authors coded 50% of the videos together (average Cohen's *kappa* = 0.805) and then coded the other 50% individually. Coding discrepancies were then discussed until consensus was reached.

Table 1: Definitions of the Categories and *k*-agreements

	Exploratory	Cumulative	Disputational
Major Characteristics	Challenge, alternative hypothesis, critical reasoning. *k*=0.714	Uncritical addition of ideas, Agreement. *k*=0.893	Disagreement without critical reasoning, *k*=0.803
Elaborated Characteristics	- Offered alternative hypothesis. - Initiations challenged, and counter-challenged followed by consensus. - Justifications given. - Joint acceptance	-Agreement without critical discussion. -Friendly and conflicts avoidance. -Positively but uncritically sharing ideas. -Superficial amendments.	-Disagreement without outcome. -Individualized decision-making. -Initiations directly rejected. -No/little constructive criticism. -Counter proposition with no consensus. -No resolutions.

Results and discussion

To answer our first research question, we examined the distribution of talk by quantitatively examining the coded frequencies of the three types of talk. We first investigated the overall level of discourse between the two configurations. The unit of overall quantitative analysis was students' talk per minute. We took each coded talk in 10 second intervals and aggregated them in number of different types of talk per minute. Non-parametric statistical test results indicated a non-significant difference, U=20.5, p=.943, between the two-computer configuration (66.9%) and one-computer (81.7%). The trend of a lower percentage of talk in the two-computer configuration could be explained by the fact that each student had equal autonomy in terms of computer access, limiting the need for negotiation. A Kruskal-Wallis test was used in an analysis into the distribution across the percentage of different types of talk per minute, and showed a statistically significant difference, H= 24.9, p<.001. Distribution was 12.7% Exploratory, 82.6% Cumulative and 4.7% Disputational. A series of Wilcoxon tests between pairs of each of the talk types revealed the following: Exploratory-Cumulative, Z=-3.18, p<.01, Cumulative-Disputational, Z=-3.18, p<.01, Exploratory-Disputational, Z=-2.85, p<.01. Previous literature indicates that, generally, without any intervention, Cumulative and Disputational talk tend to be higher than Exploratory (T'sas, 2018). While no significant differences were found between the configurations for any of the types of talk, Disputational trended higher for one-computer (7.0%) than two-computer (2.6%) (Figure 2).

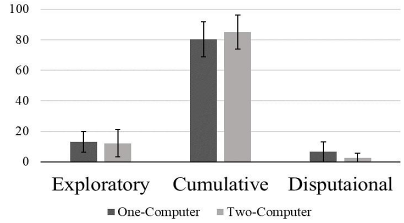

Figure 2. Percentage of each type of talk per minute, by computer configuration.

Our second research question allowed us to explore the quality of the talk across the conditions. We qualitatively analyzed the collaborative conversations and found some salient aspects of talk which elucidate differences between collaboration in one- and two-computer configurations. From the coded talk, we found a few sub-categories like, "challenge", "explain", "cooperation' etc. Three themes emerged from these categories.

Exploratory talk often preceded/followed by Disputational talk in One-computer

Results of qualitative analyses suggest that the transition between Exploratory talk and the other types of talk differed based on the configural conditions. We provide examples to better illustrate this phenomenon.

Tom and Sandy (pseudonyms) present one example of a one-computer configuration where Exploratory talk was preceded by Disputational talk. They repeatedly disagree with one another without any explanation to each other. There are disagreements like "No" and "Nope" multiple times without trying to come to a consensus. This falls into the characteristic of Disputational talk. However, after several statements of dispute, Tom hypothesize, "that is as far as it's gonna go," whereby Sandy challenges, "How do you know?" This challenge forces him to counter her question with further explanation which becomes Exploratory talk.

One-Computer	Two-Computer
Tom: Nope, not far enough, negative 150. Sandy: Negative 110. Tom: No, negative 150. Sandy: Let's go with 10, maybe. 10? Tom: Nope, five. Sandy: One. So we need to go, like, negative- Tom: Well, no, it's- it's going to X.215. Sandy: Negative ... Tom: That's as far as it's gonna go. Sandy: How do you know? Tom: Because it's a negative. You see, 'cause that's still the same number.	Tony: What'd you do? Sandy: I made, whoa, it's supposed to go right when I hit the right arrow but instead, it's like flying away. Tony: Yeah. Sandy: Why though? Tony: Oh, that happens. Sandy: Why? Tony: I don't know if it's just 'cause we don't understand what's happening. Because whenever we did that, it'd change that too much and it'd make it. I forgot how we did it last time. Sandy: It might have something to do with that.

Figure 3. Excerpts illustrating the relationship of Disputational talk with Exploratory (one-computer) or Cumulative (two-computer).

In comparison, consider these transitions between types of talk in two-computer configurations, where Cumulative talk progressed into Exploratory talk and then is followed by more Cumulative talk. In the excerpt in Figure 3, the same girl, Sandy, is paired with another boy, Tony. Here, Exploratory talk originates from Cumulative talk where Sandy requests Tony's assistance by saying "Tony, help me." This request then segues into a repetitive pattern of Exploratory phrases comprising a series of stimulating questions when each student responds to their partner's inquiry using several 'why' questions and 'because' explanations.

These examples are demonstrative of a divergent sequential form for these two conditions, where in the one-computer configuration Exploratory talk transitioned into Disputational talk while in the two-computer configuration Exploratory talk transitioned back into Cumulative talk. A possible explanation may be that because students in two-computer have equal access to their own devices, allowing them agency to express their ideas directly without requiring negotiation, the likelihood of control-based dispute is less likely to ensue.

Past research on student conflict during collaborative activities has demonstrated that this is a naturally occurring phenomenon (Jeong, 2008; Rubin, Pruitt, & Kim, 1994), and can be beneficial or harmful to outcomes depending on its characteristics. For example, Lee, Huh, and Reigegluth (2015) found task-related conflict to have a positive influence on peer collaboration as students are forced to self-reflect and build upon one another's ideas. Whereas they contend that when students are having disputes about the collaboration process itself or interpersonal conflicts, it can be detrimental to productive collaboration. There is concern that the one-computer configuration is perhaps amplifying the potential for unproductive disputes over process or based on interpersonal dynamics. The two-computer configuration still requires negotiation over the final programming artifact but removes the need for sharing input devices, perhaps striking a better balance of agency and control.

Balance of challenger and explainer in Exploratory talk

In the one-computer configuration, we found evidence that one learner may end up challenging their partner more by asking questions and demanding justification for any proposals or claims; we consider this type of learner to be a *challenger*. In the excerpts below, we can see the other partner took the role of an *explainer* and ended up justifying what she proposed or edited. As challenging is identified as a major characteristic of Exploratory talk, this pairing of *challenger* and *explainer* was found in this type of talk. Interestingly, among the excerpts, where this combination was found, it was the driver (who had control of the input) who played the role of challenger. With Steve driving and Melony navigating, Steve was more of a challenger, asking more questions to Melony who was answering them. In one instance, they were discussing deleting a block and Steve (driver) was constantly challenging Melony's idea with questions such as "Okay. So, what?", "How do you have two layers?", "Like, does it still work?", "Okay, So what?" one after another (see Figure 4). This tendency of the driver to be more challenging was found in other dyads in the one-computer configuration too. In another pair, Sandy, the driver, was challenging Anthony's ideas with a lot of 'why' and 'how' questions. Previous research suggests that 'why' and 'how' are Exploratory key words for being counter challenging and 'because', 'but', 'what if' are Exploratory keywords for explanation (Polo et al., 2015; Wegerif et al., 2005).

One-Computer	Two-Computer
Steve: Okay. So, what? Melony: And so, like here, wouldn't you do, um.. Steve: How do you have two layers? Melony: I don't know. Melony: Maybe you've to put wind direction off. Steve: Like, does it still work? Melony: Yeah, it works, but like ... Steve: Okay. So, what? Melony: And so, like here, wouldn't you do,um..	Tony: Because it was like this, when we pressed up arrow ... oh wait he has a... Steve: But I think we're actually connected these, when you press it this goes to- Tony: What happen if would go down more than he would go up? Steve: He doesn't even go down though.

Figure 4. Excerpts illustrating the balance of challenger and explainer in Exploratory talk.

In contrast, in the two-computer configuration, there was evidence that showed both the partners taking the roles of challenger and explainer in a more interleaved fashion. In Figure 4, Tony and Steve are found to be continuously challenging and explaining to each other with words such as 'but,' 'because,' 'what happens if,' 'even though' etc. Another pair, Luke and Clara were similarly questioning each other's ideas and demanding an explanation. In one instance, Clara was suggesting an 'if/else' block as well as challenging Luke's idea of inputting a 'stop' block. Luke too, was simultaneously proposing the opposite idea and kept questioning Clara.

This differing combination of *challenger* and *explainer* suggests that this relationship is not always equitable with the challenger playing more of an authoritative role in the pair. Researchers have strived to analyze and understand the equity of collaborative relationships in many subjects, including computer science (e.g., Deitrick et al., 2016; Shah et al., 2014; Stanton et al., 2002). While we are still unsure of the effects of an inequitable relationship in terms of the challengers and explainers, we hypothesize that collaboration is productive if the dyads have balanced, interleaved challenging and explaining interactions.

Cooperative work rather than collaborative work in two-computer configuration

In the two-computer configuration, both partners have access to a laptop, thus they get the opportunity to edit individually. This increase possibilities of getting involved in *cooperative* work rather than *collaborative* work, which has a clear distinction in terms of the socio-cultural learning mechanism (Davidson, 2014; Oxford, 1997). Kagan and Kagan (2009) identify a pair of critical attributes of cooperative learning: interdependence and accountability. Here, students can work in separate sections of the project, yet still simultaneously seek help from each other. For example, Rupert and Melony divided the work between themselves by saying "Yeah, we need to.

I'll do the top one, you can do the bottom one," indicating different code blocks. However, they both sought help when they came upon problems. Melony, facing difficulty, asked "In this one? And then the other way around?". Rupert then assisted her, answering "And then you can put negative 85 in the second one." This pattern of interdependence then continued. Figure 5 shows another prominent example, where Sandy and Tony start cooperating, Sandy responds to a teacher saying, "I'm gonna work on the sprite getting it to move back and forth and he's gonna work on the different enemies." However, after a few minutes of cooperative work, they start going back and forth to each other's section, hovering over their partner's computer to help them edit their sections. While doing cooperation, they were more involved in Cumulative talk. But when they started collaboration on same section of the code, more Exploratory instances were observed.

Two-Computer (Cooperative)	Two-Computer (Collaborative)
Sandy: You work on the ball, I'll work on the platforms. Tony: Let me just duplicate this last group and then add it. Sandy: Wait, I'll work on the ball actually, you keep going the enemies. Tony: Okay, let me fix. okay, so, he shouldn't be doing this forever now. Let me check. Tony: Yay, I've got enemy one. But now I have enemy two doing the exact same thing.	Sandy: Broadcast. We're gonna need some sort of broadcast in there. Melony: Okay, broadcast...Scissors... Sandy: Oof! Hello. Oof! Why is it repeating that? Oh, no, what did we do? We broke it!We broke it. We broke it. Melony: Don't worry, I know why. Sandy: Is it flashing for you, too? Melony: Yeah. I put it in forever. There. Oh, no. There we go. Now it'll only say it once. See? Sandy: But then it just disappears. What did you do, Melony? Melony: What?

Figure 5. Excerpts illustrating the individual vs. collaborative talk in two-computer configuration.

In the two-computer configuration, we also found both the students working on the same sprites but different blocks. The collaborative excerpt in Figure 5 shows Sandy and Melony working on the same section of code, challenging and helping each other as they work through it. While, as we see above that the two-computer collaboration could devolve into cooperative work, dyads seldom did work completely by themselves, rather they often verbalize what they were doing and sought help. Thus, these interactions are best characterized as moving along a collaborative-cooperative continuum. Relevant literature supports the idea that once each student has their own input device, they tend to cooperate rather than collaborate (Stanton et al., 2002).

Cooperative learning has been found to be an effective learning strategy to enhance achievement (McConnell, 2014) and thus a reason for exploration in pair-programming configurations. For future research we would explore productive outcomes of pair-programming with cooperative or collaborative interactions.

Conclusion

Guided by research informed by Mercer's (2000) framework, we believe helping students to productively collaborate with a higher percentage of Exploratory talk is a laudable goal when designing a pair-programming platform for elementary students. Exploratory talk offers potential for learning where evidence was found to extend students' learning within the ZPD (Littleton, 2005). Our findings that Exploratory talk occurred much less frequently than Cumulative were congruent with past studies (Bungnum, 2018; T'sas, 2018), which showed that without any intervention most conversation generally fall into positive but uncritical utterances, characteristic of Cumulative talk. However, our focus will be to use a combination of system and teacher level supports that enhance the quality and quantity of productive, Exploratory talk.

While there were no significant quantitative differences in the three types of talk between the configurations, qualitative analysis revealed interesting patterns that possibly point to how one and two-computer configurations shape collaborative work by students of this age. The one-computer configuration that provided direct control of editing code to only one student seemed to create an imbalance in an agency with regards to both proposing and evaluating potential solution paths. Evidence pointed to challenging ideas residing primarily with the driver, with the navigator left defending their ideas. We wonder if this is related to the pattern of Disputational talk proceeding Exploratory talk more often in the one-computer configuration. This result was not surprising because the driver is the learner that has control over the computer, thus control over the changes in the code. While we believe that the inequity of the challenger-explainer interaction may be due to the nature of the pair-programming roles, we plan to conduct statistical comparison to verify the differences between the two configurations. Our goal will be to develop interventions and adaptive support to improve the equity of these types of interactions. Examples of interventions include collaboration trainings to educate students on how to challenge their partner's idea and having intelligent collaboration agents that can urge both the students to get involved in challenging and explaining. Of course, one way to limit conflict over editing the program is to provide both students with input control. However, that configuration has the risk of dissipating the collaborative relationship, as evidence suggested that collaboration in two-computer configuration has a possibility of turning into cooperation. That said, our findings point to a healthy cooperative interdependence of the students in the two-computer configuration, as evidenced by the interleaving of challenger-explainer roles and regular consultation when working on separate codes.

In future research, we will continue to examine and operationalize Mercer's framework for measuring and analyzing dialogue in pair programming. We want to explore Cumulative talk more exhaustively to identify if there are methods of disaggregation of the Cumulative category that are parsimonious with theory and useful empirically to our research and development. Larger scale data collection with more students will also allow us to reach more conclusions about the quantity of both Exploratory and Disputational talk between the two conditions. These findings also have important implications for our future design efforts on systems for pair programming where virtual agents could be used to facilitate the use of Exploratory talk.

Limitations

Our study consisted of a relatively small, non-representative sample of students, limiting our statistical power and ability to generalize findings. Second, we were unable to analyze partner differences by condition due to instructional and logistical limitations. Relatedly, our analysis did not attempt to account for individual differences (e.g., prior programming knowledge, gender, etc.) which might have impacted dyad's collaborative dynamics. Future studies are planned to address these limitations.

References

Bakeman, R. (2000). Behavioral observation and coding. In H. T. Reis & C. M. Judd (Eds.), *Handbook of research methods in social and personality psychology* (pp. 138-159). New York, NY: Cambridge University Press.

Baines, E., Rubie-Davies, C., & Blatchford, P. (2009). Improving pupil group work interaction and dialogue in primary classrooms: results from a year-long intervention study. *Cambridge Journal of Education, 39*(1), 95-117.

Bungum, B., Bøe, M. V., & Henriksen, E. K. (2018). Quantum talk: How small-group discussions may enhance students' understanding in quantum physics. *Science Education, 102*(4), 856-877.

Carver, J. C., Henderson, L., He, L., Hodges, J., & Reese, D. (2007, July). Increased retention of early computer science and software engineering students using pair programming. In *Software Engineering Education & Training, 2007. CSEET'07*. (pp. 115-122). IEEE.

Clark, A. M., Anderson, R. C., Kuo, L. J., Kim, I. H., Archodidou, A., & Nguyen-Jahiel, K. (2003). Collaborative reasoning: Expanding ways for children to talk and think in school. *Educational Psychology Review, 15*(2), 181-198.

Cockburn, A. (2005). Crystal Clear. Addison-Wesley.

Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Thousand Oaks, CA: Sage.

Davidson, N., & Major, C. H. (2014). Boundary crossings: Cooperative learning, collaborative learning, and problem-based learning. *Journal on Excellence in College Teaching, 25*.

Deitrick, E., Shapiro, R. B., & Gravel, B. (2016). How Do We Assess Equity in Programming Pairs? In Looi, C. K., Polman, J. L., Cress, U., and Reimann, P. (Eds*.). Transforming Learning, Empowering Learners: The International Conference of the Learning Sciences (ICLS)* 2016, Volume 1. Singapore: International Society of the Learning Sciences..

Dewan, P., Agarwal, P., Shroff, G., & Hegde, R. (2009, May). Distributed side-by-side programming. In *Proceedings of the 2009 ICSE workshop on cooperative and human aspects on software engineering* (pp. 48-55). IEEE Computer Society.

Dillenbourg P., Järvelä S., Fischer F. (2009) The Evolution of Research on Computer-Supported Collaborative Learning. In: Balacheff N., Ludvigsen S., de Jong T., Lazonder A., Barnes S. (eds) *Technology-Enhanced Learning.* Springer, Dordrecht.

Echeverría, A., Améstica, M., Gil, F., Nussbaum, M., Barrios, E., & Leclerc, S. (2012). Exploring different technological platforms for supporting co-located collaborative games in the classroom. *Computers in Human Behavior, 28*(4), 1170-1177.

Fernández, M., Wegerif, R., Mercer, N., & Rojas-Drummond, S. (2001). Re-conceptualizing "Scaffolding" and the Zone of Proximal Development in the Context of Symmetrical Collaborative Learning. *The Journal of Classroom Interaction, 36/37*(2/1), 40-54.

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: a literature review. *Computer Science Education, 21*(2), 135-173.

Hsiao, H. S., Chang, C. S., Lin, C. Y., Chang, C. C., & Chen, J. C. (2014). The influence of collaborative learning games within different devices on student's learning performance and behaviours. *Australasian Journal of Educational Technology, 30*(6).

Infante, C., Hidalgo, P., Nussbaum, M., Alarcón, R., & Gottlieb, A. (2009). Multiple Mice based collaborative one-to-one learning. *Computers & Education, 53*(2), 393-401.

Jeong, H. W. (2008). *Understanding conflict and conflict analysis*. Thousand Oaks, CA: Sage.

Kagan, S. (2009). *Kagan's cooperative learning*. San Clemente, CA: Kagan Publishing.

Mercer, N. (2000). *Words and minds: How we use language to think together*. New York, NY: Routledge.

Müller, M. M. (2006). A preliminary study on the impact of a pair design phase on pair programming and solo programming. *Information and Software Technology, 4*8(5), 335-344.

Lee, D., Huh, Y., & Reigeluth, C. M. (2015). Collaboration, intragroup conflict, and social skills in project-based learning. *Instructional Science, 43*(5), 561-590.

Littleton, K., Mercer, N., Dawes, L., Wegerif, R., Rowe, D., & Sams, C. (2005). Talking and thinking together at Key Stage 1. *Early Years, 25*(2), 167-182.

Oxford, R. L. (1997). Cooperative learning, collaborative learning, and interaction: Three communicative strands in the language classroom. *The Modern Language Journal, 81*(4), 443-456.

McConnell, D. (2014). Implementing computing supported cooperative learning. Routledge.

National Research Council. (2013). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. Washington, DC: National Academies Press.

Polo, C., Lund, K., Plantin, C., & Niccolai, G. (2015). *Analyzing Exploratory Talk as a Socio-Cognitive Practice: Identity, Group Argumentation, and Class Debate Quality*. (hal-01208319)

Rubin, J. Z., Pruitt, D. G., & Kim, S. H. (1994). *Social conflict: Escalation, stalemate, and settlement*. New York, NY: McGraw-Hill.

Salleh, N., Mendes, E., & Grundy, J. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering, 37*(4), 509-525.

Scott, S. D., Mandryk, R. L., & Inkpen, K. M. (2003). Understanding children's collaborative interactions in shared environments. *Journal of Computer Assisted Learning, 19*(2), 220-228.

Shah, N., Lewis, C., & Caires, R. (2014). Analyzing equity in collaborative learning situations: A comparative case study in elementary computer science. In *Proceedings of the International Conference of the Learning Sciences,* 495-502. International Society of the Learning Sciences. Boulder, CO.

Stanton, D., Neale, H., & Bayon, V. (2002, January). Interfaces to support children's co-present collaboration: multiple mice and tangible technologies. In *Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community,* 342-351. International Society of the Learning Sciences.

Tsan, J., Lynch, C. F., & Boyer, K. E. (2018). "Alright, what do we need?": A study of young coders' collaborative dialogue. *International Journal of Child-Computer Interaction, 17*, 61-71, ISSN 2212-8689.

T'Sas, J. (2018). *Learning outcomes of exploratory talk in collaborative activities* (Doctoral dissertation, University of Antwerp).

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard University Press. Cambridge, Massachusetts.

Wegerif, R., Linares, J., Rojas-Drummond, S., Mercer, N., & Velez, M. (2005). Thinking Together in the UK and Mexico: Transfer of an Educational Innovation. *The Journal of Classroom Interaction, 40*(1), 40-48.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software, 17*(4), 19-25.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89-100.

Xu, S., & Rajlich, V. (2006, July). Empirical validation of test-driven pair programming in game development. In *Computer and Information Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering* (pp. 500-505). IEEE.

Yin, R. K. (2017). *Case study research and applications: Design and methods.* Thousand Oaks, CA: Sage.

Acknowledgements

This work is supported in part by the National Science Foundation (NSF) through the grant DRL-1721000. Everything expressed in this material are those of the authors and do not necessarily reflect the views of NSF.