ABSTRACT

ZAKARIA, ZARIFA. Collaborative Programming Practices in Elementary Classrooms: Exploring the Roles of Gender, Motivation, and Collaborative Discourse. (Under the direction of Dr. Margareta M Thomson)

Collaborative programming (i.e. pair-programming) in elementary school is a highly promising strategy for developing students' efficacy and interest in computer science, helping motivate them to excel in this skill and use it in future education and career trajectories. However, we know very little about students' current practices in programming, let alone collaborative programming environments and what influences them. This study provides new insights on the influence of gender and motivation in programming, the ground truth measure of current practices in an efficacious programming environment such as pair-programming, and how to facilitate students to better perform in pair programming contexts.

In this research we explored the role of gender and motivational beliefs on students' collaborative and programming performances in pair programming and design an intervention to support collaboration during pair programming. We conducted three studies to accomplish the goal. The first study explored how and if gender and prior experiences, mediated by motivational factors (mindset, interest and self-efficacy) influence individual students' programming related conceptual understanding; Second, we qualitatively compared different gender groupings based on the quality of dyad's collaboration and programming learning; Third, we examined a intervention of providing feedback on both productive collaboration and motivational factors aiming to help students better collaborate regardless of their gender and prior experiences.

Finding from our research shed light on how gender, prior experience and motivational beliefs and interests intercorrelate and influence students' learning in CS. The findings may direct educators and researcher toward factors related to programming practices that needs most attention for elementary classroom. This study also provided a preliminary validation of an

instrument that measures students' self-perceived ability beliefs and interests in programming at elementary level. This can be utilized and adapted by researchers to measure elementary students' motivational beliefs and interest in future studies. Finding that gender do not have strong impact on the understanding of programming, we then explored if gender impacts collaborative programming performance. This research provided insights on how different gender pairings such as girl-girl, boy-girl and boy-boy pairs' collaborative practices and programming compare. These findings will help researchers and educators decide on what aspects of pair programming male and female students may need support and design interventions accordingly. In the process of analyzing students' collaboration in pair programming, we also provided a validation of a framework for discourse analysis in pair programming. This framework can capture the more complex patterns and nuances of students' collaborative practices during programming activities. Using this framework we provided important insights into students' natural tendencies in collaborative discourse in pair programming that were critical for developing a more refined and targeted study design. Finally, we examined an intervention that utilized a feedback framework specifically designed to engage students in high quality collaboration. Educators can utilize this intervention to help students better collaborate not only to program better, also in any computer-supported collaborative environments.

Collaborative Programming Practices in Elementary Classrooms: Exploring the Roles of Gender, Motivation, and Collaborative Discourse

by
Zarifa Zakaria

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Teacher Education and Learning Sciences

Raleigh, North Carolina
2021

APPROVED BY:

Dr. Margareta M. Thomson Committee Chair	Dr. Eric N. Wiebe
Dr. John L Nietfeld	Dr. Sarah J Carrier

BIOGRAPHY

Zarifa Zakaria was born in Dhaka, Bangladesh where she grew up with her mother along with two cousin brothers and aunt. Her interest in young students' education started in high school when she volunteered at winter camps to guide and instruct students from marginalized communities. Thus, she pursued Education and earned her Masters in Education with concentration in Educational Psychology in 2014 from University of Dhaka, Bangladesh.

After her Masters degree she taught in a public primary school, as part of Teach for Bangladesh fellowship, for two years. Her experiences involved teaching in extremely overpopulated classrooms with absolutely no technology access. Majority of her students came from economically disadvantaged families. She learnt how challenging it is for students to succeed and strive in the world without any technology and computer access in this 21st century. This inspired her to pursue higher education in Educational Psychology and explore psychological mechanisms that influence learning in computer-supported environments. She got the opportunity to research young students' motivation and collaboration in Computer Science Education. During her doctoral program, Zarifa worked at Dr. Eric Wiebe's lab at Friday Institute of Educational Innovation, Raleigh, North Carolina, where she conducted her dissertation research. In her future work, she aims to continue working along the mission of providing all students equal access to learning and technology. Zarifa currently lives in North Carolina with her loving husband and a beautiful daughter.

ACKNOWLEDGEMENTS

I want to first thank my academic advisor Dr. Margareta Thomson for her guidance and encouragement throughout my doctoral journey. I am grateful for the support she provided throughout all the ups and downs of this academic journey. I also want to thank my research supervisor and committee member Dr. Eric Wiebe, for his mentorship for the past three years. He provided me the with sincere guidance in learning and conducting research, opening up immense opportunities to grow as a researcher.

I want to thank all of my former lab mates for continuously supporting me, challenging me and providing feedback. I have constantly learned from all my talented co-authors- Danielle Boulden, Jessica Vandenberg and Jennifer Tsan. Thank you Arif Rachmatullah for your feedback and suggestions. I am grateful to my friends Marina Wagemaker and Jennifer Houchins for listening to all my concerns, supporting me and helping me throughout this journey.

I thank my committee members for providing me valuable feedback and guidance. Also, thanks to Dr. Kristy Boyer and Dr. Colin Lynch for helping me conduct meaningful research. I thank all my friends and teachers at the college of education who made my doctoral experience valuable and meaningful.

Finally, I want to acknowledge and thank my family. I am grateful to my husband Samiul Haque for his encouragement, constant support on all my endeavors and bringing the best out of me. Thanks to my mother for everything. Without both of their support I could not have accomplished this achievement. Thanks to my adorable daughter Charu Mondrila for inspiring me all the time. I am also grateful to my eldest sister Zerin Zakaria for her encouragement on every step of the way.

Table of Contents

LIST OF TABLES

Chapter 2

Chapter 3

Chapter 4

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Introduction

Computer science (CS) is now an essential area of study in 21st century education. Computer science in the form of programming is important in modern lives as a career opportunity but also it belongs with reading and writing as a basic literacy for developing understanding in science, mathematics and other subject areas. In elementary level, programming has been increasingly introduced through elective and special technology courses. Literature shows strong intellectual and practical merits of learning programming in the elementary grades to increase students' problem-solving skills and learn new content in the digital era (e.g., Hank et al., 2011; Shute et al., 2017; Wing, 2006) and also helps prepare them for computationally intensive activities in the future (Navlakha & Bar-Joseph, 2011). Right now primarily CS graduates build technological tools, programs, and apps that are used by a vast majority of people, and thus they hold the preparatory privilege of controlling the lives of so many people (Barnes, 2017). Developing CS skills such as programming throughout K-12 will educate all students and provide the knowledge to make informed decisions about technologies-related problems in their everyday lives. On the other hand, programming skills are increasingly preferred in many non-CS careers such as any job with data analytics responsibility, sales, marketing, data science in any field etc. Having programming skills in early education levels would help students set the foundations of computer science concepts and skills as well as grow their interest towards the domain. Early, broad introduction of CS practices also has the important benefit of developing interest among populations of students who have historically been underserved and underrepresented in CS. Despite such potential for introducing computer science skills at elementary level, research is scarce in discerning how elementary students develop interest and motivation in common programming practices. In addition, there is a lack of understanding of what pedagogical strategies are effective at forwarding productive learning in CS for all students. These are some of the factors that significantly impact the ability to design and implement intervention strategies to support programming and learning CS concepts at this level.

Statement of Problem

Pair-programming (PP), a common and highly collaborative programming strategy, is being incorporated in many upper elementary classrooms for its pedagogical benefits in computer science education as well as other domain areas (Rodríguez et al., 2017). However, we lack studies that identify students' natural tendencies in pair-programming and external factors that influence those tendencies. Discerning students' current practices in programming and finding influential factors is key for designing and implementing effective interventions in classrooms. Studies have found that many factors play a significant role on students' motivation and interest and consequently performance, such as gender and prior experiences (Stephenson et al., 2018; NASEM, 2018). Female students' attrition exceeds and motivation in CS significantly lags that of men (Stephenson et al., 2018; NASEM, 2018). Both CS performance and quality of collaborative discourse is found to be highly influenced by students' gender (Jiang et al., 2017; Wu & Wang, 2019). Female students demonstrate a significantly low self-perceived ability, beliefs and interest towards CS Education than male counterparts (Babes-Vroman & Nguyen, 2020; Quille et al., 2017; Murphy et al., 2019). Students' programming and collaborative programming skills have been extensively studied in different domains and grade-levels but very few studied multiple factors that collectively influence programming at elementary level. In terms of pair-programming, most of the research to date has been done on college-level pair-programming instructions (Umapathy & Ritzhaupt 2017; Williams et al., 2000). In summary, even though collaborative work is, generally, a widespread practice in the elementary grades, there is a dearth of knowledge regarding psychological mechanisms influencing pair-programming practices at this grade level. This can be addressed by studying impacts of students' personal attributes (i.e., gender and motivational factors such as self-theories about abilities, self-efficacy and interest in programming) on their programming and collaboration performances and simultaneously designing instructional interventions for enhancing student performance. Such studies will help researchers, instructors, and policymakers to make informed decisions about implementation of pair programming in elementary classrooms.

Purpose Statement

The goal of our research is to explore the role of gender and motivational beliefs on students' collaborative and programming performances in pair programming contexts and design an intervention to support collaboration during pair programming. We will conduct three studies to accomplish the goal. The first study will explore how and if gender and prior experiences, mediated by motivational factors (mindset, interest and self-efficacy) influence individual students' programming related conceptual understanding; Second, we will qualitatively compare different gender groupings based on the quality of dyad's collaboration and programming learning; Second,; Third, we will examine a intervention of providing feedback on both productive collaboration and motivational factors aiming to help students better collaborate regardless of their gender and prior experiences.

Research Objectives

To accomplish the purpose of our research, we will utilize the following three research objectives:

Research Objective 1: The exploration of impact of gender and prior experience, mediated by motivational factors: Mindset, Self-Efficacy and Interest, on students' programming performances.

Research Objective 2: The case comparison of students' pair-programming performance based on different gender groupings.

Research Objective 3: The examination of a feedback intervention on students' collaboration during pair programming.

We will conduct separate studies to explore these three objectives. The three studies are represented by consecutive chapters in this dissertation.

Significance of the Research

Programming in elementary school is a highly promising strategy for developing students' efficacy and interest in computer science, helping motivate them to excel in this skill and use it in future education and career trajectories. However, we know very little about students' current practices in programming environments and what influences them. This study provides new insights on the influence of gender and motivation in programming, the ground truth measure of current practices in an efficacious programming environment such as pair-programming, and how to facilitate students to better perform in pair programming contexts.

- This research will shed light on how gender, prior experience, and motivational beliefs and interests intercorrelate and influence students' learning in CS. The findings may direct educators and researcher toward factors related to programming practices that needs most attention for elementary classroom.

- It will provide insights on how different gender pairings such as girl-girl, boy-girl and boy-boy pair’s collaborative practices and programming compare. These findings will help researchers and educators support pair programming for both male and female students through the design of effective interventions and formative assessment practices.
- This study will produce insights into students’ natural tendencies in collaborative discourse in pair programming that will help researchers further study pair-programming instructions for elementary students.
- Finally, this study will design and examine an intervention to engage students in high quality collaboration. Educators can utilize this intervention to help students better collaborate not only in programming, but also in any computer-supported collaborative environments.

Substantive Content Theories

Exploratory Talk

At the heart of the instructional impact of collaborative work is the constructive discourse between students that moves learning forward (Tudge, 1992). A framework by Mercer (2002) elaborates on students’ dialogue, distinguishing more productive conversation from the less productive ones. Drawing on Vygotsky’s Zone of Proximal Development (ZPD; Vygotsky, 1978) and Bruner’s ‘Scaffolding’ (Bruner, 1978), 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, 2002). In a teacher-student or peer-peer joint activity, three types of conversations can occur which Mercer termed Cumulative, Disputational and Exploratory talk (Table 1). In Cumulative talk, speakers build positively but uncritically on what the other has said, whereas Disputational talk is characterized by disagreement and individualized decision making. In Exploratory talk, participants engage critically but constructively with each other's ideas, which leads to improved reasoning and conceptual understanding (Mercer, 2002; Bennett & Cass, 1989). Exploratory talk has been shown to expand the joint ZPD by enabling partners to achieve a better mutual understanding of the problem (Fernández, Wegerif, Mercer, & Rojas-Drummond, 2015). Exploratory talk produces improved arguments, and critically considers alternative views during collaborative work (Rojas-Drummond & Zapata, 2004). Mercer et al. (1999) proposed a set of ground rules for Exploratory talk to use as guidance to facilitate collaborative discourse. With teachers' support, students could participate in Exploratory talk by appropriating partner's ideas and strategies, co-constructing new ideas to contribute to the collaborative activity critically and productively (Warwick, Mercer, & Kershner, 2013).

Self-theories about abilities (Mindsets)

Students' Mindset significantly impacts their view of employing effort on a particular task. Students with a growth mindset believe that effort can foster their ability, whereas students with a Fixed Mindset believe that ability is static, and effort would not change that state substantially (Blackwell et al., 2007). Thus, reactions to setbacks or failures also vary between the students of these two Mindsets (Lin-Siegler et al., 2016). According to Dweck, students with a Fixed Mindset believe that the intelligence or ability of a person cannot be substantially changed (Dweck & Bempechat, 2017). Their reactions or attribution of setbacks are more likely to accuse their innate brain capability (Lin-Siegler et al., 2016). As a result, students with a Fixed Mindset are less inclined to exert effort on a task and set challenging goals for themselves (Dweck, 2006, 2013). Students with a growth mindset hold self-beliefs very opposite of people with a fixed mindset. They believe people can substantially change their ability or intelligence through effort (Dweck, & Bempechat, 2017). Students may attribute their success to the amount and quality of effort they exert to a task (Dweck, 1999). Such a belief helps them to set learning goals and engage in strategic self-regulation more actively than students with a Fixed Mindset (Dweck & Master, 2012). Growth mindset beliefs allow them to think that they can achieve a goal regardless of its difficulty level, which gives them long-term perseverance.

Self-efficacy belief

Self-efficacy is another expectancy belief that defines an individual's perceived capabilities to perform a task and achieve specific results (Pajares, 1996). Bandura, the author of the construct, defined it as "Perceived self-efficacy is concerned with people's beliefs in their ability to influence events that affect their lives. This core belief is the foundation of human motivation, performance accomplishments, and emotional well-being (Bandura, 1997, 2006)." Unlike the self-theories or mindsets, self-efficacy judgments are task- and situation-specific (Bandura, 1986, 1989; Pintrich & Schunk, 1995). Self-efficacy is a key contributor to student attribution of success and failure (Bandura, 2013).

Interest

The expectancy-value theory conceptualizes interest (or intrinsic value) as an essential aspect of the task value (along with attainment, utility, and cost; Eccles, 1983; Wigfield et al., 2015). Comparable with the construct of intrinsic motivation (Deci et al., 1991; Collis et al., 2012), interest is defined as "the enjoyment one gains from doing the task" (Wigfield & Cambria, 2010, p. 4). Hidi and Renninger (2006) refers to interest as the preferred engagement of students with a specific topic, activity, or idea, enduring a predisposition toward these objects. According to the person-object theory of interest, interest develops through ongoing interactions between the related environment and the person (Krapp, 2000, 2005). Interest can be both situational and individual (general) interest (e.g., Hidi, 1990; Krapp et al., 1992; Krapp, 2000, 2005). Recent research has stressed that interest is a domain-specific construct and that recommends measuring interest separately for different school subjects (e.g., CS interest; Gogol et al., 2017).

Definitions

Pair programming

Pair-programming is a practice of collaborative programming where two programmers or coders work side by side on the same platform to solve a programming problem (Williams, Kessler, Cunningham, Jeffries, 2000). Block-based programming environments (i.e Scratch, NetsBlox) are generally used with younger students to introduce them to programming (Weitrop, 2019). There are different ways collaboration takes place using these environments. Students can perform one-computer pair-programming where one person, the driver, controls the programming environment, and the navigator monitors the progress and anticipates future steps (Hanks et al., 2011). Contemporary environments such as NetsBlox allow participants to interact through a shared, virtual workspace where they can collaborate by working on the same project simultaneously. In classrooms, students collaborate in computers situated proximally side-by-side, so they can engage in direct interaction with each other under the guidance of a teacher.

References

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.

Babes-Vroman, M., & Nguyen, T.D. (2020). Gender Diversity in Computer Science at a Large Research University. *ArXiv, abs/2004.13760*.

Babes-Vroman, M., Juniewicz, I., Lucarelli, B., Fox, N., Nguyen, T., Tjang, A., ... & Chokshi, R. (2017, March). Exploring gender diversity in CS at a large public R1 research university. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 51-56).

Bandura, A. (1986). The explanatory and predictive scope of self-efficacy theory. *Journal of social and clinical psychology*, *4*(3), 359-373.

Bandura, A. (1989). Regulation of cognitive processes through perceived self-efficacy. *Developmental psychology*, *25*(5), 729.

Bandura, A. (2013). Regulative function of perceived self-efficacy. In *Personnel selection and classification* (pp. 279-290). Psychology Press.

Bennett, N., & Cass, A. (1989). The effects of group composition on group interactive processes and pupil understanding. *British Educational Research Journal*, *15*(1), 19-32.

Barnes, T. (2017). C.S. for all, equity, and responsibility. *ACM SIGCSE Bulletin*, *49*(2), 18-18.

Biggers, M., Brauer, A., & Yilmaz, T. (2008). Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers. *Acm sigcse bulletin*, *40*(1), 402-406.

Blackwell, L. S., Trzesniewski, K. H., & Dweck, C. S. (2007). Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention. *Child development*, *78*(1), 246-263.

Bruner, J. (1985). The role of interaction formats in language acquisition. In *Language and social situations* (pp. 31-46). Springer, New York, NY.

Collis, J. M., Messick, S. J., & Schiefele, U. (Eds.). (2012). *Intelligence and personality: Bridging the gap in theory and measurement*. Psychology Press.

Dweck, C. S., & Bempechat, J. (2017). Children's theories of intelligence: Consequences for learning. In *Learning and motivation in the classroom* (pp. 239-256). Routledge.

Dweck, C. S. (2013). *Self-theories: Their role in motivation, personality, and development*. Psychology press.

Dweck, C. S., & Master, A. (2012). Self-theories motivate self-regulated learning. In *Motivation and self-regulated learning* (pp. 43-64). Routledge.

Fernández, M., Wegerif, R., Mercer, N., & Rojas-Drummond, S. (2015). Re-conceptualizing" scaffolding" and the Zone of Proximal Development in the Context of Symmetrical Collaborative Learning. *Journal of Classroom Interaction*, 54-72.

Eccles, J. (1983). Expectancies, values and academic behaviors. *Achievement and achievement motives*.

Gogol, K., Brunner, M., Martin, R., Preckel, F., & Goetz, T. (2017). Affect and motivation within and between school subjects: Development and validation of an integrative structural model of academic self-concept, interest, and anxiety. *Contemporary Educational Psychology*, *49*, 46-65.

Hidi, S. (1990). Interest and its contrubution as a mental resource for learning. *Review of Educational Research*, 60, 549–571

Hidi, S., & Renninger, K. A. (2006). The four-phase model of interest development. *Educational psychologist*, *41*(2), 111-127.

Krapp, A. (2005). Basic needs and the development of interest and intrinsic motivational orientations. *Learning and Instruction*, 15, 381–395.

Lin-Siegler, X., Dweck, C. S., & Cohen, G. L. (2016). Instructional interventions that motivate classroom learning. *Journal of Educational Psychology*, *108*(3), 295.

Mercer, N. (2002). *Words and minds: How we use language to think together*. Routledge.

Mercer, N., Wegerif, R., & Dawes, L. (1999). Children's talk and the development of reasoning in the classroom. *British Educational Research Journal*, *25*(1), 95-111.

Murphy, S., MacDonald, A., Wang, C. A., & Danaia, L. (2019). Towards an understanding of STEM engagement: A review of the literature on motivation and academic emotions. *Canadian Journal of Science, Mathematics and Technology Education*, *19*(3), 304-320.

National Academies of Sciences, Engineering, and Medicine (Ed.). (2018). *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments.* Washington, DC: The National Academies Press.

Navlakha, S., & Bar-Joseph, Z. (2011). Algorithms in nature: the convergence of systems biology and computational thinking. *Molecular systems biology*, *7*(1), 546.

Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of educational research*, *66*(4), 543-578.

Pintrich, P. & Schunk, D. (1996). *Motivation in Education: Theory, Research & Applications*, Ch. 3.

Englewood Cliffs, NJ: Prentice-Hall

Rojas-Drummond, S., & Zapata, M. P. (2004). Exploratory talk, argumentation and reasoning in Mexican primary school children. *Language and education*, *18*(6), 539-557.

Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142–158. https://doi.org/10.1016/j.edurev.2017.09.003

Stephenson, C., Miller, A. D., Alvarado, C., Barker, L., Barr, V., Camp, T., ... & Zweben, S. (2018). *Retention in computer science undergraduate programs in the us: Data challenges and promising interventions*. ACM.

Tudge, J. (1992). Vygotsky, the zone of proximal development, and peer collaboration: Implications for classroom practice.

Quille, K., Culligan, N., & Bergin, S. (2017, June). Insights on Gender Differences in CS1: A Multi-institutional, Multi-variate Study. In *Proceedings of the 2017 acm conference on innovation and technology in computer science education* (pp. 263-268).

Umapathy, K., & Ritzhaupt, A. D. (2017). A meta-analysis of pair-programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education (TOCE)*, *17*(4), 1-13.

Vygotsky, L. (1978). Interaction between learning and development. *Readings on the development of children*, *23*(3), 34-41.

Warwick, P., Mercer, N., & Kershner, R. (2013). 'Wait, let's just think about this': Using the interactive whiteboard and talk rules to scaffold learning for co-regulation in collaborative science activities. *Learning, Culture and Social Interaction*, *2*(1), 42-51.

Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, *62*(8), 22-25.

Wigfield, A., & Cambria, J. (2010). Students' achievement values, goal orientations, and Interest: Definitions, development, and relations to achievement outcomes. *Developmental review*, *30*(1), 1-35.

Wigfield, A., Eccles, J. S., Fredricks, J. A., Simpkins, S., Roeser, R. W., & Schiefele, U. (2015). Development of achievement motivation and engagement. In M. E. Lamb & R. M. Lerner (Eds.), *Handbook of child psychology and developmental science: Socioemotional processes* (pp. 657–700). John Wiley & Sons. https://doi.org/10.1002/9781118963418.childpsy316

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

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

CHAPTER 2: VARIANCES IN ELEMENTARY STUDENTS' PROGRAMMING CONCEPTS: IMPACT OF GENDER, PRIOR EXPERIENCE MEDIATED BY MINDSET, SELF-EFFICACY, AND INTEREST

Abstract

Background Before our education policy moves forward with scaling computer science (CS) education throughout the elementary system, it is crucial to understand constructs that have evidence in impacting CS learning such as the impact of gender and prior experience mediated by motivational beliefs and interest in CS. However, there is a dearth of research that explores these constructs altogether in the context of elementary CS education.

Objective In this study, we investigate the influence of gender mediated by prior experience and motivational factors (mindset, self-efficacy, and interest in programming) on students' CS conceptual understanding. In doing so, we also validate an instrument to measure students' self-perceived ability beliefs (mindset and self-efficacy) and interest in CS.

Method We collected data from 210 fourth and fifth grade students. First, we validated the instrument using a combination of classical test theory and item response theory focusing on construct validity, item difficulty, and reliability. We then tested a hypothesized path model and compared that to a revised model to identify the relationship between gender, prior experiences, mindset, interest, and self-efficacy with CS conceptual understanding.

Findings The study findings highlight the value of prior knowledge in CS at elementary level. Gender is found to directly impact students' interest in CS. In Terms of motivational beliefs, Self-efficacy was found to be the strongest predictor of CS performance whereas, Growth Mindset and interest-only mediates its effects.

Implications The findings of this study have substantive and methodological implications for those exploring issues relevant to students' individual differences in motivational beliefs, gender, and prior experience in the field of elementary level CS education.

Introduction

A projection by the U.S. Bureau of Labor & Statistics indicates that occupation growth in computer science and math occupations will increase by 12.7% from the 2018-28 decade, which is one of the most rapidly growing occupation fields in the US (Dubina, Morisi, Rieley, & Wagoner, 2019). A report by the National Academies of Sciences, Engineering, and Medicine (2018) also shows that despite a steady increase in computer science-related bachelor's degrees, it's not yet enough to fill the jobs available in the field. CS educators and education policymakers have long recognized that part of the solution will need to come from increasing the pool of students interested in pursuing and staying in this academic trajectory (Biggers et al., 2014; Petersen et al., 2016).

Although students at the elementary level are at an early stage of their education and may not be aware of CS-related academic and career paths. Their experience with programming can have a significant impact on their motivation to pursue higher education in this field. Literature shows that prior experiences in programming are beneficial for motivation (i.e. self-efficacy, and enjoyment) and thus retention in college-level CS courses (Biggers et al., 2008; Byrne & Lyons, 2001; Ramalingam et al., 2004). It is important that these prior experiences are well aligned with the practices employed by the discipline, in this case, programming and not simply general computer application use. Studies found that students majoring in CS often considered programming (i.e., block-based programming) as a relevant prior experience rather than other computing experiences (e.g., Microsoft Office or HTML) (Lewis et al., 2011). Studies explicitly looking at experiences of students who dropped out from CS majors found that most of them had little to no prior experiences with programming (Biggers et al., 2008).

Despite the benefits of prior experiences, it is also important to address that both the level of prior experience and its relationship to motivation towards the CS domain are not the same for all demographic categories of students. Research shows a significant difference in prior experiences based on gender; male students tend to have higher experience with programming prior to college be that of advanced courses at high school or self-taught (Babes-Vroman, M., & Nguyen, T.D. 2020). Similarly, female students' attrition and motivation in CS significantly lags that of men (Barr, 2018; NASEM, 2018). The role of motivational factors in overall CS performance is also found to be highly influenced by the student's gender (Jiang et al., 2017; Wu & Wang, 2019).

Students' gender and gender-based experiences may impact their beliefs about ability in a certain subject area which may, in turn, impact their motivation to learn that subject (Dicke et al. 2019). Self-beliefs about abilities, self-efficacy beliefs, and interest are motivational constructs that act interdependently to influence students' engagement, achievement, and attrition in a task field. These constructs are also content-specific and represent a personal significance between the individual and the object of their interest (Schiefele, 1991; Renninger and Hidi, 2002). Thus it is important to understand how individual attributes like gender and prior experience relate to motivational constructs influencing students' achievement in the CS content area. Much research has been done to identify how motivational factors impact students' learning in CS (e.g.), and the gender disparity in CS performance (e.g.), however, it is less common to find research that has explored these factors collectively and how they associate to students' performance, especially at the elementary level.

Before our education policy moves forward with scaling CS education throughout the elementary system, it is crucial to understand these relationships to design interventions that can provide equal access to CS learning. Researchers need to explore these constructs in the context of elementary CS education for three main purposes: 1) Discerning students' beliefs and it's relation to other factors as interest and self-efficacy in CS would guide educators to identify aspects that need intervening, 2) Identifying any gender disparity in students' self-perceptions of abilities and interest in young students to address equity issues regarding this content area, and 3) Understanding elementary students current experiences with programming (formally or informally) and how that experience impacts their motivation and performance is crucial in deciding the timing of introducing pair-programming activities to elementary students. Research in these areas will not only inform the research field but also guide the development of effective instructional strategies to address them.

For studying such a combination of motivational constructs, one challenge is that these motivational factors lack diverse options of validated instruments that target elementary-level CS Education. One of the instruments that were validated for elementary CS education has explored one of two constructs such as self-efficacy and outcome expectancy (Vandenberg et al., 2021). To explore other factors such as mindsets toward ability, interest, and self-efficacy with detailed CS concepts, more instruments need to be validated.

In this study, we investigate how the exogenous variable gender influence endogenous variables prior experiences and three motivational factors (mindset, self-efficacy, and interest in programming) affecting students' CS performance (i.e., CS Concept Assessment; Vandenberg et al., 2021). The findings from this study can both inform core motivational research and inform strategies for motivating students towards learning CS and reduce the gaps in students' experiences with CS. The study also provides a validated instrument to measure elementary students' motivational beliefs and interests. We explored the following research questions in this study:

1. What are the content validity, dimensionality, internal structure, and reliability of the instrument - Elementary Self-perceived Ability Beliefs and Interest in Computer Science (E-SABICS)?

2. How does the intercorrelation of gender, prior experience, and the motivational factors: Mindset, Self-Efficacy, and Interest, described by a path analysis, explain the variations in CS conceptual understanding?

Theoretical Framework

Self-theories about abilities (Mindsets)

Students' Mindset significantly impacts their view of employing effort on a particular task in order to gain knowledge or ability. Students with a growth mindset believe that effort can foster their ability, whereas students with a Fixed Mindset believe that ability is static, and effort would not change their ability state substantially (Blackwell et al., 2007). Thus, reactions to setbacks or failures also vary between the students of these two Mindsets (Lin-Siegler et al., 2016). According to Dweck, students with a Fixed Mindset believe that the intelligence or ability of a person cannot be substantially changed (Dweck & Bempechat, 2017). Their reactions or attribution of setbacks are more likely to target their innate brain capability (Lin-Siegler et al., 2016). As a result, students with a Fixed Mindset are less inclined to exert effort on a task and set challenging goals for themselves (Dweck, 2006, 2013). Students with a growth mindset hold self-beliefs opposite of people with a fixed mindset. They believe people (themselves included) can substantially change their ability or intelligence through effort (Dweck, & Bempechat, 2017). Students may attribute their success to the amount and quality of effort they exert to a task (Dweck, 1999). Such a belief helps them to set learning goals and engage in strategic self-regulation more actively than students with a Fixed Mindset (Dweck & Master, 2012). Growth mindset beliefs allow them to think that they can achieve a goal regardless of its difficulty level, which gives them long-term perseverance.

Self-efficacy belief

Self-efficacy is another expectancy belief that defines an individual's perceived capabilities to perform a task and achieve specific results (Pajares, 1996). Bandura, the lead theorizer of the construct, defined it as "Perceived self-efficacy is concerned with people's beliefs in their ability to influence events that affect their lives. This core belief is the foundation of human motivation, performance accomplishments, and emotional well-being (Bandura, 1997, 2006)." Unlike the self-theories or mindsets, self-efficacy judgments are task- and situation-specific (Bandura, 1986, 1989; Pintrich & Schunk, 1995). Self-efficacy is a key contributor to student attribution of success and failure around specific tasks and contexts (Bandura, 2013).

Interest

The expectancy-value theory conceptualizes interest (or intrinsic value) as an essential aspect of the task value (along with the values of attainment, utility, and cost; Eccles, 1983; Wigfield et al., 2015). Comparable with the construct of intrinsic motivation (Deci et al., 1991; Collis et al., 2012), interest is defined as "the enjoyment one gains from doing the task" (Wigfield & Cambria, 2010, p. 4). Hidi and Renninger (2006) refers to interest as the preferred engagement of students with a specific topic, activity, or idea) enduring a predisposition toward these objects. According to the person-object theory of interest, interest develops through ongoing interactions between the related environment and the person (Krapp, 2005). Interest can be both situational and individual (general) interest (e.g., Hidi, 1990; Krapp et al., 1992; Krapp, 2005). Recent research has stressed that academic interest is, like self-efficacy, a domain-specific construct and that recommends measuring interest separately for different school subjects (e.g., CS interest) (Gogol et al., 2017).

Hypothesized Model

Relationship between the Motivational Factors and Learning Outcome: *Mindsets, Interest and Self-efficacy*

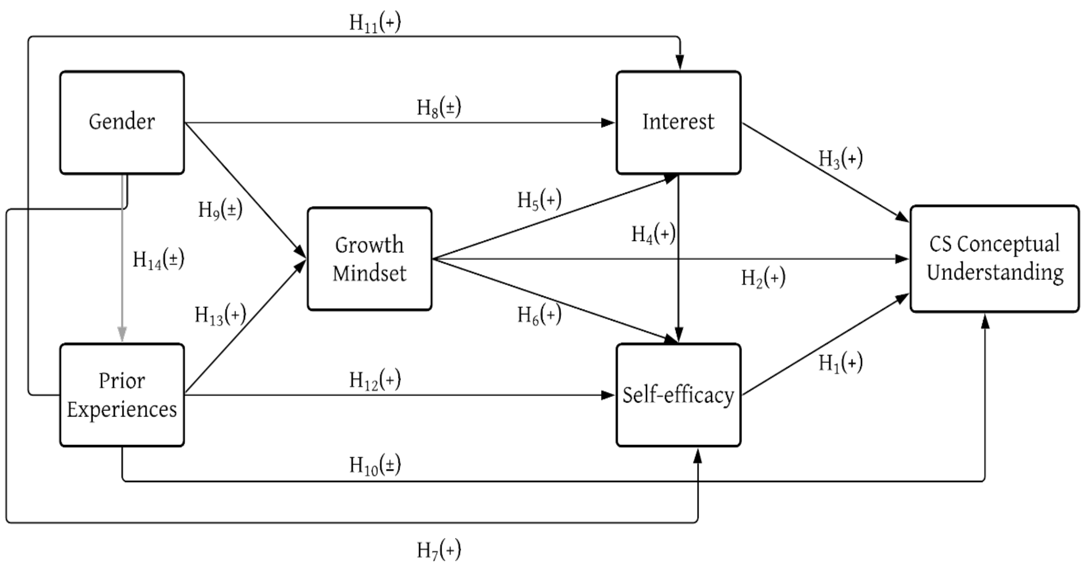


Figure 1: Hypothesized Path Model.

Student motivation can be described as a student's willingness, beliefs and desire, and compulsion to participate and be successful in the learning process (Bomia et al., 1997). Students' motivation, or the lack of it, has been attributed as one of the primary reasons for attrition from academic courses, including introductory CS (CS1) courses (Kinnunen & Malmi, 2006). Researchers found that learning to program at advanced levels requires constant practice (Settle, et al., 2014), and to do so, students' motivation is of utmost importance. Previous studies show that students perform better in CS if they are motivated with the beliefs that they can perform well in CS, the ability to grow and perform better in CS, and/or have a high Interest in CS (Flanigan et al., 2015; Scott & Ghinea 2013). Beliefs on whether one can grow their existing level of knowledge and abilities are called Mindsets about abilities by Carol Dweck (1999), which have been found to impact students' goals and motivation significantly. A nationwide study on Mindsets found that a growth mindset improves students' achievement (Yeager et al., 2019). Complementary to this, the belief that is concerned with a confidence level in the capability of executing a particular task is called Self-efficacy (Shell et al., 2013; Bandura, 2013). In a comprehensive synthesis of over 800 meta-analyses, Self-efficacy was identified as the most potent predictor of learning outcomes (Hattie, 2008). Another construct that is often related to both Mindsets and Self-efficacy is students' interest in a domain area (Huang et al., 2019; Vongkulluksn et al., 2018). Students' beliefs about their ability to perform well, confidence in showing efficacy in a task, and interest in the task can significantly predict their engagement, performance, academic subject selection, and career aspirations in that task area (Murphy et al., 2019). Similar to Mindsets and Self-efficacy, Interest is also found to positively impact students' achievement in a domain area and influence their choices of learning a particular subject (Jansen et al., 2016; Köller et al., 2001).

Recent studies regarding students' motivation towards a task are increasingly exploring these three constructs altogether because of their interdependence and mediation on student performance (i.e., Bai et al., 2021; Fryer, 2015; Wigfield & Cambria, 2010). Studies posit that Self-theories about one's own abilities (Mindset) can be considered an antecedent of students' Interest and Self-efficacy in learning (Burnett et al., 2020; Schmidt et al. 2017). What students believe about their ability to perform well in a task determines if they are interested in engaging with that task. In the same way, their or Mindsets can also influence how confident they feel they are capable of executing the task (a.k.a, self-efficacy on the task). Recent studies show that a growth mindset may increase students' Interest in the STEM disciplines (Degol et al., 2018). Researchers explored the association between the Mindset theories towards a subject with students' Self-efficacy in that subject area. Bandura and his colleagues have demonstrated a causal association between these two variables by manipulating Mindset through experiments focused on acquiring complex decision-making skills (Bandura & Wood, 1989; Jourden, Bandura, & Banfield, 1991). Results showed that a growth mindset fostered Self-efficacy while a fixed Mindset either decreased or fostered no growth in Self-efficacy. In addition, studies found that in domain areas like mathematics and English, upper elementary students with a growth mindset tend to have higher Self-efficacy than those with a fixed mindset (Huang et al., 2019; Abdullah, 2008; Young and Urdan 1993). Bråten and Stromso (2005) also demonstrated that a fixed Mindset negatively predicted Self-efficacy; students who have a fixed Mindset had lower Self-efficacy in certain problem-solving activities. At the undergraduate level Davis et al., 2011) showed that college students with a fixed mindset were more prone to feeling helpless in challenging situations and had low Self-efficacy.

This brief literature review has demonstrated interesting and compelling connections between these constructs. However, these constructs have rarely been studied together in computer science education, let alone at the elementary level. Based on this literature, we predict that all three of these motivational constructs would positively affect students' CS performance. As self-theories are not a task-specific construct, we predict that (a general) growth mindset will have an indirect effect through domain-specific interest and self-efficacy. However, CS interest and self-efficacy being more contextually targeted constructs, would have a direct effect on CS performance. Here are the hypotheses concerning the relationship of these three constructs with each other and with CS Conceptual Understanding:

H1: Self-efficacy in programming will show a positive direct effect on CS Conceptual Understanding.

H2: Interest in programming will show a positive direct effect on CS Conceptual Understanding.

H3: Growth Mindset will show a positive direct effect on CS Conceptual Understanding.

H4: Interest in programming will show a positive direct effect on Self-efficacy.

H5: Growth Mindset will show a positive direct effect on students' Interest in programming.

H6: Growth Mindset will show a positive direct effect on students' Self-efficacy toward programming.

Gender influence on CS Education

Research in the past few decades reported that CS is more male-dominated than mathematics and chemistry (Beyer, 1999). Increasing female representation in CS-related academic tracks continues to be a challenge (NASEM, 2018). Early gender research in CS Education shows mixed results. Some studies found male students performing better than female students in college-level CS courses (i.e.), while others found no such difference (Beyer et al., 2003). However there is general agreement in this body of literature that students incorrectly perceive CS as a masculine subject and believe that males perform better than females in this domain (Beyer, 1999; Heenwood, 1999; Papastergiou, M. 2008). These earlier studies were conducted more than ten years ago, and since then there has been strong policy momentum to improve exposure and experiences in K-16 CS for females and had shown to double participation in CS within 3 years (Board, 2019). More recent studies from the past five years predominantly showed that female students perform either the same or significantly higher in CS learning outcomes (Malik & Coldwell-Neilson, 2018; Sharma & Shen, 2018; McBroom, 2020). Nevertheless, enrollment of female students is still very low compared to male students (McBroom, 2020).

When looking for evidence of females students' motivation to pursue CS education, recent studies show that female students still demonstrate a significantly low self-perceived ability, beliefs, and interest towards CS Education than male counterparts (Babes-Vroman & Nguyen, 2020; Quille et al., 2017; Murphy et al., 2019). Male students have more enthusiasm for programming, which is found to influence their higher learning outcomes than females (Rubio et al., 2015). Studies exploring the reason behind the lack of motivation or enthusiasm to pursue CS suggests that the differences in prior experiences such as familiarization with computing in the home and the academic environment differentiate boys' and girls' motivation towards CS (Papastergiou, 2008). Despite being researched so widely, there is no consensus on how the motivational variables described above interact with gender to shape academic outcomes in CS (Gunbatar, 2018). Moreover, nearly all the referenced literature is contextualized in higher education or at the high school level. As previously noted, the motivational research literature, including studies focused on gender are scarce at the elementary level. Such studies can help guide educators and policymakers with regards to early CS education interventions.

Based on prior literature with older students, we predict that Gender will have a significant impact on self-perceived abilities and interests. Our hypothesis is:

H7: Students' gender will show a direct effect on students' Self-efficacy in programming.

H8: Students' gender will show a significant effect on students' Interest in programming.

H9: Students' gender will show a significant effect on students' Growth Mindset level.

Relationship between Prior Experiences and Motivational Factors

Previous experience in programming was found to have a relationship to retention, Self-efficacy, and enjoyment in college-level CS courses (Biggers et al., 2008; Byrne & Lyons, 2001; Ramalinga et al., 2004). Students majoring in CS often considered programming (i.e., block-based programming) as relevant previous experiences but also considered other computing experiences (e.g., Microsoft Office or HTML) as relevant (Lewis et al., 2011). One study found that computing outreach events have a long-term impact on developing an interest in computing (Lakanen, A. J. & Isomöttönen, 2018). Students with little to no programming experiences prior to entering college more frequently leave CS programs (Biggers et al., 2008). These students are less interested in programming and nurture a perception that the environment in CS is an asocial and coding-only field with no connection to the real world (Tafliovich et al., 2013). Aivaloglou & Hermans (2019) found that previous programming experience strongly impacts extrinsic motivation, CS career orientation, and self-efficacy at the college level. As with the earlier cited literature, the studies referenced in this section are at high school above. Still, considering them as the most relevant point of reference, we hypothesize that there is a correlational interplay at the elementary level between prior experience, the motivational factors, and CS understanding:

H10: Students' prior experience with programming will show a direct effect on students' CS Conceptual Understanding.

H11: Students' Prior Experience with programming will show a significantly positive effect on students' Interest in programming.

H12: Students' prior experience with programming will show a significantly positive effect on students' Self-efficacy in programming.

H13: Students' Prior Experience with programming will show a significantly positive effect on students' Growth Mindset level.

H14: Students' Gender with programming will show a significant effect on students' Prior Experience level.

Methodology

Participants and contexts

We conducted this study on a sample of 210 consented students attending four elementary schools in the southeastern region of the United States. 9-11 years old participants were from two grade levels: 58% 5th grade, 42% 4th grade. Demographics of the participants are presented in Table 1. We obtained parental consent and students' assent from all the students participating in this study. Although our sample size is 210, The requirement for some analyses to have complete cases reduced the effective sample (complete cases = 148) in some instances.

Table 1 Demographics and Prior Experience Frequencies of the Participants.

		f	Percentage
Gender	Female	118	56.2
	Male	92	43.8
Age	8	4	1.9
	9	75	35.7
	10	96	45.7
	11	32	15.2
	12	1	0.5
Race/Ethnicity	Asian	13	6.2
	Black	24	11.4
	Multi	12	5.7
	White	137	65.2
	Hispanic/Latino	15	7.1
	Native American	5	2.4
	Other	4	1.9
Grade	4	143	68.1
	5	67	31.9
School*	Excel Elementary	78	37.1
	Hillary Elementary	62	29.5
	Fairway Elementary	34	16.2
	Parker Elementary	36	17.1
Prior Experience	Never	66	31.4
	Sometimes	88	41.9
	Always	56	26.6

**Pseudo names.*

Data sources

Self-belief on ability, Interest, and Self-efficacy

The motivational constructs self-report instrument Elementary Self-perceived Ability Beliefs and Interest in Computer Science (E-SABICS) has 42 items, along with a demographics questionnaire administered to the students. The questionnaire included the items for three constructs: Mindset, Interest in programming, and Self-efficacy towards programming.

Eight items for Mindsets were modified from Dweck's (2014) original Mindset items. These items asked students about their self-beliefs in the malleability of abilities, both general and programming-related abilities. Four items measured students' Mindsets on general abilities, and four items measured Mindsets on programming abilities.

Twenty-four items were used to measure Self-efficacy towards programming and collaboration. These items measure students' self-efficacy on specific programming concepts (variables, conditionals, loops, algorithms), logical thinking skills, control of the programming environment, debugging programs, and collaboration. We modified all these items from validated instruments that have proven to effectively measure programming (Kukul et al., 2017; Tsai, 2019; Williams et al., 2003; Quade, 2003) at different levels of education. We had to extensively modify the language of most of the items as all these studies were on college-level students, except one, which was on middle school students.

Seven items were used to measure students' Interest in programming. As interpreted by Schiefele (1991), Interest is the "relatively long-term orientation of an individual toward a type of object, an activity, or an area of knowledge. Schiefele identified two components of Interest: feeling-related valences such as feelings of enjoyment and involvement and Value-related valences that refer to the attribution of personal significance such as competence and self-development. These items asked students about the feeling-related valence like "I like coding," or “Coding is boring” and value-related valence such as “I will need coding for my future work.” Items were modified from validated instruments made for college-level students (Williams et al., 2003; Lamb et al., 2012)

Programming performance

Students' computer science conceptual understanding was measured with a validated instrument called CS Concept Assessment (CSCA), validated by Vandenberg et., 2021 on the a sample of students that also includes part of sample used in this study. These items measured students’ understanding of the four core concepts of CS (variables, conditionals, loops, and algorithms), paralleling the same concepts measured on the self-efficacy instrument. The assessment consists of 20 multiple-choice questions related to computer programming in a block-based programming context.

Gender, and Prior Experiences

As part of demographics questions, students were asked about their gender, ethnicity, and prior experiences. In terms of prior experiences, students were asked how often they participated in coding. They responded on a 5-point Likert scale.

Instrument Validation Procedure

To answer research question 1, we will test the dimensionality, internal structure, and reliability of the E-SABICS instrument. For that, we used a combination of the classical test theory and item response theory-Rasch approaches. We first conducted an exploratory factor analysis on the entire item set of the instrument to determine if the number of factors matches the number of theoretical constructs we are measuring. We evaluated the instrument for sampling adequacy using Kaiser–Meyer–Olkin (KMO) measure of sampling adequacy test and Bartlett's test of sphericity for each of the three theoretical constructs in our instrument. KMO is a diagnostic measure for assessing the extent to which the indicators of a dimension belong together, where a value above .6 is acceptable but a value above .8 is considered meritorious (Kaiser, 1974). The factor models themselves were evaluated in terms of Scree plots, factor loadings, theoretically expected factor structure and interpretability. The factor analysis was conducted using SPSS with the Principal Axis Factoring extraction method and Oblimin with Kaiser Normalization rotation method with delta settings (Costello & Osborne, 2005). As recommended by Tabachnick et al., (2007), items that showed loadings greater than 0.32 on more than one factor were dropped from the revised instrument or had factor loading less than .4 across all the factors. We used SPSS version 26 to run the exploratory factor analysis.

After the initial construct validity check, we explored the dimensionality and items quality within the revised instrument using a multidimensional IRT-Rasch model. Using the multidimensional Rasch analysis students' raw scores for Mindset, Interest and Self-efficacy were transformed into the ratio-interval unit (logit). The logit scales are based on the "person ability" and "item difficulties." of the instrument. Because a raw Likert rating scale measure is not typically considered linear for all items and for all persons (Boone et al., 2013), we used the logit scores for the path analyses as well.

Two models were compared in this dimensionality analysis: one- and three-dimensional models. A one-dimensional model that groups all items in a single factor was used as the baseline. The three-dimensional model was based on our theoretical conceptualization of the instrument based on three factors: Mindset, Interest, and Self-efficacy in CS (Bandura, 1986; Dweck, 2002; Wigfield & Eccles, 2000). To identify the best fitting model, we used Adams and Wu's recommendation that suggests that the best fitting model is the one that has the lowest values in final deviance across three criteria: Akaike Information Criterion (AIC), Akaike

Information Criterion Corrected (AICc), and Bayesian Information Criterion (BIC). Mean-square (MNSQ), which in our software package shows as infit and outfit values were used to assess the item quality, with the assumption that a well-behaved, high-quality item has a value ranging from 0.60 to 1.40 (Wright & Linacre, 1994). All of these analyses were run in R using the 'TAM' package (Robitzsch et al., 2021).

Next, we conducted a validity test using a correlation coefficient between scores grains instruments with a total score (r-hit) by Pearson Product Moment Correlation technique.

Finally, to measure the internal consistency of the item responses three reliability values were computed using the CTT (Cronbach's alpha) and IRT-Rasch methods (item separation

Reliability (EAP), and person reliability (WLE)). We used the same cut-off value of > .70 to determine an acceptable reliability value (DeVellis, 2016; Linacre, 2012).

Path Analysis

After validation of the instrument, we ran descriptive statistics including skewness and kurtosis on the data to determine the normality. Skewness and kurtosis between -2 and 2 is an indicator of normal distribution (George and Mallery, 2010) .

We tested the hypothesized model illustrated in Figure 1 using a path analysis, a special case of Structural Equation Modeling, on gender, prior experience, converted scores of Mindset, Interest, Self-efficacy and the CS Concept Assessment raw scores. Then we evaluated the model using the cutoff values suggested by Schreiber et al. (2006): $\chi 2/df < 3$, TLI > 0.90, CFI > 0.95, RMSEA < 0.06, and RMR < 0.05. If paths are found to be non-significant, we remove those non-significant paths, rerun the analysis and, finally, compare the fit indices with that of the hypothesized model to determine the best model that fits the data.

In standard path analysis, even small amounts of misfit can lead to significant chi-square values when sample sizes are moderate to large (N>200; Chen, 2007), and therefore can lead to misinterpretations. By extension, in analyses with small to moderate sample size, complex models with full information estimation of a structural equation model (i.e. path analysis) parameters can lead to inappropriate solutions (small N ≤ 50-100, Moderate N 100, 200 and Large N ≤ 300; Devlinger, 2018). Contemporary research has found a robust alternative to traditional full information path analysis for small sample size which is called bias-corrected factor score path analysis (BCFSPA) (Devlieger & Rosseel, 2017; Kelcey, 2019, 2021). A factor score path analysis addresses small sample sizes by: 1) breaking down the system of equations that was created based on a theory, 2) using separate measurement models for each latent variable, and then 3) estimate a path analysis using the factor scores predicted by the resulting measurement models. This stepwise estimation of the components of the SEM reduces the model complexity for each stage, consequently improving estimation stability and solution admissibility (Kelcy, 2019). Furthermore, it corrects any bias arising in the structural path coefficients in FSPA using Croon's method (Croon, 2002). Using this method, we corrected covariance by dividing the covariance between the factor scores of latent variables (i.e Mindset, Interest, Self-efficacy) by the products of their factor score and loading matrices (Kelcy, 2019). For testing the goodness of fit, the BCFSPA only provides chi-square and we will use that for our study. We used R package 'lavaan' to conduct this analysis using the model codes provided by Ben Kelcy. (Kelcy, 2019; Rosseel, 2021) (Appendix C).

Results

RQ1: Instrument validation

Exploratory Factor Analysis (EFA)

We first ran an EFA model with the initial 41 items in the E-SABICS instrument: 8 items measuring Mindset, 7 items for Interest, and 27 items measuring Self-efficacy. The Kaiser–Meyer–Olkin (KMO) for each of the factors was more than .8 indicating the adequacy of the sample for running factor analysis (Table 2).

Table 2 Test of Sample Adequacy for Factor Analysis.

Constructs	Kaiser-Meyer-Olkin Measure of Sampling Adequacy.	Bartlett's Test of Sphericity *Approx. Chi-Square (df)*
Self-efficacy	0.941	2381.715 (351)
Interest	0.860	401.577(21)
Mindset	0.803	417.076(28)

EFA showed that the data converged into acceptable factor loading of three factors (Table 3). There were Items that showed loadings greater than 0.32 on more than one factor items or had factor loading less than .4 across all the factors that were dropped from the revised instrument. Table 3 highlights 13 items that had either a loading of greater than .32 for two or more items or had factor loading less than .4 across all the factors.

Table 3 Factor loading and Rasch Item Fit Statistics for Initial items.

Item ID	EFA loadings			Rasch item fit	
	Factor1	Factor2	Factor3	Outfit	Infit
M1	0.115	-0.009	0.548	1.292	1.334
M2	0.045	0.225	0.588	1.215	1.278
M3	0.112	0.226	0.504	0.929	1.063
M4	0.14	0.173	0.645	1.010	1.064
M5	0.101	0.16	0.596	1.267	1.246
M6	0.172	0.114	0.621	0.995	0.907
M7	0.238	0.198	0.535	1.362	1.35
M8	0.097	0.214	0.626	1.096	1.176
I1	0.137	0.851	0.239	0.793	0.865
I2	0.384	0.311	0.205	0.971	0.98
I3	0.395	0.368	0.363	1.087	1.042
I4	0.215	0.876	0.191	0.83	0.843
I5	0.217	0.399	0.209	0.988	0.913
I6	0.016	0.723	0.273	0.835	0.802
I7	0.155	0.799	0.194	0.982	0.923
S1	0.358	0.556	0.291	0.776	0.861
S2	0.515	0.177	0.025	1.491	1.399
S3	0.320	0.524	0.095	1.174	1.224
S4	0.479	0.374	0.267	1.087	0.807
S5	0.53	0.146	0.245	0.985	0.923
S6	0.671	0.229	0.292	0.541	0.533
S7	0.567	0.111	0.271	0.91	0.965
S8	0.634	0.213	0.257	0.843	0.992
S9	0.594	0.165	0.417	0.752	0.758
S10	0.885	0.297	0.169	1.147	1.099
S11	0.395	0.391	0.235	1.798	1.659
S12	0.349	0.579	0.320	1.991	1.949
S13	0.539	0.391	0.331	0.744	0.728
S14	0.717	0.118	0.202	0.731	0.72
S15	0.505	0.226	0.221	0.711	0.741
S16	0.349	0.364	0.217	0.805	0.859
S17	0.677	0.298	0.253	0.725	0.713
S18	0.807	0.241	0.354	0.564	0.573

Table 3 (continued)

S19	0.572	0.298	0.225	0.712	0.72
S20	0.527	0.029	0.228	1.181	0.916
S21	0.341	0.382	0.325	0.775	0.796
S22	0.347	0.691	0.318	1.229	1.136
S23	0.598	0.061	0.239	0.815	0.906
S24	0.558	0.292	0.315	0.818	0.845
S25	0.802	0.597	0.252	1.39	1.162
S26	0.773	0.276	0.052	1.089	1.014
S27	0.346	0.347	0.192	0.863	0.874

Extraction Method: Principal Axis Factoring.
Rotation Method: Oblimin with Kaiser Normalization.
Note: Items that had factor loadings more than .32 under multiple factors or a factor loading of less than .4 throughout all the factors were removed. The red blocks indicate removed items and green blocks for retained items.

Along with EFA, we also ran Rasch Analysis on the initial items. The infit and outfit of the items are provided in Table 3. Only a few items were out of the .6 to 1.4 acceptable range. Among them we only kept S2 and S6 as they were very marginally out of the infit and outfit acceptable range but had good factor loadings.

We dropped all these 13 items from our instrument and used the revised 27-item instrument for further analysis that had 8 items for Mindset, 4 items for Interest, and 17 items for Self-efficacy (Figure 2). Table 4 provides the final list of items.

Table 4 Final List of Items.

Item ID	Item	Construct
M1	You can't do much to change how smart you are.	Mindset
M2	Everyone has the ability to change how smart they are.	Mindset
M4	People can learn new things, but can't change how smart they are.	Mindset
M5	You can't change how talented you are in coding.	Mindset
M6	Everyone has the ability to change how talented they are in coding.	Mindset
M8	Even if you already have talent in coding, you can always get better.	Mindset
I1	I like coding.	Interest
I4	Coding is boring.	Interest
I6	I have no interest in solving coding problems.	Interest
I7	I'll need coding for my future work.	Interest
S2	I'm no good at coding.	Self-Efficacy
S5	I can explain my idea of the coding project step by step.	Self-Efficacy
S6	I can make plans to solve the coding problem.	Self-Efficacy
S7	I know where to build the codes.	Self-Efficacy
S8	I can make changes to the code.	Self-Efficacy
S9	I can run codes.	Self-Efficacy
S10	I can discuss the coding problem with my partner.	Self-Efficacy
S13	I can build the code to run correctly.	Self-Efficacy
S14	I can solve the coding problem with different solutions.	Self-Efficacy
S15	I can figure out the steps of coding procedures.	Self-Efficacy
S17	I can create new variables.	Self-Efficacy
S18	I can understand code that uses variables.	Self-Efficacy
S19	I can fix a coding problem that is not working correctly.	Self-Efficacy
S20	I can fix problems in the code.	Self-Efficacy
S23	I can understand code that uses loops.	Self-Efficacy
S24	I understand when I should use loops.	Self-Efficacy
S26	I understand the purpose of conditionals.	Self-Efficacy

Item-Response Theory Rasch

Table 5 Reliability of the scale and each dimension.

	Number of Items	N	WLE	EAP	Cronbach's alpha
Overall Scale	29	210	-	-	0.943
Dimension 1: Mindset	8	210	0.782	0.7841	0.787
Dimension 2: Interest	4	148	0.813	0.641	0.856
Dimension 3: Self-Efficacy	17	148	0.932	0.716	0.950

We ran both one- and three-dimensional models of the E-SABICS instrument that did not have any more misfitting items. The two-dimensional model had lower values of the final deviance criteria (AIC, AICc, and BIC) than the one-dimensional model which indicates that the three-dimensional model is a better fit (Table 5). Additionally, we also ran a Chi-square test on the two models between one- and two-dimensional models ($\chi2 = 492.7067$, $p < .000$) and found significant differences confirming the superiority of the three-dimensional model. We then used this three-dimensional model in our subsequent analysis. All the items had weighted (Outfit) and unweighted (Infit) MNSQ values within the range of acceptable values, 0.60 −1.40, as suggested by Wright and Linacre (1994). These values demonstrated that the items had a good fit of the data. Figure 3 is a Wright map from the multidimensional analysis which shows an appropriate distribution of students' responses for each dimension.

Lastly, Table 6 shows the reliability values for the overall instrument as well as each of the dimensions. We can see that all the values are >.8 (rounded), except for the EAP value on the Interest and Self-Efficacy scales. However, considering the other two values (WLE=.813/.932, alpha=.856/.950), we will conclude these scales to be acceptable with regards to reliability.

Table 6 Correlation coefficients (*r*) computed from Pearson product-moment correlation coefficient test.

	1	2	3	4	5	6
1. Growth Mindset	1	.292**	.329**	.257**	0.043	0
2. Interest		1	.642**	.255**	0.161	.324**
3. Self-efficacy			1	.432**	.295**	.232**
4. CS Concept Assessment				1	.404**	0.007
5. Prior Experience					1	-0.064
6. Gender[a]						1

*** Correlation is significant at the 0.01 level (2-tailed).*
[a]Gender was coded as (1 = Female, 2 = Male)

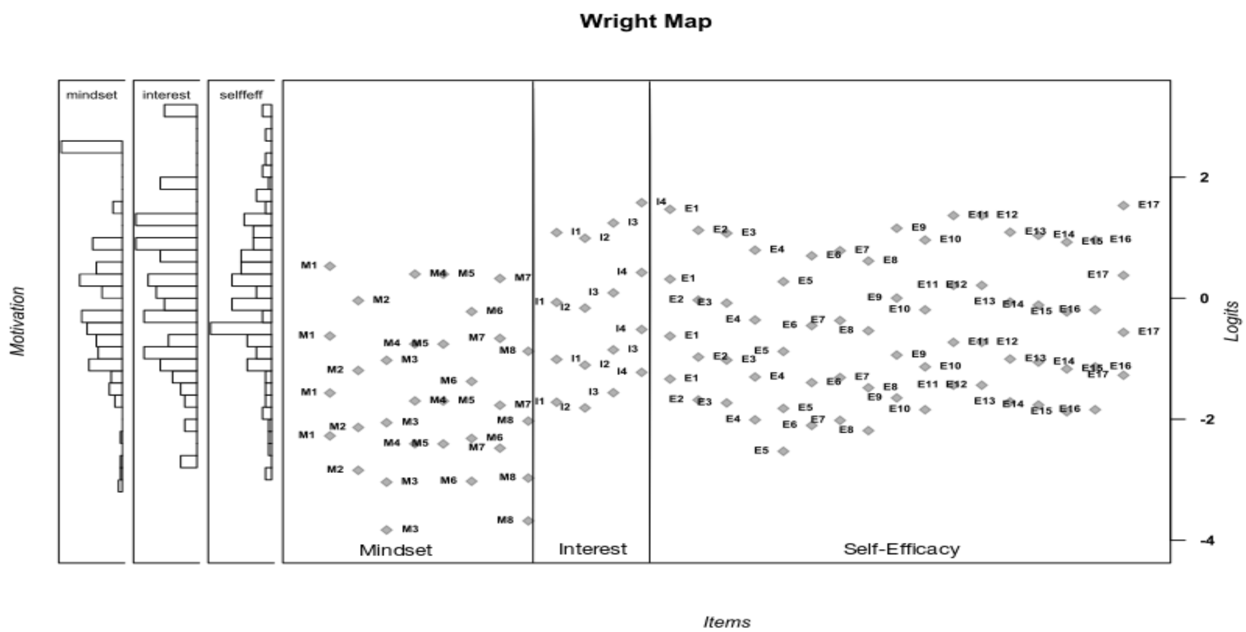


Figure 2: Wright Map of items in Multidimensional Rasch Model.

RQ2: Variances in Elementary Students' Programming Concepts

Descriptive Statistics

We present the descriptive statistics for each variable in Table 7 including the skewness and kurtosis for each construct. Removing individuals with missing values, we then run all further analyses on full information data (n=148).

Table 7 Descriptive Statistics of the Variables.

Variable	Min	Max	M	SD	Skewness	Kurtosis
Growth Mindset[a]	-3.050	2.550	0.251	1.383	0.406	-0.565
Interest[a]	-2.740	3.050	0.133	1.290	0.198	-0.031
Self-efficacy[a]	-2.940	3.180	0.142	1.166	0.100	0.370
CS Concept Assessment	0.000	14.000	6.550	3.074	0.105	-0.170
Prior Experience	0	2	.9550	0.499	-0.220	-1.979

[a] *Transformed Rasch Scores or logits.*

We can see that some values for skewness and kurtosis of growth mindset, Self-efficacy and Prior Experience are slightly out of the range of -.2 to .2. However, normality is only assumed for the dependent variable and, in this case, the dependent variable 'CS Conceptual Understanding' is in the range of normal distribution (Hair et al. 2019). Thus, we accept the non-normality of the independent variables for this analysis.

Path analysis results

We answered the second research question with a path analysis. Using the variables Gender, Prior Experiences and Rasch scores (logit) of Growth Mindset, Interest, Self-efficacy, and raw CSCA scores, we tested our hypothesized model and the revised model (i.e., the model with significant paths and superior fit indices) (Figure 5, Table 5).

Table 8 Model fit indices for two multidimensional rasch models.

Model	loglike	Deviance	Npars	N	AIC	BIC	AICc
One-dimension	-5873.289	11746.58	34	210	11814.58	11928.38	11828.18
Three-dimension	-5669.465	11338.93	39	210	11416.93	11547.47	11435.28

Removing some non-significant paths improved the quality of the model according to the fit indices (χ2=3.326, df = 2, p = .190, GFI= .993, CFI = .992, RMSEA =0.067, and AIC = 41.326, for the hypothesized model, as compared with χ2 =4.133 df=6, p=.659, GFI=.991,

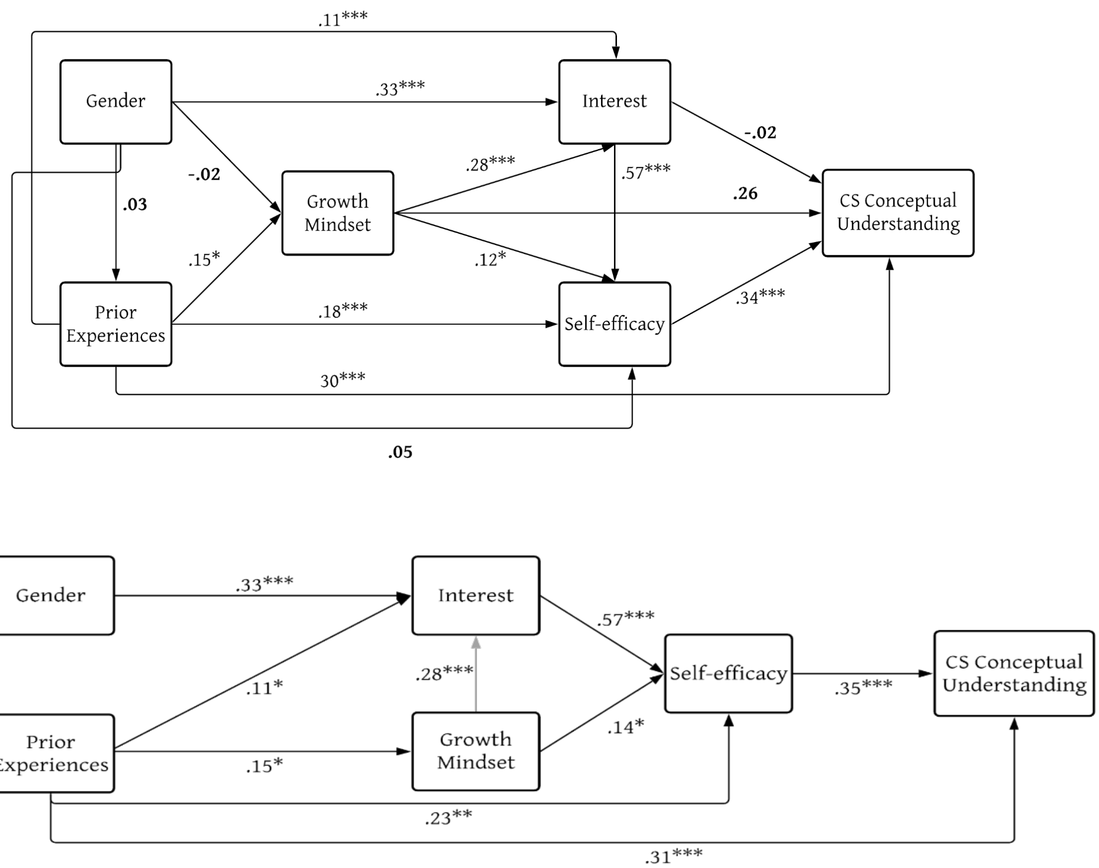


Figure 3: Hypothesized (top) and revised (bottom) path model with coefficients. *p<.05, ***p<.001

CFI=1, RMSEA =0.000 and AIC = 34.133 for the revised model). The improved fit indices, in particular the χ2 /df value of the revised model indicated that the data fit the revised model better than the hypothesized model. Both of the models with standardized path coefficients are visualized in Figure 3. Each path (arrow) represents the change in Y associated with an increase in X of one standard deviation. For example, given a change of one standard deviation in Self-efficacy belief, CSCA score statistically significantly improved by 0.35 standard deviations.

Table 9 shows all the direct and indirect effects of exogenous variable Gender and endogenous variables (Prior Experience, Growth Mindset, Interest and Self-efficacy). Results show that Gender, Prior Experience, Growth Mindset, Interest in CS have statistically significant indirect positive effects on CS Conceptual Understanding (Table 9). Gender only has a direct effect on Interest (also see Table 11), and has an indirect effect through Interest and Self-Efficacy, β=.064, p<.001. Interest has a significant indirect effect through Self-Efficacy β=.197, p<.001. Growth Mindset has a significant indirect effect through Interest and Self-efficacy, β=.101, p<.001. And finally Prior experience has a significant indirect effect through Growth mindset, Interest and Self-efficacy, β=.403, p<.001.

Table 9 Standardized direct and indirect effects of each variable in Path Analysis and Bias-Corrected Factor Score Path Analysis (BCFSPA).

	Standardized β coefficient (Path Analysis)	Bootstrapped *confidence interval* (Path Analysis)	Standardized β coefficient (BCFSPA)	Bootstrapped *confidence interval* (BCFSPA)
Gender→ Interest→ Self-efficacy→ CSCA	.064***	.02, .11	.081**	.042, .134
Prior Experience→ Growth Mindset→ Interest→ Self-efficacy→ CSCA	.403**	.268, .520	.416***	.203, .532
Growth Mindset→ Interest→ Self-efficacy→ CSCA	.101***	.044, .183	.094***	.030, .147
Interest→ Self-efficacy→ CSCA	.197***	.097, .297	.195***	.091, .265
Self-efficacy→ CSCA	.343**	.178, .474	.338**	.127, .481
Prior Experience→ CSCA	.304**	.170, .432	.616**	.406, .711

** *Estimate is significant at the 0.01 level.*
*** *Estimate is significant at the 0.001 level.*

Table 10 Gender wise descriptive statistics.

Variable	gender	N	Mean	Std. Deviation	Std. Error
csca	Female	74	6.53	2.756	0.32
	Male	74	6.57	3.38	0.393
Prior Experience	Female	118	1.71	0.455	0.042
	Male	92	1.65	0.479	0.05
Growth Mindset	Female	118	0.11	1.23	0.11
	Male	92	0.11	1.43	0.15
Interest	Female	74	-0.28	1.11	0.13
	Male	74	0.55	1.33	0.15
Self-efficacy	Female	74	-0.13	1.08	0.13
	Male	74	0.41	1.19	0.14

Here are the findings according to the hypotheses:

H1: Self-efficacy in programming **does** show a positive direct and statistically significant $\beta = .343$, $p<.001$, effect on CS Conceptual Understanding.

H2: Interest towards programming **does not** show a positive direct effect, $\beta=-.024$, $p=.796$, on CS Conceptual Understanding (path removed from the revised model).

H3: Growth Mindset **does not** show a positive direct effect, $\beta= .256$, $p=.122$, on CS Conceptual Understanding.

H4: Interest towards programming **does** show a positive direct effect, $\beta=.570$, $p<.001$, on Self-efficacy.

H5: Growth Mindset **does** show a significant positive direct effect $\beta=.280$, $p<.001$ on students' Interest towards programming. (path removed from the revised model).

H6: Growth Mindset **does** show a significant positive direct effect, $\beta=.118$, $p<.05$, on students' Self-efficacy toward programming.

H7: Students' gender **does not** show a direct effect, β=.049, p=.446, on students' Self-efficacy in programming. (path removed from the revised model).

H8: Students' gender **does** show a significant positive effect, β=.325, p<.001, on students' Interest in programming.

H9: Students' gender **does not** show a significant positive effect on, β=-.019, p=.818, students' Growth Mindset level (path removed from the revised model).

H10: Students' prior experience with programming **does** show a significant positive direct effect, β=.304, p<.001 on students' CS Conceptual Understanding.

H11: Students' Prior Experience with programming **does** show a significantly positive direct effect, β=.110, p<.001, on students' Interest in programming.

H12: Students' prior experience with programming **does** show a significantly positive effect, β=.183, p<.001 on students' Self-efficacy in programming.

H13: Students' Prior Experience **does** programming will show a significantly positive direct effect, β=.149, p<.05 on students' Growth Mindset level.

H14: Students' Gender **does not** show a significant effect β=.03, p=.742 on students' Prior Experience with programming.

Table 11 Fit Indices for Path Models.

Models	Number of Parameters	df	GFI	TLI	CFI	RMSEA	AIC	Chi-squared (*p*)
Hypothesized Model	19	2	.993	.992	.940	.067	41.326	3.326 (*.190*)
Path Model	15	6	.991	1.000	1.000	.000	34.133	4.133 (*.659*)
BCFSPA	15	6	n/a	n/a	n/a	n/a	n/a	3.049 (*.646*)

Bias-corrected Factor-Score Path Model

We tested the revised path model with a BCFSPA model. The BCFSPA model showed very similar estimates for each path along with an acceptable chi-squared fit index ($\chi 2$=3.049, df = 6, p = *.646*) (see Table 9 and 10). The purpose of this bias-corrected factor score path model was to inspect and confirm the validity of the traditional path analysis only because we have a small sample size. Because BCFSPA is biased corrected for small sample size and shows similar results as the full-information path model, we consider the path model acceptable.

Discussion

The purpose of this study was to discover how gender and prior experience of elementary students' impact CS performance and if self-perceived ability beliefs (Mindset, Self-Efficacy) and interest in CS has any mediating effect on CS performance. In doing so, we also wanted to validate an instrument that can reliably measure motivational beliefs and interest of elementary students in CS.

RQ1: How are the dimensionality, internal structure, and reliability of the instrument.

In this study, we passively addressed content validity by modifying a previously validated instrument on the constructs Self-efficacy (Bandura et al., 1999) and Interest (Deci et al., 1991; Collis et al., 2012) and Mindsets (Dweck, 2006, 2013). Although we had to modify the items to make the wordings appropriate for elementary level, we kept the concepts the same. For example, Self-efficacy items that addresses specific CS concepts such as variables, loops, conditional etc, were taken from Kukul et al. (2017), however, we simplified wordings from " I can use a loop instead of repeating instructions" to "I understand when to use loops" (Kukul et al., 2017; Tsai, 2019; Williams et al., 2003; Quade 2003). Next, we addressed individual item qualities and constructed validity first with EFA and them with IRT-Rasch Analysis. The EFA allowed us to determine how items fit within a factor. We were able to drop 13 items that did not fit well within a construct. Then, the three-dimensional Rasch model and its fit indices were used to confirm the construct validity of the revised instrument.

RQ 2: How does the intercorrelation of gender, prior experience and the motivational factors: explain the variations in CS Conceptual Understanding?

We found that for our sample, Prior Experience had a significant role in performance on the CS conceptual assessment, both directly and indirectly through other variables. This is not surprising considering that students with some level of CS experience definitely are expected to perform better than those who do not have any experience. However, it is interesting how strongly prior knowledge influences development of motivation beliefs and interest towards CS. It only emphasizes the importance of introducing students to concepts of CS early in the education level to both get acquainted with and develop interest in the subject.

Unlike previous literature, gender of a student was not found to directly influence students' level of Prior Experience, Growth Mindset or Self-efficacy for this sample. Some contemporary studies have found similar results, finding that female and male students may have different reasons for pursuing CS, but do not show significant differences in CS participation (i.e. prior experience) (Crues et al., 2018; Milesi et al., 2017; Ruthotto et al., 2020). However, all these studies are on college-age students. A possible explanation can be that current patterns of young students' exposure to technology and more inclusiveness in education lead them to have more equitable experience and performance in CS. This seems to be the case with our sample. A future longitudinal study may be able to explain changes in experience, self-efficacy and mindset over the year through middle and high school.

Gender of the students, however, only showed a significant direct influence on interest towards CS. Male students have significantly higher mean Interest levels than female students

(Table 11). Our study agrees with the finding by Margolis & Fisher (2002) that the differences in interest in CS appear very early as children. The reason may be the socio-cultural stereotyping of CS as a male-oriented subject. Investigation by Cheryan et al. 2009 revealed that stereotypical broadcasting of masculine stereotypes may discourage women's interest and sense of belongingness in CS.

Despite males and females having no difference in their Growth Mindset level, young female students do not feel less able to perform well in this topic area, they only do not have the same level of interest towards it. Thus, it is possible that they do not relate to the topic area and do not have a sense of belongingness. Future studies can qualitatively look into factors and reasons behind female students' low interest in CS.

Findings with Growth Mindset also showed no significant influence on CS Performance, where growth mindset only significantly impacted students' interest toward CS. Studies with similar findings, such as Flanigan et al., (2015), showed that student Mindset was a weak predictor of performance. Similarly, Burnette et al., 2020 reported that a Growth Mindset intervention improved interest but not CS academic performance. The strongest impact of growth mindset was found with students' Interest in CS. Degol et al., 2018 found that students' interest, or task value, can mediate the effect of Growth Mindset and posits that it is unlikely that an individual would willingly spend time on an activity they do not think they can grow or learn more. As individuals with growth mindset values improving their skills, they are found to have higher interest and enjoyment in pursuing challenging activities (Stipek and Gralinski 1991). Thus, it makes sense that students who have a higher growth mindset are more likely to find interest in CS.

Another interesting and rather confusing finding is that student's Interest in CS did not directly influence students' CS performance. It only indirectly impacted CS performance through self-efficacy. It may be that Interest is only impactful when there is an interaction of Prior experience and Self-efficacy. Students' interest may only be valuable if they have grown some threshold level of prior knowledge and self-efficacy. CS is a unique and comparatively new K-12 academic subject which is increasingly getting introduced at the elementary level. Interest often needs support to keep developing further (Renninger and Hidi, 2016, p. 3), and it makes sense that general interest without such prior knowledge and efficacy may not be effective in performance in this subject area.

Finally, student's Self-Efficacy was found to be one of the strongest predictors of CS performance, both directly and indirectly. This is in alignment with the vast majority of previous research findings on how self-efficacy predicts performance regardless of subject area. What is more interesting is the extent to which interest level impacts development of self-efficacy and how growth mindset mediates the effect. We hypothesized that a growth mindset might have a direct impact on students' performance. Instead, we found a possible mediating effect on the relationship of interest and self-efficacy. Though lacking in direct effect, the findings indicate the importance of nurturing Growth mindset and Interest in order to support raising self-efficacy which, in turn, would lead to higher performance.

Limitations and Future Directions

There were several limitations to this research. First, the relationships we found can have sampling bias. We had only a small to moderate sample size from a single urban area. We address the possible impact of the small sample size through conducting sampling adequacy tests for factor analysis and conducting a bias-corrected factor score path analysis to confirm our findings. However, researchers still should not over-generalize any of the findings but rather consider the relationships as a probability that needs to be tested and confirmed with both further experimental and longitudinal studies. The pattern of relationships among those variables also may change as students develop and move on to middle and high school. Future research may expand to middle and high school students to draw a complete picture of those variables. For the E-SABICS instruments, we adopted items from previous validated instruments aimed for older students and in other subject areas (i.e Science). Although we did conduct a number of validity and reliability tests, we did not conduct content validity at a deeper level. Future research can focus on the content validity of this instrument with cognitive interviews with upper-elementary age-groups.

Conclusion

Computer Science education is being implemented in elementary classrooms at a fast pace. Before it scales through education systems, researchers and educators need to be equipped with knowledge and interventions to support students' needs to excel in this subject area. The present research aimed to further explore the relationship of gender and prior experience along with motivational beliefs (Mindset, Interest, and Self-efficacy) on CS performance. Furthermore, in this research, we validated an instrument measuring motivational beliefs and interest catering to elementary-level CS courses. The study findings highlight the value of prior knowledge in CS even at an elementary level and how its effects are stronger than gender-based differences. Student gender was only found to significantly impact Interest toward CS, with female students having lower Interest in CS. In Terms of motivational beliefs, Self-efficacy was found to be the strongest predictor of CS performance, whereas Growth Mindset and Interest only mediates its effects. We also utilized a methodologically new but valid alternative of full information path analysis intended for a small-to-moderate sample size named Bias-corrected Factor Score Path Analysis. As this method is rather new and less common in the field but has a high potential of utilization in small sample educational research, researchers can certainly find this study a useful example. The findings of this study have substantive and methodological implications for those exploring issues related to students' individual differences in motivational beliefs, gender and prior experience in the field of elementary CS education.

References

Abdullah, M. N. L. Y. (2008). Children's implicit theories of intelligence: Its relationships with self-efficacy, goal orientations, and self-regulated learning. *International Journal of Learning*, *15*(2).

Aivaloglou, E., & Hermans, F. (2019, February). Early programming education and career orientation: the effects of gender, self-efficacy, motivation and stereotypes. In *Proceedings of the 50th ACM technical symposium on computer science education* (pp. 679-685).

Biggers, M., Brauer, A., & Yilmaz, T. (2008). Student perceptions of computer science: a retention study comparing graduating seniors with cs leavers. *Acm sigcse bulletin*, *40*(1), 402-406.

Babes-Vroman, M., & Nguyen, T.D. (2020). Gender Diversity in Computer Science at a Large Research University. *ArXiv, abs/2004.13760*.

Babes-Vroman, M., Juniewicz, I., Lucarelli, B., Fox, N., Nguyen, T., Tjang, A., ... & Chokshi, R. (2017, March). Exploring gender diversity in CS at a large public R1 research university. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 51-56).

Bai, B., Wang, J., & Nie, Y. (2021). Self-efficacy, task values and growth mindset: what has the most predictive power for primary school students' self-regulated learning in English writing and writing competence in an Asian Confucian cultural context?. *Cambridge Journal of Education*, *51*(1), 65-84.

Bandura, A. (1986). The explanatory and predictive scope of self-efficacy theory. *Journal of social and clinical psychology*, *4*(3), 359-373.

Bandura, A. (1989). Regulation of cognitive processes through perceived self-efficacy. *Developmental psychology*, *25*(5), 729.

Bandura, A. (2013). Regulative function of perceived self-efficacy. In *Personnel selection and classification* (pp. 279-290). Psychology Press.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Beyer, S. (1999). The accuracy of academic gender stereotypes. *Sex Roles*, *40*(9), 787-813.

Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. (2003, January). Gender differences in computer science students. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education* (pp. 49-53).

Board, C., 2021. *Participation in AP Computer Science Principles More Than Doubles 3 Years After Launch*. [online] Newsroom.collegeboard.org. Available at: <https://newsroom.collegeboard.org/participation-ap-computer-science-principles-more-doubles-3-years-after-launch> [Accessed 22 October 2021].

Boone, W. J., Staver, J. R., & Yale, M. S. (2013). *Rasch analysis in the human sciences*. Springer Science & Business Media.

Bomia, L., Beluzo, L., Demeester, D., Elander, K., Johnson, M., & Sheldon, B. (1997). The Impact of Teaching Strategies on Intrinsic Motivation.

Burnette, J. L., Hoyt, C. L., Russell, V. M., Lawson, B., Dweck, C. S., & Finkel, E. (2020). A growth mind-set intervention improves Interest but not academic performance in the field of computer science. *Social Psychological and Personality Science*, *11*(1), 107-116.

Blackwell, L. S., Trzesniewski, K. H., & Dweck, C. S. (2007). Implicit theories of intelligence predict achievement across an adolescent transition: A longitudinal study and an intervention. *Child development*, *78*(1), 246-263.

Bråten, I., & Strømsø, H. I. (2005). The relationship between epistemological beliefs, implicit theories of intelligence, and self-regulated learning among Norwegian postsecondary students. *British Journal of Educational Psychology*, *75*(4), 539-565.

Byrne, P., & Lyons, G. (2001, June). The effect of student attributes on success in programming. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education* (pp. 49-52).

Cheryan, S., Plaut, V. C., Davies, P. G., & Steele, C. M. (2009). Ambient belonging: how stereotypical cues impact gender participation in computer science. *Journal of personality and social psychology*, *97*(6), 1045.

Costello, A. B., & Osborne, J. (2005). Best practices in exploratory factor analysis: Four recommendations for getting the most from your analysis. *Practical assessment, research, and evaluation*, *10*(1), 7.

Cordova, J. R., Sinatra, G. M., Jones, S. H., Taasoobshirazi, G., & Lombardi, D. (2014). Confidence in prior knowledge, self-efficacy, interest and prior knowledge: Influences on conceptual change. *Contemporary Educational Psychology*, *39*(2), 164-174.

Croon, M. (2014). Using predicted latent scores in general latent struc ture m odels. In *Latent variable and latent structure models* (pp. 207-236). Psychology Press.

Crues, R. W., Henricks, G. M., Perry, M., Bhat, S., Anderson, C. J., Shaik, N., & Angrave, L. (2018). How do gender, learning goals, and forum participation predict persistence in a computer science MOOC?. *ACM Transactions on Computing Education (TOCE)*, *18*(4), 1-14.

Collis, J. M., Messick, S. J., & Schiefele, U. (Eds.). (2012). *Intelligence and personality: Bridging the gap in theory and measurement*. Psychology Press.

Davis, J. L., Burnette, J. L., Allison, S. T., & Stone, H. (2011). Against the odds: Academic underdogs benefit from incremental theories. *Social Psychology of Education*, *14*(3), 331-346.

Deci, E. L., Koestner, R., & Ryan, R. M. (1999). The undermining effect is a reality after all—Extrinsic rewards, task interest, and self-determination: Reply to Eisenberger, Pierce, and Cameron (1999) and Lepper, Henderlong, and Gingras (1999).

Degol, J. L., Wang, M. T., Zhang, Y., & Allerton, J. (2018). Do growth mindsets in math benefit females? Identifying pathways between gender, mindset, and motivation. *Journal of youth and adolescence*, *47*(5), 976-990.

DeVellis, R. F. (2017). Scale Development: Theory and Applications (4th ed.). Thousand Oaks, CA: Sage.

Devlieger, I., & Rosseel, Y. (2017). Factor score path analysis: An alternative for SEM?. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, *13*(S1), 31.

Dicke, A. L., Safavian, N., & Eccles, J. S. (2019). Traditional gender role beliefs and career attainment in STEM: A gendered story?. *Frontiers in psychology*, *10*, 1053.

Dubina, K. S., Morisi, T. L., Rieley, M., & Wagoner, A. B. (2019). Projections overview and highlights, 2018-28. *Monthly Lab. Rev.*, *142*, 1.

Dweck, C. S., & Bempechat, J. (2017). Children's theories of intelligence: Consequences for learning. In *Learning and motivation in the classroom* (pp. 239-256). Routledge.

Dweck, C. S. (2013). *Self-theories: Their role in motivation, personality, and development*. Psychology press.

Dweck, C. S., & Master, A. (2012). Self-theories motivate self-regulated learning. In *Motivation and self-regulated learning* (pp. 43-64). Routledge.

Fan, T. S., & Li, Y. C. (2005). Gender issues and computers: college computer science education in Taiwan. *Computers & Education*, *44*(3), 285-300.

Flanigan, A. E., Peteranetz, M. S., Shell, D. F., & Soh, L. K. (2015, August). Exploring changes in computer science students' implicit theories of intelligence across the semester. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 161-168).

Fryer, L. K. (2015). Predicting self-concept, Interest and achievement for first-year students: The seeds of lifelong learning. *Learning and Individual Differences*, *38*, 107-114.

Gogol, K., Brunner, M., Martin, R., Preckel, F., & Goetz, T. (2017). Affect and motivation within and between school subjects: Development and validation of an integrative structural model of academic self-concept, interest, and anxiety. *Contemporary Educational Psychology*, *49*, 46-65.

Gunbatar, M. S. (2018). Examination of Undergraduate and Associate Degree Students' Computer Programming Attitude and Self-Efficacy According to Thinking Style, Gender and Experience. *Contemporary Educational Technology*, *9*(4), 354-373.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). Multivariate Data Analysis (Eight). *Hampshire, UK: Cengage Learning*.

Hattie, J. (2008). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. routledge.

Heenwood, F. (1999). Exceptional women? Gender and technology in UK higher education. *IEEE Technology and Society Magazine*, *18*(4), 21-27.

Hidi, S. (1990). Interest and its contrubution as a mental resource for learning. *Review of Educational Research*, 60, 549–571

Hidi, S., & Renninger, K. A. (2006). The four-phase model of interest development. *Educational psychologist*, *41*(2), 111-127.

Krapp, A. (2005). Basic needs and the development of interest and intrinsic motivational orientations. *Learning and Instruction*, 15, 381–395.

Huang, X., Zhang, J., & Hudson, L. (2019). Impact of math Self-efficacy, math anxiety, and growth Mindset on math and science career Interest for middle school students: the gender moderating effect. *European Journal of Psychology of Education*, *34*(3), 621-640.

Jiang, M., Li, Y., Zheng, J., & Han, X. (2017). Gender group differences on behavior patterns in collaborative problem solving through LEGO. *Journal of Computers in Education*, *4*(2), 127-145.

Kaiser, H. F. (1974). An index of factorial simplicity. *Psychometrika*, *39*(1), 31-36.

Kelcey, B. (2019). A robust alternative estimator for small to moderate sample SEM: Bias-corrected factor score path analysis. *Addictive behaviors*, *94*, 83-98.

Kelcey, B., Cox, K., & Dong, N. (2021). Croon's bias-corrected factor score path analysis for small-to moderate-sample multilevel structural equation models. *Organizational research methods*, *24*(1), 55-77.

Kinnunen, P., & Malmi, L. (2006, September). Why students drop out CS1 course?. In *Proceedings of the second international workshop on Computing education research* (pp. 97-108).

Köller, O., Baumert, J., & Schnabel, K. (2001). Does Interest matter? The relationship between academic Interest and achievement in mathematics. *Journal for research in mathematics education*, *32*(5), 448-470.

Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, *7*(1), 158-179.

Margolis, J., & Fisher, A. (2002). *Unlocking the clubhouse: Women in computing*. MIT press.

Jansen, M., Lüdtke, O., & Schroeders, U. (2016). Evidence for a positive relation between Interest and achievement: Examining between-person and within-person variation in five domains. *Contemporary Educational Psychology*, *46*, 116-127.

Lakanen, A. J., & Isomöttönen, V. (2018). Computer science outreach workshop and Interest development: A longitudinal study. *Informatics in Education*, *17*(2).

Lamb, R. L., Vallett, D., & Annetta, L. (2014). Development of a short-form measure of science and technology self-efficacy using Rasch analysis. *Journal of Science Education and Technology*, *23*(5), 641-657.

Lewis, C. M., Yasuhara, K., & Anderson, R. E. (2011, August). Deciding to major in computer science: a grounded theory of students' self-assessment of ability. In *Proceedings of the seventh international workshop on Computing education research* (pp. 3-10).

Lin-Siegler, X., Dweck, C. S., & Cohen, G. L. (2016). Instructional interventions that motivate classroom learning. *Journal of Educational Psychology*, *108*(3), 295.

Linacre, J. M. (2017). Winsteps (Version 4.0. 0)[Computer software]. *Beaverton: Oregon. Recuperado de http://www. winsteps. com*.

Malik, S. I., & Coldwell-Neilson, J. (2018). Gender differences in an introductory programming course: New teaching approach, students' learning outcomes, and perceptions. *Education and information technologies*, *23*(6), 2453-2475.

McBroom, J., Koprinska, I., & Yacef, K. (2020, February). Understanding gender differences to improve equity in computer programming education. In *Proceedings of the Twenty-Second Australasian Computing Education Conference* (pp. 185-194).

Milesi, C., Perez-Felkner, L., Brown, K., & Schneider, B. (2017). Engagement, persistence, and gender in computer science: Results of a smartphone ESM study. *Frontiers in psychology*, *8*, 602.

Murphy, S., MacDonald, A., Wang, C. A., & Danaia, L. (2019). Towards an understanding of STEM engagement: A review of the literature on motivation and academic emotions. *Canadian Journal of Science, Mathematics and Technology Education*, *19*(3), 304-320.

National Academies of Sciences, Engineering, and Medicine (Ed.). (2018). *Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments.* Washington, DC: The National Academies Press.

Papastergiou, M. (2008). Are computer science and information technology still masculine fields? High school students' perceptions and career choices. *Computers & education*, *51*(2), 594-608.

Petersen, A., Craig, M., Campbell, J., & Tafliovich, A. (2016, November). Revisiting why students drop CS1. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research* (pp. 71-80).

Pajares, F. (1996). Self-efficacy beliefs in academic settings. *Review of educational research*, *66*(4), 543-578.

Pintrich, P. & Schunk, D. (1996). *Motivation in Education: Theory, Research & Applications*, Ch. 3

Ramalingam, V., LaBelle, D., & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. In *Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education* (pp. 171-175).

Renninger, K. A., & Hidi, S. (2002). Student interest and achievement: Developmental issues raised by a case study. In *Development of achievement motivation* (pp. 173-195). Academic Press.

Rosseel, Y. (2012). Lavaan: An R package for structural equation modeling and more. Version 0.5–12 (BETA). *Journal of statistical software*, *48*(2), 1-36.

Rottinghaus, P. J., Larson, L. M., & Borgen, F. H. (2003). The relation of self-efficacy and interests: A meta-analysis of 60 samples. *Journal of Vocational behavior*, *62*(2), 221-236.

Robitzsch, A., Kiefer, T., Wu, M., Robitzsch, M. A., Adams, W., Rcpp, L., & LSAmitR, R. E. (2021). Package 'TAM'.

Rubio, M. A., Romero-Zaliz, R., Mañoso, C., & Angel, P. (2015). Closing the gender gap in an introductory programming course. *Computers & Education*, *82*, 409-420.

Ruthotto, I., Kreth, Q., Stevens, J., Trively, C., & Melkers, J. (2020). Lurking and participation in the virtual classroom: The effects of gender, race, and age among graduate students in computer science. *Computers & Education*, *151*, 103854.

Schiefele, U. (1991). Interest, learning, and motivation. *Educational psychologist*, *26*(3-4), 299-323.

Schmidt, J. A., Shumow, L., & Kackar-Cam, H. Z. (2017). Does Mindset intervention predict students' daily experience in classrooms? A comparison of seventh and ninth graders' trajectories. *Journal of youth and adolescence*, *46*(3), 582-602.

Scott, M. J., & Ghinea, G. (2013). On the domain-specificity of mindsets: The relationship between aptitude beliefs and programming practice. *IEEE Transactions on Education*, *57*(3), 169-174.

Settle, A., Vihavainen, A., & Sorva, J. (2014, June). Three views on motivation and programming. In *Proceedings of the 2014 conference on Innovation & technology in computer science education* (pp. 321-322).

Sharma, R., & Shen, H. (2018). Does education culture influence factors in learning programming: A comparative study between two universities across continents. *International Journal of Learning, Teaching and Educational Research*, *17*(2), 1-24.

Stipek, D. J., & Gralinski, J. H. (1991). Gender differences in children's achievement-related beliefs and emotional responses to success and failure in mathematics. *Journal of educational psychology*, *83*(3), 361.

Tabachnick, B. G., Fidell, L. S., & Ullman, J. B. (2007). *Using multivariate statistics* (Vol. 5, pp. 481-498). Boston, MA: Pearson.

Tsai, M., Wang, C., & Hsu, P. (2019). Developing the Computer Programming Self Efficacy Scale for Computer Literacy Education.

Quade, A. (2003, April). Development and validation of a computer science self-efficacy scale for CS0 courses and the group analysis of CS0 student self-efficacy. In *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing* (pp. 60-64). IEEE.

Quille, K., Culligan, N., & Bergin, S. (2017, June). Insights on Gender Differences in CS1: A Multi-institutional, Multi-variate Study. In *Proceedings of the 2017 acm conference on innovation and technology in computer science education* (pp. 263-268).

Vandenberg, J., Rachmatullah, A., Lynch, C., Boyer, K. E., & Wiebe, E. (2021). Interaction effects of race and gender in elementary CS attitudes: A validation and cross-sectional study. *International Journal of Child-Computer Interaction*, *29*, 100293.

Vandenberg, J., Rachmatullah, A., Lynch, C., Boyer, K. E., & Wiebe, E. (2021, June). 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* (pp. 46-52).

Vongkulluksn, V. W., Matewos, A. M., Sinatra, G. M., & Marsh, J. A. (2018). Motivational factors in makerspaces: a mixed methods study of elementary school students' situational Interest, Self-efficacy, and achievement emotions. *International journal of STEM education*, *5*(1), 1-19.

Wiebe, E., Williams, L. A., Yang, K., & Miller, C. S. (2003). *Computer science attitude survey*. North Carolina State University. Dept. of Computer Science.

Wigfield, A., & Cambria, J. (2010). Students' achievement values, goal orientations, and Interest: Definitions, development, and relations to achievement outcomes. *Developmental review*, *30*(1), 1-35.

Wigfield, A., Eccles, J. S., Fredricks, J. A., Simpkins, S., Roeser, R. W., & Schiefele, U. (2015). Development of achievement motivation and engagement. In M. E. Lamb & R. M. Lerner (Eds.), *Handbook of child psychology and developmental science: Socioemotional* *processes* (pp. 657–700). John Wiley & Sons. https://doi.org/10.1002/9781118963418.childpsy316

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, *12*(3), 197-212.

Wright, B. (1994). Reasonable mean-square fit values. *Rasch Meas Trans*, *8*, 370.

Wu, S. Y., & Wang, S. M. (2020). Exploring the effects of gender grouping and the cognitive processing patterns of a Facebook-based online collaborative learning activity. *Interactive Learning Environments*, 1-15.

Yeager, D. S., Hanselman, P., Walton, G. M., Murray, J. S., Crosnoe, R., Muller, C., ... & Dweck, C. S. (2019). A national experiment reveals where a growth mindset improves achievement. *Nature*, *573*(7774), 364-3.

Young, A. J., & Urdan, T. C. (1993). The Influence of Perceived Classroom Goals and Prior Beliefs on Aspects of Student Motivation.

CHAPTER 3: Gender Dynamics in Collaborative Programming: A Multiple Case Study of Group Composition

Abstract

Background The quality of collaborative discourse is found to be highly influenced by students' demographics such as gender (Jiang et al., 2017; Wu & Wang, 2019). Research essentially needs to determine how gender changes the dynamics of collaborative discourse and learning activities when students as young as elementary age participate in pair programming. There is a scarcity of studies influencing the grouping base of gender impact conversational variances along with learning outcomes in a computer-supported collaborative learning environment.

Objective In this current study, we compare three cases of gender based pairing : girl-girl, girl-boy and boy-boy dyads' collaboration and programming performance. In doing so, we have utilized data on students' dialogue and their game artifacts to measure the programming functionality of games programmed by students.

Method We used a multiple case-study approach to stage our analysis around the cases with six dyads from an elementary classroom. Both the qualitative and quantitative content analysis is embedded into these cases for in-depth and holistic comparison of the collaboration.

Findings Our findings are in accord with the idea that there are differences in nuanced characteristics of collaborative programming between boys and girls in pair programming. Mixed gender dyads showed the least harmonious collaboration. Compared to the same-gender dyads they conversed less and mostly had unproductive conflicts. Same gender dyads were more collaborative, while girl-girl dyads showed more Exploratory talk as well as more synced in programming.

Implications Findings hold implications for how teachers pair students, design differentiated instructions to provide sufficient opportunities and conditions to achieve constructive collaborative communication.

Introduction

Pair-programming is an effective learning method in computer science education because of its reported benefits under the collaborative setting. Originating in industry, pair programming as a pedagogical method has paved its way to classrooms with young students by showing benefits on retention, better achievement and engagement. Early research sought to find productivity advantages of collaborative, interactive activities over individual ones. More recently, it is found that the quality of discourse is found to be an important antecedent of any collaborative performance (Chi & Menekse, 2015; Curşeu et al., 2018). How students talk, such as types of questions they ask, types of responses they get from partners, sharing and receiving different ideas from partners can highly impact cognitive processes in any collaborative activity (Mercer, 2019). Moreover, pair programming as a computer-supported collaborative environment, adds another fold of complexity on how students interact in this environment. The dynamics of such a complex collaborative environment can be influenced by many independent factors that we need to consider when implementing such interventions in classrooms. The quality of collaborative discourse is found to be highly influenced by students' demographics such as gender (Jiang et al., 2017; Wu & Wang, 2020). Research essentially needs to determine how gender changes the dynamics of collaborative discourse and learning activities when students as young as elementary age participate in pair programming.

In a collaborative setting such as pair programming, we need to consider the gender of both the partners, in other words, the gender composition of a pair. Research shows that differences in gender composition (i.e male-male, female-male, female-female) can impact pairs' motivation towards the task, the level of interaction and consequently the pairs' performance in a collaborative activity (Zhan et al., 2015). A greater volume of studies has explored the impact of gender on quantitatively measuring and comparing their learning outcome on assessment tests (e.g.). However, fewer studies integrate different factors such as conversational variances along with learning outcomes related to collaborative learning effectiveness, especially in a computer-supported collaborative learning environment.

In this current study, we compare three cases of gender based pairing : girl-girl, girl-boy and boy-boy dyads' collaboration and programming performance. In doing so, we have utilized data on students' dialogue and their game artifacts to measure the programming functionality of games programmed by students.

Such integrative research will help further extend research by better understanding if and how different gender compositions perform differently (or the same) in a computer-supported collaborative setting. In this case study, we specifically address the following three questions:

How do three cases of different gender-grouped dyads (boy-girl, boy-boy, girl-girl) compare in terms of collaborative programming?

a) How does the quality of collaboration compare between the cases of different gender groupings?

b) How does the program functionality compare between the cases of different gender groupings?

Theoretical Framework on Discourse

Collaborative discourse encompasses cognitive processes such as knowledge sharing and critical thinking, as well as engaging emotional processes (Isohätälä et al., 2018). For example, as part of this collaborative process, students may suggest ideas in some situations and explain themselves in others (Ruvalcaba et al., 2016). Despite considering task-centric critical conversation as productive, intense reactions or insensitivity may engender an unfavorable socioemotional learning environment (Gilbert, 2004; Plantin, 2004). Although we might all agree that such scenarios are not uncommon in social situations, theories that explain why, where, and when such occurrences unfold with young pair programmers is still in its formative stages of development. Thus, we attempt to incorporate a theoretical framework, Inter-Developmental Zone (IDZ) by Mercer and Littleton (2007) that describes students' dialogue in classrooms in the context of pair programming. In this framework, Mercer defines three types of talk that can occur in the course of a conversation: Cumulative, Disputational and Exploratory. In Cumulative talk, speakers build positively but uncritically on what their interlocutor has said. Thus, partners use cumulative talk to construct 'common knowledge'. Cumulative talk is characterized by repetitions, confirmations, and elaborations. Disputational talk, by contrast, is characterized by disagreement and individualized decision-making. Participants "... defend their self-identity and show little interest in mutual." (Wegerif & Mercer, 2000) Disagreements emerging from Disputational talk are usually short without further elaboration; participants defend their self-identity and show little interest in mutual understanding. Finally, in Exploratory talk, participants engage critically but constructively with each other's ideas, which leads to improved reasoning and conceptual understanding (Bennett & Cass, 1989; Mercer and Littleton, 2007). According to Wegerif and Mercer (2000), "In exploratory talk, knowledge is made publicly accountable and reasoning is visible in the talk". Group members challenge each other with questions, respond to the challenges with appropriate justifications, and offer alternative hypotheses during this type of talk. Thus, with exploratory talk, the pair engage in conflict, but it is bilaterally embraced through constructive engagement in order to move forward with a problem solution.

Pair programming

Pair-programming is a practice of collaborative programming where two programmers or coders work side by side on the same platform to solve a programming problem (Williams et al., 2000). Pair-programming has been practiced in the software industry for quite a while with empirical research demonstrating many benefits over individual programming. Some of these benefits include higher level of product quality, less time consumption in solving a problem and higher satisfaction in the programming process (Williams et al. 2000; Nagappan, et al., 2003). Because of its reported benefits in the software industry, it has been introduced in college level computer science courses as well. Studies that explored pair-programming at the college level suggest that pair-programming is an effective pedagogical approach for teaching students programming skills (e.g., Hank et al. 2011; Williams et al., 2002). What work has been done with college students indicates that, pairing in introductory computer science course bolsters completion rates, greater persistence, confidence and enjoyment in computer science related majors and more importantly, producing higher quality programs (Mcdowell et al., 2003; Braught et al., 2008; Williams et al. 2000; Nagappan, et al., 2003). In terms of inclusiveness, pair-programming also helps female students to stay engaged and persist in computer science educational pathways (McDowelletal., 2002). Additionally, the ability to perform effectively in a pair-programming environment helps prepare students for the future workforce (National Research Council, 2013).

Block-based programming environments (i.e Scratch, NetsBlox) are commonly used with younger students to introduce them to programming (Weitrop, 2019). Collaboration in these environments can take place in several different ways. Much like documented in the previously cited studies, they can perform one-computer pair-programming where one person, the driver, controls the programming environment, and the navigator monitors the progress and anticipates future steps (Hanks et al., 2011). Newer environments such as NetsBlox allow participants to interact through a shared, virtual workspace where they can collaborate by working on the same project simultaneously. While this collaborative programming could be done distally, the more common instantiations in younger grades is to have students situate their computers proximally, side-by-side in the classroom, so they can continue direct verbal and paraverbal interaction under the guidance of a teacher.

Gender dynamics in programming

Discourse

In computer supported environments, research has shown a distinction between the interactions of gendered pairs relating to gender composition. The combination of girl-girl dyads can lead to a greater proportion of activity completion as well as participant agreement that the pair exerted good effort towards the activity (Jarratt et al., 2019). Same-gender female pairs may spend more time integrating ideas and less time on off-topic discussions than other pairings (Lin et al., 2020). The inclusion of at least one female participant within a dyad (boy-girl or girl-girl) can produce higher levels of participant confidence in their final products, but can also lead to a lower rate of assignment completion in a classroom setting (Jarratt et al., 2019). Same-gender pairings (boy-boy or girl-girl) may exhibit better performance in the application of computer-science knowledge to problem solving than mixed-gender pairings (boy-girl) (Lin et al., 2020).

Where same-gender pairings can produce more focused interactions, mixed-gender pairings can lead to male-participant domination in discussion or difficulty coordinating problem-solving interactions (Lin et al., 2020). Overall literature on gender pairings in collaborative computer supported environments suggests higher rates of successful collaboration between same-gender pairs, particularly for female participants. Although, in a recent experimental study, boy-boy pairs were found to score higher in computational thinking skills in an adapted pair programming setting than the mixed-gender, girl-girl or students in individual programming setting (Wei et al. 2021). This study, however, had students from a very different cultural context and did not take into account the within-pair collaborative processes, thus calling for further research on both students' programming skill as well as their collaborative practices for each gender composition separately. This case study would shed light on the discrepancies, if any, lies in the collaborative behavior of students in pair-programming context.

Program functionality in programming projects

Studies measuring differences in how male and female students individually program shows mixed findings. Some studies like Baytak et al., (2011) found that female students, despite having less prior experience with programming, improved their programming concepts in block-based programming settings more than male students. However, others like Grover et al. (2018) showed that male and female students' scores on block-based programming projects did not have any significant difference. However, the same study found that there can be differences in the content of the programming artifacts. If given the opportunity, females used more stories in their programming project whereas males incorporated games. This study used a qualitative rubric to measure students' learning in programming. Studies qualitatively measuring students programming learning outcomes through analyzing their programming artifacts are comparatively new and found to be effective in identifying multidimensions in learning such as overall proficiency, design elements and coding & CS constructs (Basu, 2019; Grover, 2021; Saito, 2020). Our case study utilizes a rubric tailored to the programming project assigned to students and measures their program functionality to find how and if there are any patterns in their programming.

Method

Study Context

The study took place at a suburban elementary school in the Southeastern United States. The students were from four classrooms, but taught by the same technology teacher following the same curriculum. Students used a block-based coding environment Scratch to collaboratively program. Dyads were seated side-by-side (where they were instructed to create identical programs on each of their computers in separate workspaces). The teacher directed the task, which required the pairs to collaboratively create a game wherein the user follows rhyming clues to click on certain sprites that move or speak in response.

The participants were 62 fourth grade students (9-10 years) who formed 31 dyads (formed by the teacher). In-depth analysis was conducted on 18 purposively selected participants forming 9 dyads, three from each type of gender composition (female-female, female-male, male-male) (9 boys and 9 girls, total) from all the pairs that had recordings of good video and audio clarity. Activity sessions occurred once per week for four weeks (average of 26 active minutes/week). We did not analyze all of the lessons. Rather, we selected to analyze the third day of the collaboration based on the presumption that the task had a higher cognitive involvement than other days, as thus was likely to elicit the richest conversation. We used video cameras with synced headsets to capture audio and video of the dyads. Verbatim transcripts of pairs' verbal interactions were created. We analyzed students' turns of talk ($n = 3152$) to examine their collaboration and their submitted Scratch program artifacts (n = 18) to examine their program functionality.

Measure and Analysis

We use a multiple case-study approach to stage our analysis around the cases (Yin, 2017). Both the qualitative and quantitative content analysis is embedded into these cases for in-depth and holistic exploration of the cases. The cases in this study were defined a priori based on the three variations of gender groups used in this pair-programming implementation: boy-girl, girl-girl, boy-boy.

To fully characterize each case of pair programming in different gender-groups, we triangulated data by integrating materials and evidence collected using two methods: video and transcribed data of students' discourse, and game artifacts (Kohlbacher, 2006). We use a sequential mixed method approach to analyze discourse data (Creswell and Clark, 2011). In this approach, we code transcribed video data to quantitatively measure students' discourse and follow up with a qualitative template analysis identifying themes in their discourse. To measure students' program functionality as an indicator of learning, game artifacts were qualitatively analyzed with a coding rubric.

To answer our research questions, we first describe our findings for each case and then thematically compare case-characteristics. For each case description, we first describe the collaborative performance of each case using descriptive statistics of conversation categories. The quantitative descriptives are followed by the thematic characteristics of collaboration found through qualitative template analysis. Finally, each of the cases is synthesized with the findings regarding program functionality found through qualitative scoring of game artifacts. Such multifaceted exploration of students' collaboration helps in understanding and explaining each of our cases of gender composition (Creswell, 2009). For case comparison, we report on both the important unique and common characteristics of the case found through our analysis.

<u>Discourse data</u>

To support our analysis, we collected webcam video and screen capture data from each pair of students in a single file. Participating students wore headsets with microphones to ensure high-quality audio in their video recordings. Our final sample consisted of 9 videos that were on average 35 minutes each, with a total of 315 minutes of conversation. We transcribed the videos verbatim. The transcripts of students' conversations were segmented into dyads' turn of talk to create an appropriate level of granularity and more coherence to the segments of text (Schegloff, 1991). The analyzed transcripts along with videos of conversation consisting of 3152 turns of talk in the data analyzed for this study.

Data coding

The theoretical framework of Inter-Developmental Zone (IDZ) by Mercer provides a promising start to distinguish between productive and less productive conversation during pair programming. That is, conversation that forwards learning for both students. However, it needs to be operationalized in a manner that allows for a structured method of analyzing conversation data (i.e. video data of students collaborating in a computer-supported environment) and also catering to the unique setting of pair programming. In our previous studies, we applied Mercer's framework in analyzing students' dialogue in pair-programming activities. Through iterative studies, we developed and modified an analytical framework to code and analyze student dialogue in pair programming activities (Zakaria et al., 2019; Zakaria et. al., 2021).

Two authors coded the videos using a coding scheme developed across multiple studies (Zakaria et al., 2019; Zakaria et. al., 2021). This coding scheme was adapted from a scheme originally developed by Ruvalcaba et al. (2016) for coding conversations during pair programming. To better differentiate productive from unproductive conversation, we explicitly operationalized Mercer and Littleton's (2007) three types of talk in a pair programming context. We considered characteristics of the types of talk and added or clarified a number of conversation categories that would help identify productive and unproductive talk more explicitly. Thus, we modified the coding scheme throughout three previous studies, and it resulted in a new coding scheme that we used in this study as well.

The coding scheme had categories that reflect Exploratory talk: sharing alternative ideas, asking higher-order questions (questions that challenge partner's ideas, such as 'why' questions), and providing justification. For Cumulative talk, the categories were: *simple questions* (any questions that are not explicitly challenging partners), *suggestion*, *comments*, *self-explanations, coordinatio*n, *agreement*. We also had categories of *disagreemen*t and other. For this study, disagreements were further re-coded into two categories. Disagreements that did not include any justifications were named *unjustified disagreement*, which we surmised might lead to Disputational talk (we explored this in our analysis). Disagreements that had *justifications* in the same turn of talk or the following four turns of talk were re-coded to *justified disagreement* and considered as Exploratory talk.

For our analysis, we first applied quantitative content analysis to the students' conversation that consisted of a total of 3152 turns of talk. Each turn of talk was then coded with conversation categories that helped identify the patterns of type of talk (i.e. Exploratory) (Hannessey et al., 2020). Two coders coded 22.2% of the data using the final coding scheme. We coded each partner's turns of talk in a conversation (Schegloff, 1991). The categories were not mutually exclusive; thus, *kappa* was calculated for each category separately. The resulting average *kappa* indicated substantial agreement (*kappa:* average = 0.859, min = .783, max = .967)

(Landis & Koch, 1977) (see Table 1). The disagreements were resolved through discussion followed by consensus. Both coders coded equal proportions of the remaining data.

In the next step, we did a template analysis on the coded transcripts of collaborative dialogue (King, 2004). Template analysis is a type of thematic analysis done on an already coded transcript with the purpose of finding patterns regarding how the codes unfolded in the text. This analysis will allow us to qualitatively describe and compare the cases of different gender-group collaborations.

Data on programming functionality

Figure 1: Screenshot of a I-SPY game created by a student.

By programming functionality, we mean the extent to which the games students programmed has fulfilled the requirements of the programming project. We collected Scratch program outputs of the game students submitted to the teacher at the end of the last session of the project activity (Figure 1). Table 1 describes the components we used to measure their program functionality to code their Scratch game artifacts: Number of Sprites, I-Spy Riddle, Completion, Types of motion, Advanced features. Each component has a scale of target parameter value (i.e. 2,1,0). We added scores on all components to get a final total score for each student's game artifact that could range from 0 - 12. The components were created both deductive and inductively. First, we considered the requirements the teacher had for the programming project and then two coders open coded four student Scratch program outputs based on the assignment teacher requirements and accepted best coding practices. The components are important in creating a functional I-SPY game and indicate the quality of the program, for example, without the I-SPY riddle and the characters (sprites) in the riddle, one cannot navigate through the game, types of motions and features would indicate if and how loops and conditional were used to program the game. Two coders coded 22.2% of the Scratch game artifacts with a *kappa* of .931 ensuring the reliability of the coding scheme (Cohen, 1960). The remaining artifacts were coded by one of the coders.

Table 1 Scoring Rubric for Program Functionality.

Component	Target parameter value	Description
Number of Sprites	2	More than or equal to 4 sprites (I-Spy objects) are present on the screen.
	1	Less than or equal to 3 sprites are present.
	0	Do not have any sprites
I-Spy Riddle	2	Has a full sentence of I-Spy riddle.
	1	Has an incomplete sentence of I-Spy riddle.
	0	No sentence of an I-Spy riddle is present.
Completion	2	Has all the sprites that the riddle mentions.
	1	Has some but not all the sprites the riddle mentions.
	0	Do not have any sprites mentioned in the riddle.
Types of motion	3	Used more than or equal to 3 types of motions when the sprites are clicked.
	2	Used 2 types of motions when the sprites are clicked.
	1	Used only 1 type of motions when the sprites are clicked.
	0	No motion when the sprites are clicked.
Advanced features	3	Used more than or equal to 3 types of features (appearing texts, sounds etc.) when the sprites are clicked.
	2	Used 2 types of features (appearing texts, sounds etc.) when the sprites are clicked.

Table 1 (continued)

1	Used 1 type of features (appearing texts, sounds etc.) when the sprites are clicked.
0	No other features when the sprites are clicked.

Note: Used the same rubric with each level of the game.

Results

Overall, we see that throughout all groups, the percentage of *self-explanation* and *simple-question* use were higher than the other categories (Table 2, Figure 2). On the other hand, categories reflecting Exploratory talk (*Disagreement with justification, alternative idea, justification, higher-order questions*) were used proportionately the lowest throughout all pairs. Table 3 provides the program functionality scores for each students in a dyad. The average program functionality score for girls' game artifacts was 8.38 (min=3, max= 14) and for boys, 7.38 (min =1, max=17).

Table 2 Proportion of conversation categories used by dyads in each three cases.

Conversation Category	Gender Composition		
	Boy-Girl	Girl-Girl	Boy-Boy
Unjustified Disagreement	40.4	33.7	25.9
Justified Disagreement	32.0	37.9	30.1
Justification	24.2	48.1	27.7
Alternative idea	14.1	42.2	43.7
Higher-order question	8.6	44.8	46.6
Simple question	33.4	32.0	34.6
Suggestion	25.5	37.3	37.2
Comment	53.2	20.4	26.4
Self explanation	40.7	31.3	28.1
Coordination	13.1	67.8	19.1
Agreement	54.9	24.9	20.3
Other	25.6	34.1	40.3
Off task	20.2	12.4	67.3

Table 3 Program functionality score and gender of the students.

Dyad	Name (Pseudonym)	Gender	Program Functionality Score
Dyad 1	Susan	Girl	8
	David	Boy	5
Dyad 2	Stephen	Boy	1
	Linda	Girl	3
Dyad 3	Noah	Boy	17
	Jack	Boy	9
Dyad 4	Lucas	Boy	8
	Harry	Boy	3
Dyad 5	Asher	Boy	10
	Logan	Boy	6
Dyad 6	Mia	Girl	10
	Evelyn	Girl	11
Dyad 7	Elena	Girl	11
	Abigail	Girl	14
Dyad 8	Chloe	Girl	4
	Aria	Girl	6
Dyad 9	Mary	Girl	4
	James	Boy	7

Case 1: Boy-girl groups

In the boy-girl collaboration, resonating to the general findings, *self-explanation* (35%) occurred most of the turns of talk. The next highest use was *simple questions* (16%). Exploratory categories such as *Justified disagreement* (.34%), *Justification* (1.69%), *Alternative idea* (.58%), *Higher-order questions* (.16%) occurred lower than the cumulative categories like *Suggestion* (9.54%), *Comment* (3.60%), *Coordination* (1.75%), *Agreement* (6.78%), and they involved in *off-task* talk 3.69% of the time. Dyads also had frequent *disagreements* (6.3%) among which, most were *unjustified disagreements* (85%).

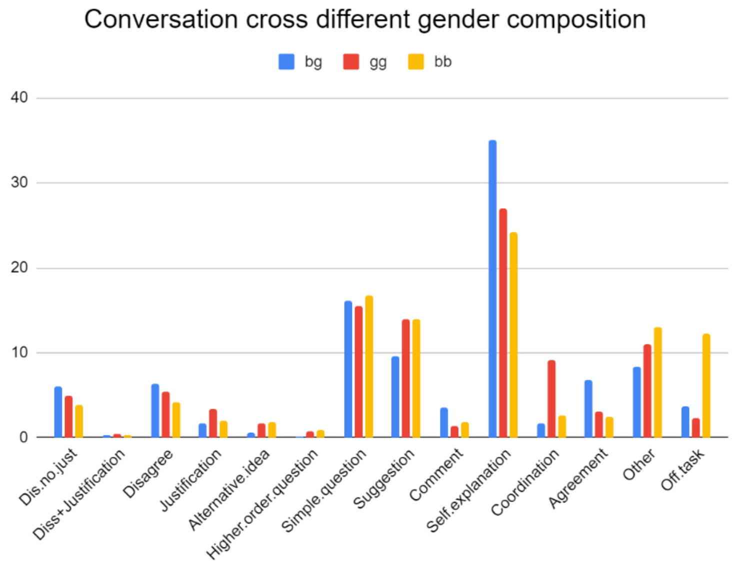


Figure 2: Frequency of Conversation categories between boy-girl, girl-girl and boy-boy pairs.

We explored the coded transcripts in excel to get a deeper understanding of how dyads in this case were collaborating. The first thing we noticed was the use of *suggestion.*

Excerpt 1

4:42	Mary:	I'm doing a beach theme.	Suggestion, Self-explanation
4:45	James:	0-8-1-8-2. Let's do... enter.	Suggestion
4:51	Mary:	I want mine to spin.	Suggestion
5:01	James:	Oh, crap.	Other
5:09	Mary:	This is going to bite me in the butt. I definitely doing this.	Other, Self-explanation
5:28	James:	Oh, mine is way too small. My fish is very small.	Self-explanation
5:40	Mary:	Yeah.	Agreement
5:51	James:	My griffon will be flying on the doughnut.	Suggestion
5:51	Mary:	Yay!	Agreement
6:54	James:	Okay, now I've got to do the griffon. I'll have the griffon fly, or glide.	Self-explanation
6:56	Mary:	I'm just going to do this.	Self-explanation
6:57	James:	No! I can't do the griffon, I just forgot. Okay, what's my next one. King. I forgot.	Self-explanation
7:04	Mary:	Okay, I'm going to try this. What?	Self-explanation,Simple question
7:13	James:	Trying a king. You've got to be kidding me. No king.	Self-explanation, Other
7:24	Mary:	That sucks. Too bad.	Comment

In Excerpt 1 from the dyads James and Mary (Pseudonyms), we can see high frequencies of both *suggestion* and *self-explanation*. Here, at the beginning, we can see that Mary *self-*

explains that she is doing a beach theme which is also a new idea that she introduces, thus we consider it a *suggestion*. However, when we watched the video records, we saw that she immediately executed what she said. Thus, her intention probably was more to *self-explain* what she was doing instead of *suggesting* and deciding together. On the other hand, he edited the size of his own theme which is not a beach theme. Moving forwards, James looked at his fish sprite and explained that it is too small. To this, Lucy agreed saying "yeah", but never looked at James's computer to check on the size. Followed by that, this dyad had consecutive *suggestions* for some time. They continued to introduce new ideas as *suggestions* but in an individualized manner, not including the partner in the decision making.

Excerpt 2

11:33	Susan:	Look. So we're going to draw our vase. This how I'm going to draw the vase.	Self-explanation, Suggestion
11:45	David:	I'm going to draw mine like this. You can draw your own. I want mine like this. And then I'm going to...	Self-explanation, Suggestion
11:52	Susan:	You want yours to look symmetrical?	Simple question, Suggestion
11:52	David:	And then... more of this.	Self-explanation
12:00	Susan:	I'm not going to boss you around, I'm just saying, like... Okay, I made my vase.	Self-explanation, Justification
12:19	David:	Oh no. What..	Other
12:19	Susan:	It's good.	Comment
12:26	David:	I know it might not look that good, but it's better than nothing. It's better than not having a vase. How do I make it bigger?	Self-explanation, Simple question

We found this trend in the other two dyads as well. Such as in Excerpt 2, Susan and David were talking about a vase sprite. Susan *suggested* a way to draw the sprite. Then David was drawing something which did not look like a vase and *suggested* that he wanted his like that. Though Susan *suggested* David make his sprite different, asking if he wanted to make it symmetrical, David kept on doing what he decided to do. We found many such instances in the three boy-girl dyads where the partners were doing their own thing.

Boy-girl collaboration also showed frequent use of *disagreements*. When we explored those instances, we found that they are short disagreements like "No." "No, you need that." And there are instances when there is no response to disagreement. Or responses like "I don't care" "No, it's not" to "Yes, it is". Excerpt 3 from Susan and David's talk gives an example of what such *disagreements* look like.

Excerpt 3

22:14	David:	No, what are you doing?	Disagree, Simple question
22:14	Susan:	I'm trying to make it block.	Self-explanation
22:19	David:	Don't need that. Susan, what are you doing?	Simple question, Disagree
22:21	Susan:	Wait... I'm trying to make it block.	Self-explanation
22:27	David:	What do you need a block for? Everything we need is here.	Simple question, Justification
22:37	Susan:	No it's not.	Disagree
22:37	David:	Yeah it is.	Disagree
22:39	Susan:	No, it's not.	Self-explanation
22:41	David:	Yeah	Disagree
22:48	Susan:	Let's stop after 1 time. Let's see.	Self-explanation,Alternative idea
22:48	David:	No.	Disagree

In this episode, Susan is trying to stop a sprite from moving. While she tries to do that, David keeps on asking what she is doing and *disagreeing* with her actions. However, Susan does not provide any justification as to why she is doing that and David too says they do not need that action, but does not provide a reason why they do not need that. This excerpt is part of a longer episode where they keep on disagreeing with each-other with short utterances like "No, stop." or "No, I need to." Another example of such short *disagreements* is from another boy-girl dyad Stephen and Linda.

Excerpt 4

28:16	Stephen:	Oh, it's a blank field. Oh yeah, it is.	Self-explanation
28:19	Linda:	Yeah.	Agreement

28:20	Stephen:	But maybe you have to do when [inaudible]	Alternative idea
28:25	Linda:	Shh. No.	Disagree
28:26	Stephen:	Meanie. No, Maybe-	Disagree
28:30	Linda:	Shh..	Disagree
28:49	Linda:	What is happening? Oh there it is, it's in purple.	Simple question, Suggestion
28:52	Stephen:	No.	Disagree.

As they were trying to edit a code block together, Linda started work on a sprite on her own. As Stephen shared *ideas* with her, Linda stopped him, saying "Shh" without much explanation. This episode of their conversation led Stephen to make his decision about the block himself and he continued to work individually on that for some time.

In the case of boy-girl coding activity, we noticed that all three of the dyad's Scratch outputs were very different within dyads. Both the teacher and the researchers emphasized to the students that dyads needed to make identical games by making joint decisions. However, Scratch outputs among boys and girls in a dyad were different in terms of most of the components measured in the Program functionality measurement scale. Mary got a score of 4 and James 7; Linda 5 and Stephen 0; Susan 5 and David 8. The difference in their score shows that their game did not have identical components. Figure 3 shows one example of their game output in Scratch. Outputs in this figure are from Mary (Top artifact) and James's (Bottom artifact) pair programming. We can see that their background, sprites (characters) are all different from each other.

<u>*Case 2: Boy-boy collaboration*</u>

Dyads in boy-boy also had uttered *self-explanation* (24.18%) in most of the turns of talk. The next highest use was *simple questions* (16.73%) and *suggestions* (13.92%). Some Exploratory categories such as *Justified disagreement* (.32%), *Higher-order questions* (.86%) occurred less than other categories, however, *justification* (1.93%), *alternative idea* (1.80%) amd

Cumulative categories like *comment* (1.78%) occurred less than other categories. Cumulative categories like *agreement* (2.5%), *coordination* (2.56%). Dyads in this case had a higher amount of *other* (13%) and *off-task* (12.29%) talk.

Conversation in the dyads of this case has a comparatively higher percentage of *off-task* talk. To understand how and when these *off-task* conversations are occurring, we explored all these instances. We found that the *off-task* talk did not occur randomly between other categories of conversation, rather, they were episodic and occurred consecutively for several turns of talk (6 turns of talk on average). For example, similar to the following example (Excerpt 5) of Noah and Jack, they were found to talk about personal incidents several times through their activity. The video shows they were still editing code in their computer while talking off task, although sometimes taking individual decisions on certain aspects of the game they were building. For example, when Noah and Jack were having the conversation presented in Excerpt 5, Noah was adding a sprite in his workspace and Jack was editing code for the motion of a sprite. While they talked, they were continuously working on their computer. We found this pattern to be similar in other instances of *off-task* episodes.

Excerpt 5

13:38	Noah	That's what it's called, the toxic towers. If you enter, you'll be toxic.	Off task
13:43	Jack	Yeah. You'll be toxic as [inaudible 00:13:46].	Off task
13:45	Noah	My cousin is so toxic. He's like, "Your mom."	Off task
13:51	Jack	There's this toxic kid on [inaudible 00:13:56] monkey liquid?	Off task
13:57	Noah	Monkey liquid? Cool. Yeah. Well this kid at my old school…[talks about personal incident for about 30s]	Off task
14:25	Jack	Oh my eyes.	Other
14:26	Noah	That's toxic at its finest, man. I said…[keeps talks about this personal incident for about 40s.]	Off task

One of the *off-task* episodes of another boy-boy dyad Lucas and Harry was about some music they could hear from the classroom. They talked about it for almost two minutes. Parts of it are presented in Excerpt 6. Similar to the other example, the video showed that they were simultaneously working in their coding workspace. However, they were not working on the same aspect of the game. Lucas was drawing a sprite and Harry was editing a conditional block.

Excerpt 6

11:07	Lucas	Yeah. It's like loud.	Off task
11:11	Harry	It's like, yeah, it's too loud.	Off task
11:11	Lucas	I hear that a lot. We hear that every time we're here.	Off task
11:14	Harry	No, I hear it at the gym a lot because when we're running you can hear it. But then like...	Off task
11:19	Lucas	Yeah, but from here you won't be able to hear it.	Off task
11:21	Harry	Yeah, it's kind of weird. Is it music class? Because music is like...	Off task
11:25	Lucas	Music is right there.	Off task
11:27	Harry	It's coming that way. It's coming that way though.	Off task
11:32	Lucas	Wait, is music right there? Yeah, music's right there. I don't know. But that's weird.	Off task
11:37	Harry	It's not any music I've heard from music class.	Off task

Next, we explored the frequently occurring categories in this case - *self-explanation*, *simple questions,* and *suggestions.* While instances of *self-explanation* and *simple-questions* occurred as we expected, use of *suggestion* was interesting. Instances when *suggestions* occurred in this group, students were not directly suggesting but rather informing their partner about the idea they are implementing. Such as saying "I'm using a gumball, actually," which sounds like *self-explanation* but actually is informing the partner about a new-idea. We found many instances of such indirect *suggestion* in conversation of all the three dyads.

In terms of the Exploratory categories, the boy-boy pair used a minimal amount of *justification, higher-order questions, alternative ideas, and justified disagreements.* The template analysis did not show any significant qualitative differences in the use of these categories than usual ones.

In terms of the program functionality, the Scratch output of dyads in this case showed that partners were at different levels in their activity. The differences in score between the partners were high and showed that one of the partners in all three dyads were at an advanced level. In the boy-boy case of collaboration, Asher got a score of 10 and Logan 6; Lucas got 8 and Harry 3; Noah got 17 and Jack 9. The scratch game artifacts show that in all the three cases, the partner with the lower score had the same number of sprites, the I-SPY riddle was also written similarly like the other partner, however, the details of the sprite-like motion when a sprite is clicked were missing. For example, in Asher's scratch output, there were sprites such as a pigeon and a fish which when clicked would move in different directions. Logan's scratch output had the same sprites, but they did not move when clicked. From this, we can interpret that Asher used conditionals as 'when clicked,' but Logan did not use them. Similarly, Noah's scratch output had two levels for the game where each level had a different I-Spy game, and each level had a riddle and sprites which moved when clicked. Noah's partner Jack also has two levels, however, sprites on the second level did not move when clicked. Here too, we can interpret that Jack could not use motion code for his game.

Case 3: Girl-girl collaboration

In the girl-girl collaboration, like the other two cases, *self-explanation* (26%) occurred most of the turns of talk. The next highest use was *simple questions* (15.45%) and *suggestions* (13.97%). Some Exploratory categories such as *Justified disagreement* (.40%), *Higher-order questions* (.83%) occurred less than other categories, however, *justification* (3.36%) and *alternative idea* (1.74%) in this case were used in similar frequency like some Cumulative categories like *comment* (1.38%) and *agreement* (3.07%). In girl-girl cases, *Coordination* (9.07%) occurred more times than many other Cumulative and Exploratory categories. Also, dyads in this case had an average of 11% *other* type of talk and 2.27% *off-task.*

As we did a template analysis of the coded transcripts, we found interesting patterns on using *coordination* categories of conversation. One of the dyads, Mia and Evelyn, made this *coordination* protocol of counting from three to one to make sure they are on the same page. Excerpt 7 shows an example of how Mia and Evelyn were trying to *coordinate* their work. We observed using this countdown system throughout their collaboration. It also showed in their Scratch output which were very similar where Mia scored 6 and Evelyn 5.

Excerpt 7

6:34	Mia	Okay, lets make it hide. And then what if I clicked show.	Suggestion, Agreement, Simple question
6:35	Evelyn	Wait for me!	Coordination
6:36	Mia	I am trying to.	Self-explanation
6:37	Evelyn	Countdown from three and I'll be ready.	Coordination
6:37	Mia	I'm not even ready yet.	Coordination
6:39	Evelyn	Okay. Lets save. Three, two, one	Suggestion, Coordination

The other two pairs were also seen to be using *coordination* very frequently. For example Elena and Abigail waited for each other when the other lagged behind. On one instance Elena said “Hey, wait for me,” after which Abigail paused and looked at Elena’s screen and helped her by *suggesting* she fix a conditional block. 85% of the *coordination* utterances of this dyad had the word ‘wait.’ The other dyad Choe and Aria had a different *coordination* pattern where they checked on each other to get informed on what the other person was doing. Except 8 shows an example of this where Choe asks Aria if she is making a basket (a sprite) and keeps following with questions to know where Aria is at. The scratch outputs of these two dyad also were very similar if not identical.

Excerpt 8

6:09	Choe	Oops. Why can I not get this like a good circle? Okay, there you go. Now let's make a circle. Are you making your basket?	Self-Explanation, Coordination
6:10	Aria	Nope, I'm not.	Other
6:11	Choe	What are you making? What? What are you making?	Simple question, Coordination
6:11	Aria	One second.	Other
6:12	Choe	I can not hear you.	Coordination

6:12	Aria	I know.	Other
6:13	Choe	What are you making?	Simple question, Coordination
6:13	Aria	My flashlight.	Other

The next notable observation on transcripts of dyads in this case are their use of *disagreements*. The three dyads rarely used short *disagreements,* but rather they either elaborated with a *justification* for their *disagreements* or shared an *alternative idea* or a *suggestion.* For example, when Mia *suggests* to move the sprite ten degrees when clicked, we see in the video that Evelyn immediately puts 10 and checks if it worked and after seeing that it did not work, she says an *alternative* suggestion to use five. We saw this pattern of adding further information with a *disagreement* all the time in Mia-Evelyn.

Excerpt 9

10:11	Evelyn	Hiccup, hiccup, hiccup, why? That's not what you wanted it to do. So is it not going to move steps?	Justification, Higher order question, Simple question
10:13	Mia	Oh yeah move steps after it turns ten degrees.	Suggestion
10:13	Evelyn	No, five maybe.	Alternative idea, Disagree
10:14	Mia	Five. Okay, do you want to try it out?	Simple question, Agreement

Similarly with the other dyad Elena and Abigail, they almost all the time had addition to their *disagreements*. Excerpt 10 gives an example of that. Abigail *disagrees* with what Elena was doing and *suggests* several steps to help her fix a code block. Elena follows her, but then *disagrees* and adds a *justification* by referring to a previous incident.

Excerpt 10

9:52	Abigail	No. Pull forever off of when this sprite is clicked. Now push the flag and now put your zebra back behind the rock. Not up there, like down, down.	Suggestion, Disagree
10:06	Elena	I know what you mean.	Agreement
10:07	Abigail	Okay. Now go here. Now. Go here. Go to costumes. Go to costumes. Go to... Wait a minute. Delete the top zebra. Go to the top. There. Click that. No, click that. Push X. Now go to sprites and push the paintbrush. No. Don't click it. Just go on it. Now go to the paintbrush. Now we can paint our hearts.	Suggestion, Disagree

10:40	Elena	No, remember we did this last time and we couldn't program them.	Disagree, Justification
10:43	Abigail	We couldn't program them because it was our background. We didn't make them a character. See, do you get what I mean?	Justification, Simple question
10:51	Elena	Mm-hmm [affirmative].	Agreement

The third dyad, Aria and Chloe had only five *disagreements* and all of them had a *justification* or an *alternative idea* with it.

In terms of the program functionality in the case of girl-girl collaboration, Mia and Evelyn scratch output was scored 10 and 11, Abigail and Elena 14 and 11, Aria and Chloe 6 and 4. The dyad's outputs were very similar if not identical in terms of functionality score. The minimal differences they had were either related to the number of sprites or types of motion. Between Mia and Evelyn, Mia had three sprites and Evelyn had more allowing an extra score point. Abigail had more sprites in the level two of her I-SPY game, but Elena had only two, thus getting three score points less than Abigail. Aria and Chole both had only one level of the game. Aria had two sprites with motion when clicked but Chloe had only one sprite with motion. Each partner in a dyad in this case has the same riddle and same background, type of sprites and motions when clicked as their partner.

Case Comparison

Individualized decision-making in boy-girl collaboration

Between all three cases, mixed gender collaboration showed more individualized decision making than the other two cases. Figure 2 shows they had more *disagreements* than the same-gender dyads. However, analysis of their conversation showed that their *disagreements* were short without further elaboration on their thoughts. In contrast, girl-girl collaborations were found to have more substantive, but productive *disagreements,* having follow-up *suggestions* or *alternative ideas* or *justification* with a *disagreement.* Boy-boy dyads did not show any such trend in how they disagreed with each other. We found both short *disagreements* as well as ones that had additional information along with those *disagreements.*

Mixed-gender dyads also were *suggesting* in a self-explaining manner where they were telling the other person what they were doing. Additionally in many such instances we found that each partner was telling each-other what they were doing but seldom mutually decided on solution paths. Although, boy-boy collaboration had such individualized *suggestions*, we found one partner was typically following the *suggestions* of the other throughout the collaboration. In contrast, girl-girl collaboration mostly had *suggestions* in a collaborative manner, using collective words or phrases like 'let's,' 'how about,' 'we can' etc.

Figure 3: Differences in output of Programming activity in a boy-girl dyad.

Finally, the Scratch output in mixed-gender dyads were very different from each other (Figure 3). Their I-SPY riddle, sprites, motions and backgrounds were very different from each-other. Same gender dyads had relatively similar scratch outputs. Although, boy-boy dyads had more differences in score as compared to girl-girl and boy-girl dyads, the components like number of sprites, I-SPY riddle, backgrounds were similar. However, it was not the case of boy-girl pairs, where most components of the dyads were different, again indicating individualized decision-making.

Distraction and imbalance of activity in boy-boy collaboration

Compared to other two cases, boy-boy dyads' conversation had a higher proportion of *off-task* conversation (Figure 2). As we presented above, we found boy-boy dyads talking about topics unrelated to the tasks while simultaneously working on their programming task. Thus, conversation was used less to move project work forward. However, similar to mixed-gender dyads, did make use of *suggestion.* Like boy-girl dyads, they were introducing new ideas (*suggestions*) by *self-explaining* what they were doing instead of waiting for the partner's response. However, unlike boy-girl pairs, only one partner in each boy-boy dyad was sharing such *suggestions.* In all three of the boy-boy dyads, the other partner was either following (two dyads) or ignoring (one dyad) those *self-explaining suggestions.*

Differences in scores of program functionality between partners in boy-boy dyads were higher than girl-girl or boy-girl pair indicating an imbalance between the partners' activities. The average difference between partners in boy-boy dyads is 5.3 (8, 5, 3), whereas boy-girl dyads' is 2.6 (3,2,3) and girl-girls' is 2 (1,3,2).

High coordination and Reasoning in girl-girl collaboration

Figure 2 shows that the mean proportion of *coordination-related* utterances in girl-girl dyad conversation is higher than boy-boy or mixed-gender dyad conversation. The conversation analysis shows that these dyads have consistently tried to align their coding activity, whether by having a structured *coordination* system (i.e. countdown) or waiting and checking with the partner. On the other hand, *coordination* in the boy-boy or boy-girl dyads were less frequent and, when they did occur, often indicating they were out of sync, using words like "wait," or "wait a sec."

In addition to being coordinated, the girl-girl conversation included more *justification* for their actions. As presented above, we see that girl-girl dyads *justify* their *disagreements* and/or

Two screenshot of games created by girls in a girl-girl dyad. elaborate their thoughts through adding *suggestions* or *alternative ideas* with the *disagreement.* This is different from boy-girl collaboration, where we found brusque unresolved *disagreement* without much *justification* to that *disagreement* or addition of *suggestion* to move forward in resolving the disagreements. Boy-boy dyads on the other hand had the lowest proportion of *disagreement*, the existing ones involving both short *disagreements* but also a few ones that had follow up conversation with *suggestions* and *justifications.* However, it seems the boy-boy dyads were less committed to producing an aligned pair of programs for the game.

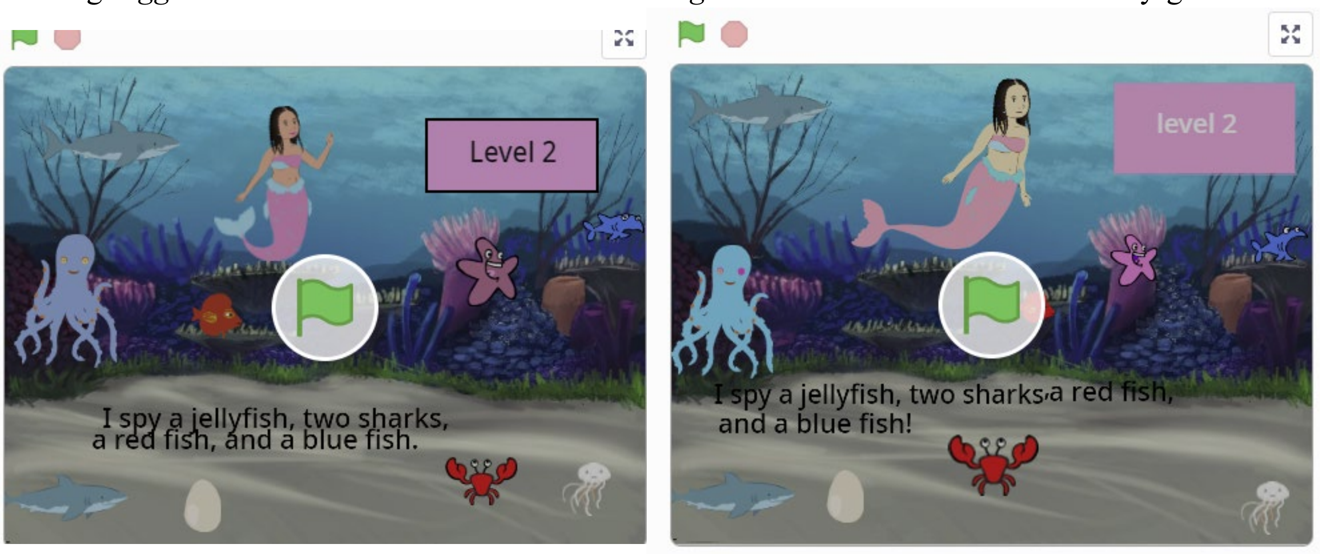


Figure 4 Screenshots of games programmed by each partner in a girl-girl pair.

Compared to the other two cases, scratch output of girl-girl dyads' were closest to being identical (Figure 4). However, the mean functionality score (9.6) was similar to boy-boy dyads (mean = 9).

Discussion

As we explored how different gender-grouped cases of collaborative programming compare in terms of discourse and program functionality, we found distinguishing patterns supporting previous studies and raising a few issues in terms of gender-grouped pairings. In summary, three interesting patterns distinguish collaborative practices of the three cases from each other. Boy-girl collaboration involved more individualized decision making, boy-boy collaboration more imbalanced program functionality with more occurrence of off-task talk while girl-girl collaboration involved high coordination and well-justified decision-making.

Literature shows that balanced-gender groups have the most diverse cognitive exchanges (Wu & Wang, 2020) , however, our findings on this sample indicate that mixed gender dyads lack cognitive congruence when collaborating . Mixed gender dyads spoke the least amount of utterances compared to the other two cases, had blunt short disagreements and uttered what they were doing instead of taking shared decisions. To some extent, this individualized nature is reflected in their programming outputs. While they were required to produce identical programs, the game artifacts we measured were often very different from each other. Studies exploring collaborative interaction under a computer-supported environment also have similar findings (Fitzpatrick, & Hardman, 2000; Underwood et al., 2000). Underwood, Underwood and Wood (2000) found that boy–girl pairs showed lower levels of verbal interaction and less keyboard co-operation. Additionally, their findings indicate that effective group work fails to occur when one or more of the members dominates the discussion or when the interaction tends to be negative and socially divisive. These conditions are more likely to occur in both mixed-gender groups and larger groups of males with uneven dominance. We also saw trends of one boy dominating the suggestions and developing code in the boy-boy dyads. However, the boy did not necessarily dominate code development in the mixed-gender dyads.

We found the two same gender cases to be more collaborative, specially girl-girl dyads who were well-coordinated and having substantive cognitive exchange through exploring ideas and reasoning. Studies support this finding that girl-girl dyads are more interactive and balanced in any type of collaborative setting whether computer- supported or not (Ding et al., 2011; Fitzpatrick & Hardman, 2000; Underwood et al., 2000). On the other hand, our findings show that boy-boy dyads are imbalanced in outcome measures and involved in more off-task conversations. Teachers may expect some level of off-task talk for socialization purposes and research shows that a certain amount of off-task in collaboration can set the tone for better collaborative behavior (Langer-Osuna et al., 2020). However, it is also important to note that, boy-boy dyads had imbalance in programming performance, meaning, one partner's program functionality was much higher than the other one. It may have to do with accepting one partner from the same gender to be the more knowledgeable one, further research needs to include other factors such as, students' prior knowledge and attitude to evaluate this assumption.

Our findings resonated with previous findings that female-female pairs are more active in social learning than male-male pairs (Jiang et al., 2017; Lin et al., 2020; Wu & Wang, 2020). Similarly, previous studies indicate girl-girl pairs participated in constructing game artifacts more uniformly as well (Jiang et al., 2017). More broadly, studies indicate that the performance of female students in computer-supported environments has been found to be greater when female students are the majority in the classroom and the instructor is female which was also true for our study settings (Bailey et al., 2020).

These findings hold several implications for teaching programming in elementary classrooms. Teachers can differentiate instructions for same and mixed gender pair. Monitoring progress may include checking with boy-boy pairs if they are on task, prompting mixed gender pairs to participate in exploratory talk, and girl-girl pairs to challenge each-other for higher quality programs. Overall, it might be worth teachers noticing two things regarding collaboration: First, checking if dyads are in line with expectations on collaborative behavior. Setting expectations can work as ground rules for using specific forms of collaboration (i.e. Exploratory talk). Second, Monitoring progress of individuals within dyads to check if specific students need more help in the coding process.

Three note-worthy limitations should be kept in mind while interpreting and implementing the findings of this research. First, the generalization of the results found here is limited as our findings are based on collaborative programming of only nine dyads from only one urban school. Future studies could replicate the current study on qualitatively different samples with demographic diversity. Second, several potentially confounding factors were not taken into account. For instance, it is possible that some of the differences were results of mutual liking between dyad members' (reciprocated) friendship, dyad members likeability status, classroom culture, task structure, teacher behavior, communication skills of the individual students, and so forth. Third, although great efforts were taken to construct a valid and reliable discourse analysis framework covering the complete range of dimensions related to the different types of collaborative talk, there is always the risk of interpretation bias. Future researchers could therefore replicate the current study testing the analytical framework on different computer supported collaborative environments. Another valuable direction for future research is to learn more about the beliefs and interests of different genders and how that impacts their collaboration and performance in pair programming.

Conclusion

Our multiple case study on the collaboration of different gender-grouped dyads provide an in-depth understanding of the collaborative programming practices. Our findings are in accord with the idea that there are differences in nuanced characteristics of collaborative programming between boys and girls in pair programming. Mixed gender dyads showed the least harmonious collaboration. Compared to the same-gender dyads they conversed less and mostly had unproductive conflicts. Same gender dyads were more collaborative, while girl-girl dyads showed more Exploratory talk as well as more synced in programming. However, boy-boy pairs had much within differences in programming outcome measure with comparatively high frequency of off-task episodes. Findings hold implications for how teachers pair students, design differentiated instructions to provide sufficient opportunities and conditions to achieve constructive collaborative communication.

Reference

Bailey, E. G., Greenall, R. F., Baek, D. M., Morris, C., Nelson, N., Quirante, T. M., ... & Williams, K. R. (2020). Female in-class participation and performance increase with more female peers and/or a female instructor in life sciences courses. *CBE—Life Sciences Education*, *19*(3), ar30.

Basu, S. (2019, February). Using rubrics integrating design and coding to assess middle school students' open-ended block-based programming projects. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 1211-1217).

Baytak, A., & Land, S. M. (2011). Case study: Advancing elementary-school girls' programming through game design. *International Journal of Gender, Science and Technology*, *3*(1).

Braught, G., Eby, L. M., & Wahls, T. (2008, March). The effects of pair-programming on individual programming skill. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education* (pp. 200-204).

Bennett, N., & Cass, A. (1989). The effects of group composition on group interactive processes and pupil understanding. *British Educational Research Journal*, *15*(1), 19-32.

Chi, M. T., & Menekse, M. (2015). Dialogue patterns in peer collaboration that promote learning. *Socializing intelligence through academic talk and dialogue*, 263-274.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, *20*(1), 37-46.

Creswell, J. W., Klassen, A. C., Plano Clark, V. L., & Smith, K. C. (2011). Best practices for mixed methods research in the health sciences. *Bethesda (Maryland): National Institutes of Health*, *2013*, 541-545.

Curşeu, P. L., Chappin, M. M., & Jansen, R. J. (2018). Gender diversity and motivation in collaborative learning groups: the mediating role of group discussion quality. *Social Psychology of Education*, *21*(2), 289-302.

Ding, N., Bosker, R. J., & Harskamp, E. G. (2011). Exploring gender and gender pairing in the knowledge elaboration processes of students using computer-supported collaborative learning. *Computers & Education*, *56*(2), 325-336.

Fitzpatrick, H., & Hardman, M. (2000). Mediated activity in the primary classroom: girls, boys and computers. *Learning and Instruction*, *10*(5), 431-446.

Gilbert, M. A. (2004). Emotion, argumentation and informal logic. *Informal Logic*, *24*(3).

Grover, S. (2021, March). Toward A Framework for Formative Assessment of Conceptual Learning in K-12 Computer Science Classrooms. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (pp. 31-37).

Grover, S., Basu, S., & Schank, P. (2018, February). What we can learn about student learning from open-ended programming projects in middle school computer science. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 999-1004).

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.

Hennessy, S., Howe, C., Mercer, N., & Vrikki, M. (2020). Coding classroom dialogue: Methodological considerations for researchers. *Learning, Culture and Social Interaction*, *25*, 100404.

Isohätälä, J., Näykki, P., Järvelä, S., & Baker, M. J. (2018). Striking a balance: Socio-emotional processes during argumentation in collaborative learning interaction. Learning, Culture and Social Interaction, 16, 1-19.

Jarratt, L., Bowman, N. A., Culver, K. C., & Segre, A. M. (2019, July). A large-scale experimental study of gender and pair composition in pair programming. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 176-181).

Jiang, M., Li, Y., Zheng, J., & Han, X. (2017). Gender group differences on behavior patterns in collaborative problem solving through LEGO. *Journal of Computers in Education*, *4*(2), 127-145.

Cassell, C., & Symon, G. (Eds.). (2004). *Essential guide to qualitative methods in organizational research*. sage.

King, N. (2004). Using Templates in the Thematic Analysis of Text. In C. Cassell & G. Symon (Eds.), *Essential guide to qualitative methods in organizational research* (pp. 256–270). Sage.

Kohlbacher, F. (2006). The use of qualitative content analysis in case study research. In *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research* (Vol. 7, No. 1, pp. 1-30). Institut für Qualitative Forschung.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, 159-174.

Langer-Osuna, J. M., Gargroetzi, E., Munson, J., & Chavez, R. (2020). Exploring the role of off-task activity on students' collaborative dynamics. *Journal of Educational Psychology*, *112*(3), 514.

Lin, Y. T., Wu, C. C., Chen, Z. H., & Ku, P. Y. (2020). How Gender Pairings Affect Collaborative Problem Solving in Social-Learning Context: The Effects on Performance, Behaviors, and Attitudes. *Journal of Educational Technology & Society*, *23*(4).

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2003, May). The impact of pair programming on student performance, perception and persistence. In 25th International Conference on Software Engineering, 2003. Proceedings. (pp. 602-607). IEEE.

Mayring, P. (2015). Qualitative content analysis: Theoretical background and procedures. In *Approaches to qualitative research in mathematics education* (pp. 365-380). Springer, Dordrecht.

Mercer, N. (2019). *Language and the joint creation of knowledge: The selected works of Neil Mercer*. Routledge.

Mercer, N., & Littleton, K. (2007). *Dialogue and the development of children's thinking: A sociocultural approach*. Routledge.

Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Yang, K., Miller, C., & Balik, S. (2003). Improving the CS1 experience with pair programming. *ACM SIGCSE Bulletin*, *35*(1), 359-362.

National Research Council. (2012). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. National Academies Press.

Plantin, C. (2004). On the inseparability of emotion and reason in argumentation. *Amsterdam Studies in the Theory and History of Linguistic Science Series 4*, *248*, 265-276.

Ruvalcaba, O., Werner, L., & Denner, J. (2016, February). Observations of pair programming: Variations in collaboration across demographic groups. In *Proceedings of the 47th ACM technical symposium on computing science education* (pp. 90-95).

Saito, D., Kaieda, S., Washizaki, H., & Fukazawa, Y. (2020). Rubric for Measuring and Visualizing the Effects of Learning Computer Programming for Elementary School Students. *Journal of Information Technology Education: Innovations in Practice*, *19*, 203-227.

Schegloff, E. A. (1991). Conversation analysis and socially shared cognition.

Underwood, J., Underwood, G., & Wood, D. (2000). When does gender matter?: Interactions during computer-based problem solving. *Learning and Instruction*, *10*(5), 447-462.

Wei, X., Lin, L., Meng, N., Tan, W., Kong, S.-C., & Kinshuk. (2021). The effectiveness of partial pair programming on elementary school students' Computational Thinking skills and self-efficacy. *Computers & Education*, 160, 104023. doi: 10.1016/j.compedu.2020.104023

Wegerif, R and Mercer, N (2000) Language for Thinking: a study of children solving reasoning test problems together. In Cowie, H, Aalsvoort, D. *Social Interaction in Learning and*

Instruction: the meaning of discourse for the construction of knowledge. Oxford: Elsevier, pp 179-193. ISBN 0080435971

Weintrop, D. (2019). Block-based programming in computer science education. *Communications of the ACM*, *62*(8), 22-25.

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

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, *12*(3), 197-212.

Wu, S. Y., & Wang, S. M. (2020). Exploring the effects of gender grouping and the cognitive processing patterns of a Facebook-based online collaborative learning activity. *Interactive Learning Environments*, 1-15.

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

Zakaria, Z., Boulden, D., Vandenberg, J., Tsan, J., Lynch, C., Wiebe, E., & Boyer, K. E. (2019). Collaborative Talk Across Two Pair-Programming Configurations. *In Proceedings of 13th International Conference on Computer Supported Collaborative Learning* (CSCL) 2019 (vol. 1, pp. 224-231). Lyon, France: International Society of the Learning Sciences.

Zakaria, Z., Vandenberg, J., Tsan, J., Boulden, D. C., Lynch, C. F., Boyer, K. E., & Wiebe, E. N. (2021). Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students. *Computer Science Education*, 1-28.

Zhan, Z., Fong, P. S., Mei, H., & Liang, T. (2015). Effects of gender grouping on students' group performance, individual achievements and attitudes in computer-supported collaborative learning. *Computers in Human Behavior*, *48*, 587-596.

CHAPTER 4: TWO-COMPUTER PAIR PROGRAMMING: EXPLORING A FEEDBACK INTERVENTION TO IMPROVE COLLABORATIVE TALK IN ELEMENTARY STUDENTS.

A version of this study can be found at:

Zakaria, Z., Vandenberg, J., Tsan, J., Boulden, D. C., Lynch, C. F., Boyer, K. E., & Wiebe, E. N. (2021). Two-Computer Pair Programming: Exploring a Feedback Intervention to improve Collaborative Talk in Elementary Students. *Computer Science Education*, 1-28.

Abstract

Background and Context: Researchers and practitioners have begun to incorporate collaboration in programming because of its reported instructional and professional benefits. However, younger students need guidance on how to collaborate in environments that require substantial interpersonal interaction and negotiation. Previous research indicates that feedback fosters students' productive collaboration.

Objective: This study employs an intervention to explore the role

instructor-directed feedback plays on elementary students' dyadic collaboration during 2-computer pair programming.

Method: We used a multi-study design, collecting video data on students' dyadic collaboration. Study 1 qualitatively explored dyadic collaboration by coding video transcripts of four dyads which guided the design of Study 2 that examined conversation of six dyads using MANOVA and non-parametric tests.

Findings: Result from Study 2 showed that students receiving feed-

back used productive conversation categories significantly higher than the control condition in the sample group considered. Results are discussed in terms of group differences in specific conversation categories.

Implications: Our study highlights ways to support students in pair programming contexts so that they can maximize the benefits afforded through these experiences.

Introduction

Researchers and practitioners have begun to incorporate collaboration in programming activities because of its reported instructional and professional benefits. Studies suggest that collaboration is an effective pedagogical approach for programming instruction for undergraduates (Hanks et al., 2011) and to prepare students for further education and workforce needs (National Research Council, 2013). It fosters higher-order thinking skills (Williams et al., 2002) and facilitates effective knowledge sharing through productive dialogue (Kavitha & Ahmed, 2015). Pair programming is where two individuals solve a programming challenge together. Though originating in industry, pair programming is often incorporated in college-level CS courses where the bulk of research on the pedagogical approach has occurred (Umapathy & Ritzhaupt, 2017; Williams et al., 2000). Fewer studies can be found at the high school and middle grades, and only a few studies can be found on elementary students' pair-programming practices. The research done with elementary students indicates that pair-programming with block-based programming environments can foster problem-solving skills, develop their understanding of programming concepts, and work as a motivating environment to explore other domain related concepts (Calder, 2010; Lai & Yang, 2011). Given the general acceptance of group work in the elementary grades as a teaching and learning strategy (i.e. collaborative inquiry, project-based problem-solving etc.) (Adams & Hamm, 1998; De Lisi & Golbeck, 1999) and the growing popularity of block-based programming for this age-range (Franklin et al., 2015), it seems worthwhile to explore pedagogical practices related to pair programming with elementary-aged students as well.

With the more common, popularized form of pair programming referred to here as 1-computer pair programming (1C), dyads share one computer and have a defined role in the collaboration: the driver controls the programming environment, and the navigator monitors the progress and anticipates future steps (Hanks et al., 2011). These defined roles in 1C pose several challenges in students' collaboration, such as imbalances in dialogue and work distribution, in addition to equity issues for elementary students (Lewis & Shah, 2015; Shah & Lewis, 2019; Shah et al., 2014; Tsan et al., 2018). Prior work has demonstrated that elementary students can show a clear disinclination towards 1C because of the restrictions under navigator role (Bradbury et al., 2019). As 1C involves a high degree of interpersonal negotiation and reconciliation during the turn-taking of driver and navigator roles, this is perhaps not surprising (Zakaria et al., 2019). Studies on children's social practices show that elementary-aged students are still developing necessary socio-emotional skills such as social problem-solving and showing empathy (Beauchamp & Anderson, 2010), which are needed when sharing a computer and negotiating turn-taking (Shah et al., 2014). These findings point to a need to explore alternative pair-programming strategies that better leverage the benefits of collaborative learning strategies. Some studies have also explored an alternative setting, referred to here as 2-computer pair programming (2C) (Figure 1), in which pairs of students collaborate side-by-side with individual computers on the same programming project (Bradbury et al., 2019; Zakaria et al., 2019). Such a setting is also referred to in the literature as distributed programming, concurrent-coupled programming or side-by-side programming (e.g., Dewan et al., 2009; Nawrocki et al., 2005;

Zakaria et al., 2019). Although such an environment seems promising given the independence it provides for pairs to learn and contribute to each other's work equally, it isimportant to examine the quality of collaboration in such a setting. In the 2C setting, partners do not have roles. Thus, supporting students during their collaboration needs to be structured in a way that scaffolds appropriate discourse so students can productively collaborate. Designing effective support for students in upper elementary grades(9–11 years) will require in-depth exploration of the collaborative processes of these students as they engage in 2C. By first determining the characteristics of collaboration during 2C, we can then design and implement instructional interventions that expand the scope of quality collaboration in this setting.

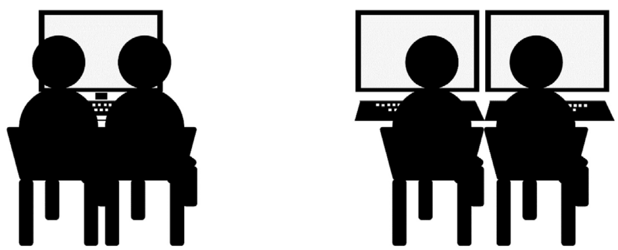


Figure 1: 1C and 2C Pair Programming (Zakaria et al., 2019).

Previous work

Collaborative talk

At the heart of the instructional impact of collaborative work is the constructive discourse between students that moves learning forward (Tudge, 1992). A framework by Mercer (2002) elaborates on students' dialogue, distinguishing more productive conversation from less productive ones. Drawing on Vygotsky's Zone of Proximal Development (ZPD) (Vygotsky, 1978) and Bruner's "Scaffolding" (Bruner, 1978), 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, 2002). In these joint activities, three types of conversations can occur, which Mercer termed Cumulative, Disputational and Exploratory talk (Table 1). In

Cumulative talk, speakers build positively but uncritically on what the other has said, whereas Disputational talk is characterized by disagreement and individualized decision making. In Exploratory talk, participants engage critically but constructively with each other's ideas which leads to improved reasoning and conceptual understanding (Bennett & Cass, 1989; Mercer, 2002). Exploratory talk has been shown to expand the joint ZPD by enabling partners to achieve a better mutual understanding of the problem (Fernández et al., 2015).

Previous studies that employed this framework to analyze students' conversations during pair programming showed that elementary and middle school students organically use Cumulative talk more than the other two types (Campe et al., 2020; Zakaria et al.,2019). Despite the knowledge that students' use of Exploratory talk is improved with teacher guidance (Mercer et al., 2004, 1999), Rojas-Drummond and Mercer (2003) foundthat teachers rarely instruct students in effective collaborative talk. However, when taught how to engage in Exploratory talk, students produced improved arguments, and critically considered alternative views during collaborative work (Rojas-Drummond & Zapata, 2004). Mercer et al. (1999) proposed a set of ground rules for Exploratory talk to use as guidance to facilitate collaborative discourse. With teachers' support, students could integrate partner's ideas and strategies, co-construct new ideas to contribute to the collaborative activity critically and productively (Warwick et al., 2013). Although 2C offers balanced autonomy for students–a departure from traditional driver-navigator roles–it involves a higher level of technology coordination as both partners in 2C have control ofa computer. It envisions more verbal negotiation for decision-making while students simultaneously edit in the workspace. Because 2C presents new dynamics in the collaborative affordances of pair programming, it is important to examine students' discoursein order to best leverage the affordances of this setting and enhance Exploratory talk.

Feedback on collaboration

Mercer and others believe that productive collaboration cannot simply emerge naturally but rather has to be taught and supported through modeling and feedback (Mercer, 1995; Mercer et al., 1999). Similar to cognitive skills, collaboration skills include procedural knowledge that inform students how to perform in a collaborative environment (Deiglmayr & Spada, 2010). Feedback is found to be effective when focused on different aspects of collaboration like group awareness (Pifarré et al., 2014), knowledge sharing breakdown (Soller, 2001; Zumbach et al., 2006), or unproductive communication (Gweonet al., 2006). It is also found that along with cognition, affective dimensions such as motivation can affect collaboration (Meier et al., 2007). Feedback on "social grounding"which Zumbach et al. (2006) summarize as interactions, collaboration, and motivational processes appear to be essential for learning. In terms of motivation towards collaboration, it is found that students with a growth mindset, which emphasizes the belief on effort exertion in utilizing different strategies to be successful in a task, value group work and its creative potential more (Alpay & Ireson, 2006; Dweck, 2013), and perform significantly better on programming with feedback focused on mindset (Cutts et al., 2010).

Preliminary work on elementary pair programming

A set of preliminary studies undergirds the studies reported here. Our initial classroom observations and other literature began to raise concerns with 1C with elementary students. These findings led to an exploration of 1C and 2C settings where we conducted post-hoc focus groups to gather more information on student experience and preference. Students who participated in both settings expressed they had less autonomy and opportunity to equally contribute to the activity in 1C (Bradbury et al., 2019). Prior research has demonstrated the advantages of using one's own computer rather than sharing a single computer during collaboration. Evidence shows that elementary students completed tasks with more timeefficiency and higher levels of enjoyment and engagement (Infante, 2009; Lewis, 2011; Scott et al., 2003). Moreover, 2C pair programmers have been found to produce higher quality industry products (Bandukda & Nasir, 2010). Similar advantages have also been found in studies conducted with undergraduates (Cockburn, 2004; Nawrocki et al., 2005).

Table 1 Conversation categories in the coding scheme and the inter-rater agreement (kappa) for Study 1.

Categories	Sub-categories	Description of sub-categories	Examples	*k*
Question	*Question*	Any question, concern, request for assistance, clarification, seeking or check confirmation.	"Where can I find the 'if' block?"	.85
	Respond to question	Any response to partner's question or concerns.	"You can search under categories"	.93
Discussion	*Agreement*	Agreement on any opinion, edits	Yeah, I think so too.	.83
	Disagreement	Disagreement on any edits or opinion.	No, this should be here. No, I think (pause)	.66
	Explanation	Explain what step they are taking or what edits they are doing or random iteration of what they are doing themselves. (have edited or editing in the present)	"if touching sprite 4, then stop" (while editing on the screen).	.70
	Suggestion	Any suggestions when directly talking to the partner and suggesting before taking an action.	"maybe we should add this block."	.88
	Reminder	Reminding, redirecting or warning about conducting any steps or to be on-task.	"You have to save it before exiting"	1.0
	Antagonistic action	Actions interactions that cause tension including hurtful comments instigating fights, prodding, putting down partner contributions, and showing annoyance with partner	"you are being ridiculous".	(no occurrence)
Manipulation	*Edits with consent*	Edits after discussion, agreement or getting a positive response from the partner.		(no occurrence)
	Edits without consent	Edits without discussing or getting any positive response.		.93

Current work

Exploring nuances of dyadic conversation may give us an in-depth understanding of elementary students' needs and help us design appropriate interventions to support effective collaboration during pair programming. Prior literature and our own work have pointed to the potential advantages of elementary students working in a 2C setting. We define 2C as a setting where student pairs each have their own computer screen, keyboard, and mouse (thus lowering points of conflict) but sit side by-side where they can leverage discursive practices to create a single solution to a programming problem. Depending on the programming environment (i.e. Netsblox, Scratch etc.) used in 2C, students could be working in a linked environment where their programming space was synchronized by the software, or in an unlinked environment where the nature of their physical proximity allowed them to visually and verbally compare and synchronize their programs manually. We conducted two consecutive studies with the goal of both more deeply understanding elementary students' collaborative discourse in 2C settings, but also to look at how feedback shapes this discourse. More specifically, the goal of Study 1 was to refine the coding scheme used in the preliminary studies reported above to adapt it for use to discern students' collaborative process in a 2C setting. The prior coding scheme was designed for use in a 1C context by Ruvalcaba et al. (2016) to analyze videos of dyadic collaboration and thus needed to be modified for a 2C setting. Utilizing those findings, a second study explored an intervention program to support students' collaboration in a 2C setting. We have not compared or contrasted these two studies, rather Study 1 was considered informative for designing and conducting Study 2.

Results and coding experiences in Study 1 helped us to modify and refine discourse analysis methods and design a feedback intervention to improve students' collaborative talk during pair-programming activities. We consider talking to be productive when it includes characteristics of Exploratory talk, less productive with Cumulative talk and unproductive with Disputational talk. Feedback was provided to the dyads during their collaboration to enhance their use of Exploratory talk. Study 2 utilized this revised and refined coding scheme to explore a feedback intervention informed both by Mercer's (2002) Exploratory talk as well as Dweck's (2013) growth mindset theories.

Study 1

For Study 1, we explore the following research questions: 1) How do dyads utilize different categories of collaborative talk? 2) What are the discursive characteristics of dyadic collaboration in 2-computer pair programming?

Method

The study took place at a suburban elementary school in the Southeastern United States. The participants were 15 fifth grade (10–11 years) students from which we used video data of 8 students, forming 4 dyads (formed by the teacher); 4 girls and 4 boys (Table 2).This subset of participants was chosen in terms of audio and video clarity. All the students were in an academically gifted student program and had been participating in programming lessons designed by the authors.

Table 2 Demographics of the students in Study 1.

Student pairs	Gender	Race
Sandy	Female	White
Anthony	Male	White
Dorothy	Female	White
David	Male	White
Melony	Female	White
Rupert	Male	White
Clara	Female	Biracial
Luke	Male	Biracial

Table 3 Frequency of conversation categories in Study 1.

Categories	Sub-categories	Melony and Rupert (%)	Dorothy and David (%)	Sandy and Anthony (%)	Clara and Luke (%)	Total (%)
Question	*Question*	6.7	10.0	27.3	19.4	18.1
	Respond to question	6.7	0.0	13.6	11.1	9.6
Discussion	*Agreement*	13.3	0.0	0.0	8.3	6.0
	Disagreement	0.0	0.0	4.5	8.3	4.8
	Explanation	53.3	60.0	36.4	36.1	42.2
	Suggestion	20.0	30.0	18.2	13.9	18.1
	Reminder	0.0	0.0	0.0	2.8	1.2
	Antagonistic	0.0	0.0	0.0	0.0	0.0

The lessons covered topics such as conditionals, loops, debugging, and game design throughout three days of activities; however, they were in 2C setting only one day during which they did several debugging activities. Fromthese activities, we used one debugging activity that required students to use all the other concepts taught before. Students were tasked with fixing the code so that a mouse (Sprite 1) stopped walking back-and-forth when a snowflake (Sprite 2), falling from the sky, touched it. In this 2C setting, students coded in NetsBlox (Broll et al., 2017). NetsBlox provided a linked, synchronous workspace (akin to Google docs) viewed on each of the dyad's computer. On average students took 6.3 mins to solve the problem and we analyzed a total of 25.53 mins of video data (Table 4). For this study, we used OpenBroadcaster Software (OBS) (Bailey, 2017) to align webcam, screen capture, and audio data(captured through headsets) into a single file. In addition, we collected the students' programs and log data using a mixed-methods approach (Creswell & Creswell, 2017), first, video-recorded dialogue was qualitatively analyzed as multiple cases of dyadic conversation (Toerien, 2013;

Yin, 2017), then descriptives on coded data were explored. Grounded on the IDZframework, we conducted a template analysis (King, 2004), a type of thematic analysis on the coded data.

Table 4 Each student's contribution and teacher's feedback on collaboration in Study 1.

Student pairs	Contribution (%)	Teacher's feedback on collaborate (f)	Time to solve the problem (min:sec)
Sandy	60.6	5	5:30
Anthony	39.4		
Dorothy	59.2	1	3:53
David	40.8		
Melony	41.7	0	2:12
Rupert	58.3		
Clara	44.4	3	14:00
Luke	55.6		

For qualitative analysis, initially two researchers dual-coded 46% of the video data (along with verbatim transcripts) using the coding scheme (Ruvalcaba et al., 2016). Because the scheme was originally designed for 1C, challenges immediately emerged during the trial coding sessions with 2C data. For example, we lacked a category that captured students' explanations of what they were doing when working on their own screen. This resulted in several modifications so that it would align with the 2C setting. The modified scheme had three major categories: Question, Discussion, and Manipulation. Question and respond to questions are under the category Questions. Agreement, disagreement, explanation, suggestion, and antagonistic action are under the category Discussion. Manipulation had the sub-categories: edit with consent and edit without consent (Table 1). The modified scheme was tested through pilot (dual) coding 24% of the data and then revised again, resulting in the final coding scheme (Table 1). We used time-based video segmentation to code where we divided videos of each dyadic collaboration into ten-second intervals and coded each interval with conversations categories, where the categories are not mutually exclusive. We finally dual coded 21% of the video data (using different data was used at each step of coding) that had an average kappa of the interrater agreement was 0.847. All the subcategories had a kappa above or equal to 0.70 except for disagreement which had a kappa of 0.66 (Table 1). The codersresolved disparities through discussion, and one of the coders then coded the rest of data. We then reflected on the coded data through template analysis and found several themes which we then related to the characteristics of Mercer's types of talk.

For the quantitative analysis, the coded frequencies of sub-categories from the datawere used to corroborate the qualitative case studies and thematic analysis. Log data from NetsBlox activity that recorded the sequences of programming steps were also used to determine each partner's contribution to the code by calculating the total number ofmoves each partner had (Table 2). Both data sources were derived in parallel, then synthesized, to support findings from the discourse analysis (Tables 3 and 4).

Although there was not an explicit feedback intervention in this study, we felt it wasimportant to capture what feedback teachers did provide. Thus, one coder open-codedinstances of teacher feedback on collaboration (Table 4).

Results

We first provided quantitative descriptive findings, then a qualitative study of cases based on our interpretation of students' use of conversation categories during collaboration which subsequently followed by thematic findings.

Among all categories (Table 3), the most frequent sub-category was an explanation (42.2%); question (18.1%) and suggestion (18.1%) were used moderately; respond to question (9.6%), agreement (6%) and disagreement (4.8%) were minimally used, and antagonistic action was not identified. In this 2C setting, students edited without their partner's verbal consent almost 100% of the time.

Dorothy and David

Dorothy and David both actively searched for blocks that would help to solve the problem, which they did in about 4 minutes. David gave a lot of *suggestions* and Dorothy seemed to sincerely respond to the *suggestions* and *explanations* by acknowledging and implementing David's proposals. They also used *questioning*, and both were observed to *respond to each other's questions*. A seemingly concerned Dorothy said, "I found the 'when,' but I don't know" and David peeked at Dorothy's screen to respond and said, "Oh I think I know what it is, maybe it's the 'if' one" and then corrected his idea to "no, if else." This pair listened to each other's *explanations* closely by stopping their own work and looking at each-others screen while also responding very frequently. To confirm what they were supposed to do, Dorothy read the instruction "if touching sprite then stop." David took the hint and said, "oh, stop," but then *suggested*, "maybe we should take out this 'forever' then."

Although David did many of the editing tasks, the pair communicated through *suggestions* and *explanations* to solve the problem together. Both stayed on task and neither got any feedback from the teacher.

Clara and Luke

Clara initially struggled to log into NetsBlox. During that time (50 seconds), Luke edited the script and when he thought he was finished said, "I did it." The teacher then asked if he worked with his partner, to which Luke said no. At the teacher's prompting, Luke stopped editing and waited for Clara to get ready. It was three and a half minutes into the activity and there had been no conversation within the dyad until this point. Clara then found that her "snowflakes" sprite was not visible and although she asked Luke for help first; then within 2 s, she asked the teacher. After solving the problem, they scrolled through the block categories to search for blocks to solve the problem. Unlike David and Dorothy, this dyad did not explain to each other what they were editing and searching for. After several more minutes, Clara explained to Luke that she could not find the exact conditional block that would solve the problem. When she found the "if-else" block, she began to work without any explanation. Luke seemed to agree with her work by nodding and began to edit but did not explain what he was doing. At one point, the pair disagreed on the use of one block and, without coming to consensus, continued editing. Because the pair used little explanation with each other, they often got confused about each other's work. For example, when Clara questioned, "what are you doing?" Luke explained, "because it says forever if touching." Then, Luke's screen froze, and he exited from the collaborative space, though Clara continued working on her screen without talking. Meanwhile, Luke sat and watched another group work. After five minutes of sitting, Luke seemed to get agitated, shaking hands and rapidly moving on his chair. He reminded Clara to re-link the project with him saying, "Invite me in, invite me in." After several reminders, Clara noticed and reshared the project. After getting back into the project, Luke immediately started editing without explanation of what he was doing and finished the project.

Although the log data show that the pair almost equally contributed to the editing, their synchronous collaboration was interrupted because of a technical problem with Luke's screen as well as the lack of explanation from either (Table 4). As Table 3 shows, they are the only pair using most (all but antagonistic) of the conversation categories. The teacher's feedback was frequent for this dyad. The teacher encouraged them to communicate with each other three times during this activity and each time the dyad immediately engaged in conversations after the feedback (Table 4).

Anthony and Sandy

From the beginning, Anthony showed enthusiasm for participating in the activity. He said with a heightened voice, "I can't get in," to which Sandy calmly replied, "because I haven't yet invited you." After getting into the project, Anthony said "oh, I know this one, I can figure out how to do this one" and then *suggested*, "when sprite four touches sprite three, sprite three run." Anthony talked a great deal and at a faster pace while offering many *suggestions*; however, Sandy was unresponsive to Anthony's *suggestions*. Noticing the lack of communication, the teacher gave them feedback to divide work between themselves and talk saying "it's good to talk to each other, like, say I'll do this part and you'll do that part." Followed by the feedback, Sandy *explained* what she would do." The teacher tried to reconfirm "so what is he doing?" Sandy did not respond and looked at Anthony. The teacher asked how the pair were planning to work and suggested to work collaboratively. Here, both of them separately talked about certain aspects of the problem. After the teachers' feedback, the pair began talking about the task together. Anthony edited the "forever" block and Sandy disagreed with his edit by saying "I don't think that's 1000" to which Anthony wanted to play the program *suggesting*, "let's reactivate the program." When Sandy resumed editing again, Anthony said, "wait, stop for a sec so I can, I'm trying to retype this." Then Anthony edited the "glide" block and provided an *explanation* of what he was doing.

Unlike Dorothy and David who had more *explanation* (60% of the pair's conversion), Sandy and Anthony (36.4% of the pair's conversation) did not explain as often what they were doing unless prompted to do so by the teacher. The teacher gave feedback five times during their collaboration. They demonstrated all but three conversation categories (not *agreement, reminder* and *antagonistic*) during their collaboration (Table 3).

Melony and Rupert

Suggestions and *explanations* from Rupert were frequent during this collaboration. Although Rupert made *suggestions*, he did not seem to expect responses. Instead, he edited the blocks without waiting for Melony's confirmation. Rupert said, "so first we have to do the 'if'" and then edited the script. Again, he *suggested*, "so we can just do 'glide' then." At one point he *suggested* splitting up the work saying, "I'll do the top one and you can do the bottom," but edited both parts. When Melony began to edit two minutes into the activity, Rupert said, "I got it" and finished the work. Melony almost never talked or edited the code blocks during that time.

Rupert is the only student from this sample who participated in two sessions, thus he had prior experience solving a similar problem. Table 3 shows that, compared to other pairs, this pair used the lowest amount of *questioning*. Although there were many instances of *explanation* and *suggestion*, most were uttered by Rupert.

Themes found in dyadic collaboration

Analyzing the conversation categories, we could see that students were sometimes Exploratory by being constructively critical–providing productive challenge–about their collective work, and sometimes Cumulative by only confirming partners' work. Four themes emerged for the thematic template analysis on the coded videos. The first resonates with Exploratory talk where students critically question and discuss with each other; the following two are characterized by one student dominating and are less generative. The fourth pattern indicates that with teacher feedback, students could be encouraged to be mutually responsive.

Question and discuss while responding. We found this to be the most active collaboration mode. This is reflected in much of David and Dorothy's conversations. In this type of collaboration, partners respond to each other's explanations, suggestions, or questions with additional suggestions or explanations. While this type of questioning might be characterized as Exploratory talk, the coding scheme did not allow us to differentiate its definitive characteristics, as we did not have specific categories such as use of challenging questions, justifications, or sharing alternative ideas to be able to characterize it as Exploratory.

Edit without responding. In this type of collaboration, partners only use explanations and suggestions while working independently. We observed one partner explaining what they were doing or making suggestions without the presence or expectation of any response or one partner being passive by only listening or even ignoring their partner. We could see this in much of Melony and Rupert's collaboration, as Rupert took charge of the activity and Melony provided little input. As Rupert had previous experience in programming, Melony might have considered Rupert the authority in the dyad (Lewis & Shah, 2015). This type of conversation where one person is trying to keep an assertive relationship is closer to Cumulative talk in IDZ. However, in

Cumulative talk, collaborators do respond to each other through agreements to suggestions. Since the 2C setting allows partners to work simultaneously, dominating partners may assume they are collaborating through explanation and suggestion without having to confirm or negotiate with their partner.

Does not question, discuss, or respond. Sometimes there was little or no communication between partners. Even when a partner asked questions, explained or suggested, the other partner continued to edit without responding. Both Melony-Rupert and Clara-Luke dyads showed this characteristic in many instances. Although there was no Disputative talk like antagonistic actions or disagreements between partners, it was not productive, as partners did less knowledge sharing.

More conversation with teacher feedback. We found that when teachers reminded students to collaborate, students often improved their level of productive conversation, if only briefly. Asking questions, explaining their actions and making suggestions were common immediately after teacher feedback. During the total time the dyads took to solve the problem, Anthony-Sandy was reminded five times to collaborate, Luke-Clara three times, Dorothy-David once while Melony-Rupert were never reminded (Table 4). In Anthony and Sandy's collaboration, we observed how they became engaged in conversation every time the teacher reminded them. These findings parallel general findings in the literature that we need to explicitly teach students how to collaborate and model patterns of productive collaboration (Mercer et al., 2004, 1999; Rojas-Drummond & Zapata, 2004).

Discussion

In this first study, we wanted to investigate the collaborative talk students engaged in during 2C and, in doing so, refine our coding instrument.

Study 1 offered a preliminary validation of methods of analyzing 2C field data usingthe Ruvalcaba et al. (2016) coding scheme. The scheme, refined through trial coding sessions, helped us to parse students' conversation into a detailed analyzable format. However, there were two major challenges with the coding method we used. First, it was difficult to differentiate Cumulative and Exploratory talk, which restricted us from identifying productive conversation during collaboration. Thus, for Study 2, we needed to explore how to operationalize these two types of talk through the coding scheme. The second challenge was with the technique we used to segment the video data. Priorresearch using analyzing conversation has employed both time-based (such as the type we used in Study 1) (e.g., Baines et al., 2009; Ruvalcaba et al., 2016) and event-based videosegmentation to code (Schegloff, 1991). Although the 10-s interval-based coding provided a systematic method for quantifying the conversations, we lost important contextual and qualitative information. In Study 2, we switched to event-based coding, wherewe coded turns of talk by each partner – allowing us to code with higher specificity andgranularity in utterances.

The themes derived in Study 1 showed, as would be expected, that dyads varied in terms of their level and quality of discourse. Some dyads were not responding to ordiscussing work productively with their partners. This lack of task-related dialogue may be related to the 2C setting where two sets of computers lowers the need to negotiatechanges to the code. Thus, pairs may not be leveraging the potential benefits of collaborative work through verbally negotiating the problem-solving process. Previous research on students' computer-based collaboration using platforms that provided opportunities to divide and conquer tasks show that partners may vary in patterns of interaction, marked by individual rather than joint work (Barron, 2000). In our previousstudy comparing 1C and 2C, we found that students' collaboration can turn into cooperation by working separately on individual sections, thus restricting the opportunityto work collaboratively on coding challenges (Zakaria et al., 2019). Although cooperative work through a division of labor is useful in some contexts, a primary goal of pair programming in an educational context is the shared work on a common problem that can both further understanding through negotiated meaning (intersubjectivity), and theways in which ZPD can be leveraged through dyad members' unique abilities (Fawcett & Garton, 2005). Dyads who had difficulty with collaboration tended to increase their talkwhen the teacher reminded the dyad of the expectations for collaboration. However, teacher feedback was not consistent in this study as we did not purposefully implementa structured support system for students. The teacher merely reminded students to collaborate without providing explicit guidelines on how to collaborate, sometimes even asking students to divide work among themselves. Studies point to the problemof learning to collaborate versus learning through collaboration and suggests that teachers often may not understand the mechanisms of productive collaboration or do not recognize the necessity to instruct students on the norms of collaboration (Murphy & Henessey, 2001). To be effective, teacher feedback needs to specifically target learning strategies that cater to the task on hand (Chan & Lam, 2010; Hattie & Timperley, 2007;Zimmerman & Kitsantas, 2002).

Thus, for Study 2, we incorporated an instructor-driven support system for students to encourage the use of quality Exploratory talk through collaborative modes. The main goalof Study 2 was to explore how explicitly guiding students in productive collaboration andconsistently giving feedback on the quality of collaboration can impact the quality of their discourse.

Study 2

Guided by the insights and subsequent recommendations that emerged from Study 1, our goal was to examine if a structured feedback intervention could help students use more Exploratory talk in their conversation. Accordingly, in Study 2, we explored the following research question: *What role does feedback play in students' collaborative talk during 2C pair programming?* A secondary goal for this study was to continue to reviewand modify the coding scheme to identify Mercer's (2002) three types of talk moreaccurately. Thus, the focus is on how feedback does (or does not) assist with promptingstudents' collaborative problem solving and help counteract the tendencies to work individually without communication as seen in some dyads in Study 1.

Method

The study took place at a suburban elementary school in the Southeastern United States. The students were from four classrooms but taught by the same technology teacherfollowing the same curriculum. Study 2 made use of Scratch (Resnick et al., 2009), as the teacher and students were most familiar with this environment. This variation of 2C with Scratch had unlinked workspaces which did not allow for a synchronized workspace. Dyads were seated side-by-side (per the 2C setting) where they were instructed to create identical programs on each of their computers in separate workspaces. The teacher directed the task that required the pairs to collaboratively create a game wherein the user follows rhyming clues to click on certain sprites that move or speak in response.

Table 5 Expectations for collaboration presented to the students.

Control and intervention group	Additional terms for the Intervention group
1. This work needs to be happening collaboratively. 2. Each partner is responsible for the coding, problem-solving, and debugging. 3. Make sure you both agree on your plans before you make changes. 4. The partners talk through the problem the entire time.	1. Information is shared openly. 2. Listen to each-other and consider each-others suggestions. 3. Actively encourage your partner to contribute to the discussion. 4. Welcome challenges from your partner. 5. Challenge ideas with questions. 6. Justify your ideas to your partner. 7. Appreciate your partner's effort.

In this intervention study, we randomly selected two classrooms to constitute our feedback group while the other two were our control group. The participants were 62 (control = 30; intervention = 32) fourth grade students (9–10 years) who formed 31 dyads (formed by the teacher). In-depth analysis was conducted on 12 randomly selected participants forming 6 dyads, three from each group (10 girls and 2 boys) from all the pairs that had recordings of good video and audio clarity. Activity sessions were once per week for four weeks (average of 26 active minutes/week) and the dyads were the same across all the sessions. We used video cameras with synced headsets to capture audio and video of the dyads. Verbatim transcripts of pairs' verbal interactions were created.

Table 6 Feedback Structure.

Session	Serial	Feedback
1	First	Sticky note: Try challenging your partner's idea by asking questions. Verbal addition: Questions with a 'why' like "why do you want to do that" would help to challenge each other's ideas.
	Second	Sticky note: Put effort on sharing alternative ideas. Verbal addition: Sharing alternative ideas to your partner would help to create the best idea together.*
2	First	Sticky notes: Questions asking for explanations help you decide the best action. Verbal addition:*
	Second	Sticky note: Justify your idea with 'because'. Verbal addition: When you have an idea or you are answering a question try to provide a reason. Like "I think we should change the color of the text because it is too bright."
3	First	Sticky note: Check yourself how much are you challenging your partner. Verbal addition: Do you have a question to ask right now?*
	Second	Sticky note: Check yourself how much you are sharing alternative ideas. Verbal addition: Do any of you have a different idea to share now?*
4	First	Sticky note: Ask your partner for justification for an idea. Verbal addition: Can any of you ask for a justification right now?*
	Second	Sticky note: The more you practice coding with collaborative strategies, you get better at it. Verbal addition: What strategies make good collaboration? (If no response, told the list of the three strategies challenging with questions, sharing alternative ideas. justifying own ideas)

Note. *Provided contextual examples from what they were doing at that moment.

At the beginning of each session, we verbally presented to the whole class the expectations for collaboration to both the control and intervention groups (see Table5). In the feedback group, the terms also included information on what constitutes goodcollaboration. As the students programmed, we provided pre-structured feedback tostudents in the feedback group focusing on the vital characteristics of Exploratory talk(Mercer, 2002) (i.e., *challenging partners with questions, sharing alternative ideas, justifyingideas, or disagreeing with justification*). Informed by growth mindset theory (Dweck &Leggett, 1988), the importance of effort and use of appropriate strategies was incorporated into the feedback as well. The feedback–a different set statement for each session–was provided on a sticky note twice per session (10 minutes into the activity and then10 minutes after the first feedback) by the first author (Table 6), with students asked toread the feedback aloud from the note upon delivery. In addition, at the time the stickynote was provided additional verbal feedback was also given related to students' currentcollaboration practices. It took around 30 seconds to a minute to complete each feedback.In the last week of the intervention, all participants (n = 32) in the intervention group completed anonymous reflections on the feedback. They responded to three questions(*Was the feedback helpful? What was helpful? and What was not helpful?*) about the feedback on a printed questionnaire.

Data coding and analysis

Based on our pilot study and Study 1, we developed a new coding scheme (Table 7). Our revisions resulted in new categories reflecting Exploratory talk: *sharing alternative ideas*, asking *higher-order questions* (questions that challenge partner's ideas such as "why" questions), giving *justification;* three categories to Cumulative talk: *seeking help, coordination, controlled direction* and we considered conversation as Disputational talk when *disagreements* occurred but that did not include any *justifications*. We added *Disagreement with justification* to Exploratory talk. We removed two categories *editing with consent* and *editing without consent*, as they never got used in Study 1. We believed these added categories would help us to differentiate Cumulative and Exploratory talk much more explicitly, and thus better be able to identify productive conversation during collaboration. Each category was placed into one of Mercer's (2002) three types of talk (Table 7). In terms of feedback identification, as the feedback statements were pre-structured and time-specific, we decided not to code them.

Two pairs of coders, dual coded 47% of the data using the final coding scheme (Table 7). We coded each partner's turns of talk in a conversation (Schegloff, 1991). The categories were not mutually exclusive; thus, *kappa* was calculated for each category separately. Also, kappa was computed for each coder pair then averaged to provide a single index of interrater reliability (Light, 1971). The resulting average *kappa* indicated substantial agreement (*kappa*: average = 0.795, min = .749, max = .857) (Landis & Koch, 1977) (see Table 7). Due to differences in session lengths, we normalized the data by calculating the total of each coded category per minute as a proportion of overall talk for that dyad. This normalized score of each category from video data was then used to explore significant differences in collaborative talk between the groups.

Table 7 Conversation categories in the coding scheme and the inter-rater agreement (kappa) for Study 2.

Type of Talk	Categories	Description	Example utterances
Cumulative Talk	*Agreement*	Show explicit agreement with partner's idea, suggestion, or change. *k*=.8	"Yeah, good idea", "Okay".
	Self-Explanation	Starts explaining themselves what they are doing or thinking while working on own platform. Like a think aloud. *k*=.716	"Okay, so move 10 steps.", "I've to do the sailboat"
	Suggestion	Student makes a suggestion or shares an idea for the next step. Also, reminders would fall under suggestion. *k*=.767	"You have to change the number to 80." Words like 'we/you/I can', 'we/you need', 'let's', 'we/you have to'
	Controlled Direction	Gives step-by-step direction to their partner. *k*=.799	"Glide 2 sec, then go back to the sailboat",
	Seeking help	Student directly or indirectly seeks help from partner. *k*=.734	"I'm confused", "What shall we do?"
	Simple question	Question about a process or fact. (any type of questions which are not higher order) *k*=.903	"That?", "Which one?" "What are you doing?,"
	Coordination	Monitoring task or group progress, Coordination of task or group process. *k*=.822	"You're always one step ahead," "Wait for me please," "Marshall, focus!"
Exploratory talk	*Higher-order question*	Student questions to challenge their partner's ideas. These questions should be asking for reasoning. *k*=.811	"Why did you move it?", "Why?", "What happens if you keep it that way?"
	Alternative idea	Share an idea as an alternative to what the partners was suggesting or editing. *k*=.695	"No, Let's change it to blue", "May be try three seconds"
	Justification	Students justify their idea with reasons. Or evaluates a step as reasoning. May include words like 'because'/ 'cause'. *k*=.773	"Five seconds is too long", "It has to be the glide block cause it needs to move"

Results

To answer the research question, we utilized quantitative methods to directly compare the level of discourse from the dyads' collaboration between conditions. Figure 2 presents the overall distribution of the mean rate of conversation categories for each group. *Self-explanation* was most common throughout students' collaboration for both groups. *Simple question* (any questions not challenging the partner) was the next highest category. *Coordination, suggestion*, and *seeking help* were moderately used. However, categories that characterize Exploratory talk–*justification, sharing alternative idea*, and *higher-order question*–were minimally used by all students.

Each data point represents the mean rate of each conversation category used by dyads each week. In terms of growth in conversation rate throughout the study, a repeated measures ANOVA on four data points (one per week) per dyad (24 total across the sample) showed no significant difference in mean rate of conversation categories between feedback and control groups. Figure 3 shows progressions of estimated mean rate of Exploratory categories and *simple questions* from week one to four indicating there is growth in some of these categories in the first two or three weeks but then a decrease in the last week. We then followed up with a MANOVA test to find if there were significant mean differences between the groups regardless of time. Because of the small sample of data used, we also conducted tests to check if the assumptions of MANOVA were met. We found Cronbach's *alpha* of .781 indicating that the categories were sufficiently correlated to be considered as dependent variables in MANOVA. As the sample sizes were equal for both groups, we assumed the homogeneity of covariance was met. Shapiro-Wilk statistics for the dependent variables (conversation categories) shows that except for *off-task, controlled direction* and *self-explanation* categories, all other conversation categorieswere normally distributed (Table 8).

Table 8 Shapiro-Wilk test of normality distribution for conversation categories.

Conversation categories	Shapiro-Wilk Statistic	
	Intervention	Control
Alternative idea	.940	.885
Justification	.888	.936
Higher-order question	.935	.901
Disagreement with justification	.964	.911
Disagreement without justification	.953	.879
Agreement	.866	.939
Controlled direction	.972	.872**
Suggestion	.968	.947
Self-Explanation	.898**	.916
Seeking help	.916	.947
Simple question	.974	.948
Coordination	.926	.898
Off-task	.511***	.559***
Other	.920	.903

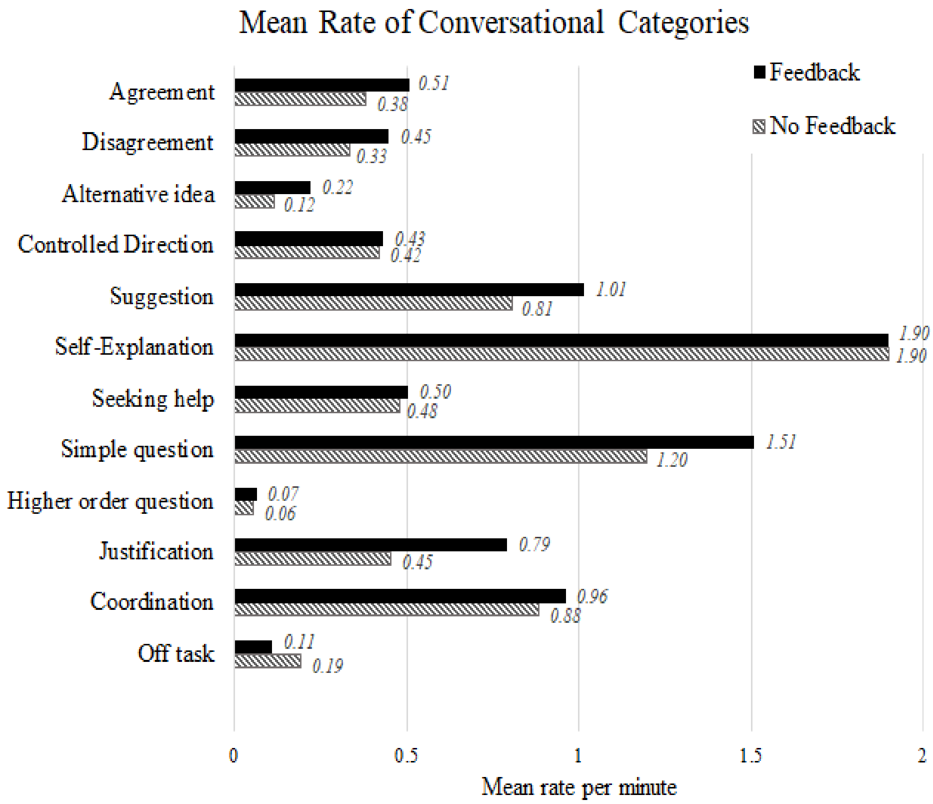


Figure 2: Mean rate of conversational categories per minute in both feedback intervention and control groups.

Research shows that when the assumption of normality is untenable, but the homogeneity of covariance is met, the parametric statistic is more robust, and even slightly outperforms the nonparametric statistic in terms of Type I error rate (Finch, 2005). Thus, we conducted a MANOVA that showed a significant difference in the use of conversation categories (all the categories as DVs) between the groups (IV), $F(14, 9) = 9.090$, $p = 0.001$; Wilk's $\Lambda = .066$, $\eta_p^2 = .934$. Because MANOVA statistics did not show which conversation categories were significantly different and to what level, we explored each category separately with Kruskal-Wallis H tests, a non-parametric alternative of univariate ANOVA (Table 9). Among the exploratory categories, justification, $\chi2(1) = 13.026$, $p = 0.00$ and sharing alternative idea, $\chi2(1) = 3.992$, $p = 0.046$ were significantly higher in the feedback group.

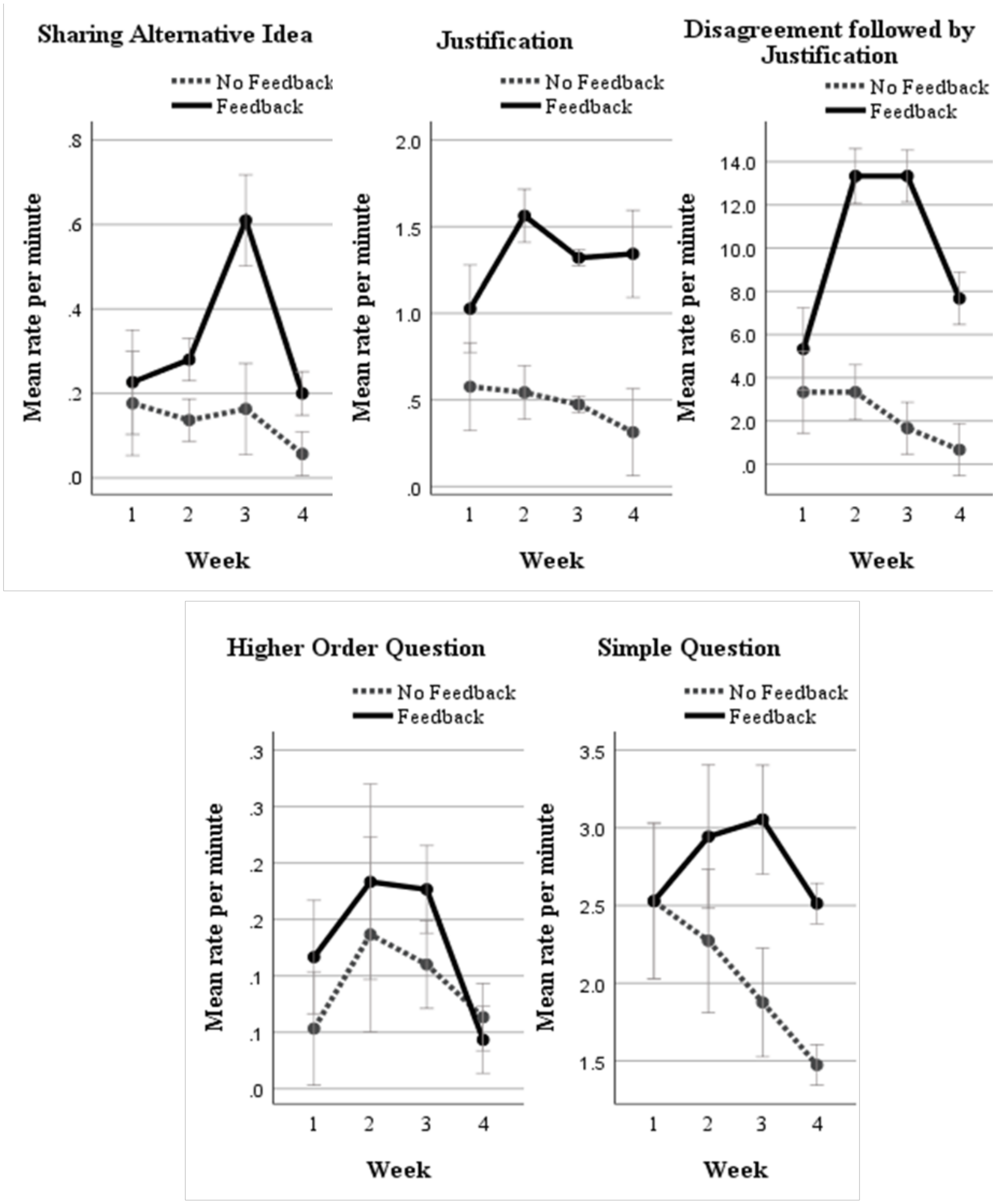


Figure 3: Mean rate of categories reflecting Exploratory talk and simple question.

However, despite that a substantial part of the feedback and pre-activity instruction emphasized asking challenging questions (i.e.“why” questions seeking justification) to the partner, students in both groups used minimal *higher-order questions* and did not show any significant differences.

In contrast, the use of *simple questions* was significantly higher in the feedback group, $\chi2(1) = 5.749, p = 0.017$.

Because we were interested in the impact of instructor-directed feedback on student student conversation, immediate uptake by students was examined. We found 26% of the time they used a *higher-order question* within two minutes of receiving feedback aboutasking challenging questions. For example, right after feedback ("Try challenging your partner's idea by asking questions."), Dyad F1 engaged in the following conversation:

A: *I have a question for you. Why do you want to do that color?*

B: *That would look really weird because it would blend in.*

A: *No, it won't. Because this is dark enough. See?*

To similar feedback, Dyad F2 responded:

C: *I think we should put "when the sprite clicked" high*. (paused) *Say, "why?"*.

D: *Why?*

C: *Because and then when somebody finds it, then it would disappear.*

We analyzed another sequence of categories that reflects a desired category of Exploratory talk: *disagreement* followed by *justification*. Example from Dyad F3:

E: *Polar bear (suggestion).*

F: *No. (Disagreement) Because it will run off (justification).*

Students in the feedback group had a significantly higher rate of *disagreement* followed by *justification*, $\chi2(1) = 15.460$, $p = 0.00$, than the control group (Table 9).

Table 9 Test-statistics for univariate ANOVA main effects of Feedback intervention.

Main effect (between)	Kruskal Wallis H	*df*	Mean Rank	
			Intervention	Control
Alternative idea	3.992*	1	15.38	9.63
Justification	13.026***	1	17.71	7.29
Higher-order question	.660	1	13.67	11.33
Disagreement with justification	15.460***	1	18.17	6.83
Disagreement without justification	1.022	1	13.96	11.04
Agreement	8.344***	1	16.57	8.33
Controlled Direction	1.842	1	14.46	10.54
Suggestion	10.839***	1	17.25	7.75
Self-explanation	.563	1	13.58	11.52
Seeking help	1.470	1	14.25	10.75
Simple question	5.743**	1	15.96	9.04
Coordination	2.430	1	14.75	10.25
Off-task	1.847	1	10.54	14.46
Other	.008	1	12.63	12.38

**** The mean difference is significant at level .01*
***The mean difference is significant at level .05*

Student reflection on feedback

Two pairs of coders open-coded students' reflections on feedback where we found that 90.3% of students reported that feedback was helpful. Investigating the strengths and limitations of the feedback, there were three themes related to strengths. First, most students' found that feedback helped collaboration by reminding them to ask questions,communicate with their partner, and give partner feedback. According to a student (S14), "It was very helpful. I was talking to my partner way more." Second, they felt the feedbackhelped their motivation and task progression by mentioning that it was "fun," (S21) "encouraging," (S7) and helped them work "faster" (S19) and "stay focused" (S6). Third,a few students found that the feedback enhanced their code practices stating that it "improved work" (S29) and that it "was helpful because if some of the coding was wrong we could help each other to make it right." (S12).

We also found three themes related to limitations. First, some students felt that thefeedback lacked the necessary detail, mentioning that there were "not enough details and examples"(S5) and "some of the feedback on the sticky notes did not make sense" (S13).Second, two of the students found the feedback to be a distraction, with one noting "thatit would stop us from what we were doing and make us lose our train of thought" (S27).Third, four of the students felt that they already knew the main strategies for collaborationsaying, "We already know the target points" (S3). Also, of note, another student mentioned "My partner did not help me. So, no [the feedback was not helpful]!" (S21) servingas evidence that whoever is providing the feedback needs to remind both members of the pair to collaborate.

Discussion

In Study 2, we examined if structured feedback on dyadic collaboration along with explicit instructions on the expected characteristics of collaboration can impact students' level of collaborative conversation in this 2C setting. The findings suggested that, for this sample, the conversation rate was significantly higher in dyads who were given feedback and instruction. This supports findings in other research that showed feedback on the use ofcertain strategies and effort improved performance (Dweck & Leggett, 1988), and helped students to incorporate those strategies into their work (Binglan & Jia, 2010; Burnett, 2002).

Perhaps most importantly, nearly all of the Exploratory categories appeared at significantly higher levels in the intervention group. In particular, students who received feedback had higher use of *justification* for the new and *alternative ideas* they shared as well as the *disagreements* that they had. While this might be expected as students weregiven explicit feedback on their use of *justification*, it is encouraging to see the level ofuptake. Since programming problems can have many solutions, it is important for students to explore potential solutions and justify why they think it could be correct. Such critical reasoning processes may help them explore and make rationalized decisions (Besnard & Hunter, 2008). This process is found to be beneficial both for students' learning outcomes (Wegerif & Dawes, 1998) and their problem-solving capacity (Rojas-Drummond& Zapata, 2004).

The use of *higher-order questions*, which would indicate if they were asking challenging questions to each other, was not found to have any difference between the groups.However, use of *simple questions* – questions not challenging the partner, but rather asking about simple information – was found to be used significantly more by the experimental group. Reflecting back on the structure of the feedback system, we believe that our prompt related to using *higher-order questions* may have been unclear to students and they may have merely perceived it as a cue to ask any type of question. Previous literature suggests that feedback can easily be misinterpreted by students due to inadequacy of the feedback or other factors (i.e. consistency, accuracy, or comprehensibility of the feedback) not investigated in this study (Lee, 2008). In further research, weneed to carefully word the feedback regarding questioning as well as demonstrate howchallenging questions would be utilized in this context. Considering how challenging it isfor students at this age to participate in argumentative collaboration (Bell, 2004), it is also plausible that this more cognitively challenging discursive move requires longer and consistent training with the teacher and their students to be utilized effectively.

Discussing with students the feedback prompts they received is likely to provide useful data for future refinement of our feedback strategy (cf., Hedgcock & Lefkowitz, 1994).While students were, overall, positive about the feedback, a few of them felt that the feedback was distracting and/or redundant. These thoughts shared by the students spurred us to reflect on the structure of the feedback, which confirmed that perhaps some of the feedback content could be perceived as repetitive. Although we specificallywanted to reiterate the strategies as a reminder to students, we now believe it may have been more beneficial to make the feedback personalized and adaptive by incorporatingexamples grounded in their specific activity, thus helping students better understand what we were striving for.

In both Study 1 and 2, we found the use of *self-explanation* (*explanation* in Study 1) to be significantly higher than the other categories. The high rate of *self-explanation* was not surprising given the opportunity in these 2C pair-programming settings to work concurrently (Zakaria et al., 2019), thus heightening the need to coordinate among themselves by explaining what they were working on and thinking. While considered Cumulative talk, we see this as a positive quality as more *self-explanation* often leads togreater understanding (Chi et al., 1994) as well as more communication between partners to monitor their progress.

Reflecting on the revised coding scheme in Study 2, adding new categories helped usto classify Exploratory talk in more detail. Although we needed to sub-code categories like *disagreement* with *justification* when *disagreement* code was immediately followed by*justification* code and *disagreement* without *justification*, it was important to keep primarycodes *disagreement* and *justification* separate.

Mercer’s (2002) framework emphasizesstudents’ ability to disagree with appropriate reasoning as an important characteristic of Exploratory talk whereas only disagreeing without any justifying elaboration is Disputational and considered unproductive. Thus, this differentiation in the coding helped us explore these characteristics in detail. Additionally, analysis segmented by turnof talk in Study 2 helped to get a more accurate count of the number of instances for each category used and to better contextualize the discourse.

Limitations

As with all qualitative analysis, there are risks of interpretation bias, which we diligently tried to minimize through multi-coder corroboration and by striving for a high level of objectivity in our interpretations. The analytic-intensive nature of mixed-methods studies that involve verbatim transcripts and multi-coder concurrence, limited the total number of students we were able to study. We consider such a small sample size as a limitation of Study 2; however, this sample size is common for this type of annotation-intense video analysis work (e.g., Deitrick et al., 2017; Lewis & Shah, 2015). Still, this analysis of the intervention study provided enough statistical power to guide the development of future intervention studies utilizing 2C pair programming. Additionally, we chose to randomly select dyads from the data that had good quality audio and video rather than developing a purposeful or stratified sample based on factors such as competence level or gender. Our aim was to explore collaborative practices and the impact of feedback on students more broadly in this initial set of studies. However, in our future studies, we intend to pursue research questions that might benefit from such a sampling method.

Future studies should strive to enlarge both the number and diversity of students studied. Another limitation is that we did not measure the existing collaboration norms of the classrooms. Although there was a single teacher of record for the students in each study, these classes all had different primary teachers that they spent most of the school day with. Both studies took place at suburban elementary schools, with Study 1 in an academically gifted classroom while Study 2 was with classrooms of students with weekly exposure to technology/coding experiences. Thus, our populations may not be representative of a larger body of students of this age. Finally, there can also be a possible Hawthorne Effect for the feedback; students with microphone headsets on, cameras in their space, and extra lessons on how to appropriately talk to each other might talk differently as a result of the attention and not the feedback themselves.

Conclusion

Our studies support the notion that we cannot assume that students inherentlywill know how to engage in productive collaboration practices and discourse. These two investigations demonstrate that before we can begin to design effective student scaffolds, it is imperative to characterize what collaborative discourse looks like in a relatively new and underexplored context, 2C pair programming with elementary students. Although Mercer's (2002) framework provided an excellent foundation based in constructivist theory, iterative development was needed to create a coding scheme that was contextually responsive enough to provide insight into elementary students' collaborative programming practices. Appropriately, this was achieved by integrating discursive moves specific to programming (Ruvalcaba et al., 2016) with Mercer's overarching framing. Our work demonstrates that before one is able to design thoughtful interventions and support, pilot studies are essential in developing effective methodological frameworks that capture the nuances of collaborative practices and discourse. Results from Study 1 helped gain important insights into students' natural tendencies in collaborative discourse in 2C learning environments that were critical for developing a more refined and targeted study design. Study 1 showed discourse was highly varied by dyad and characterized mostly by one student explaining what they were doing as they took the libertyto edit their program without engaging in conversation with their partner. We also learned that teacher feedback has a possibility to influence the quality of students' conversation. We used these findings to implement an intervention study that utilized a feedback framework that was specifically designed to engage students in higher levels of collaboration as exemplified in Mercer's Exploratory talk. Findings showed the intervention did influence the small sample of dyads' discourse; however, there are still refinements that need to be made to our feedback system, such as modeling higher-order questioning.These two studies also helped us develop an enhanced analytic approach through our revised coding rubric that captured the more complex patterns and nuances of students' collaborative practices in 2C environments. We encourage others to utilize or adapt our framework, as more studies in thisarea are needed with larger and more diverse populations. We hope this workhas begun to shed light and start a fruitful conversation on the promise ofsupporting elementary students who are engaging in collaborative practices in pair-programming contexts so that they can maximize the benefits afforded through these experiences.

References

Adams, D., & Hamm, M. (1998). *Collaborative inquiry in science, math, and technology*. Heinemann,361 Hanover Street, Portsmouth, NH 03801-3912.

Alpay, E., & Ireson, J. (2006). Self-theories of intelligence of engineering students. *European Journal of Engineering Education*, *31*(2), 169–180. https://doi.org/10.1080/03043790600567027

Bailey, J. (2017). Open Broadcaster Software. *Software, OBS Project*, *62*. https://obsproject.com/ 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. https://doi.org/10.1080/03057640802701960

Bandukda, M., & Nasir, Z. (2010, June). Efficacy of distributed pair programming. In *2010 International Conference on Information and Emerging Technologies* (pp. 1–6). IEEE.

Barron, B. (2000). Achieving coordination in collaborative problem-solving groups. *The Journal of the Learning Sciences*, *9*(4), 403–436. https://doi.org/10.1207/S15327809JLS0904_2

Beauchamp, M. H., & Anderson, V. (2010). SOCIAL: An Integrative Framework for the Development of Social Skills. *Psychological Bulletin*, *136*(1), 39–64. https://doi.org/10.1037/a0017768

Bell, P. (2004). Promoting students' argument construction and collaborative debate in the science classroom. *Internet environments for science education, 3,* 115–143.

Bennett, N., & Cass, A. (1989). The effects of group composition on group interactive processes and pupil understanding. *British Educational Research Journal*, *15*(1), 19–32. https://doi.org/10.1080/0141192890150102

Besnard, P., & Hunter, A. (2008). *Elements of argumentation* (Vol. 47). MIT press.

Binglan, Z., & Jia, C. (2010). The Impact of Teacher Feedback on the Long-term Improvement in the Accuracy of EFL Student Writing. *Chinese Journal of Applied Linguistics (Foreign Language Teaching & Research Press)*, *33*(2), 18–34. http://www.celea.org.cn/teic/90/10060802.pdf

Bradbury, A., Wiebe, E., Vandenberg, J., Tsan, J., Lynch, C., & Boyer, K. (2019). The Interface Design ofa Collaborative Computer Science Learning Environment for Elementary Aged Students. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vols. 63, No. 1, pp.493–497). SAGE Publications.

Broll, B., Lédeczi, A., Volgyesi, P., Sallai, J., Maroti, M., Carrillo, A., Weeden-Wright, S.L., Vanags, C., Swartz, J.D., & Lu, M. (2017, March). A visual programming environment for learning distributed programming. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 81–86).ACM

Bruner, J. S. (1978). The role of dialogue in language acquisition. In *A. Sinclair, R. J. Jarvelle, & W. J. M. Levelt (Eds.), The child's conception of language* (pp. 241–256). Springer-Verlag.

Burnett, P. C. (2002). Teacher praise and feedback and students' perceptions of the classroom environment. *Educational Psychology*, *22*(1), 5–16. https://doi.org/10.1080/ 01443410120101215

Calder, N. (2010). Using scratch: An integrated problem-solving approach to mathematical thinking. *Australian Primary Mathematics Classroom*, *15*(4), 9–14. ISSN: ISSN-1326-0286.

Campe, S. et al. (2020). "Pair programming in middle school: variations in interactions and beha- viors." Computer Science Education 30(1), 22–46.

Chan, J. C., & Lam, S. F. (2010). Effects of different evaluative feedback on students' self-efficacy in learning. *Instructional Science*, *38*(1), 37–58. https://doi.org/10.1007/s11251-008-9077-2

Chi, M. T., De Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439–477. https://doi.org/10.1016/0364-0213(94)90016-7

Cockburn, A. (2004). Crystal clear *A human-powered methodology for small teams*. Pearson

Education.

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

Cutts, Q., Cutts, E., Draper, S., O'Donnell, P., & Saffrey, P. (2010, March). Manipulating mindset to positively influence introductory programming performance. In *Proceedings of the 41st ACM technical symposium on Computer science education* (pp. 431–435). ACM.

De Lisi, R., & Golbeck, S. L. (1999). Implications of Piagetian theory for peer learning. In *A. M. O'Donnell and A. King (Eds.), The Rutgers Invitational Symposium On Education Series, Cognitive perspectives on peer learning* (pp. 3–37). Lawrence Erlbaum Associates Publishers.

Deiglmayr, A., & Spada, H. (2010). Developing adaptive collaboration support: The example of an effective training for collaborative inferences. *Educational Psychology Review*, *22*(1), 103–113. https://doi.org/10.1007/s10648-010-9119-6

Deitrick, E., Wilkerson, M. H., & Simoneau, E. (2017, August). Understanding student collaboration in interdisciplinary computing activities. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 118–126). ACM.

Dewan, P., Agarwal, P., Shroff, G., & Hegde, R. (2009). Distributed side-by-side programming. In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*. IEEE.

Dweck, C. S. (2013). *Self-theories: Their role in motivation, personality, and development*. Psychology press.

Dweck, C. S., & Leggett, E. L. (1988). A social-cognitive approach to motivation and personality. *Psychological Review*, *95*(2), 256. https://doi.org/10.1037/0033-295X.95.2.256

Fawcett, L. M., & Garton, A. F. (2005). The effect of peer collaboration on children's problem-solving ability. *British Journal of Educational Psychology*, *75*(2), 157–169.

Fernández, M., Wegerif, R., Mercer, N., & Rojas-Drummond, S. (2015). Re-conceptualizing "scaffold- ing" and the zone of proximal development in the context of symmetrical collaborative learning. *Journal of Classroom Interaction*, *36/37*(1/2), 54–72. https://www.jstor.org/stable/44735711

Finch, H. (2005). Comparison of the performance of nonparametric and parametric MANOVA test statistics when assumptions are violated. *Methodology*, *1*(1), 27–38. https://doi.org/10.1027/1614-1881.1.1.27

Franklin, D., Hill, C., Dwyer, H., Iveland, A., Killian, A., & Harlow, D. (2015, February). Getting started in teaching and researching computer science in the elementary classroom. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 552–557). ACM.

Gweon, G., Rose, C., Carey, R., & Zaiss, Z. (2006, April). Providing support for adaptive scripting in anon-line collaborative learning environment. In *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 251–260). Montréal Québec, Canada.

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in educa- tion: A literature review. *Computer Science Education*, *21*(2), 135–173. https://doi.org/10.1080/08993408.2011.579808

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81–112. https://doi.org/10.3102/003465430298487

Hedgcock, J., & Lefkowitz, N. (1994). Feedback on feedback: Assessing learner receptivity to teacher response in L2 composing. *Journal of Second Language Writing*, *3*(2), 141–163. https://doi.org/10.1016/1060-3743(94)90012-4

Infante, C., Hidalgo, P., Nussbaum, M., Alarcón, R., & Gottlieb, A. (2009). Multiple Mice based collaborative one-to-one learning. *Computers & Education, 53*(2), 393-40.

Kavitha, R. K., & Ahmed, M. I. (2015). Knowledge sharing through pair programming in learning environments: An empirical study. *Education and Information Technologies*, *20*(2), 319–333. https://doi.org/10.1007/s10639-013-9285-5

King, N. (2004). Using Templates in the Thematic Analysis of Text. In C. Cassell & G. Symon (Eds.), *Essential guide to qualitative methods in organizational research* (pp. 256–270). Sage.

Lai, A. F., & Yang, S. M. (2011). The learning effect of visualized programming learning on 6 th graders' problem solving and logical reasoning abilities. In *2011 International Conference on Electrical and Control Engineering* (pp. 6940–6944). IEEE.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 159–174

Lee, I. (2008). Student reactions to teacher feedback in two Hong Kong secondary classrooms. *Journal of Second Language Writing*, *17*(3), 144–164. https://doi.org/10.1016/j.jslw.2007.12.001 Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students? *Computer Science Education*, *21*(2), 105–134.

Lewis, C. M., & Shah, N. (2015). How equity and inequity can emerge in pair programming. In *Proceedings of the eleventh annual International Conference on International Computing Education Research* (pp. 41–50). ACM.

Light, R. J. (1971). Measures of response agreement for qualitative data: Some generalizations and alternatives. *Psychological bulletin*, *76*(5), 365.

Meier, A., Spada, H., & Rummel, N. (2007). A rating scheme for assessing the quality of computer-supported collaboration processes. *International Journal of Computer-Supported Collaborative Learning*, *2*(1), 63–86. https://doi.org/10.1007/s11412-006-9005-x

Mercer, N. (1995). *The guided construction of knowledge: Talk amongst teachers and learners*. Multilingual matters.

Mercer, N. (2002). *Words and minds: How we use language to think together*. Routledge.

Mercer, N., Dawes, L., Wegerif, R., & Sams, C. (2004). Reasoning as a scientist: Ways of helping children to use language to learn science. *British Educational Research Journal*, *30*(3), 359–377.https://doi.org/10.1080/01411920410001689689

Mercer, N., Wegerif, R., & Dawes, L. (1999). Children's talk and the development of reasoning in the classroom. *British Educational Research Journal*, *25*(1), 95–111. https://doi.org/10.1080/0141192990250107

Murphy, P., & Hennessy, S. (2001). Realising the potential – and lost opportunities – for peer collaboration in D&T setting. *International Journal of Technology and Design Education*, *11*(3),203–237.

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

Nawrocki, J. R., Jasiński, M., Olek, Ł., & Lange, B. (2005). Pair programming vs. side-by-side programming. In *European Conference on Software Process Improvement* (pp. 28–38). Springer. Pifarré, M., Cobos, R., & Argelagós, E. (2014). Incidence of group awareness

information on students' collaborative learning processes. *Journal of Computer Assisted Learning*, *30*(4), 300–317. https://
doi.org/10.1111/jcal.12043

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. B. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

Rojas-Drummond, S., & Mercer, N. (2003). Scaffolding the development of effective collaborationand learning. *International Journal of Educational Research*, *39*(1–2), 99–111. https://doi.org/10.1016/S0883-0355(03)00075-2

Rojas-Drummond, S., & Zapata, M. P. (2004). Exploratory talk, argumentation and reasoning in Mexican primary school children. *Language and Education*, *18*(6), 539–557. https://doi.org/10.1080/09500780408666900

Ruvalcaba, O., Werner, L., & Denner, J. (2016). Observations of pair programming: Variations in Collaboration Across Demographic Groups. In *Proceedings of the 47th ACM Technical Symposiumon Computer Science Education (SIGCSE '16)* (pp. 90–95).

Schegloff, E. A. (1991). Conversation analysis and socially shared cognition. *Perspectives on Socially Shared Cognition*, 150–171. https://doi.org/10.1037/10096-007

Scott, S. D., Mandryk, R. L., & Inkpen, K. M. (2003). Understanding children's collaborative interactionsin 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*. International Society of the Learning Sciences.

Shah, N., & Lewis, C. M. (2019). Amplifying and attenuating inequity in collaborative learning: Toward an analytical framework. *Cognition and Instruction*, *37*(4), 423–452. https://doi.org/10.1080/07370008.2019.1631825

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education (IJAIED)*, *12* (1), 40–62. https://iaied.org/journal/980

Toerien, M. (2013). Conversations and conversation analysis. In *Uwe Flick The SAGE handbook of qualitative data analysis* (pp. 327–340).

Tsan, J., Rodríguez, F. J., Boyer, K. E., & Lynch, C. (2018, February). I Think We Should . . . : Analyzing Elementary Students' Collaborative Processes for Giving and Taking Suggestions. In *Proceedingsof the 49th ACM Technical Symposium on Computer Science Education* (pp. 622–627). ACM.

Tudge, J. R. (1992). Processes and consequences of peer collaboration: A Vygotskian analysis. *Child Development*, *63*(6), 1364–1379. https://doi.org/10.2307/1131562

Umapathy, K., & Ritzhaupt, A. D. (2017). A meta-analysis of pair programming in computer programming courses: Implications for educational practice. *ACM Transactions on Computing Education(TOCE)*, *17*(4), 16. https://doi.org/10.1145/2996201

Vygotsky, L. (1978). Interaction between learning and development. In *Readings on the Development of Children* (pp. 34–41).

Warwick, P., Mercer, N., & Kershner, R. (2013). 'Wait, let's just think about this': Using the interactive whiteboard and talk rules to scaffold learning for co-regulation in collaborative science activities. *Learning, Culture and Social Interaction*, *2*(1), 42–51. https://doi.org/10.1016/j.lcsi.2012.12.004

Wegerif, R., & Dawes, L. (1998). Encouraging Exploratory Talk Around Computers. In M. Monteith (Ed.), *IT for Learning Enhancement* (pp. 10–29). Intellect Press.

Williams, L., Kessler, R. R., Cunningham, W., & Jeffries, R. (2000). Strengthening the case for pair programming. *IEEE Software*, *17*(4), 19–25. https://doi.org/10.1109/52.854064

Williams, L., Wiebe, E., Yang, K., Ferzli, M., & Miller, C. (2002). In support of pair programming in the introductory computer science course. *Computer Science Education*, *12*(3), 197–212. https://doi.org/10.1076/csed.12.3.197.8618

Yin, R. K. (2017). *Case study research and applications: Design and methods*. Sage publications. Zakaria, Z., Boulden, D., Vandenberg, J., Tsan, J., Lynch, C., Wiebe, E., & Boyer, K. E. (2019). Collaborative Talk Across Two Pair-Programming Configurations. In *Proceedings of 13th International Conference on Computer Supported Collaborative Learning (CSCL) 2019* (vol. 1, pp. 224-231). Lyon, France: International Society of the Learning Sciences.

Zimmerman, B. J., & Kitsantas, A. (2002). Acquiring writing revision and self-regulatory skill through observation and emulation. *Journal of Educational Psychology*, *94*(4), 660. https://doi.org/10.1037/0022-0663.94.4.660

Zumbach, J., Reimann, P., & Koch, S. C. (2006). Monitoring students' collaboration in computer-mediated collaborative problem-solving: Applied feedback approaches. *Journal of EducationalComputing Research*, *35*(4), 399–424.

CHAPTER 5: CONCLUSION

This research has explored the role of gender and motivational beliefs on students' collaborative and programming performances in pair programming followed by designing and testing a feedback intervention to support collaboration during pair programming. We conducted three studies reported here addressing three research objectives. First, we explored how gender, prior experiences and motivational factors (mindset, interest and self-efficacy) influence individual students' programming related conceptual understanding. The second study qualitatively contrasted the collaboration practices and programming learning of three gender groupings (girl-girl, girl-boy, boy-boy); For the third study, we designed and tested an intervention that provided feedback on productive collaboration aiming to help students better collaborate during pair programming regardless of their gender and prior experiences. Each study is represented by a chapter and in the following section we provide a brief summary of each chapter.

Summary

Chapter 2

In this study, we investigated how the exogenous variables gender and prior experience influence three endogenous motivational factors (mindset, self-efficacy, and interest in programming) affecting students' CS conceptual understanding. In doing so, we also validated an instrument to measure students' self-perceived ability beliefs and interest in CS using data from 210 fourth and fifth grade students. First, we validated the instrument using a combination of classical test theory and item response theory focusing on construct validity, item difficulty, and reliability of the items. Then, we tested a hypothesized path model and compared that to a revised model to identify the relationship between gender, prior experiences, mindset, interest, and self-efficacy with CS conceptual understanding.

The study findings highlight the value of prior knowledge in CS even at an elementary level and how its effects are stronger than gender-based differences. In terms of motivational beliefs, Self-efficacy was found to be the strongest predictor of CS performance whereas, Growth Mindset and Interest only mediates its effects.

The findings of this study have substantive and methodological implications for those exploring issues relevant to students' individual differences in motivational beliefs, gender, and prior experience in the field of elementary level CS education.

Chapter 3

Identifying that gender has a moderate impact on motivational beliefs related to programming, in this chapter we explored if gender-grouping would show different patterns of collaborative programming. We compared three cases of gender based pairing: girl-girl, girl-boy and boy-boy dyads' collaboration and programming performance. In doing so, we have utilized data on students' dialogue and their game artifacts to measure the programming functionality of games programmed by students.

We used a multiple case-study approach to stage our analysis around the cases with six dyads from an elementary classroom. Both the qualitative and quantitative content analysis is embedded into these cases for in-depth and holistic comparison of the collaboration.

Our findings are in accordance with the idea that there are nuanced differences in characteristics of collaborative programming between boys and girls in pair programming. Mixed gender dyads showed the least harmonious collaboration. Compared to the same-gender dyads they conversed less and mostly had unproductive conflicts. Same gender dyads were more collaborative, while girl-girl dyads showed more Exploratory talk as well as more synced in programming.

These findings point to both CS-specific and perhaps larger cultural forces shaping gender interactions with collaborative work.The findings also hold implications for how teachers pair students, design differentiated instructions to provide sufficient opportunities and conditions to achieve constructive collaborative communication.

Chapter 4

After the exploratory studies in the previous chapter identified how external factors like gender, prior experiences and motivational beliefs impact pair-programming, we finally report on an examination of an empirically-guided intervention to improve students' collaboration during pair programming. This intervention explored the role instructor-directed feedback plays on elementary students' dyadic collaboration during pair programming.

We used a multi-study design, collecting video data on students' dyadic collaboration. Study 1 qualitatively explored dyadic collaboration by coding video transcripts of four dyads which guided the design of Study 2 that examined conversation of six dyads using MANOVA and non-parametric tests.

Results showed that students receiving feedback used productive conversation categories at a significantly higher level than the control condition. Results are discussed in terms of group differences in specific conversation categories. This study points to specific ways to support students in pair programming contexts so that they can maximize the benefits afforded through these experiences.

Contributions

Each of the chapters explored different aspects of pair-programming for elementary students. Thus, there are distinct intellectual and methodological contributions from each of the chapters. In the following section we list are main contributions coming from the three studies reported in this dissertation:

- This research provided insights on how different gender pairings such as girl-girl, boy-girl and boy-boy pairs' collaborative practices and programming compare. These findings will help researchers and educators support aspects of pair programming male and female students and design interventions accordingly.
- Our research shed light on how gender, prior experience, and motivational beliefs and interests intercorrelated and influence students' learning in CS. The findings may direct educators and researchers toward factors related to programming practices that need most attention for elementary classrooms.
- This research also provided a preliminary validation of an instrument that measures students' self-perceived ability beliefs and interests in programming at elementary level. This can be utilized and adapted by researchers to measure elementary students' motivational beliefs and interest in future studies.
- This work provided a preliminary validation of a framework for discourse analysis in pair programming that can capture the more complex patterns and nuances of students' collaborative practices during programming activities.
- This work helped gain important insights into students' natural tendencies in collaborative discourse in pair programming that were critical for developing a more refined and targeted study design.

- Finally, this study examined an intervention that utilized a feedback framework specifically designed to engage students in high quality collaboration. Educators can utilize this intervention to help students better collaborate not only to program better, also in any computer-supported collaborative environments.

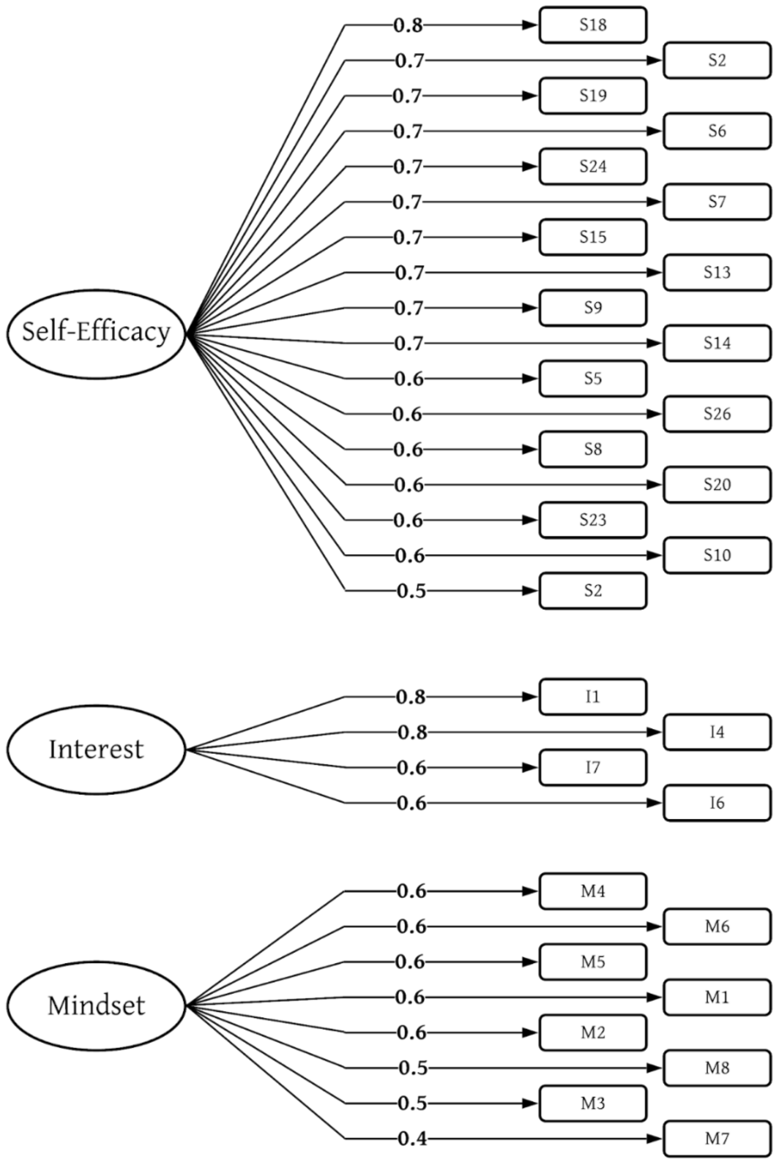
Future Research

The pattern of relationships among those variables also may change as students develop and move on to middle and high school. Future research may expand to middle grades and high school students to draw a complete picture of those variables. Future researchers could therefore replicate the current study testing the analytical framework on different computer supported collaborative environments. Another valuable direction for future research is to learn more about the beliefs and interests of different genders and how that impacts their collaboration and performance in pair programming. Future studies should strive to enlarge both the number and diversity of students studied. Another limitation is that we did not measure the existing collaboration norms of the classrooms. Although there was a single teacher of record for the students in each study, these classes all had different primary teachers that they spent most of the school day with. Both studies took place at suburban elementary schools, with Study 1 in an academically gifted classroom while Study 2 was with classrooms of students with weekly exposure to technology or coding experiences. Thus, our populations may not be representative of a larger body of students of this age. As already noted, the validation work done on instruments developed and used in these studies was preliminary and should be considered only the first step in a more thorough refinement and validation process.

APPENDICES

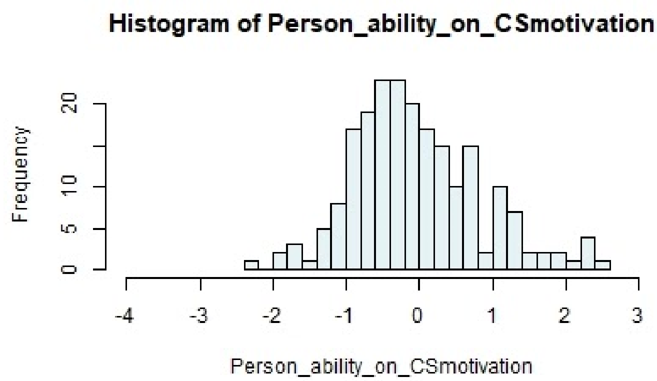
Appendix A

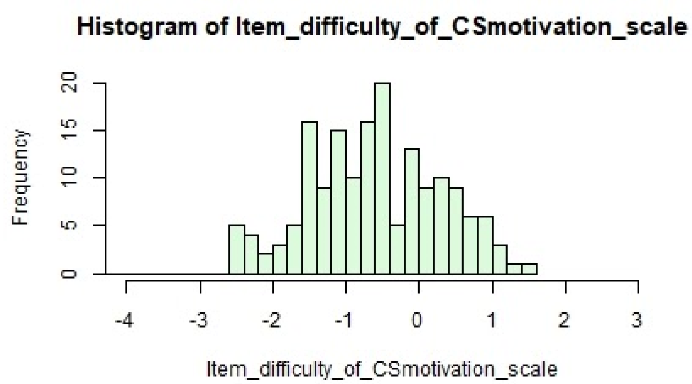
Reliability of each item in the final list (Chapter 2).



Appendix B

Item difficulty and person ability distribution for the multidimensional Rasch model (Chapter 2).





Appendix C

R code for Bia Corrected Factor Score Path Analysis (Chapter 2)

```
library(MASS)
library(lavaan)
library(haven)
Raw_data_for_FSPA <- read_csv("Raw data.csv")
data_n<-Raw_data_for_FSPA

##true relationships among latent variables
ntruesem<-'
        csca~ selfeff+pre_exp
        selfeff~ interest+mindset+pre_exp
        interest~mindset+pre_exp+gender
        mindset~pre_exp+gender'
ntsem<-sem(ntruesem,data=data_n,std.ov=T)
summary(ntsem)

##true sequential mediation effect S-D-U-A
#coef(ntsem)["interest~mindset"]*coef(ntsem)["selfeff~interest"]*coef(ntsem)["csca~selfeff"]*c
oef(ntsem)["csca~selfeff"]*coef(ntsem)["csca~selfeff"]
##specify SEM
semodeln<-'
      fm=~ M1+M2+M3+M4+M5+M6+M7+M8
      fm ~~ 1*fm

      fi=~ I1+I2+I3+I4
      fi ~~ 1*fi

      fs=~ E1+E2+E3+E4+E5+E6+E7+E8+E9+E10+E11+E12+E13+E14+E15+E16+E17
      fs ~~1*fs

      fc=~ 1*csca
      fg=~ 1*gender
      fp=~ 1*pre_exp

      fc~ fs+fp
      fs~ fi+fm+fp
      fi~ fm+fp+fg
      fm~ fp+fg
      fc~~fc
      fs~~fs
      fi~~fi
      fm~~fm
```

```
##estimate SEM with typical concurrent estimation of parameters
sem1n<-sem(semodeln,data_n)
summary(sem1n)

##sequential mediation effect S-DU-A under full info SEM
#coef(sem1)["fd~fs"]*coef(sem1)["fu~fd"]*coef(sem1)["fa~fu"]

## estimate SEM with BCFSPA

##step a-- estimate individual measurement models
cfa_m<- cfa('fm=~ NA*M1+M2+M3+M4+M5+M6+M7+M8
   fm ~~ 1*fm',data_n)
summary(cfa_m)
cfa_i<- cfa(' fi=~ NA*I1+I2+I3+I4
   fi ~~ 1*fi',data_n)
summary(cfa_i)
cfa_s<- cfa('fs=~
NA*E1+E2+E3+E4+E5+E6+E7+E8+E9+E10+E11+E12+E13+E14+E15+E16+E17
   fs ~~1*fs',data_n)
summary(cfa_s)
cfa_c<- cfa(' fc=~ NA*csca
   fc ~~ 1*fc',data_n)
summary(cfa_c)

##step b-- estimate factor scores and covariance
fs_m<- lavPredict(cfa_m,method="regression")
fs_i<-lavPredict(cfa_i,method="regression")
fs_s<-lavPredict(cfa_s,method="regression")
fs_c<-lavPredict(cfa_c,method="regression")
fscov_n<-((n-1)/n)*cov(data.frame(fs_m,fs_i,fs_s,data_n$gender,data_n$pre_exp,fs_c))

##step c--esimtatecorrected covariance
bcfscov_n<-fscov_n
#divide covariances by factor score and loading matrices
bcfscov_n[1,]<-bcfscov_n[,1]<-
 bcfscov_n[1,]/(attr(lavPredict(cfa_m,method="regression",fsm=T),"fsm")[[1]] %*%
        lavInspect(cfa_m,what="est")$lambda)
bcfscov_n[2,]<-bcfscov_n[,2]<-
 bcfscov_n[2,]/(attr(lavPredict(cfa_i,method="regression",fsm=T),"fsm")[[1]] %*%
        lavInspect(cfa_i,what="est")$lambda)
bcfscov_n[3,]<-bcfscov_n[,3]<-
 bcfscov_n[3,]/(attr(lavPredict(cfa_s,method="regression",fsm=T),"fsm")[[1]] %*%
        lavInspect(cfa_s,what="est")$lambda)
bcfscov_n[4,]<-bcfscov_n[,3]<-
 bcfscov_n[4,]/(attr(lavPredict(cfa_c,method="regression",fsm=T),"fsm")[[1]] %*%
         lavInspect(cfa_c,what="est")$lambda)
```

```
##correct variances (fixed to one in standardized appraoch)
diag(bcfscov_n)<-1

##step d-- estimate path model with corrected covariance
pathmodel_n<- '
     fc~ fs+data_n.pre_exp
     fs~ fi+fm+data_n.pre_exp
     fi~ fm+data_n.pre_exp+data_n.gender
     fm~ data_n.pre_exp+data_n.gender
     fc~~fc
     fs~~fs
     fi~~fi
     fm~~fm   '
bcfspa1_n<- sem(pathmodel_n,sample.cov=fscov_n,sample.nobs = n)
summary (bcfspa1_n.
```