How Early Does the CS Gender Gap Emerge? A Study of Collaborative Problem Solving in 5th Grade Computer Science

Jennifer Tsan
Computer Science
North Carolina State Univ.
Raleigh, NC, USA
jtsan@ncsu.edu

Kristy Elizabeth Boyer
Computer Science
University of Florida
Gainesville, FL, USA
keboyer@ufl.edu

Collin F. Lynch
Computer Science
North Carolina State Univ.
Raleigh, NC, USA
cflynch@ncsu.edu

ABSTRACT

Elementary computer science has gained increasing attention within the computer science education research community. We have only recently begun to explore the many unanswered questions about how young students learn computer science, how they interact with each other, and how their skill levels and backgrounds vary. One set of unanswered questions focuses on gender equality for young computer science learners. This paper examines *how the gender composition of collaborative groups in elementary computer science relates to student achievement.* We report on data collected from an in-school 5th grade computer science elective offered over four quarters in 2014-2015. We found a significant difference in the quality of artifacts produced by learner groups depending upon their gender composition, with groups of all female students performing significantly lower than other groups. Our analyses suggest important factors that are influential as these learners begin to solve computer science problems. This new evidence of gender disparities in computer science achievement as young as ten years of age highlights the importance of future study of these factors in order to provide effective, equitable computer science education to learners of all ages.

CCS Concepts

•Social and professional topics → Computer science education; Student assessment; K-12 education;

Keywords

K-12, elementary, gender diversity, collaboration

1. INTRODUCTION

The benefits of introducing computer science to students at a young age are starting to be recognized [14]. Students often choose not to take computer science classes or enter a career in the field based upon misconceptions [3]. By reaching students earlier in their school years, we may be able to prevent students from forming these misconceptions and help them to make more informed decisions about a future career in computer science. However, bringing effective computer science learning opportunities to K-5 students is a challenging proposition [11]. There are many unanswered questions about effective teaching practices and how to assess students' skills, practices, and knowledge in age-appropriate ways [13, 20].

A very important category of unanswered questions centers around gender equity. The challenges of fostering and sustaining a diverse group of computer science learners has been long recognized at the postsecondary, high school, and even middle school levels, where evidence of gender disparity is clear [2, 3]. However, there has been very little research on these issues at the elementary school level. Research in math has shown that gender disparities in that area can begin as early as first grade [18]. Math gender stereotypes also emerge in elementary school and may have an effect on how students of each gender relate to math [7]. The community has yet to collect a strong body of evidence on gender disparities in computer science at the elementary school level.

This paper reports on a project to develop and implement an in-school computer science elective course at an urban elementary school in the southeastern US. We co-designed the course with an elementary school technology teacher, piloted it as an after-school course, and then offered the 30-contact-hour elective in school four quarters during the 2014-2015 academic year. We initially hypothesized that if we could recruit girls into this elective, there would be no gender gap in achievement between girls and boys. Our team conducts research on diversity in computer science education; the team itself is diverse; and we designed the curriculum, projects, and activities to have broad appeal to both genders. However, our in-class observations suggested that there were indeed differences in the quality of artifacts produced by students depending on the gender composition of their collaborative groups.

This paper investigates the following research question: *How is gender composition of collaborative groups in elementary computer science related to student achievement?* We examine the context of collaboration based on the long-recognized importance of collaboration for computer science practice [4, 9] and in light of evidence that learning collab-

SIGCSE '16 Memphis, Tennessee USA
© 2016 ACM. ISBN 978-1-4503-3685-7/16/03...$15.00
DOI: http://dx.doi.org/10.1145/2839509.2844605

oratively is beneficial to students in computer science [16].

We collected students' programs from one project in the 5th grade computer science course. These programs were rated for quality, and the results show that all-female groups produced significantly lower-scored artifacts than groups with at least one male. This finding is consistent with field observations made by the team and suggests that there is a gender disparity in computer science performance even at the young age of ten years old. We present the code quality comparison as well as a set of case studies to illustrate factors that are influential in the achievement of collaborative groups of computer science students at the elementary level, and suggest ways in which computer science education research can make advances in addressing this gender gap.

2. RELATED WORK

Recent years have seen an increase in efforts to engage elementary students in computer science, both in formal education and outreach programs. These projects span in-school and after-school interventions.

2.1 Elementary School Interventions

The KELP curriculum is a modular computer science curriculum for upper elementary school students [10]. The curriculum was designed around LaPlaya, a programming language and environment based on Scratch [13]. LaPlaya was built specifically to scaffold 4th-6th grade students in learning programming. The development of this curriculum and the preparation required to make the implementation successful have yielded informative insights. For example, some of the research findings suggest that it is important to reinforce content knowledge from other subject areas, construct an environment that fosters collaborative work, and provide careful feedback that emphasizes that there may be many solutions to a particular problem [11]. In interviews students were asked predictive questions such as what would happen when a user pressed the run button [8]. The results showed that some visual cues can be confusing: for example, some students believed that the location of blocks in the editing area determined the order in which the blocks would run.

Code Club, an after-school programming club in the UK, reached children in 1000 elementary schools in 2014 [22]. The goal of Code Club was to get elementary students excited about programming. Survey data showed that students were comfortable with the programming concepts they learned, and a third of the schools reported that over two thirds of the students in the club continued to use some form of digital creation outside of the club.

In Scotland, an in-depth study of 29 games created by pairs and groups of 60 students between the ages of 8 and 11 revealed the types of games students chose to create, the computer concepts used within the game, and how the code was organized and the game was designed [23]. The games were designed over the course of eight weeks. The authors' discussion suggested that the students were able to learn and apply their programming knowledge. 97% of the games had at least some functionality and 28% of the games had full functionality. The authors also note that, given more lessons and time to work, it was likely that students would have created more complex games with increasing functionality.

One question that arises with the introduction of computer science to young students is how the students' knowledge and skills should be evaluated. The PECT (Progression of Early Computational Thinking) Model was introduced to assess the computational thinking skills of students from the ages of 6-12 through their Scratch programs [20]. The model was applied to 25 projects found online in the Scratch community. The authors found that students in lower grades modified the sprites' looks and had them move and converse, but they often did not have any other features in their projects, such as collision and scoring. Upper elementary students had the same features as the lower elementary students but they also integrated collision, interaction, and scoring into their projects more often.

2.2 Collaboration in Elementary School

Research on how to foster, practice, and evaluate collaboration in computer science classrooms is ongoing for many ages of learners. For elementary computer science learning, some work has investigated collaborative problem solving. Notable work has examined the equity of students' relationships as they solve computer science problems in pairs [21, 15]. That work focused on elementary students who participated in a summer computer science course, and the authors analyzed the paired dialogues in terms of how much the students spoke to each other and the content of their conversations. The authors concluded that equity is contextualized and may be a result of certain students' desire to complete projects quickly.

Outside of computer science, a study of elementary students' collaboration revealed that female-male relationships in subjects such as physical education, reading, language art, social studies, and art are not equal [17]. Boys appear to benefit more from female-male pairings by developing their leadership skills and increasing their self-efficacy in problem solving, whereas these pairings can reinforce gender stereotypes for girls.

Given these findings, the importance of investigating gender and collaboration for elementary computer science education is clear. The remainder of this paper reports on a study to examine one aspect of gender and collaboration, the quality of artifacts produced. We also examine some possible factors influencing artifact quality through case studies.

3. ELEMENTARY CLASS & STUDY

In 2014 a team consisting of two computer science Ph.D. students, one computer science undergraduate, and a local elementary school technology teacher collaboratively designed a 30 hour 5th grade computer science curriculum. In this class students engage in problem solving, writing, and collaborative projects that span a series of topics in algorithmic thinking and programming, robotics, AI, and computers in society. The content of the curriculum was inspired by the intersection of CS Principles [1], Exploring CS [12], and the CSTA K-8 guidelines [5], which were adapted with the help of the teacher for 5th grade. The teacher had no formal background in computer science and the team introduced necessary topics to the teacher as they collaboratively designed and built the curriculum.

The curriculum was first piloted as a 13-day after-school club in Spring 2014, then was expanded and refined during Summer 2014 to fit the in-school 30-day, 45-minute class slot requested by our partner elementary school.[1] The school's

[1]The curriculum is modularly designed, so others who wish to use it may adopt smaller portions if desired.

demographics were 53.1% African American, 32.6% Caucasian, and 14.3% Hispanic, Latino, Native American, Asian, or mixed race. 47.4% of students received free or reduced lunch.

Each quarter during the 2014-2015 school year, teachers enrolled students in electives based on the students' expressed preferences. There was a total of 55 students enrolled in the computer science elective (16 girls, 39 boys).

Table 1: Class gender by quarter.

Quarter	**Girls**	**Boys**	**Total**
1	6	8	14
2	3	12	15
3	5	10	15
4	2	9	11

As part of exploratory work on the effectiveness of the curriculum, we collected the following: pre- and post-interviews, videos of students working in class, screen recordings of student programming, student short essays, and handwritten design artifacts including conditional trees and storyboards. A researcher observed the class an average of three times per week and made field notes.

The students in the class completed three major projects: two Scratch programming projects, and one research project on computer science and society. The analysis reported in this paper investigates the first programming project, in which students selected a fairy tale to implement as an interactive program. The teacher made the decision as to how the students could be paired. For some quarters he allowed the students to choose their partners and intervened only when he felt it was necessary. For other quarters he choose the pairs himself. Although most of the students worked in pairs, there were also cases where students worked in groups of three or four, or, as seen in one case study in Section 5, individually. The students were required to incorporate two instances of cause and effect from the fairy tale, and two instances of user input where the user's input decided the path the story took.

The project spanned approximately six class days and it began after students received between four and six days of instruction on using conditionals and loops, broadcasting and receiving signals, and moving sprites in Scratch. The project days were structured, with design and planning time required before students started programming. The paper-based design packet included a sheet for determining collaborative tasks and tracking their completion. On the first day the students selected and read their fairy tales, identified two instances of cause and effect and user input, and drew conditional trees. The next day the students identified the characters and backgrounds they would use in their program, drew a storyboard, and created a rough code plan. The third day was the only day reserved for the students to choose or make their characters and backgrounds. On the remaining days the students pair programmed, switching the "driver" at 15 minute intervals.

4. ANALYSIS AND RESULTS

The goal of this analysis is to examine the relationship between gender composition of groups and the quality of the final project they submitted. The sample size for this analysis is 18 pairs or groups (and in one case an individual): 4 all-female, 4 female-male, and 10 all-male.

First, quality of the submitted projects was rated. We created and iteratively refined a rating scheme inspired by previous work using the SOLO (Structure of the Observed Learning Outcome) taxonomy to rate 4th grade student programs [19]. Given students' completed work for assigned learning tasks, the SOLO taxonomy measures the complexity of the work. Each student's work is assigned to one of five levels depending on the criteria it meets. We found that for some important factors of the student programs, this taxonomy was not fine-grained enough, while in other cases it was too fine-grained. We ultimately rated the project based on three factors:

- Requirements: How well the project runs and meets the requirements given by the teacher (includes two instances of cause and effect and two instances of user-input).
- Consistency: Whether the project runs consistently (resetting appropriately before or after the code runs).
- Usability: Whether the project is intuitive for users (e.g., uses prompts to clearly guide the user).

Table 2 displays the four-level coding scheme for the Code Requirements factor. Consistency and Usability were rated on a binary scale.

Table 2: Coding scheme for Code Requirements.

Level	**Criteria**
0	Did not attempt to write the program
1	The program contains blocks but does not function, or it functions but does not show cause and effect or utilize user input
2	The program shows an instance of cause and effect and user input
3	The program shows two instances of cause and effect and user input

After rating the 18 projects, their overall weighted score was computed as $(0.7 * Requirements) + (0.2 * Consistency) + (0.1 * Usability)$. We determined the weights before starting our analysis. The weights were based on the emphasis the teacher placed on each criterion. The median of all project scores was 0.62, with a minimum of 0.23 and a maximum of 1.0.

We divided the groups as follows: all-female (FF), female-male (FM), and all-male (MM). Figure 1 shows a scatter plot of each pair's code score. The programs of all four all-female groups scored below the median, and their scores were clustered together. The scores of the female-male groups were also relatively close to each other, with their lowest scores being equal to the highest score of the all-female groups and their highest score being slightly lower than most of the all-male groups. The all-male groups displayed the widest range of scores, with both the highest and lowest scoring groups out of all the categories (however, note that the all-male group was also largest, with n=10). Seventy percent of the all-male groups scored 0.77 or higher.

We ran a Fisher's exact test to investigate whether there were differences in average code quality score between the groups. Because of the small sample, we first binned into high and low scores based on median, and considered a

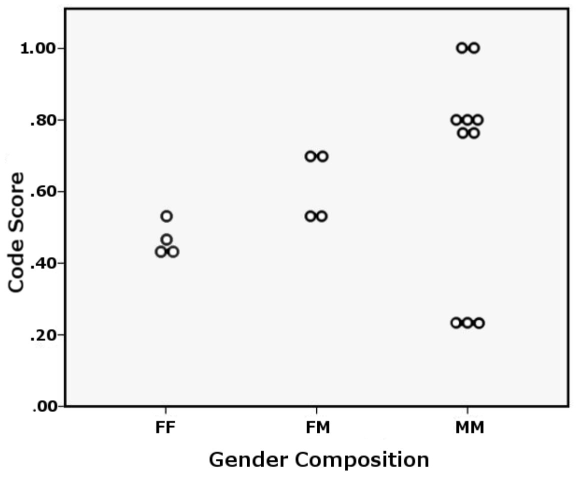


Figure 1: Code score vs. group gender composition.

Table 3: Results of the Fisher's exact test on gender composition and code score.

	Code Score		
	Low	**High**	**Total**
All Female	4 (100%)	0 (0%)	4 (100%)
1+ Male	5 (35.7%)	9 (64.3%)	14 (100%)
Total	9 (50%)	9 (50%)	18 (100%)

two-way split of gender composition: all-female (n=4) and at-least-one-male (n=14).[2] The results revealed that *none* of the all-female groups, and 64% of the at-least-one-male groups, achieved a high code score. This difference was statistically significant (p=0.04).

We also examined the all-male (n=10) and at-least-one-female (n=8) groups separately. In this split, 70% of all-male groups and 25% of at-least-one-female groups achieved high code scores, and this difference was not statistically significant (p=0.08).

5. CASE STUDIES

In order to better elucidate group dynamics, we examine selected case studies. These case studies are based on collected videos and the first author's field notes from observing the class. The cases were selected to illustrate the results and are not comprehensive.

5.1 Individual: Xavier

Xavier[3] took the class during the second quarter. The teacher mentioned that Xavier had experience from participating in the after-school pilot in Spring 2014; however, the project analyzed here was not part of that pilot. At first Xavier was paired with another student with less experience. However, the teacher noticed that Xavier's partner quickly became disengaged. The teacher moved Xavier's partner to another team and Xavier completed the project individually.

[2]Although this split resulted in unbalanced group sizes, it was necessary to investigate the hypothesis which arose from our classroom observations, that the presence of a male substantially altered group dynamics.

[3]All student names are pseudonyms.

When completing his design packet, Xavier bypassed the sprites, stages, and outline sections. However, his was one of six groups that completed the code plan, and he produced one of the two highest-rated conditional trees. Most groups drew two separate conditional trees that illustrated two instances of cause and effect. Although those trees met the requirements the teacher gave, they did not show any connection between two major events that occurred in their chosen fairy tale. In contrast, Xavier's conditional tree was unified and resembled a flow chart. Xavier's code plan was also detailed, included drawing chunks of Scratch blocks in the order he would later put them in, and he drew arrows to show how the program should flow. Xavier completed his project implementation work early, and when the teacher suggested he extend his project, Xavier added an alternate stage to the program when the player selected a specific ending. Xavier's program was one of two programs that received a perfect score.

5.2 Female-Male Pair: Mia and John

Mia and John participated in the class the during the final quarter of the 2014-2015 school year. Prior to the project, they were observed working well together (spontaneously) on basic programming exercises. However, sharing a computer proved more difficult for them. Mia often overrode John's ideas, choosing instead to ask the teacher for help. The team rarely solved a problem on their own due to the frequency with which they summoned the teacher. Mia was also reluctant to switch from the driver to the navigator role when the teacher announced that it was time to switch; she sometimes even took the mouse from John when he was the driver. The pair also spent much of their time modifying their sprites in Scratch and less time programming and problem solving.

Mia and John's program met the requirements of conditional branching. However, it did not run consistently and had a low usability score. To run the program from beginning to end, users had to take actions in addition to answering questions, but there were no prompts or user guidance. Still, Mia and John, along with one other pair, achieved the highest project scores out of the four female-male pairs. Even this score was only slightly higher than the overall median.

Figure 2 displays a screen capture of Mia and John's program running and Figure 3 shows the code for one of the characters, a prince. Mia and John did not use some of the concepts they learned in class, such as loops, broadcasting, and receiving messages. They only accounted for one user answer to their first question, and some actions were triggered by clicking on the sprite or pressing a key, but these available actions were not made apparent to the user. Overall, Mia and John failed to form a unified vision of their work and it appears that they did not successfully apply their potential when they were placed into a collaborative pair.

5.3 Female-Female Pair: Willow and Daphne

During the first quarter, Willow and Daphne were two of the six girls in the class. The girls asked to work as a team because they were friends and had worked together during the prior Scratch lessons. Willow and Daphne worked well together and were creative. Like Xavier, they also bypassed the sprites, stages, and outline section, completing

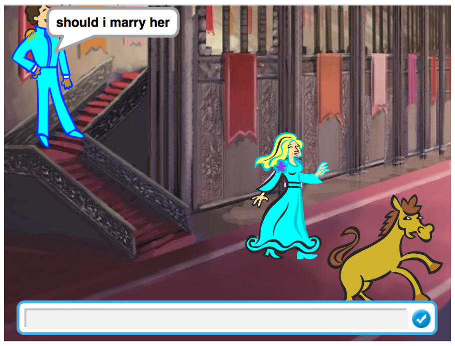
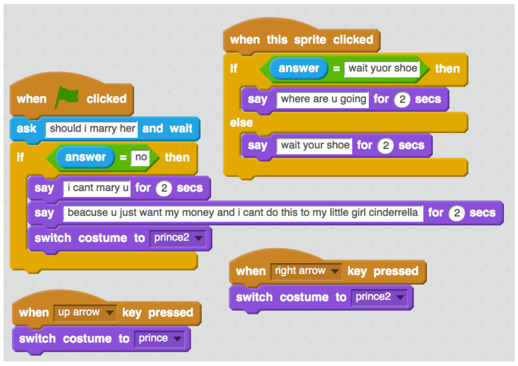


Figure 2: Screen capture of Mia and John's stage.



```
when [flag] clicked
ask [should i marry her] and wait
if <(answer) = [no]> then
    say [i cant mary u] for (2) secs
    say [beacuse u just want my money and i cant do this to my little girl cinderrella] for (2) secs
    switch costume to [prince2 v]

when this sprite clicked
if <(answer) = [wait yuor shoe]> then
    say [where are u going] for (2) secs
else
    say [wait your shoe] for (2) secs

when [right arrow v] key pressed
switch costume to [prince2 v]

when [up arrow v] key pressed
switch costume to [prince v]
```

Figure 3: Screen capture of Mia and John's code.

the other components instead. The conditional trees Willow and Daphne drew expressed two conditionals based on user input, but like many of their peers, the two conditions were not linked in an overall flow. Like Xavier's, their code plan included some hand-sketched Scratch blocks, but theirs included only the user input options, not the condition that resulted from each one.

When it came to implementation, Willow and Daphne did not implement the required cause-and-effect. Their code did run consistently and the user prompts were clear, but their characters' conversations were incoherent. The missing requirements were reflected in their score of 0.53, which although it was the highest among all of the all-female groups, was still well below the overall median.

5.4 Male-Male Pair: Greg and Harry

Greg and Harry participated in the last quarter of the school year. Both students had enthusiasm for programming and they often took creative liberties during the Scratch lessons. Although the pair worked well together and seemed to follow a similar line of thinking, Harry would sometimes get distracted by a friend in a different group, even leaving his seat to visit the other group. They were animated while brainstorming ideas for their "Jack and the Beanstalk" game, but spent less time on their design document, only completing about half of it.

This trend continued through their implementation, when they were engaged in off-task behavior with unrelated blocks. They only introduced one instance of cause and effect, and their program scored low for usability and consistency. Their code score was the lowest, shared by two other all-male groups.

6. DISCUSSION AND LIMITATIONS

Our results suggest that there is a relationship between gender and the artifacts produced by collaborative groups in a 5th grade computer science elective. While our analyses do not definitively answer our original research question, they do suggest factors that are influential and are important for future study.

Supporting Design and Planning. One important factor is the extent to which students engaged in design and planning. While these activities were scaffolded with a carefully designed paper-based packet, we observed a phenomenon that is often seen in computer science education. Namely, students often preferred to bypass design and planning, instead preferring to implement immediately or "tinker." While tinkering has been shown to have potential benefits, our results suggest that it is important to investigate how to foster the collaborative design and planning process for young learners.

Fostering Collaborative Skills and Practices. In addition to these concerns regarding design and planning, collaboration skills and practices are a very important area for future study. Young learners have had little opportunity to practice collaboration, and they display tendencies to separate rather than collaborate (as we saw with Xavier and his partner, as well as with Mia and John). This tendency, of course, is often observed with more sophisticated learners, as one of the authors of this paper can attest: even in second and third programming courses, she finds herself needing to remind students constantly not to just attempt to *divide* the implementation tasks in a collaborative project, but to work *together* and share ideas. If we can identify ways to support good collaborative habits as early as elementary school, perhaps those habits can propagate into students' future education and careers.

Investigating Cognitive and Social Constructs. Differences in performance at young ages is almost certainly related in part to influences that occur prior to the class itself. For example, there is a systematic gender difference between toys marketed to young children. Legos and robots are more likely to be marketed for boys, while dolls and kitchen sets are more likely to be marketed for girls. It is possible that the traditionally "boy" toys foster spatial and programmatic reasoning skills more than traditionally "girl" toys, and recent literature suggests a correlation between spatial reasoning and computer science performance [6].

Limitations. As mentioned above, there are important limitations due to the small sample size and exploratory nature of the study described here. The sample size, n=18, was partially due to the fact that not all projects were submitted. This points to a bigger-picture discrepancy between required versus elective in-school classes. Ours was offered as an elective, which was not permitted to include exams, and for which grades were assigned in a much less structured way than required courses such as mathematics and language. It will be important to explore this phenomenon on a larger dataset in the future.

Another limitation to the study stems from the evolution of the course across quarters. As is often the case, the course was refined and improved over time. The course was also confined to a single instructor and thus the teacher effect cannot be eliminated. Finally, it is important to note that the researcher making field notes was not always able to achieve the ideal role as a silent observer in the classroom. Because this classroom teacher was still developing computer science skills and knowledge, the researcher provided input and support when needed.

7. CONCLUSION AND FUTURE WORK

This paper has examined the first of a series of projects created by students in an in-school 5th grade computer science elective. We examined 18 group projects and investigated whether there was a difference between gender composition of groups and the quality of the final artifact they produced. Classroom observation suggested that a difference did exist, and results demonstrated that all-female groups achieved significantly lower program quality than groups with at least one male. The analyses suggest that numerous factors play important roles, including the extent to which students were willing to engage in design, students' collaborative practices, and other cognitive and social factors.

There are several important directions for future work. The finding of a possible gender discrepancy in computer science achievement as early as elementary school highlights the need for research on diversity and equality in K-5. More broadly, there is a need for increased computer science education research, curriculum building, and integration with other subject areas at the elementary level. As a computer science education community of practitioners and researchers, we have the opportunity to expand our focus on computer science learners to include younger children, as they make their way toward becoming the next generation of computer scientists.

ACKNOWLEDGEMENTS This work is supported in part by the Wake County Public School System and the National Science Foundation through Grant CNS-1453520. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

8. REFERENCES

[1] AP Computer Science Principles Draft Curriculum Framework, 2014.

[2] S. Beyer, K. Rynes, J. Perrault, K. Hay, and S. Haller. Gender differences in computer science students. *SIGCSE Bulletin*, 35(1):49–53, 2003.

[3] L. Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *SIGCSE '06*, pages 27–31, 2006.

[4] B. Coleman and M. Lang. Collaboration Across the Curriculum: A Disciplined Approach to Developing Team Skills. In *SIGCSE '12*, pages 277–282, 2012.

[5] I. Computer Science Teachers Association, Association for Computing Machinery. Computer Science K-8: Building a Strong Foundation, 2012.

[6] S. Cooper, K. Wang, M. Israni, and S. Sorby. Spatial Skills Training in Introductory Computing. In *ICER'15*, pages 13–20, 2015.

[7] D. Cvencek, A. Z. Meltzoff, and A. G. Greenwald. Math-Gender Stereotypes in Elementary School Children. *Child Development*, 82(3):766–779, 2011.

[8] H. Dwyer, C. Hill, A. Hansen, A. Iveland, D. Franklin, and D. Harlow. Fourth Grade Students Reading Block-Based Programs: Predictions, Visual Cues, and Affordances. In *ICER'15*, pages 111–119, 2015.

[9] K. Falkner, N. J. Falkner, and R. Vivian. Collaborative Learning and Anxiety: A phenomenographic study of collaborative learning activities. In *SIGCSE '13*, pages 227–232, 2013.

[10] D. Franklin, D. Harlow, H. Dwyer, J. Henkens, C. Hill, A. Iveland, A. Killian, and Staff. Kids Engaged in Learning Programming (KELP-CS) -Module 1 Digital Storytelling. A computer science curriculum for elementary school students. https://discover.cs.ucsb.edu/kelpcs/educators/KELPCSIntro.pdf, 2014.

[11] D. Franklin, C. Hill, H. Dwyer, A. Iveland, A. Killian, and D. Harlow. Getting Started in Teaching and Researching Computer Science in the Elementary Classroom. In *SIGCSE '15*, pages 552–557, 2015.

[12] J. Goode and G. Chapman. Exploring Computer Science: A High-School Curriculum Exploring what Computer Science is and What it Can Do. https://www.exploringcs.org/curriculum, 2013.

[13] C. Hill, H. A. Dwyer, T. Martinez, D. Harlow, and D. Franklin. Floors and Flexibility: Designing a Programming Environment for 4th-6th Grade Classrooms. In *SIGCSE '15*, pages 546–551, 2015.

[14] E. Kazakoff and M. Bers. Programming in a Robotics Context in the Kindergarten Classroom: The Impact on Sequencing Skills. *Journal of Educational Multimedia and Hypermedia*, 21(4):371–391, 2012.

[15] C. M. Lewis and N. Shah. How Equity and Inequity Can Emerge in Pair Programming. In *Proceedings of ICER'15*, pages 41–50, 2015.

[16] C. M. Lewis, N. Titterton, and M. Clancy. Using collaboration to overcome disparities in Java experience. In *ICER '12*, pages 79–86, 2012.

[17] M. E. Lockheed and A. M. Harris. Cross-Sex Collaborative Learning in Elementary Classrooms. *American Educational Research Journal*, 21(2):275–294, 1984.

[18] J. P. Robinson and S. T. Lubienski. The Development of Gender Achievement Gaps in Mathematics and Reading During Elementary and Middle School: Examining Direct Cognitive Assessments and Teacher Ratings. *American Educational Research Journal*, 48(2):268–302, 2011.

[19] L. Seiter. Using SOLO to Classify the Programming Responses of Primary Grade Students. In *SIGCSE '15*, pages 540–545, 2015.

[20] L. Seiter and B. Foreman. Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *ICER '13*, pages 59–66, 2013.

[21] N. Shah, C. Lewis, and R. Caires. Analyzing Equity in Collaborative Learning Situations: A Comparative Case Study in Elementary Computer Science. In *ICLS*, pages 495–502, 2014.

[22] N. Smith, C. Sutcliffe, and L. Sandvik. Code Club: Bringing Programming to UK Primary Schools through Scratch. In *SIGCSE '14*, pages 517–522, 2014.

[23] A. Wilson, T. Hainey, and T. Connolly. Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In *Proceedings of the European Conference on Games Based Learning*, pages 549–558, 2012.