International Journal of Computer Science Education in Schools, November 2024, Vol. 7, No. 1
ISSN 2513-8359

Investigating the Impact of Introducing Pair Programming to Primary Computing Education on Female Pupils' Attitudes Towards Computing

Katharine CHILDS
Raspberry Pi Foundation, Cambridge

Sue SENTANCE
Raspberry Pi Computing Education Research Centre,
University of Cambridge, Cambridge

DOI: 10.21585/ijcses.v7i1.218

Abstract

Gender balance in computing education is a decades-old issue that has been the focus of much previous research. In K-12, the introduction of mandatory computing education goes some way to giving all learners the opportunity to engage with computing throughout school, but a gender imbalance still persists when computer science becomes an elective subject. The study described in this paper investigates whether introducing pair programming would make a difference to primary-aged girls' attitudes to computing and intent to study the subject in the future. A randomised controlled trial (RCT) was designed and implemented around a 12-week intervention with 785 female pupils between the ages of 8 and 10 years, alongside a qualitative evaluation investigating teachers' and pupils' experience of the interventions and the development of materials and teacher preparation resources. The results of the RCT showed no statistically significant changes in student attitudes towards computing or intent to study further, although the qualitative data indicated that both teachers and pupils found the interventions engaging and enjoyable. Themes emerging from the qualitative data point to the importance of collaboration in supporting a development in pupil confidence. Overall, these results emphasise the societal and systemic barriers around computer science and technology engagement across genders that persist despite many initiatives being implemented over many years.

Keywords

K-12 computing education, pair programming, gender balance

1. Introduction

There is a decades-old problem with gender imbalance in computing education, at all levels of schooling and university, with an equally long history of research into the reasons why this is the case (Butler, 2000). Global statistics show that gender imbalances persist into the technology workforce, particularly in digital transformation roles such as data and cloud computing (Global Education Monitoring Report Team, 2024). Some evidence has shown that the gender stereotypes around computing affecting choices on further study develop early (Eccles, 2015), while there is less research available on specific approaches that can be used to address these stereotypes. Universal access to inclusive quality education offers a route to eliminating gender and wealth disparities (United Nations Development Programme (UNDP), n.d.) so in theory, mandatory computing education at primary and lower secondary phases of education provides a leveller to gender balance.

In school, girls do well in computing. An analysis of recent examination results in England demonstrates that girls are more likely to achieve higher grades than boys in formal computer science (CS) education (Kemp et al., 2019). However, girls may be less likely to be encouraged to continue with computing education or to take it in the first place (Cheryan et al., 2009) and amongst students who have chosen to study GCSE Computer Science, girls are less likely to aspire to be a computer scientist compared with boys (Hamer et al., 2023). Other studies have identified gender differences between learners in their attitudes towards computing. Male students are generally more confident using computers (Beyer et al., 2003) because they have more access and exposure to computers at home (Varma, 2009) and these gender disparities affect students' achievement in computing from as young as ten years old (Tsan et al., 2016).

This paper discusses the design, implementation, and evaluation of an intervention designed to investigate whether using pair programming – as part of primary computing education – could reduce barriers to female pupils' uptake of computing. The context for these studies is England, which has mandatory computing in Grades K-8 (primary and lower secondary education) with elective qualifications in CS offered for pupils at Grades 9-12 (upper secondary). Mandatory computing education ensures that computing is offered to all children but when it becomes elective and young people choose a reduced number of subjects, educational statistics from 2022 report that only 21% of pupils taking CS at age 15/16 were female, and 15% at age 17/18. Therefore, this study was motivated by the need to maintain girls' interest during the mandatory stage of education.

2. Related work

2.1 Gender balance in computing

Work to research gender balance in computing is by no means new. A review conducted of gender balance research in computing from the 1980s and 1990s suggested that boys may have access to more role models in computing, may receive more encouragement to pursue the subject, and that software may be developed with a bias towards interests traditionally considered to be male (Butler, 2000). Given that this review dates from over two decades ago, gender balance in computing is an issue we've been trying to address for many years.

Initial studies framed the low number of female students as a puzzling phenomenon and used an abductive approach to create a metaphor to describe the imbalance. An example of this is the "shrinking pipeline" metaphor which illustrates the decreasing number of girls studying computing at each stage of education (Camp, 1997). Research that references this viewpoint aims to increase the likelihood of girls remaining engaged with computing (Porter et al., 2013), although it does not address the issue that there are too few girls entering the pipeline at the outset. A separate metaphor suggests that improving girls' participation in computing is like "unlocking the clubhouse" to create more inclusive spaces (Fisher & Margolis, 2002). Through their research and work at Carnegie Mellon University, Margolis and Fisher increased the proportion of female computing students from 7% in 1995 to 42% in 2000 through a multi-faceted set of interventions that included contextualising learning and broadening computer science stereotypes. However, the clubhouse metaphor is not widely understood outside of the context of the USA, which limits the generalisability of the findings.

2.2 Pair programming

Pair programming has been recognised as an important collaborative approach in industry and education for the past two decades (Hanks et al., 2011). The approach has been used in both agile software development and education to improve code quality (McDowell et al., 2006). Similar findings in K-12 (primary and secondary) environments demonstrated that pair programming generally increased pupil attitudes and confidence toward computing (Denner et al., 2014). Pair programming activities in school have a defined structure in which two pupils work together at a single computer to jointly create a computer program (Werner & Denning, 2009). One pupil takes on the role of 'driver', has control of the keyboard and mouse, and writes the code (Denner et al., 2014). The second pupil is the 'navigator', who reads out any instructions, monitors the code for errors, and points these out to the 'driver' (McDowell et al., 2006). Pupils regularly swap roles after a designated period of time, so that they both perform both roles equally. The teacher's role includes training the pupils in successful pair interactions and ensuring that pairs rotate regularly and fairly. The success of pair interactions is actively managed by the teacher as well as being evaluated by the pairs themselves (Williams et al., 2008).

Research that builds initial studies into pair programming is largely concerned with either improving the *process* of pair programming, or the analysing the *behaviours* that pupils exhibit when collaborating. Studies that explore the *process* of pair programming seek to better understand instructional strategies that can be used. For example, students in 4th and 5th grades were found to prefer two-computer pair programming because they felt more independent (Tsan et al., 2020) and a study with 6th grade students found that semi-free switching between roles in pairs led to increased achievement in a post-test of computing concepts (Zhong et al., 2017). However, the focus on student outcomes in these studies means that the role of the teacher in setting up effective pair programming activities has not yet been fully explored. Although the pair programming approach is a specific pedagogy employed in computing, primary (K-5) teachers use paired work in other subjects too and less is known about how teachers can use their pedagogical content knowledge (PCK) (Carlson et al., 2019) to assign effective pairings.

Recent work suggests that both teachers and students perceive pair programming as an equitable method for learning to write programs (Graßl & Foster, 2024), although when pair programming interactions have been investigated, some disparities emerge. Four different types of behaviour have been observed between pairs in an after-school setting with pupils aged 11 - 14: game interaction, when the pair discussed designing or programming the game; non-game interaction, when the pair discussed or reacted to non-task activities; third person present, when the pair interacted with a teacher or peer; and no interaction, when the pair did not interact (Campe et al., 2020). The most common interaction was about the game, although this made up less than half of all the interactions. Pair programming has also been found to lead to inequitable relationships between 11 - 12-year-old students if pairs decide to focus on completing programming tasks quickly (Lewis & Shah, 2015). These variations in pair interactions suggests that pair programming requires careful training of teachers to ensure that all elements of the pedagogy are understood and applied, without bias.

2.3 Pair programming and gender

An emerging body of evidence suggests that collaborative teaching approaches can engage more girls with computing (Tsan et al., 2016). This is of particular interest when learning to write computer programs, which can be seen as the most difficult aspect of the computing curriculum for learners (Kallia & Sentance, 2018). Introducing a shared, group approach requires a shift from traditional computing pedagogy. Learning to code changes from a series of tasks undertaken by individuals, to a sociocultural experience in which pupils work together to create and share digital content (Kafai & Burke, 2013). Talk and discussion promote a social construction of knowledge; according to sociocultural theory, a child's development involves social interaction, dialogue, and mediated activity between learners and with their teachers (Vygotsky, 1978). Gender differences have been observed where girls are more likely than boys to be motivated by social dimensions of learning (Korpershoek et al., 2021) and to express a preference for working toward social goals (Hijzen et al., 2006).

Research focusing on girls' engagement has shown that pair programming particularly impacted K-12 girls' enjoyment (Liebenberg et al., 2012) and interest (Werner et al., 2004) in programming. This suggests that pair programming has the potential to be used as an inclusive pedagogy to benefit girls’ perceptions of computing, whilst also supporting all learners. Female undergraduate students also report positive outcomes from using a pair programming approach on their confidence levels, although the literature attributes this to two different mechanisms: either through social engagement and peer learning (Ying et al., 2019) or through girls observing a similarity in knowledge with peers (Yates & Plagnol, 2022). Less is known, however, about K-5 teachers’ and pupils’ perspectives on the mechanisms that cause these outcomes.

The research described above indicates that pair programming may be an effective way to improve the motivation of girls to continue to study computing at school. However, many of these studies are small in scale. In this study, the first intention was to consider the impact of pair programming in a larger-scale study using a rigorous quantitative method, and then to explore a smaller group of teachers’ and pupils’ views about which aspects of pair programming are most beneficial for girls to contribute towards better understanding the mechanisms that increase girls’ confidence through pair programming activities.

The study described in this paper has the following research questions:

RQ1: How does introducing pair programming in Grades 3-5 impact on girls’ attitudes towards computing?

RQ2: What are the experiences of teachers and pupils in computing lessons when using a pair programming approach?

RQ3: In what ways do teachers and female pupils think that pair programming activities increase girls’ confidence in computing?

3. The Study

To address the research questions, a cross-organisational research group designed a mixed-methods study to explore the effect of teaching computing using pair programming on primary school pupils' attitudes toward computing. The study had three components:

- A small-scale preliminary study, akin to a pilot, to trial ways of introducing teachers to pair programming and evaluate the impact on a small scale.
- A large-scale main study, designed as a randomised controlled trial (RCT).
- A qualitative evaluation of the implementation fidelity and experience of the intervention with a small sample of schools.

The study was externally funded and formed part of a larger programme of separately delivered interventions. The three parts of the study were conducted by a multi-organisation research team, of which the authors are a part, between 2019 and 2022. The authors are part of the organisation that conducted the preliminary study, recruited schools, designed the intervention and provided support to participating schools for the large-scale main study. Data were collected and analysed by a second organisation. The two organisations worked closely together, and the authors recruited and liaised with participants throughout the duration of the study; in this way, all partners were familiar with and consulted about the processes used for data collection and analysis. In this paper, we have drawn on interpretations made by the second organisation, and added our own where appropriate, to situate the findings within the pair programming literature.

3.1 Preliminary study

A preliminary study, prior to the RCT, was conducted in 2019 with eleven primary schools to identify teachers' perspectives and experiences of using pair programming and incorporate these into the training materials for the main study (Leonard et al., 2021). One or two teachers from each school attended a one-day, in-person training day. The training was designed and delivered by members of the wider research team who had previous experience teaching in primary and secondary schools, introducing teachers to the pair programming methodology chosen for this study. Teachers then worked in pairs and practised the approach with a sample activity using the visual programming language Scratch[1]. Scratch was chosen because it is commonly used in primary schools in England.

With some attrition, the final sample included 10 primary schools, comprising 2 independent and 8 state-funded institutions, all of which were mixed-sex. Among them, 9 schools were situated in urban locations. These schools taught their usual programming lessons using the pair programming approach to 356 Year 6 pupils (171 female and 185 male) between January and March 2020.

The research design included an interview with participating teachers in March 2020 once they had used the pair programming approach for a minimum of six weeks in lessons. However, due to the onset of the Covid-19 pandemic, a period of emergency school closures meant that only one teacher was interviewed.

3.2 Main study: randomised controlled trial

A randomised controlled trial (RCT) is a rigorous tool to examine causal relationships between an intervention and outcome. The act of randomisation balances participant characteristics (both observed and unobserved) between the groups, allowing attribution of any differences in outcome to the study intervention (Hariton & Locascio, 2018). In this case, the RCT was designed to evaluate the impact of a 12-week intervention in school. It was funded by England's ministry of education, positioned as one pilot study amongst other studies to establish whether there was any evidence for a single intervention that might reduce gender imbalance in computing. The intervention is described in Section 4.1, and was developed by the authors and other colleagues, with the RCT itself being designed and implemented by another organisation within the broader team. The outcome measures for the RCT were: a) pupil scores on the Student Computer Science Attitudes Survey (SCSAS) (Haynie & Packman, 2017) which captures pupils' attitudes towards computing and b) pupils' response to a single item survey measure of whether the pupil plans to continue study computing.

---

[1] https://scratch.mit.edu/

3.3 Qualitative evaluation study

The qualitative study was, in essence, a process evaluation (Humphrey et al., 2016), which was conducted with a small number of schools to check for implementation fidelity and schools' experience of the intervention. It examined the mechanisms of change and the diversity of implementation and intervention delivery. A case study approach (Merriam, 1998) was used which aimed to capture the range and diversity of participant experiences. Individual, in-depth semi-structured interviews lasting between 30 – 45 minutes were conducted with one teacher from each of the case study schools (see Table 1), to explore their experiences of the intervention and any factors that influenced their ability to implement the intervention with their pupils. Group discussions were conducted with pupil focus groups at two case study schools, each lasting about 20 minutes. Pupils ranked different skills by importance for computing, discussed whether statements about computing (e.g., "boys and girls are equally likely to have computing as their favourite subject") were true or false, and completed sentence starters related to pair programming and computing lessons more generally. Additionally, lesson observations were conducted in the same two schools just before the focus groups took place, so that the pair programming lessons could be referenced in the discussions.

3.4 Participants

To assist recruitment to the RCT, a third-party paid-for marketing campaign was used to maximise representative coverage across schools in England. All primary schools in England were eligible, as long as they had female pupils from either Year 4 (aged 8 - 9 years old) or Year 6 (aged 10 - 11 years old). All schools that entered the sample did so voluntarily.

The 116 participating schools were randomly divided into a control group of 58 schools, who taught computing to a `business as usual' model, and a treatment group of 58 schools, who delivered an intervention. Researchers conducting the evaluation used school reference numbers as unique identifiers to assign schools randomly to either group. Following randomisation, balance checks on other school-level variables were carried out using the school's performance status and a standard proxy for measuring socioeconomic status. Pupils were blind to allocation during the programme and during outcome data collection. Teachers were not blind to allocation; they were responsible for delivering the materials, and were aware that there was both a control and a treatment group and of which group their school was in. Data was collected for both boys and girls, but only data from girls was analysed for primary and secondary analyses in the main study.

Ethical processes were adhered to for all three parts of the study. Schools shared information sheets and withdrawal forms with all parents of pupils in participating classes. These letters explained the purpose of the research and what taking part would involve for parent (where relevant) and their child. Parents were invited to withdraw their child from the RCT if they did not want them to take part. Pupils whose parents had withdrawn them from the RCT were still able to engage in the computing activities themselves, but did not take part in any of the evaluation activities and no data on these pupils was collected. All participants in the interviews and focus groups were informed that their participation was entirely voluntary and that they could withdraw at any point.

For the qualitative evaluation, the authors recruited four state-funded schools from the treatment group using sampling criteria including geographical location and a proxy indicator of pupil socioeconomic status to ensure a mix. Quality assurance mechanisms indicated schools had good or excellent provision. Participant details are shown in Table 1.

Table 1. Sample of Schools, Teachers and Pupils Participating in the Qualitative Study

School	Teacher	Pupils
S01	Computing specialist teacher More than 10 years' teaching experience Male	6 year 4 pupils (8 - 9 years old) 4 female (range of confidence in computing) 2 male (range of confidence in computing)
S02	Computing specialist teacher More than 10 years' teaching experience Female	5 year 4 pupils (8 - 9 years old) 3 female (range of confidence in computing) 2 male (range of confidence in computing)

S03	Classroom teacher with curriculum responsibilities for computing Fewer than 10 years' teaching experience Female	None
S04	Classroom teacher with curriculum responsibilities for computing Fewer than 10 years' teaching experience Male	None

3.5 Data analysis

As described above, the RCT used the SCSAS survey tool, which is a validated survey tool designed to measure pupil attitudes towards computing (Haynie & Packman, 2017). It has a high level of within-construct consistency with the alpha value of the five sub scales ranging from 0.85 to 0.93. The language of the SCSAS questions was adapted to a) ensure the questions used language that pupils in schools in England would be familiar with and b) ensure that pupils at primary levels would be able to independently understand what the questions were asking them (e.g. by replacing the word 'peers' with 'friends').

Data cleaning of survey data ensured any data points deemed potentially unreliable were removed; for example, all data was deleted for pupils who had answered in a straight pattern (e.g. a survey with the answer 'Strongly disagree' for every question of the SCSAS). The final data set consisted of (1) data from female pupils who had completed the endline survey matched to their baseline data and (2) data from female pupils who had completed only the endline survey, using a multistep matching process to match as many baseline and endline surveys as possible (Kelly et al., 2022).

The model analysing the SCSAS scores used a linear OLS (Ordinary Least Squares) regression, with the model which had as outcomes the intention to study computing using logistic regression. All analyses were conducted on an Intention to Treat (ITT) basis, meaning that outcomes were analysed on the basis of the groups that teachers and pupils were randomly allocated to, regardless of their compliance with the intervention. All models included the following covariates: baseline SCSAS score, school performance rating, and a proxy value for socioeconomic status. Their inclusion increases the precision of the impact estimates. All planned covariates were checked for missing data pre-analysis. Given that the endline data would likely include some pupils who were not included in the baseline dataset, pre-trial decision rules were specified for dealing with missing data as baseline scores on the SCSAS were to be used as a covariate in the analysis. More details on the data analysis are available in a technical report (Kelly et al., 2022).

For the qualitative data analysis, the data were transcribed where necessary and analysed using the Framework Approach (Ritchie et al., 2003). This involved summarising transcripts and notes into a matrix organised by themes and sub-themes (columns) as well as by individual cases (rows). This is particularly useful when multiple researchers are working on data (Gale et al., 2013). Lesson observation data was used to triangulate themes arising from interview data that related to pupil engagement. The second organisation conducted case and theme analyses with a focus on providing rich descriptions of participating experiences, whilst looking for explanations and linkages within and across participant groups, and these were then synthesised against the literature by the paper authors.

3.6 Reliability and Validity

Efforts were made to ensure the RCT was sufficiently powered to detect an effect size reliably. For each of the trials, power calculations were conducted based on informed assumptions about: the number of girls per-school per-year and the proportion of girls studying CS as an elective (source: Ministry of Education in England); the intraclass correlation coefficient (ICC) (source: comparable clustered RCTs previously carried out by the team conducting the RCT evaluations); the explanatory power of the variables included in the model at baseline (source: baseline data). The target number of schools to recruit included with a 40% attrition buffer to detect an estimated minimum detectable effect size of .05 along the 1-4 SCSAS scale, and a 10 percentage point increase in intention to study computing. However, due to higher than expected levels of attrition amongst the control group schools, the final analytical sample was not sufficient to detect an effect at the originally targeted effect size.

The Framework Approach (Ritchie et al., 2003) used for the qualitative data analysis ensures reliability through collaborative consensus, rather than using inter-rater reliability (Gale et al., 2013). Team discussions, transparent documentation and reflexivity were used to iteratively refine and agree codes, which allows for a nuanced understanding of the data while maintaining rigor.

All schools that entered the samples for both the main study and the qualitative research did so voluntarily, which has implications for the external validity of the findings. Schools that volunteer to participate in research are likely to be more enthusiastic about the intervention than an average school, and this may interact with the treatment effect to compound any effects.

4. The pair programming intervention

4.1 Intervention design

For the RCT intervention, a full set of teaching materials, including lesson plans, slide decks, example Scratch projects and seating plan templates were written by learning experience designers who had previously taught computing in schools. A panel of current primary educators reviewed the teaching materials for age-appropriateness and accuracy, and following this review some minor revisions were made.

The materials comprised twelve 1-hour lessons, which were divided into two units of work, each containing six lessons. Unit 1 started with an introductory lesson where pupils learned about pair programming, practised working in pairs, and were introduced to the pair programming map. This was followed by five lessons focussing on the creation of drawings in Scratch. In Unit 2, pupils took part in six lessons learning how to create simple animations in Scratch. In every lesson, there was a check-in for pupils to reflect on the effectiveness of their paired working. After each check-in, each pair of pupils were prompted to build up a set of rules for successful pair programming work. An overview of the two units is shown in Tables 2 and 3.

Table 2. Drawing Shapes (Unit 1)

No	Lesson	Learning objective	Example activity
1	Introduction to pair programming	To explain how working in pairs can help you learn to program	Pupils work in their allocated pairs to create a name for their pair and identify some ways that they plan to work together successfully.
2	Shapes	To explain how basic shapes can be drawn using a series of movements in a computer program	Pupils work in pairs to predict what shape a Scratch program will make and then modify it to create their own shape.
3	Pen marks	To explain how an input triggers an output	Pupils work in pairs to use an Event block and the Pen blocks to draw a shape. Pairs also think how they helped each other and how they might help each other more in the next task.
4	Repeat	To explain how to create shapes and patterns through repetition using the ‘repeat’ Control block	Pupils work in pairs to use the Repeat block to draw a square.
5	Tidy up time	To explain how to organise a program using subroutines from My Blocks	Pupils work in pairs to create a procedure to draw a shape using the My Blocks feature. They also discuss how both partners can contribute equally to the programming tasks.
6	Geometric line art	To explain how to create patterns through repetition using the ‘repeat’ Control block	Pupils work in pairs to use the skills they have learned in this unit to create geometric art shapes.

4.2 Implementation of intervention

The pair programming approach used in the main study was 'driver-navigator' (Zhong et al., 2017) with two pupils sharing a computer. Teachers were asked to make sure pupils switched roles every five minutes and guided to use a digital timer to ensure this was done accurately.

Pairs were decided on in advance by the teacher, using guidance such as pairing pupils of similar skill levels or pupils who had similar attitudes towards classwork. Teachers were advised to avoid swapping pairs and instead to plan bespoke activities to improve pair work if they noticed that pairs were working ineffectively.

Table 3. Programming Animation (Unit 2)

No	Lesson	Learning objective	Example activity
7	Two sprites	To explain how to create animation using two or more sprites and the 'move' and 'turn' Motion blocks	Pupils work in pairs to create a Scratch program that animates two sprites to dance together.
8	Park life	Pupils create simple interactive animations and explore selection using the 'if () then' Control block	Pupils investigate a pre-made Scratch program which uses selection to determine which animation effect is run.
9	Sea life	To explain how to create animations using the 'if () then, else' Control block	Pupils work in pairs to animate an underwater scene. They also discuss the 'Navigator' role and identify ways that they can improve giving instructions.
10	Artist's life	To explain how to create a detailed graphic animation using costumes	Pupils draw inspiration from real-life 'flip books' to add costumes to their animations. They also generate a list of six reasons to use pair programming.
11	Give life to	To explain how to prepare to create an animation using sequence in a storyboard	Pupils use a pre-made storyboard template to plan an animation of their choice. They also review another pair's map of pair programming rules.
12	Let's create	To explain how an interactive animation incorporates the three programming constructs — sequence, repetition, and selection	Pupils celebrate their successes in pair programming. They also program a Scratch animation based on their storyboard from lesson 11.

Each pair was given a printed document called 'Map to successful pair programming' to complete during the twelve lessons (see Figure 1 for an example). Rules were set out for the 'driver' role, the 'navigator' role and for the pair overall.

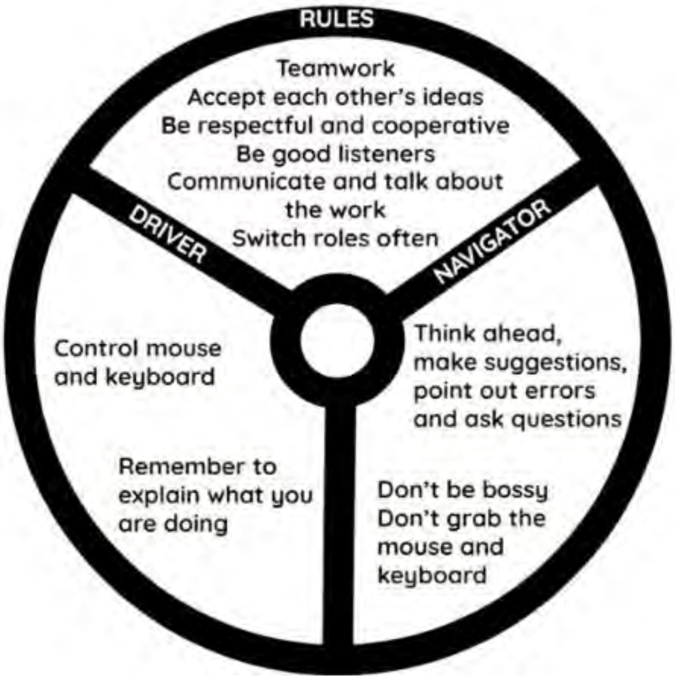

Figure 1. An example of a completed steering wheel with guidance for successful pair programming work

4.3 Teacher preparation

The planned face-to-face training for teachers participating in the RCT was moved to an online format due to Covid-19 restrictions affecting travel and the mixing of households. All teaching staff from schools in the treatment group were required to take a mandatory, 2-hour online training course, and schools could claim back any incurred costs. Professional development and reimbursement were available to all teachers and teaching assistants involved in teaching the lessons to ensure consistent use of the pair programming approach.

The training course was divided into two parts. In part 1, the study was introduced alongside existing research into the effectiveness of the pair programming approach. In part 2, teachers were introduced to the units of work and prompted to explore the teaching materials. A range of resources were used, including a video of pair programming in the primary classroom from the preliminary study. Teachers were then given clear instructions on how to administer the evaluation surveys at the beginning and the end of the 12 week unit.

5. Results

In this section, the findings from the intervention are shared. The first research question sought to explore how girls' attitudes to computing were impacted by introducing pair programming, and is answered by the RCT results presented in section 5.1. We address research question two about teacher and pupil experiences in computing lessons using a pair programming approach through qualitative findings presented in section 5.2. Finally, in section 5.3 we present data relating to our third research question about teachers' and pupils' perspectives concerning the impact of pair programming activities on girls' confidence.

5.1 RCT results

From the 116 schools, the analysis of the RCT used 990 responses of pupil data of attitudes towards computing, and 994 responses of pupil data of intent to study computing. These are reported separately in section 5.1.1. and section 5.1.2

5.1.1 Effect on attitudes towards computing

Results from the SCSAS survey comparing data from pupils in the control and treatment groups are shown in Table 4. A pre-specified multiple imputation model to account for missing baseline or endline data (Kelly et al., 2022) found a difference of 0.046 points (p=0.331) on a 1-4 scale for the intervention, which was not statistically significant at conventional significance levels. While the difference was positive, it was small in magnitude relative to the 1-4 scale and might not represent a meaningful shift in girls' attitudes even if it were significant. A standard error of 0.047 was calculated which indicates a high precision in the sample mean estimate and suggests that the data closely approximates normality.

Therefore, the intervention's impact on girls' attitudes toward computing, as measured by the SCSAS, was not statistically significant compared to the control group.

Table 4. Study Results for Girls' Attitudes Towards Computing

Measure	Control group mean	Treatment group mean	ETE (SE)	*p*
SCSAS score	2.80	2.92	0.046 (0.047)	0.331

Note. *N* = 990. SCSAS = Student computer science attitude survey. ETE = estimated treatment effect. SE = standard error.

5.1.2 Effect on stated intention to study computing in the future

Results for the stated intention to study computing in the future are shown in Table 5. The pre-specified multiple imputation model to account for missing baseline or endline data (Kelly et al., 2022) found a 4.3 percentage points (p=0.453) difference in favour of the intervention compared to the control group, which, although positive, is not statistically significant by conventional standards, and cannot be confidently attributed to the intervention rather than random chance. Therefore, the RCT revealed no conclusive evidence that the intervention positively affected girls' intentions to pursue computing studies in the future, as compared to the control group.

Table 5. Study Results for Girls' Self-Reported Intent to Study Computing in the Future

Measure	Control group mean	Treatment group mean	ETE	*p*
Positive self-reported intent to study computing in the future	45%	49.8%	4.3pp	0.453

Note. *N* = 994. SCSAS = Student computer science attitude survey. ETE = estimated treatment effect.

5.2 Themes from qualitative evaluation study

For the qualitative evaluation, the framework analysis generated four overarching themes relating to the implementation of the intervention, encompassing: fidelity of the paired work, feasibility of the paired work, teacher and pupil experiences of paired working, and mechanisms activated by the intervention.

Fidelity of paired work: Schools in the treatment group largely implemented the pair programming approach as it had been intended. Interviewed teachers generally set up the pairs so that they were mixed-gender, although that was not always possible given the gender balance of the class. All four teachers explained that they spent time carefully pairing pupils according to a variety of different criteria, including matching pairs by computing attainment, creating mixed attaining pairs or using the same pairs as in maths lessons.

"I thought very carefully about pairing the children up from my knowledge of, how they are, their current abilities, where they progressed to in terms of computing." (Teacher: S01)

Within the four case study schools, fidelity to having a driver and navigator with clearly defined roles was also high: pupils understood the different responsibilities that went with the different roles and responded promptly to the indication that it was time to swap roles (normally by changing seats).

"Now they've got used to the system, this is the lesson and this is what we're doing. That's been good." (Teacher: S02)

Feasibility of paired work: Characteristics of the learning environment such as pupil behaviour and pupil familiarity with routines affected the ease with which teachers could implement the intervention. Observations that took place towards the end of the twelve-week unit showed that pupils knew where they should be sitting, who their partners were, which partner was starting in which role, where to find the Scratch project and how to switch places. It is likely that earlier in the unit, teachers spent more time establishing routines and expectations.

“There were a few times...where they weren’t collaborating properly...At the beginning [of the unit] we had to have at least two or three [lessons]....just reminding them what the role of the navigator and what the driver was.” (Teacher: S04)

Overall, comments from the teachers implied that pair programming was well suited to computing because the clearly defined roles provided a strong collaborative environment that supported learning.

“I don't know why I've never thought to do computing like that, actually because it's a really good vehicle for the fact that there are two roles, clearly defined. There's all your conversation and knowledge comes through that, and then they're both equally having a turn." (Teacher: S04)

Teacher and pupil experiences of paired work: Data collected from discussions with pupils and teachers, and observations of lessons, showed that pupils enjoyed working with a partner. For example, a teacher in school S04 explained that *“they [the pupils] really liked working together as driver and navigator. They liked the aspect of swapping around.”*

Observations of the paired work showed that the discussions between pairs often focussed on task planning, including naming the blocks of code to use in Scratch, for example:

Boy: “What do you want him to say? Shall we try..let’s set off?”

Girl: “Yeh but we need a start block first” (Male and female pupils: S02)

Pupils’ comments suggested that using the pair programming approach had been enjoyable, although they also critically evaluated the circumstances that might mean paired work was less effective.

“I like working with both [both in a partner and by yourself] because when you do pair programming you’re collaborating with your partner, making links and you have to tell them what to do. But if you have a really good idea and then they put the wrong thing in the wrong place, it’s quite annoying.” (Female pupil: S01)

“Sometimes when you're in your pairs...it's trickier in different pairs because you don't pick your pairs - they get picked for you. Sometimes it might be easier working in one pair than another." (Female pupil: S02)

All four interviewed teachers stated that following their experiences using the pair programming approach, they intended to use it in the future for other computing lessons. For example,

"Even those who are maybe a little bit more reluctant...those who put their hands up today and said they still prefer to work independently, they are still all engaging quite clearly in that with their pair and doing it really, really well. However much they say they prefer working independently, I think they clearly showed how much they enjoy it, engage with it. And you know they're achieving with it - so we should be doing this." (Teacher: S01)

“I felt like some of my girls were really quite [makes bored noise] at the beginning, and by the end of it, especially when we did the second unit about the animation, loved it. They absolutely loved it.” (Teacher: S03)

5.3 Confidence

We aimed to better understand the mechanisms through which pair programming activities would increase girls’ confidence in programming and so synthesised the data collected from the qualitative evaluation study by the independent evaluators to gather evidence on student and teacher views.

Both interviewed teachers and pupils felt that having the support of a partner boosted girls’ confidence because they could work together to overcome challenges.

"I do think that having that equal time to have a go at both, thinking of the girls I've got, will have helped my girls, because they lack a bit of confidence. They were learning very quickly that actually 'Yes, we are sure. We can do this.’ " (S03)

Girls said that they liked having a partner because *"someone else ... could help me if I needed help"* (S02: female pupil) and that *"if you're stuck your partner can be helpful "* (S01: female pupil).

We had hypothesised that by feeling more confident, girls would engage more in discussions with their partner and increase their subject knowledge. One interviewed teacher made this connection explicitly, explaining that the confidence that girls had through working with a partner enabled them to learn more than if the approach had not been used.

"The pair programming definitely helps. I think it boosted their confidence. They [the girls] had a partner to work with so immediately that makes it more interesting for them. I don't know, I feel like they just acquired more knowledge." (S03: Teacher)

Girls tended to be positive about the collaborative elements of the Pair Programming lessons because of the opportunity to build relationships with peers.

"If someone was your friend you'd make them be more of your friend because you'd be talking with them more, sharing their interest and knowing what they like." (S02: female pupil)

Interviewed teachers had different opinions about whether the intervention led to the same outcomes for both boys and girls. One teacher described how the engagement had increased indiscriminately of gender:

"Nothing stands out in particular. I'm not going to try and kind of conjure something. I'm pleased to say that there's been equal engagement and an equal impact on both male and female." (S01: Teacher)

On the other hand, another teacher did feel that her female pupils had started with lower confidence and so the intervention had particularly appealed to girls.

"I think the girls would have come out better from it, because of their confidence towards the subject. The boys, they liked it, but I feel like the girls were more engaged with it. I don't know if I would have seen that level of engagement from the girls, if it wasn't taught that way, because I do think a lot of mine go in on themselves when they don't know." (S03: Teacher)

6. Discussion

6.1 Explaining the RCT findings

The results of the RCT (RQ1) show that there are no statistically significant results for either of the two outcomes. These results can be explained in a number of ways. Firstly, it may be surmised that the results may not be reliable due to either the impact of the coronavirus pandemic or the design of the study. For example, one interpretation of the results could be that the COVID-19 pandemic, which had an impact on school recruitment and retention, as well as on implementation fidelity, has affected the results; this would imply that conducting the trials again might produce significant results for some of the trials. However, power calculations were used and the trials only went ahead once the number of schools needed had been recruited. In addition, some expected attrition was built into the modelling. Thus, the size of the trial should still be adequate.

Another interpretation is that the endline data might have been collected too soon to show any impact. Collecting data from students a few months later might have given more opportunity for a significant change in attitude and behaviour. Change may still occur for those pupils (and indeed their teachers) involved in the set of interventions. As often in research projects, pressure to conclude the research dictated the collection of only one set of endline data at the end of each intervention: collecting attitudinal and behavioural change later on might have given more opportunity for significant change to be visible.

Without any evidence that the design or implementation of the study was faulty, the study's findings do indicate that including a pair programming methodology alone for 12-weeks does not make a significant difference to girls' attitudes towards computing. It may be that making more changes to pedagogical approaches, as well as for a longer period of time, might be needed to make any noticeable impact. However, the study as implemented has not been able to show any positively significant (or positively negative) results.

6.2 The use of RCTs in education

In some fields, including education policy, the RCT is seen as the `gold standard' of evaluation as randomisation eliminates much of the bias inherent with other study designs (Xiao et al., 2020). The RCT approach, drawn from a positivist view of education, is not universally popular in education, as others arguing that while we need to provide reliable evidence for research findings, the RCT should not be elevated above any other methodology (Morrison, 2020).

In classrooms and schools, there are multiple variables that are difficult to control for, and a more interpretivist lens is commonly used to interpret findings (Torgerson & Torgerson, 2001). In the study described in this paper, models of analysis were carefully developed to control for multiple factors such as whether the school was in a low-income area, the overall performance of the school as judged by an external body, and the baseline scores of the pupils. There will always be other variables that cannot be controlled for; for this reason it is useful to consider the qualitative research as triangulation for the quantitative findings. Indeed even positive RCT results need to be regarded carefully in education:

"*Just because research has shown that such-and-such might `work' in a such-and-such research setting, be it contrived or naturalistic, this is no reason to believe that it will work in a different temporal, locational, contextual setting, or even the same setting, a second time.*" (Morrison, 2020, p. 11).

6.3 Reporting non-significant RCT results

Researchers hesitate to report inconclusive findings in academic publications (Coldwell & Moore, 2024). However, the authors consider that reporting non-significant findings will support further research in this area. Not reporting inconclusive or negative findings can lead to publication bias (Haden, 2019, Coldwell & Moore, 2024). Selective reporting of scientific findings is sometimes known as the file drawer problem (Rosenthal, 1979), describing the tendency of researchers to publish positive results much more readily than negative results, which `end up in the researcher's drawer'. This leads to publication bias, where only positive results are reported in the literature. In this case, it is important to be able to suggest some interpretations of the quantitative results, and explanations for some discord with the (albeit low volume) qualitative results, which will be discussed next.

6.4 Aspects of pair programming emerging from the qualitative evaluation

Despite the inconclusive results of the RCT on girls' attitudes towards and intent to study computing, the qualitative data did identify some mechanisms through which the intervention might have led to the intended effect on girls' attitudes towards computing. This inconsistency may be due to the reliance on pupil-reported quantitative data immediately after the intervention had taken place rather than observable, independent indicators of effect of the RCT. Furthermore, the high stated intention to study computing scores suggest that the sample of participants could have already have had relatively high engagement with computing, thus making it more difficult for the RCT to detect an impact. The qualitative data, although small in scale, gives us another set of data with which to investigate aspects of pair programming and highlight in more detail which mechanisms may have been most effective.

In the case studies, teachers were enthusiastic about using the pair programming approach and provided insights which have implications for teacher professional development and future implementations of pair programming with K-5 learners. Thus the qualitative research indicates that introducing pair programming can effect some changes and reflections for teachers and pupils. These can be described under three headings:

1. Development of Pedagogical Content Knowledge (implementation)
2. Confidence and self-efficacy (teacher and pupil experiences)
3. Collaboration and teamwork (teacher and pupil experiences)

6.4.1 Pedagogical content knowledge

RQ2 sought to better understand how teachers pair pupils together for pair programming activities, In the same way as other studies (e.g. Tsan et al., 2020; Vandenberg et al., 2023), the instructional design of our resources included the freedom to pair pupils by drawing on teachers' prior knowledge and insights into their pupils' behaviours and teachers' existing pedagogical knowledge of classroom management (Magnusson et al., 2002).

This resulted in a variety of criteria for pairing (mixed attainment pairs, mixed gender pairs and reusing pairs from maths lessons) but a high level of fidelity to the pair programming approach across all four schools.

When viewed through the lens of the Refined Consensus Model of Pedagogical Content Knowledge (PCK) (Carlson et al., 2019), teachers used pedagogical reasoning that drew on their *enacted PCK* to pair pupils based on their existing knowledge of what works in other subjects (e.g. mathematics), or their *personal PCK* to create mixed gender pairs based on their beliefs of improving the gender balance in computing. Similarly, teachers also used their *enacted PCK* of embedding classroom routines to help lessons run smoothly. Professional development in the pair programming approach can mobilise teachers' existing *enacted PCK* of successful collaboration strategies and classroom routines to increase the potential for effective implementation.

6.4.2 Confidence and self-efficacy

Findings from the qualitative study showed having a partner's support bolstered girls' confidence, a finding that is consistent with prior research (e.g. Werner et al., 2004). Positive emotions such as confidence towards a subject, coupled with a belief in one's ability to succeed in related tasks, signify self-efficacy (Bandura, 1977). In computing in general, there are gender differences evident when pupils self-assess their abilities, with girls underestimating their performance compared with boys, who demonstrate more accurate self-assessments (Kallia & Sentance, 2018). Our results showed that girls benefited from working with a partner to boost their confidence through social engagement and knowledge-building, rather than comparing themselves with peers. This may be because of the age of the pupils involved in the study, and stereotypes about boys' superior ability in computing have not yet developed. Interviewed teachers and pupils provided evidence to support findings from prior work, that the girls felt more confident engaging with computing and took part in discussions during paired work which led to increased subject knowledge (Ying et al., 2019).

Pupil and teacher comments from our findings suggest that using the pair programming approach may enhance both boys' and girls' self-efficacy. This emergent finding is noteworthy as a strong sense of self-efficacy in computing is connected with pupils' decisions to pursue further studies in the field in both K-12 and undergraduate education (Mishkin, 2019; Aivaloglou & Hermans, 2019). It is crucial that computing teachers are trained in and can use approaches that aim to benefit girls' engagement in computing education and careers to ensure an equitable approach for all pupils, and that these approaches begin in K-5 education, so that all pupils have positive initial experiences of computing.

6.4.3 Collaboration and teamwork

The qualitative findings also showed that girls valued collaboration and teamwork, an additional mechanism for engaging girls in computing lessons that we had not planned to investigate, but which was reported by teachers and pupils during interviews and focus groups. This result is in keeping with general education research indicating females from some Western cultures may be more motivated by a social rather than a competitively oriented learning context (Korpershoek et al., 2021; Hijzen et al., 2006), and builds on sociocultural theories proposing that pupils use social interactions to learn with and from their peers and educators (Vygotsky, 1978). Similar indications of females preferring to learn in a more social and collaborative setting in CS have been inferred from instructional approach research into pair programming (Denner et al., 2014; Liebenberg et al., 2012). This reinforces calls in the field to take a more sociocultural approach in CS (e.g. Guzdial & Tew, 2006; Kafai & Burke, 2013; Faraon et al., 2020; Vrieler & Salminen-Karlsson, 2022).

6.5 Limitations

This study has potential limitations, including the lack of gender-specific analysis to compare boys' and girls' attitudes. This introduces the potential risk of causing a 'backfire' effect, where well-intentioned interventions have adverse outcomes for certain students. Future research should address these by examining gender differences and considering diverse educational contexts. By addressing these areas, future work can develop more inclusive and effective pair programming strategies that promote collaborative learning and student success.

7. Conclusion

In summary, this paper describes a study involving 116 primary schools and 990 female pupils which investigated the impact of a 12-week pair programming intervention on pupil's attitudes to computing and their intention to study it further. The main study (RCT) did not find any statistically significant results in terms of the two outcome measures. However the qualitative evaluation showed that both teachers and pupils enjoyed working in pairs.

We can speculate about the null result from the RCT that while it may be that there is no causal link between using the pair programming approach and an increase in girls' attitudes towards computing, it may also need a longer period of time (greater than 12 weeks) to be evident. This might mean that the use of pair programming is implemented for longer, or that the survey and follow-up evaluation is conducted later on, forming a more longitudinal study. Another suggested explanation is that the pair programming approach needs to be combined with other strategies to achieve a positive effect. Further research could help us in establishing which of these explanations may be most likely. In any case, this research study reaffirms the fact that gender balance in computing is a deeply systemic and societal issue that requires change beyond the classroom to improve.

Although the RCT showed no statistically significant changes in attitudes or intent, qualitative data revealed that the interventions were engaging and enjoyable, with increased confidence and engagement in discussions among the girls. These findings highlight the ongoing societal and systemic barriers in computing education while indicating the positive reception of pair programming among participants.

Acknowledgements

We are grateful to the Department for Education in England for funding this research. Additionally, we would like to thank Julia Ryle-Hodges and the team at Behavioural Insights Team for their partnership on the project, Hayley Leonard, Oliver Quinlan, Emma Posey, Caitlyn Merry and Rik Cross for their contributions towards the project management and design of the intervention resources, and the teachers involved for their active engagement.

References

Aivaloglou, E., & Hermans, F. (2019). Early programming education and career orientation: The effects of gender, self-efficacy, motivation and stereotypes. *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 679–685. https://doi.org/10.1145/3287324.3287358

Bandura, A. (1977). Self-Efficacy: Toward a Unifying Theory of Behavioral Change. *Psychological Review*, *84*(2), 191–215. https://doi.org/10.1037/0033-295x.84.2.191

Beyer, S., Rynes, K., Perrault, J., Hay, K., & Haller, S. (2003). Gender differences in computer science students. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 49–53. https://doi.org/10.1145/792548.611930

Butler, D. (2000). Gender, girls, and computer technology: What's the status now? *The Clearing House*, *73*(4), 225–229. https://doi.org/10.1080/00098650009600095

Camp, T. (1997). The incredible shrinking pipeline. *COMMUNICATIONS OF THE ACM*, *40*(10). https://doi.org/10.1145/262793.262813

Campe, S., Denner, J., Green, E., & Torres, D. (2020). Pair programming in middle school: Variations in interactions and behaviors. *Computer Science Education*, *30*(1), 22–46. https://doi.org/10.1080/08993408.2019.1648119

Carlson, J., Daehler, K. R., Alonzo, A. C., Barendsen, E., Berry, A., Borowski, A., Carpendale, J., Kam Ho Chan, K., Cooper, R., Friedrichsen, P., Gess-Newsome, J., Henze-Rietveld, I., Hume, A., Kirschner, S., Liepertz, S., Loughran, J., Mavhunga, E., Neumann, K., Nilsson, P., … Wilson, C. D. (2019). The Refined Consensus Model of Pedagogical Content Knowledge in Science Education. In A. Hume, R. Cooper, & A. Borowski (Eds.), *Repositioning Pedagogical Content Knowledge in Teachers' Knowledge for Teaching Science* (pp. 77–94). Springer Singapore. https://doi.org/10.1007/978-981-13-5898-2_2

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. https://doi.org/10.1037/a0016239

Coldwell, M., & Moore, N. (2024). Learning from failure: A context-informed perspective on RCTs. *British*

Educational Research Journal, *50*(3), 1043–1063. https://doi.org/10.1002/berj.3960

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, *46*(3), 277–296. https://doi.org/10.1080/15391523.2014.888272

Eccles, J. S. (2015). Gendered Socialization of STEM Interests in the Family. *International Journal of Gender, Science and Technology*, *7*(2), 116–132.

Faraon, M., Rönkkö, K., Wiberg, M., & Ramberg, R. (2020). Learning by coding: A sociocultural approach to teaching web development in higher education. *Education and Information Technologies*, *25*(3), 1759–1783. https://doi.org/10.1007/s10639-019-10037-x

Fisher, A., & Margolis, J. (2002). Unlocking the clubhouse: The Carnegie Mellon experience. *SIGCSE Bull.*, *34*(2), 79–83. https://doi.org/10.1145/543812.543836

Gale, N. K., Heath, G., Cameron, E., Rashid, S., & Redwood, S. (2013). Using the framework method for the analysis of qualitative data in multi-disciplinary health research. *BMC Medical Research Methodology*, *13*(1), 1–8. https://doi.org/10.1186/1471-2288-13-117

Global Education Monitoring Report Team. (2024). *Global education monitoring report 2024, gender report: technology on her terms*. UNESCO. https://doi.org/10.54676/WVCF2762

Graßl, I., & Fraser, G. Equitable Student Collaboration in Pair Programming *2024 IEEE/ACM 46th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, Lisbon, Portugal, 2024, pp. 274-285. https://doi.org/10.1145/3639474.364008

Guzdial, M., & Tew, A. E. (2006). Imagineering inauthentic legitimate peripheral participation: An instructional design approach for motivating computing education. *ICER 2006 - Proceedings of the 2nd International Computing Education Research Workshop*, *2006*, 51–58. https://doi.org/10.1145/1151588.1151597

Haden, P. (2019). Inferential Statistics. In S. A. Fincher & A. V. E. Robins (Eds.), *The Cambridge Handbook of Computing Education Research* (pp. 133–172). Cambridge University Press. https://doi.org/10.1017/9781108654555.007

Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., & Zander, C. (2011). Pair programming in education: A literature review. *Comput. Sci. Educ.*, *21*(2), 135–173. https://doi.org/10.1080/08993408.2011.579808

Hariton, E., & Locascio, J. J. (2018). Randomised controlled trials—The gold standard for effectiveness research. *BJOG: An International Journal of Obstetrics and Gynaecology*, *125*(13), 1716. https://doi.org/10.1111/1471-0528.15199

Hamer, J. M. M., Kemp, P. E. J., Wong, B., & Copsey-Blake, M. (2023). Who wants to be a computer scientist? The computing aspirations of students in English secondary schools. *International Journal of Science Education*, *45*(12), 990–1007. https://doi.org/10.1080/09500693.2023.2179379

Haynie, K. C., & Packman, S. (2017). AP CS Principles Phase II: Broadening Participation in Computer Science Final Evaluation Report. In *Prepared for The College Board and the National Science Foundation*. The College Board and National Science Foundation.

Hijzen, D., Boekaerts, M., & Vedder, P. (2006). The relationship between the quality of cooperative learning, students' goal preferences, and perceptions of contextual factors in the classroom. *Scandinavian Journal of Psychology*, *47*(1), 9–21. https://doi.org/10.1111/j.1467-9450.2006.00488.x

Humphrey, N., Lendrum, A., Ashworth, E., Frearson, K., & Buck, R. (2016). Implementation and process evaluation (IPE) for interventions in education settings: An introductory handbook. *Education Endowment Foundation*.

Kafai, Y. B., & Burke, Q. (2013). The Social Turn in K-12 Programming: Moving from Computational Thinking to Computational Participation. *Proceeding of the 44th ACM Technical Symposium on Computer Science*

Education, 603–608. https://doi.org/10.1145/2445196.2445373

Kallia, M., & Sentance, S. (2018). Are Boys More Confident than Girls? The Role of Calibration and Students' Self-Efficacy in Programming Tasks and Computer Science. *Proceedings of the 13th Workshop in Primary and Secondary Computing Education*. https://doi.org/10.1145/3265757.3265773

Kelly, S., Ryle-Hodges, J., & Bisserbe, C. (2022). *Evaluation of Teaching Approach: Pair Programming intervention.* The Behavioural Insights Team. https://www.raspberrypi.org/app/uploads/2023/02/Gender-Balance-in-Computing-Evaluation-Report-Pair-Programming.pdf

Kemp, P. E. J., Wong, B., & Berry, M. G. (2019). Female Performance and Participation in Computer Science. *ACM Transactions on Computing Education*, *20*(1), 1–28. https://doi.org/10.1145/3366016

Korpershoek, H., King, R. B., McInerney, D. M., Nasser, R. N., Ganotice, F. A., & Watkins, D. A. (2021). Gender and cultural differences in school motivation. *Research Papers in Education*, *36*(1), 27–51. https://doi.org/10.1080/02671522.2019.1633557

Leonard, H. C., Quinlan, O., & Sentance, S. (2021, September). Female pupils' attitudes to computing in early adolescence. In *Proceedings of the 2021 Conference on United Kingdom & Ireland Computing Education Research* (pp. 1-6). https://doi.org/10.1145/3481282.3481289

Lewis, C. M., & Shah, N. (2015). How Equity and Inequity Can Emerge in Pair Programming. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research*, 41–50. https://doi.org/10.1145/2787622.2787716

Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education*, *22*(3), 219–236. https://doi.org/10.1080/08993408.2012.713180

Magnusson, S., Krajcik, J., & Borko, H. (2002). Nature, Sources, and Development of Pedagogical Content Knowledge for Science Teaching. In J. Gess-Newsome & N. G. Lederman (Eds.), *Examining Pedagogical Content Knowledge* (Vol. 6, pp. 95–132). Kluwer Academic Publishers. https://doi.org/10.1007/0-306-47217-1_4

McDowell, C., Werner, L., Bullock, H. E., & Fernald, J. (2006). Pair Programming Improves Student Retention, Confidence, and Program Quality. *Commun. ACM*, *49*(8), 90–95. https://doi.org/10.1145/1145287.1145293

Merriam, S. B. (1998). *Qualitative Research and Case Study Applications in Education. Revised and Expanded from Case Study Research in Education*. Jossey-Bass Publishers.

Mishkin, A. (2019). Applying Self-Determination Theory towards Motivating Young Women in Computer Science. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1025–1031. https://doi.org/10.1145/3287324.3287389

Morrison, K. (2020). *Taming randomized controlled trials in education: Exploring key claims, issues and debates*. Routledge. https://doi.org/10.4324/9781003042112

Porter, L., Bailey-Lee, C., & Simon, B. (2013). Halving Fail Rates using Peer Instruction: A Study of Four Computer Science Courses. *SIGCSE'13 : Proceedings of the 44th ACM Technical Symposium on Computer Science Education : March 6-9, 2013, Denver, Colorado, USA*, 778. https://doi.org/10.1145/2445196.2445250

Ritchie, J., Spencer, L., & O'Connor, W. (2003). Carrying out Qualitative Analysis. In J. Ritchie & J. Lewis (Eds.), *Qualitative Research Practice: A Guide for Social Science Students and Researchers*.

Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological Bulletin*, *86*(3), 638. https://doi.org/10.1037/0033-2909.86.3.638

Torgerson, C. J., & Torgerson, D. J. (2001). The need for randomised controlled trials in educational research. *British Journal of Educational Studies*, *49*(3), 316–328. https://doi.org/10.1111/1467-8527.t01-1-00178

Tsan, J., Boyer, K. E., & Lynch, C. F. (2016). How early does the CS gender gap emerge? A study of collaborative problem solving in 5th grade computer science. *SIGCSE 2016 - Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 388–393. https://doi.org/10.1145/2839509.2844605

Tsan, J., Vandenberg, J., Zakaria, Z., Wiggins, J. B., Webber, A. R., Bradbury, A., Lynch, C., Wiebe, E., & Boyer, K. E. (2020). A Comparison of Two Pair Programming Configurations for Upper Elementary Students. *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 346–352. https://doi.org/10.1145/3328778.3366941

United Nations Development Programme (UNDP). (n.d.). *Goal 4: Quality Education*. https://www.globalgoals.org/goals/4-quality-education/

Vandenberg, J., Lynch, C., Boyer, K. E., & Wiebe, E. (2023). "I remember how to do it": Exploring upper elementary students' collaborative regulation while pair programming using epistemic network analysis. *Computer Science Education*, *33*(3), 429–457. https://doi.org/10.1080/08993408.2022.2044672

Varma, R. (2009). Gender differences in factors influencing students towards computing. *Computer Science Education*, *19*(1), 37–49. https://doi.org/10.1080/08993400902819006

Vrieler, T., & Salminen-Karlsson, M. (2022). A Sociocultural Perspective on Computer Science Capital and Its Pedagogical Implications in Computer Science Education. *ACM Transactions on Computing Education*, *22*(4), 1–23. https://doi.org/10.1145/3487052

Vygotsky, L. S. (1978). *Mind in society: The Development of Higher Psychological Processes*. Harvard University Press.

Werner, L., & Denning, J. (2009). Pair programming in middle school: What does it look like? *Journal of Research on Technology in Education*, *42*(1), 29–49. https://doi.org/10.1080/15391523.2009.10782540

Werner, L. L., McDowell, C., & Hanks, B. (2004). Pair-Programming Helps Female Computer Science Students. *ACM Journal on Educational Resources in Computing*, *4*(1), 4. https://doi.org/10.1145/1060071.1060075

Williams, L., McCrickard, D. S., Layman, L., & Hussein, K. (2008). Eleven guidelines for implementing pair programming in the classroom. *Proceedings - Agile 2008 Conference*, 445–452. https://doi.org/10.1109/Agile.2008.12

Xiao, Z., Hauser, O. P., Kirkwood, C., Li, D. Z., Jones, B., & Higgins, S. (2020). Uncovering Individualised Treatment Effect: Evidence from Educational Trials. *IDEAS Working Paper Series from RePEc*. ABI/INFORM Collection; Publicly Available Content Database. https://doi.org/10.31219/osf.io/8nsw4

Yates, J., & Plagnol, A. C. (2022). Female computer science students: A qualitative exploration of women's experiences studying computer science at university in the UK. *Education and Information Technologies*, *27*(3), 3079–3105. https://doi.org/10.1007/s10639-021-10743-5

Ying, K. M., Pezzullo, L. G., Ahmed, M., Crompton, K., Blanchard, J., & Boyer, K. E. (2019). In Their Own Words: Gender Differences in Student Perceptions of Pair Programming. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 1053–1059. https://doi.org/10.1145/3287324.3287380

Zhong, Baichang, Wang, Qiyun, Chen, Jie, & Li, Yi. (2017). Investigating the Period of Switching Roles in Pair Programming in a Primary School. *Educational Technology & Society*, *20*(3), 220–233. https://www.jstor.org/stable/26196132