# PAIR PROGRAMMING STRATEGIES FOR MIDDLE SCHOOL GIRLS

Linda L. Werner
Computer Science Department
University of California, Santa Cruz 95064, USA
linda@cs.ucsc.edu

Jill Denner and Steven Bean
ETR Associates
Scotts Valley, California 95066, USA
{jilld,steveb}@etr.org

## Abstract

A large gender gap exists with who designs and produces new computer technology; men overwhelmingly dominate the field. Studies have shown that women are more likely to pursue and persist in computer science when they have the confidence to problem-solve and explore without fear of breaking the computer, see social aspects of computing, and see a value to computing consistent with their self identity. Pair programming, developed for use in software design as part of the extreme programming methodology, puts two people working together at all phases of software development. Pair programming is also a collaborative, instructional approach, shown to increase learning, investment, and interest in computer science among both male and female university students, in particular females. We are studying pair programming with middle school girls for the purpose of devising a controlled study of the benefits. This paper will describe how we have implemented pair programming in the middle school environment and what we have learned so far about how this approach may benefit girls. We describe our current strategies so that others may use this collaborative approach both for experimentation and for practical application.

**Key words:** pair programming, middle school.

## Introduction

Currently, there are no gender differences among computer users in the USA [1] but a large gender gap exists among computer producers; those who design and produce new computer technology are overwhelmingly male. The lack of women producers and leaders in technology is partly attributable to early gender differences in interest in computing, resulting in differential participation in higher education. For example, in preschool, boys and girls have a similar level of interest in computers [2], but around fifth grade, boys' interest increases and girls' interest decreases [3, 4, 5, 6]. A study of 2,500 8th-11th grade students in Silicon Valley (USA) found that 23% of females and 42% of males are interested in a technology career [7]. Gender differences continue throughout school into the choice of career. Females and Hispanics are the least likely to be interested in or aware of high tech careers [7]. From 1988 to 1997, the proportion of computer science (CS) bachelor's degrees earned by women in the Silicon Valley local area colleges and universities declined from 33% to 29% [8].

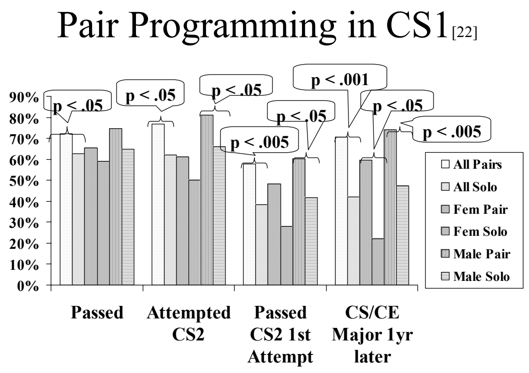
Research cites several factors that might explain the gender gap in STEM (Science, Technology, Engineering, and Mathematics) studies. First, compared to boys, girls are less confident, and less likely to see computers as useful [9]. Second, girls are less likely to pursue and persist in a high tech career because they tend to believe it involves solitary work, entails competition rather than collaboration, and has little social value [10,11,12]. A third factor is how males and females react differently to challenges on the computer—among those with high levels of computer anxiety, females are more likely to drop out of CS courses, while males are more likely to persist [13].

Research also shows that females consistently underestimate their level of computer fluency in relation to men. The US NRC [14] defines computer fluency as the ability to understand and use information technology (IT) productively and to adapt to changes in IT, all of which require a deeper understanding and mastery than simply being computer literate. The perception by women of their lack of computer fluency dissuades them from engaging with computers in ways that would further improve their fluency.

There are many research-based recommendations about how to increase girls' initial participation, and continued retention, in STEM studies. Overall, people agree that increasing girls' skills is not enough [15]; there is a need to change systems to make technology more interesting and relevant to all racial and ethnic groups and to retain boys' interest levels. Some suggest offering hands-on experiences and real world applications, such as linking course material to improving the world [16,17,18]. Others suggest offering computing in a range of classroom settings, thus challenging stereotypes about who computes [10]. Cassell [19] suggests it is important to allow girls to collaborate while constructing a social identity and attaining computer fluency (p. 299).

Pair programming is a promising approach that has not been tested before at the middle school level. It was developed for use in the software methodology called extreme programming by Kent Beck and his colleagues, Ron Jeffries and Ward Cunningham, and puts two people at one computer station [20]. One person acts as the "driver" and works the keyboard and mouse. The other person acts as a "navigator" who guides his or her partner, actively choosing the best methods for coding the software, while simultaneously scanning for errors in the code. After a period of time designated by the partners, these roles are switched. Code that is created by only one of the partners needs to be collaboratively reviewed for inclusion or redone. Pair programming has been used in industry to increase productivity and in education to increase learning [21,22].

Research involving over 500 students at the university level suggests that pair programming holds promise for increasing women's participation in STEM courses and careers. Pair programming in introductory CS courses, CS1 and CS2, resulted in greater course completion and pass rates for all participants. Performance on independently taken exams was just as good for those who worked in pairs as those working alone [22]. The findings were particularly dramatic for women. A key finding is that confidence, satisfaction, enjoyment in the experience of programming, and persistence in pursuing a CS-related major were all higher for those who work in pairs [22]. In fact, pair programming appears to hold promise for closing the gender gap in CS. Three academic quarters after their study, McDowell et al. [22] looked at whether students who had participated declared a major in CS. Significantly more women from the pair section went on to declare majors in CS than those in the solo section. These results indicate that pair programming increases learning, investment, and interest for both men and women, and that these increases are more dramatic for women.



Statistically significant differences are indicated with relevant χ2 p values.

An increasing number of university instructors are trying pair programming. Some are doing so scientifically; others are doing so informally. To date, all of the pair programming studies have some positive impact on enjoyment or performance and none have seen a negative impact (a bibliography on pair programming is at www.cse.ucsc.edu/~brianh/PairProgramBib.html).

There are several reasons to believe that pair programming has the potential to increase computer fluency and enhance a computer identity for middle school students, consequently increasing their participation and persistence in STEM. First, pair programming allows for interactions with a peer, increasing enjoyment and challenging the belief that computing is a solitary activity. It demonstrates the social value of IT by emphasizing the importance of collaboration in working with computers, an importance often unrecognized by women looking at careers in IT. Second, when done correctly, pair programming promotes dialogue about the learning process so that students must articulate concepts about, and take ownership of what they know as they work with their partner to program together. This kind of dialogue has been reported to increase conceptual understanding [23]. Third, pair programming allows participants to learn other students' strategies for programming and problem-solving, and requires students to negotiate ideas and preferences with a partner. Fourth, in pair programming, students learn to share tasks with another person on the computer, a fluency skill that is highly valued in the high technology industry. Pair programming can also help teachers with limited computers to overcome this major barrier to integrating technology into classrooms.

We expect that the benefits will be even greater for girls, as working in pairs directly addresses some of the barriers to female participation in STEM. In addition, when working in partnerships with others of unequal skill, girls are more open to renegotiating the role of the novice as her skill increases [23].

Previous research suggests that students will benefit more from same gender pairs than they do from mixed-sex pairs, and more from self-selected partners rather than randomly chosen partners. Several problems can arise in mixed-gender pairings at this age level. For example, a study of seven and nine year-olds found cross-sex antagonism and found that boys dominated the computer [24]. In another study involving LEGO/Logo projects, researchers found more collaboration within all girl groups than in mixed gender or all boy groups at the middle school level [25]. Research on older students suggests that in mixed sex pairs or teams, men take control of the technology components. As a result, the men are perceived by others as the most skilled, even when they are not [26]. Also, when working with two mice but only one cursor on one computer where transfer of control between the mice needs to be negotiated, the protocol used to gain control of the cursor appeared to be gender dependent. The mice were modified to receive transfer of control messages from the right mouse button. Boys preferred and were much more successful at their given task using a 'take' control passing protocol. The 'take' protocol passed control to whoever pressed the rightmost button. Conversely, it was found that girls preferred and were much more successful at their given task using a 'give' protocol. The 'give' protocol passed control to the partner when the rightmost button was pressed. This research suggests that the mouse control protocol used for successful collaborative work is distinctly different and is gender dependent [27]. Azmitia gives three reasons for using friendship based partnerships. First, in the face of complex tasks, collaboration between friends lasts longer than between acquaintances. Second, friends collaborate using dialogues where they build on each other's reasoning, which improves problem-solving accuracy. Third, friends are more open to increasing the role of the novice in pairs of unequal skill [23]. Self-selected partners will promote friendship based partnerships.

The potential of pair programming is supported by a body of research on university students using pair programming and by a small body of research on girls working together in computing or other technology projects. In a study of 9-13 year olds, pairs who worked together on one computer were more successful at solving computer puzzles and more motivated to continue playing than solo players or children playing on two side-by-side computers [28]. Another study suggests that collaborative learning is useful for helping girls learn complex software. The GREAT project at the Miami Museum of Science has seen small teams of two or three girls work successfully with complex 3D modeling software [29].

## Middle school and pair programming

In this section, we describe our efforts to implement and study pair programming with middle school girls. This research is based in an after-school and a summer program where middle school girls program computer games using Macromedia's Flash. Using Williams and Kessler's paper [30] as a starting point, and observations of the middle school girls in our program and observations of university students practicing pair programming in an introductory programming course, we have identified several behaviors that we believe are essential for effective pair programming at the middle school level. These include: sharing time using mouse and keyboard; sharing decision making; the navigator playing an active role and guiding the driver by pointing; asking each other for clarification and confirming understanding; and using the words "we" and "us" to talk about the decisions and work. We have found that most girls work well paired, but benefit further from basic instruction on pair programming and from support in the form of adult coaching and modeling, and positive feedback from both peers and adults.

In our current project, we use game design and production as a strategy to gain the interest and investment of girls. The girls participate in activities designed to increase their knowledge about high technology careers, and they work in pairs on the computer to create games using Flash, an environment for creating rich multimedia software. Girls use an object-oriented programming language based on the Java programming language that exists within Flash. See http://webtecc.etr.org/gcgweb/protected/login/splash.cfm for the games. We are collecting process data from the girls in the program in order to assess their identity development, problem-solving styles, and experience with pair programming. The data collection methods include interviews, activity sheets, electronic notebook entries, and observations of pair programming.

## Our strategies for pair programming

As discussed earlier, our observations have enabled us to identify some characteristics of effective pair programming. We have also developed, tested, and refined instructional strategies that can be used to train and support children to work in pairs. These include role modeling effective and ineffective pair programming, having girls identify pair programming rules, and publicly recognizing effective pair programmers of the week.

Our role modeling activity consists of two scripts, one each for 'bad' and 'good' pair programming behaviors, 'performed' by the program leaders. We start with the 'bad' script. When done with both, we ask the students to identify pair programming rules. The rules that are identified by the students are built into a large poster, signed by the students, and displayed on the wall in the computer lab for the rest of the program. The scripts follow.

| | |
|---|---|
| A | Okay, we're going to start the first skit. This skit shows bad pair programming, so watch carefully to see what NOT to do. I'm going to start as the driver and B is going to be the navigator. |
| B | NAV*: (NAV is sitting or standing directly in front of the keyboard and mouse. This will make it difficult for DRV to reach.).*Hey A … |
| A | DRV: Hi B… Ummm… B, could I be the driver first today? |
| B | NAV: Okay. |
| A | *DRV: (NAV doesn't move, so DRV has to reach across her body to use the keyboard and mouse.)* All right, **I**'m ready. *(to NA*V) I missed the last session. What do **I** need to build in the game today? |
| B | NAV*: (Stares at the ceiling, looks at nails, watches other people)* **I** dunno… |
| A | *DRV: (Angrily)* God, **you**'re so stupid. *(Reaches across further to get the pair's binder)* |
| B | NAV*: (Still staring at the ceiling, looking at her nails, watching some other girls) (Under her breath)* What*EVER*... |
| A | DRV: Let's see... **I** need to put a button on **my** Welcome page. *(Starts clicking away-she's opening the library and looking at buttons)(To herself)* That one is ugly...That's nice...ugly...cool. So it's the pretty green button or the cool flashing one... **I** like the flashing one. *(Clicks some more, placing the button on the stage(Half to herself)* Wouldn't it be cool if the buttons on **my** game looked like people, so when **I** click them **I** could hear them tell **me** what |

| | |
|---|---|
| | **my** choices are? *(Looks at NAV, then shrugs.)* |
| B | NAV*: (Gets up from her seat and walks over to audience...)* Hey, are you going to the movies tonight? *(Waits for an answer. If girls in the audience don't respond, NAV shrugs her shoulders and says...)* What***EVER***... *(Turns and starts paying a LITTLE more attention to DRV).* |
| A | DRV*: (To herself)* Now **I've** got to put ActionScript on the button to make it go to **my** credits page... |
| B | NAV turned DRV: No **STUPID**, you have to make a new layer first! Here, let **me** do it! *(Leans in and takes over the mouse and keyboard. She adjusts the computer equipment and "boxes out" her partner.)* |
| A | Now the NAV*: (Watches disinterestedly for a while.) (Half to herself)* **I** had an idea for the buttons but **I** can't remember what it was... |
| B | DRV*: (Under her breath)* What*EVER*… |
| | *(Girls are instructed by an adult to get up from seats and switch roles but these two don't)* |
| B | DRV: (*Talking to the adult*) We already switched... |
| A | NAV: But I didn't get my full turn -- |
| B | DRV: What*EVER*… |

| | |
|---|---|
| A | Okay, now we're going to do the second skit, and show you GOOD pair programming. *(Carrying binder, she walks up to her partner and in a friendly tone)* Hi B! |
| B | Hi A. *(Concerned)* Where were you on Tuesday? |
| A | I was sick. |
| B | Bummer. Well, why don't you drive first today. *(Moves over so her partner can sit in front of the computer. The mouse and keyboard are STILL in FRONT of her.)* |
| A | DRV*: (Sits down in front of the monitor.)* Cool. |
| B | NAV: Let's put the keyboard and the mouse where you can reach them, okay? *(She moves the mouse and keyboard over to her partner.)* |
| A | DRV*: (Positions the keyboard in front of her and makes sure that the mouse is on the side AWAY from the navigator.)* |
| B | NAV: Is my chair okay here? I want to show you what I did on Tuesday. |
| A | DRV: Yeah, that's fine. **LET'S** turn the screen so you can see... *(Adjusts screen.)* How's that? |
| B | NAV: Good. |
| A | DRV: Oh, and here, I got our binder out of the closet. *(Hands the binder to B)* |
| B | NAV: Thanks. *(NAV opens up the binder to the flowchart)* Here, **LET'S** open it to the flowcharts for right now... |
| A | DRV: *(Gives an affirmative nod)* |
| B | NAV: ...but help me remember that I wrote some ideas in our design notebook that I want to show you later, okay? *(Alternately pointing at the screen and looking at her partner)* Go ahead and open our game file...no, that folder... yeah. See how on Tuesday I made a new scene for us and called it "welcome," 'cause it's going to be the first scene that someone will see when they play **our** game. *(Alternately pointing at the screen and looking at her partner)* See how I typed in the direction for **our** game? I used the text that they gave **us**. I put it on its own layer and I called it "directions." |
| A | DRV: *(Alternately looking at the screen and at her partner)* How come you didn't just call it "text" like we did before? |
| B | NAV: Well, see, that's one of my ideas. *(Alternately pointing at the screen and looking at her partner)* Look at what I wrote in our design notebook. Since the directions are the same for everybody's game, I thought it would be cool if **we** made our directions *DO* something, you know, like flash different colors or something? I thought it would make sense to put the directions on a separate layer from other text, like the title of **our** game. What do **you** think? |
| A | DRV: I like it. So what are **we** doing on **our** game today? |
| B | NAV: Let me look at the assignment sheet... **We** have to put a button on **our** welcome page. **We**'re going to make it so when someone clicks on the button, it takes them to **our** credits page, where they can see **our** names, because **we**'re the fabulous builders of this game! Let me get out the flowchart about buttons from Tuesday. *( Pointing at the screen then looking at her partner)* Okay, so **we** need to go up to windows and click, no click and hold it down, yeah, like that. Now click on common libraries and then on buttons. Now **we** can click on any of these buttons... No, on the little symbol there... yeah... and **we** can see what they look like. When **we** find one **we** like, **we** can drag it on to the stage |
| A | DRV: Okay, I'm clicking on different buttons to look at them... *(Clicks and shows several buttons.)* How about this one? *(Points to the screen)* It looks just like a button on an arcade game. See the shading, it looks real. |
| B | NAV: I don't like the color. How about the same button, but in red? |
| A | DRV: I like the green one better, but...okay. How do **we** put it on the stage? |
| B | NAV: Just click on it... no click and hold... now drag the button on to the stage. |
| A | DRV: Okay, I'm clicking, holding... dragging... *(Looks at her partner when she asks the question.)* How about here? |
| B | NAV: That's pretty good. **We** can just drop it and we'll move it later. |
| A | DRV: Okay, I'm dropping it -- |
| B | NAV: *(Interrupts)* Wait, wait, wait! **We** forgot to create a new layer! |
| A | DRV: Ohhh, that's right. Okay, I'm inserting a new layer and then **we**'ll name it "button..." |
| B | NAV: Good job! |

| | |
|---|---|
| | *(Girls are instructed by an adult to get up from seats and switch roles. Both get up out of their seats, high five and switch places.)* |
| B | DRV: Okay, now **we** can drag the button on to **our** stage...*(Starts clicking, then stops)* Hey, you didn't get to finish. Do you want to drive a little longer? |
| A | NAV: No, it's okay, I'll get another chance later, or... *(casually)* whatever... |

The following is the list of rules that we have built into the pair programming scripts and into all other instructional materials that we provide for the instructors and the students.

Rules for Pair Programming

1. The driver operates the key board and mouse.
2. The navigator follows what the driver is executing on screen and prevents mistakes. The navigator is in charge of programming language reference materials.
3. Positions: The driver is in front of the keyboard with the mouse in hand. The mouse is positioned on the side of the keyboard furthest away from the navigator. The screen is angled so the navigator can clearly view it. The navigator positions her chair so that she can clearly see the screen and reach it to point. The navigator has reference materials organized on the available table space to ensure easy access.
4. Partners physically get up and move positions when switching roles.
5. Partners pay close attention to each other when pair programming, including: looking at each other when talking and listening carefully.
6. Partners work hard to make sure each person understands what is being created, including: driver checks for agreement on operations before executing on the screen; partners point at the screen to support clear communication; drivers describe what they are doing while executing on the screen; navigators show notes recorded in the design notebook to the driver to check for agreement; if one partner is absent, then in the subsequent session, that partner gets a description of the work completed while she was absent; a partner who was absent during the previous session starts off as driver in the current session which renews absent partner's sense of ownership.
7. Partners are respectful of each other: navigators do not handle the mouse or keyboard; drivers do not grab for reference materials; disagreement is natural and should be resolved respectfully.
8. Partners share ownership over the project.
9. Partners help each other, create opportunities for each other to learn, promote trading off of pair programming roles, share the creation of their project.

## Results and future work

Our initial results suggest that pair programming holds promise for engaging and sustaining the interest of middle school girls in becoming producers of technology. Results from the electronic notebooks and from individual interviews suggest girls saw both the benefits and challenges of working in pairs. One girl described what she liked about working with a partner: "I liked that she'd have ideas too. Like if I couldn't figure something out she could tell me that this was there and that wasn't there. I also liked the fact that if you weren't there on time or not able to figure something out, she could do it."

Students also respond to questions in an online notebook. Included here are two pairs' responses to the question: "Please describe what it is like to work with a partner on the computer. Do you like being the navigator or the driver? Why?" We think these responses are similar to most that we receive.

> "We like working with a partner (duh we're best friends) and since I'm a fast typer and she's quick on the other programming stuff and we both give good directions and listen to each other we like both driver and navigator!"

This previous response was a collaborative response signed by both members of one pair. The following response was separated into two pieces with each student's name after the part they authored.

> "Sometimes it's good to have a partner because they can help you when your lost, but sometimes it can be irratating. I like to be the driver. I chose that because it's fun to be driver." " I also like to be driver. I admit that I'm kind of a mouse-grabber. I do slip up, so I'm glad to have a navigator sometimes, even though I have to share the mouse (gasp!)"

The concept of "intrepid exploration", a term coined by Sherry Turkle [31], has emerged from our current project as a useful way to address the gender gap in STEM. Our data show that the process of pair programming and game design may increase exploration and problem solving on the computer. Self-report data from activity sheets suggest that what girls like least about computers at the start of the program is problem solving. But after working in a pair with her best friend, one girl stated, "I never really used to like computers and now I don't like really hit the computer when I'm mad. I sort of try and find out what's wrong."

What remains to be explored is whether: 1) the effective pair programming behaviors we have observed contribute to improved performance with, and interest in, computers, 2) whether the set of effective collaborative behaviors is the same for boys and girls, and 3) whether these behaviors look the same in a classroom setting as they do in an after school setting. We plan to explore these questions in future research.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants No. 0217221 and EIA 0089989. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We wish to thank Shannon Campe and Audrey Blumeneau for their assistance as the instructors of the program and we want to thank all of the girls of T.N.T.

## References

1. US Department of Education, National Center for Education Statistics, National Assessment of Educational Progress, http://nces.ed.gov, 2000.
2. S.W. Williams & S.M. Ogletree, Preschool children's computer interest and competence: effects of sex and gender role, *Early Childhood Research Quarterly, 7*, 135-143.
3. R. Anderson, *Computers in American schools* (Minneapolis, MN: Minnesota University Press, 1993).
4. B. Holis, Sex related differences in attitudes toward computers: implications for counselors, *The School Counselor, 33*, 1985,120-130.
5. M. Fetler, Sex differences on the California statewide assessment of computer literacy, *Sex Roles, 13*(3-4), 1985, 181-191.
6. K. Chappel, Investigating the impact of elements of educational mathematics software on girls' attitudes., *Journal of Educational Computing Research, 17*, 1997, 119-133.
7. A.T. Kearney, *2002 Workforce study* (Silicon Valley: Joint Venture, 2002).
8. Women of Silicon Valley, Unfinished Business: Women in Silicon Valley Economy, 2001, see http://www.womenofsiliconvalley.org/publications.html.
9. C. Comber, A. Colley, D.J. Hargreaves, & L. Dorn, The effects of age, gender, and computer experience upon computer attitudes, *Educ. Research, 39*, 1997, 123-133.
10. AAUW, *Tech-savvy: educating girls in the new computer age* (Washington, DC: American Association of University Women, 2000.
11. P. Lightbody, G. Siann, L. Tait, & D. Walse, A fulfilling career? Factors which influence women's choice of profession., *Educational Studies, 23*, 1997, 25-37.
12. J. Margolis & A. Fisher, *Unlocking the clubhouse: women in computing* (Cambridge, MA: MIT Press, 2002).
13. L.J. Nelson, G.M. Weise, & J. Cooper, Getting started with computers: experience, anxiety, and relational style. *Computers in Human Behavior, 7*, 1991, 185-202.
14. National Research Council, *Being Fluent with Information Technology* (Washington, D.C: National Academy Press, 1999).
15. B.C. Clewell, & P.B. Campbell, Taking stock: where we've been, where we are, where we're going, *Journal of Women and Minorities in Science and Engineering, 8*(3, 4), 2002.
16. P.B. Campbell, E. Jolly, L. Hoey, & L. Perlman, *Upping the numbers: using research-based decision making to increase diversity in the quantitative disciplines* (Fairfield, CT: GE Foundation, 2002).
17. S. Hansen, J. Walker, & B. Flom, *Growing smart: what's working for girls in school* (New York: American Association of University Women, 1995).
18. V. Lee, Gender equity and the organization of schools, B. Bank & P. Hall (Eds.), *Gender, equity, and schooling*, New York: Garland Publishing, 1997.
19. J. Cassell, Storytelling as a nexus of change in the relationship between gender and technology: a feminist approach to software design, J. Cassell & H. Jenkins (Eds.), *From Barbie to mortal kombat: gender and computer games*, Cambridge, MA: MIT Press, 1999, 298-326.
20. K. Beck, *Extreme programming explained: embrace change* (Reading, MA: Addison-Wesley, 2000).
21. L. Williams, & R. Upchurch, In support of pair programming*, Proceedings of CSE Conference on CS Education*, Charlotte, NC, 2001.
22. C. McDowell, L. Werner, H. Bullock, & J. Fernald, The impact of pair programming on student performance, perception, and persistence, *Proceedings of the 25th International Conference on Software Engineering,* Portland, OR, 2003, 602-607.
23. M. Azmitia, Peer interactive minds: developmental, theoretical, and methodological issues, P. B. Baltes, U.M. Staudinger (Eds.), *Interactive minds: Life-span perspectives on the social foundation of cognition*, New York: Cambridge University Press, 1996, 133-162.
24. H. Fitzpatrick, & M. Hardman, Mediated activity in the primary classroom: girls, boys and computers, *Learning and Instruction, 10*, 2000, 431-446.
25. L. Edwards, A. Coddington, & D. Caterina, Girls teach themselves, and boys too: peer learning in a computer-based design and construction activity, *Computer Education, 29*, 1997, 33-48.
26. J. Wolfe, personal communication, January 30, 2004.
27. K. Inkpen, S. Gribble, K.S. Booth, & M. Klawe, Give and take: children collaborating on one computer, *Proceedings of CHI '95: Human Factors in Computing Systems*, ACM press, 1995, 258-259.
28. K. Inkpen, K.S. Booth, M. Klawe, & R. Upitis, Playing together beats playing apart, especially for girls, *Proceedings of Computer Supported Collaborative Learning,* Hillsdale, NJ: Lawrence Erlbaum Associates, 1995, 177-181).
29. J. Santer, personal communication, February 1, 2004.
30. L. Williams, & R.R. Kessler, All I really need to know about pair programming I learned in kindergarten, *Communications of the ACM, 43*(5), 2000, 108-114.
31. S. Turkle, *The second self: computers and the human spirit* (New York, NY: Simon and Schuster, 1984).