Collaboration and Gender Equity in Game-Based Learning for Middle School Computer Science

Philip Sheridan Buffum and Megan Frankosky | North Carolina State University
Kristy Elizabeth Boyer | University of Florida
Eric N. Wiebe, Bradford W. Mott, and James C. Lester | North Carolina State University

Game-based learning environments can deliver robust learning gains as well as engagement, but they can unintentionally disadvantage students with less prior game experience. A design-based research study investigates the nature of collaboration in the Engage game-based learning environment, measuring the system's effectiveness at supporting all students during computer science learning.

Computing education has recently become a focus of attention among both policymakers and researchers. Even though computing skills are increasingly integral to 21st century jobs, computer science is studied by only a fraction of students in the US, and this fraction typically isn't diverse. To address the national need for a computationally skilled workforce, rigorous computer science learning must go hand in hand with increased participation of students from underrepresented groups. Accordingly, the computer science education research community has identified the critical need to create a classroom climate that fosters student learning and retention for these diverse learners.[1] At the K–12 level, many current initiatives in the US seek to broaden participation in computing, including the development of innovative pre-college curricula such as Exploring Computer Science[2] and the Computer Science (CS) Principles Advanced Placement (AP) course.[3] Fundamental to these initiatives is the mission

 1521-9615/16/$33.00 © 2016 IEEE Copublished by the IEEE CS and the AIP March/April 2016

to engage students who are historically underrepresented in computer science, and also to support learning in a measurable way. An increasingly central element that pervades these curricular innovations is collaborative learning, in which students work together to solve problems.[4]

Evidence suggests that collaborative learning provides many benefits for computer science learners, including improved performance and lower attrition,[5] particularly for women,[6] fewer "stuck" moments while problem solving,[7] earlier application of critical thinking,[8] and the ability to solve problems that might have been just beyond the reach of students individually.[9] Collaboration has become a central and highly valued skill for the 21st century,[10] so much so that efforts are underway to develop collaboration assessment frameworks as part of the CS Principles AP course and within a multinational K–12 assessment program (www.oecd.org).

Our research team has embarked on a middle school initiative in which we've integrated the benefits of collaborative learning with the engaging nature of game-based learning environments. We hope to leverage the benefits of these two strategies to spark interest in computing and also to promote significant learning gains. Over the past three years, we've developed Engage, an immersive game-based learning environment that adapts learning objectives from the CS Principles AP course for the middle school level. In the game, students take on the role of computer scientists as they develop computational thinking skills[11] while solving a mystery. We've iteratively held a series of classroom studies of the game and associated classroom activities in multiple middle schools with diverse student populations, leading to refinements of both the game and the way we deploy it in a larger curricula. Students choose partners at the beginning of the intervention and then play the game in pairs over the course of several weeks.

To measure the intervention's effectiveness, we've administered several instruments during classroom studies, including a novel knowledge assessment that we're in the process of validating.[12] This article examines the results of this knowledge assessment, providing insight to the potential benefits of collaborative gameplay, particularly for girls. In addition to examining overall performance on this assessment instrument, we examine here one exemplar in detail to illustrate our findings: a segment of block-based code that the student must use algorithmic thinking to interpret. In this particular item, the algorithmic thinking centers on the concept of broadcasting, a key concept found in many other K–12 computer science interventions that use block-based programming.[13]

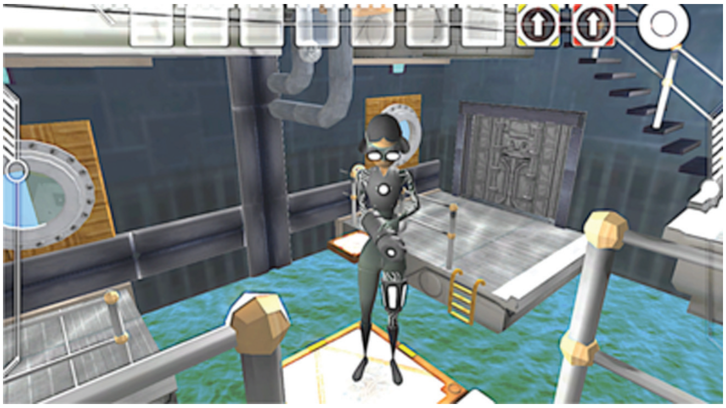

Figure 1. Screenshot of the Engage game-based learning environment.

Pilot Study: Single-Player versus Paired Gameplay

As part of our user-centered development, we conducted an exploratory study in which middle school students played the introductory level of Engage.[14] Students take on the role of a computer scientist sent to an underwater research station to solve a socially relevant mystery. To accomplish their goals in the game, students write programs for various devices to help their avatars advance through the 3D game environment. Figure 1 shows a screenshot of Engage from the segment of the game that students played during this pilot study. The study took place over one week in spring 2014, with each student participating in two hour-long sessions on separate days. Students participated during the school day as part of a regularly scheduled class and were randomly assigned to play the game either individually or collaboratively in pairs.

We designed the pilot study to investigate the impact of collaborative paired gameplay on student outcomes and experiences. Although our project team is keenly interested in all aspects of student learning in the game (such as affective outcomes and computer science attitudes), space doesn't allow for a full discussion of all those results. Instead, we focus on students' use of computational thinking that we assessed through field observations, survey responses, and an early version of our knowledge assessment instrument. We also used the pilot study to refine this instrument, which ultimately became our primary

In effect, we encouraged a style of gameplay similar to pair programming, in which students alternate between being the driver (at the keyboard) and being the navigator (advising the driver).

tool for measuring the learning gains of students in full studies of Engage. For the full classroom study described later, we used the more refined version of the instrument to track students' learning throughout a multiweek intervention. In this current section, we discuss the findings of the pilot study that informed the design of that full classroom study.

Participants

For the pilot, we worked with two seventh-grade middle school teachers and their classrooms in an urban middle school. After we obtained consent and assent, 28 seventh-grade students were randomly assigned to either the paired (N = 14) or unpaired condition (N = 14) and then played the Engage game across the span of two separate gameplay days. The random assignment might have resulted in more male-female pairings than we've seen in other studies (in which students choose their partners): three of the seven pairs for this pilot study were mixed gender. Of the 28 total students, 26 completed a postsurvey on engagement (two students assigned to the paired condition arrived very late and didn't have time to finish the game or postsurvey).

By conducting the pilot study during normal school hours and asking all students in the two classes to participate, we expected to achieve a more representative subject pool than a self-selecting, afterschool study might provide. This strategy proved successful, as the participants included 14 female students and 14 male students. The demographic composition was 7 African-American students, 7 Caucasian students, 6 Latino students, 2 multiracial students, and 1 Asian student (with 5 unknown). Using a survey item that asked, "Have you ever done any activities that involve computer science or computer programming?," we classified 11 students as having prior programming experience and 12 students as not having prior programming experience. Using a survey item that asked, "How often a week do you play computer or video games?," we classified 12 students as frequent video game players (those who responded to the item with "daily" or "almost daily"), and 14 as less frequent video game players.

Task

Once the students were randomly assigned to their workstations, we had them log on to laptop computers and individually complete the early version of the knowledge assessment, which consisted of six multiple-choice questions on programming concepts that we expect students to master in the specific segment of game played in this study. Students in the single-player condition were allowed to begin the game immediately upon completion of this assessment. Students in the paired player condition waited until both partners completed the assessment before briefly receiving instructions on how paired gameplay can work.

When played by two people, the game allows each player to select an avatar to represent him- or herself. Only one avatar is visible in the game environment at any given time, switching at predefined intervals. We encouraged students to alternate who controlled character movement based on which avatar was visible. In effect, we encouraged a style of gameplay similar to pair programming, in which students alternate between being the driver (at the keyboard) and being the navigator (advising the driver). While we did notice some alternative approaches to pair gameplay among students in other grades, the seventh-graders in this study all seemed comfortable adhering to this style.

We designed the study for two hour-long sessions on separate days; no student in either condition finished the game during the first session, so they all had to stop mid-game and resume two days later. During the second session, students completed the segment of game used in the study. Upon completion, they then took the knowledge assessment again, followed by a survey on their game experience, which included items addressing specific game strategies that they might have used while playing.

Results

We captured a rich, multifaceted corpus of data, including survey data, field observations, and learning gains derived from the knowledge assessment.

Student self-reported strategy results. After completing the gameplay session, students completed a survey that included several items concerning their reaction to the game, as well as items on their prior relevant computing experiences and demographic information. In analyzing the data, we looked for differences in game strategies, some of which seemed especially beneficial to students as they used their computational thinking skills to progress through the game environment. Collaboration might lead to wider use of these useful tactics, as students share their "best practices." Without collaboration, students with less gaming experience seem to be at a disadvantage, with results showing them less likely to take advantage of these strategies.

One survey item in particular illustrates this challenge, an item we refer to as test platform. This item asked participants to respond to the following question on a five-point Likert scale: "How often within the game did you test the program for the moving platform without being on the platform?" This item refers to certain locations in the game where students needed to program a moving platform device, which (if programmed correctly) can transport the player's avatar to a desired location. If the player is standing on the moving platform and the platform crashes into an object (the program was flawed), the player will fall off and be forced to repeat some prior gameplay (see Figure 2). This can be frustrating, but it's often avoidable. The beneficial strategy here is to run the program for the moving platform before getting on it. In this case, the player doesn't risk falling off the platform if it crashes.

Figure 2. Test platform. When an error in a program causes the moving platform to crash, the player's avatar will fall off (if it's currently riding the platform). Players can run the program for the moving platform before getting on it so they don't risk falling off the platform if it crashes.

Our results show that female students used the test platform strategy less frequently than male students. Whereas male students responded to the five-point Likert scale item on this strategy with a 3.54 (SD = 1.450), female students responded with an average 2.21 (SD = 1.251). A one-way ANOVA found this to be statistically significant ($F(1, 26) = 6.482$, $p < .05$). Although we can view the test platform strategy as an example of computational thinking, it's interesting to note that no significant differences were found between students who reported having had previous programming experience and those who reported none.

However, while prior programming experience might not have had an influence on how often students used the test platform strategy, prior gaming experience did. Students who reported playing video games daily or almost daily responded to the item on this strategy with a 3.58 (SD = 1.621), while those with less frequent video game experience responded with an average 2.27 (SD = 1.100), a statistically significant difference ($F(1, 26) = 6.302$, $p < .05$). It must be noted that female students reported a lower amount of weekly video game experience, responding with an average 3.21 on a five-point Likert scale, compared to an average for male students of 4.0. A one-way ANOVA found this to be statistically significant ($F(1, 26) = 5.667$), $p < .05$).

Taken together, these preliminary results suggest that students' prior experiences may inform the actions that they take within a game-based learning environment. These prior experiences may help students quickly discover effective game-based learning strategies. If playing alone, an inexperienced student might eventually discover these strategies over time. However, that student might be more likely to do so if collaborating with other students with diverse prior experiences.

Observations of single-player and paired gameplay. Field observations support the claim that students can gain significant benefits when playing collaboratively with a partner. Overall, observations of interactions within the game indicated that students from both conditions had an enjoyable experience, and their postgame comments echoed this sentiment (from a paired team member, "I think that this game was awesome and that I had fun playing with my partner," and from a single player, "I had lots of fun playing it by myself"). The teachers were in attendance for the entire duration and played the game themselves, and commented that the students seemed particularly focused compared to a typical day.

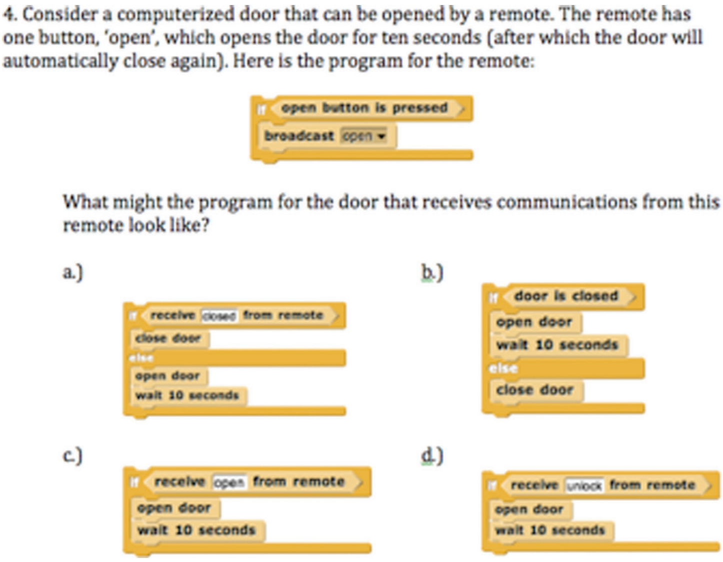

Figure 3. Question 4 on the knowledge assessment (administered on both pretest and posttest). This item primarily addresses the concept of broadcasting.

Throughout the overall gaming experience, observations revealed advantages for paired gameplay versus single gameplay—for example, a given student might use the partner as a "sounding board" or the partner might provide suggestions for what to try next. One student put it succinctly: "I enjoyed working with my partner because he helped me when I was trying to figure the game out." When one member of a pair had difficulty with character navigation (likely due in part to lack of gaming experience), the other partner could take over. Finally, students playing in pairs had the potential to receive superior affective support from their partners—if one student started feeling frustrated or discouraged, the partner could revive the frustrated student's spirits through social collaboration.

We also paid close attention to potential disadvantages. Although single-player students didn't have the advantage of a peer who could provide encouragement and support, they also didn't have anyone criticizing their actions. While most instances were in jest, there was some element of cross-partner frustration. Because the student pairings were random, it wasn't surprising to see that some partnerships led to more successful interactions than others.

Knowledge assessment results. Considering the benefits of collaboration, we hoped to see superior learning gains from those students who played the game in pairs. At the time, we were still developing our knowledge assessment,[12] which students took both before and after playing the game. Because the instrument hadn't been fully validated at this point, we must interpret the score results (which were similar for both conditions) conservatively. An examination of the results from individual questions, however, can provide specific insight into how well the students mastered certain concepts we hoped they would learn. In particular, we were interested in how well they mastered the concept of broadcasting: field observations revealed that many students found the introduction of this concept in the game to be particularly challenging.

Many computer science interventions in K–12 have included broadcasting as a key concept,[13] prompted in part by its prominence in block-based programming languages such as Scratch.[15] Roughly defined, *broadcasting* is a specific type of event in which one device (or, in Scratch's terms, "sprite") sends a message to another device that's listening for some message. Our game features this concept toward the end of the segment that students played in the pilot study. The virtual room in which broadcasting is the primary component requires the student to broadcast the message "open" to the exit door. In solving this challenge, we expect students to first review the read-only program for the door, revealing that it's waiting for the "open" message. Students must then use a pressure pad device that can broadcast messages, modifying the pressure pad's program to broadcast the correct message. Finally, students must run the programs for both the door and the pressure pad, at which point the door will open and they can exit the room. Later challenges in the complete version of the game reinforce this concept with further complexity, but students in the pilot study didn't confront those additional challenges in the segment of the game available to them.

Figure 3 shows the knowledge assessment item that addressed this concept. On the posttest, every student in the single-player condition answered this item correctly, yet only 8 out of 14 students in the paired player condition did. Although based on a small sample, this difference might illustrate one of the potential concerns of a paired gameplay approach: a stronger partner can advance the pair through a challenge without the weaker partner understanding how the challenge got solved. Of the seven partnerships, this problem might have manifested in four, as evidenced by one of the partners answering the

We expected that some students would enter with more knowledge than others. Indeed, even at the middle school level, students can have widely varying exposure to computer science.

broadcast item correctly and the other answering it incorrectly (in one partnership, both students answered incorrectly). Addressing this drawback is a major open question for designing game-based learning environments that support paired gameplay.

Full Study: Learning Gains among Students Playing in Pairs

Having revised the game-based learning environment and the knowledge assessment instrument following the results of our pilot study, we then conducted a study of Engage with two urban middle schools in Raleigh, North Carolina, in fall 2014. In contrast to the pilot study, which we conducted in a controlled environment, we deployed this full study in the context of a quarterly science elective. At each school, a cohort of students attended the elective five days a week during their regular school day. One of the school's full-time teachers taught the elective, with members of our research team attending the gameplay sessions to provide support and record field observations. Over the course of the quarter (approximately two months), several class sessions a week were given for students to interact in the game-based learning environment.

Each student chose a partner on the first day and then collaboratively played the game with that same partner throughout the quarter. This paired gameplay model was motivated by the results from the pilot study and also by logistical concerns. Because of limited technology in the computer labs at the two schools (a common issue in underresourced schools), it would have been infeasible to have every student play the game individually on separate computers. The paired gameplay model thus allows deployment of the game-based learning environment with half as many working computers as there are students in the class.

Participants

Of the 48 students who played the game in pairs during quarter 1 of the elective (and gave consent for their data to be used), 26 were male and 22 were female. The demographic composition was 21 Caucasian students, 13 Asian students, 8 African-American students, 2 Latino students, 1 Middle Eastern student, and 3 classified as "other." On the survey item asking about prior computer programming experience, we classified 11 students as having prior programming experience and 29 as not having prior programming experience. On the survey item asking how often they play computer or video games, 21 students responded "every day" or "almost every day," while the remaining 27 students responded "occasionally" or "almost never."

Task

During the first week of quarter 1, before the initial introduction of the game-based learning environment, students completed the refined version of the knowledge assessment instrument. We used this as a pretest to measure their incoming computational thinking skills. The full test consisted of 23 items that covered the entire gameplay, which is distributed over three distinct game levels. However, since only the first two game levels were deployed for this study (level three was still in development), we report only on the 15 items aligned with the first two game levels.

Students then played the game in pairs during the class sessions scheduled for gameplay (roughly every other day was reserved for gameplay, with the interceding days full of complementary science activities). When a pair finished level one of Engage (which occurred after three to five gameplay sessions, on average), both partners individually completed an interim posttest that included items that we expected students to learn while playing level one.

Upon completion of that test, the pair would then resume gameplay in level two, which both is longer and demands more complex computational thinking, taking students approximately 7 to 10 additional gameplay sessions. After a pair finished this level, they completed the full knowledge assessment instrument as a posttest. For the purpose of this article, we break down the knowledge assessment into "level one content" and "level two content," depending on where in the game we expected students to learn the concept targeted by an

Table 1. Comparison of scores on pretest.

	Performance on pretest	Significance
Gender	Female: .383 (SD = .188) Male: .517 (SD = .227)	$p < .05$
Prior programming experience?	No: .42 (SD = .152) Yes: .63 (SD = .243)	$p < .01$
Frequent video game experience?	No: .385 (SD = .187) Yes: .558 (SD = .225)	$p < .01$
Overall	.458 (SD = .219)	

Table 2. Comparison of computational thinking knowledge at multiple stages.

	Female (n = 19)	Male (n = 21)
Level one		
Pretest (level-one content)	.51 (SD = .282)	.58 (SD = .266)
Interim posttest	.434 (SD = .261)	.691 (SD = .315)
Posttest (level-one content)	.645 (SD = .268)	.726 (SD = .315)
Level two		
Pretest (level-two content)	.36 (SD = .183)	.53 (SD = .261)
Posttest (level-two content)	.63 (SD = .246)	.70 (SD = .212)

individual assessment item. Four items assess concepts introduced in level one, and 11 items assess concepts from level two.

Results

Overall, the results offer compelling evidence of the intervention's effectiveness. We first looked at learning gains of underrepresented students to investigate how equitably Engage serves all students.

Investigation of learning gains for underrepresented students. The computational thinking knowledge assessment addresses our need to evaluate how well the game-based learning environment serves all students. By administering it as a pretest, we were able to assess the extent to which students already had these targeted computational thinking skills. We expected that some students would enter with more knowledge than others. Indeed, even at the middle school level, students can have widely varying exposure to computer science. Moreover, we hypothesized that students traditionally under-represented in computer science would score lower on the pretest than their peers. This article is focused on underrepresentation based on gender, as well as whether there are differences based on students' prior programming or gaming experiences. Table 1 illustrates the differences we found along these three metrics. Overall, students scored an average of .458 (SD = .219) on the pretest (a perfect score would be 1.0), but significant differences were found between female students and male students.

To better understand the difference in pretest scores between male and female students, we ran a one-way ANOVA in IBM SPSS Statistics, and the results showed a statistically significant difference ($F(1, 43) = 4.486$, $p < .05$). The disparity was even greater between students based on their prior experience with programming ($F(1, 39) = 10.456$, $p < .01$) and video games ($F(1, 43) = 7.952$, $p < .01$). Thus, just as we found during the pilot study, that frequent video game experience correlated with more frequent use of the beneficial test platform strategy, our assessment instrument revealed a similar disparity in preknowledge of computational thinking concepts.

Having established that underrepresented students as a whole did indeed enter with less knowledge than their peers, we next compared pretest and posttest scores to examine learning progressions. Table 2 shows the average scores for female and male students on each of these tests, displayed by level.

The results indicate that while female students demonstrated less knowledge early on, they made great gains as they progressed in the game. A one-way ANOVA found the difference between genders on the interim posttest for level one to be statistically significant ($F(1, 39) = 7.735$, $p < .01$). The difference between genders on the pretest of level two content was also statistically significant ($F(1, 39) = 5.193$, $p < .05$). Thus, at this early stage of the gameplay, we don't see the gender gap closing. Indeed, the normalized learning gain here is disheartening. We calculated learning gain as (post – pre)/(1 – pre). When using the interim posttest, this calculation showed the male students as having a higher learning gain (.387 for males, compared to .110 for females). A one-way ANOVA found this to be statistically significant ($F(1, 39) = 5.684$, $p < .05$).

Yet, the longer students interacted in the game-based learning environment, the less these differences manifested themselves. On the posttest, no statistical differences were found between male and female students, as female students' greater learning gains leveled the playing field. Indeed, female students mastered the level-one concepts as they saw them and applied them more often during level two, and this didn't take away from their learning of the level-two concepts. Here we see the importance of persistence. We hypothesize that the collaborative nature of the gameplay better enabled this persistence.

Illustrative example: broadcasting. To better understand these results, it's useful to examine an example of level-one content. As described earlier, the pilot study revealed that the most challenging level-one content involved the concept of broadcasting. We assessed students' understanding of this concept in question 4 on the knowledge assessment. As seen in Table 3, 60 percent of students answered this item correctly on the pretest, indicating that the overall population of students for this study came in with relatively high prior knowledge of this concept, perhaps due to having been previously exposed to block-based programming. Similar to the overall pretest scores for the level-one content (as seen in Table 2), there was a slight gender difference for this particular item, with only 55 percent of female students answering this pretest item correctly, compared to 64 percent of male students.

Table 3. Percentage of students who correctly answered question 4 (see Figure 3).

	Female (n = 20)	Male (n = 25)	Total (n = 45)
Pretest	55	64	60
Interim posttest	50	72	62
Posttest	75	84	80

Students in this study took the interim posttest immediately after completing the same early segment of gameplay that students in the pilot study had played. The pilot study revealed that all students who played this segment individually mastered the broadcasting content, as evidenced by correctly answering the corresponding item on the posttest. Only 57 percent of those who played in pairs did so. In this full study with paired gameplay, a similar 62 percent of students mastered this concept by the time they reached the interim posttest (which corresponds to the posttest in the pilot study).

After taking the interim posttest, students resumed playing the game. The subsequent game content included many more complex challenges that require a base understanding of broadcasting in order to solve them. For example, in learning about how binary numbers can represent letters, students must navigate their in-game avatar to step on a sequence of binary numbers that get interpreted to form a textual message, which is then broadcast to an exit door to open it. A chief concern with using a paired gameplay approach is that a weaker partner might never master basic content, leading to the stronger partner dominating the learning experience. However, as shown in Table 3, students of both genders improved on the basic content as they advanced through the more complex challenges of the game. By the posttest, 80 percent of students correctly answered this item.

Table 3 illustrates that the gender disparity on the broadcasting test item widened from pretest to interim posttest. Whereas the population of male students does show some improvement on this content by the time they take the interim posttest, the rate of correct responses from female students stagnated. However, while female students didn't on average exhibit improvement on the concept

Bringing a collaborative approach to game-based learning environments for computer science thus presents a paradox familiar to researchers who study pair programming: How do we create optimal student pairings?

of broadcasting early on (as evidenced by only 50 percent correctly answering the item on the interim posttest), 75 percent were able to correctly answer this item on the posttest.

Examination of pretest knowledge differences between partners. As mentioned earlier, a major concern for the paired gameplay approach arises when student A has more prior knowledge than student B. The potential exists for student A to then dominate the interaction, solving problems rapidly and leaving little opportunity for student B to explore and learn. With this concern in mind, we sought to examine if and how differences in pretest knowledge between two partners correlated to learning. Unsurprisingly, we found a range of differences between partners. In only three pairs did both partners score exactly the same on the pretest, while the greatest difference between the pretest scores of two partners was .467. To aid this investigation, we defined a new variable, *Difference_Pretest* as (students' pretest score – part-ner's pretest score). Students who scored lower on the pretest than their partners thus have a negative *Difference_Pretest*, while those who scored higher than their partners have a positive *Difference_Pretest*.

A test for correlation between *Difference_Pretest* and learning gains found no statistically significant differences. Students therefore didn't seem to be disadvantaged if their partners had more prior knowledge. In fact, they might have benefitted from having a stronger partner. Classifying each student as either *Negative Difference_Pretest* or *Nonnegative Difference_Pretest*, we found that *Negative Difference_Pretest* students achieved a superior learning gain of .247 (SD = .191) compared to their peers' .113 (SD = .113). A one-way ANOVA found this result to be statistically significant ($F(1, 39) = 4.714$, $p < .05$). It should be noted that a ceiling effect could limit the learning gains of some students in the *Nonnegative Difference_Pretest* category. Regardless, the positive learning gains of the students with less prior knowledge further support the paired gameplay approach, as it addresses the potential concern that such students will be left behind if their partners dominate the learning experience.

Discussion

Game-based learning environments, in practice, call on students to master two forms of competency, in gameplay as well as in the given subject matter. For a well-designed game, students might not even consciously consider the effort they're devoting to mastering gameplay mechanics. Regardless of their prior game experience, a well-designed game will in principle engage them enough so that the challenge of learning the game mechanics don't unduly hinder their learning experience. Yet, to some degree, students who lack prior game experience will inevitably face greater cognitive load when starting a new game.[14] This is a critical issue for designers of games for computer science education, as female students might be less likely to have prior experience with analogous games.

In Engage, the broadcasting example illustrates how this issue can manifest itself. The early segment of gameplay available to the students in the pilot study includes a challenge that explicitly teaches the concept of broadcasting. To broadcast the message "open" from a pressure pad to the exit door, the student must use a crane device to move a box onto the pressure pad. For students who have prior experience with similar games, the mere sight of the crane device, box, and pressure pad informs their next actions. They immediately know to use the crane device to move the box onto the pressure pad, without even reading the read-only program for the exit door (the ultimate objective being to open that door). Having accomplished that subtask, they can then turn their full attention to interpreting the read-only program of the door. Students without prior gaming experience lack this advantage: they have to put more cognitive effort into the gameplay mechanics, taking some of their resources away from the computational thinking.

A paired gameplay approach can perhaps address this challenge, with potential benefits for the novice regardless of the partner's experience with analogous games. When novices have an experienced partner, they can learn from that partner's prior gaming experience. If their partner is a fellow novice, they can learn from watching their partner explore the gameplay in ways they might not otherwise have considered. Additionally, when both partners have low gaming expertise, neither is likely to jump ahead to the computational thinking while the other remains confused about some gaming mechanic. This benefit that collaboration brings to mastering the gameplay mechanics comes in addition to the potential benefits that paired students can gain in learning the subject domain.

On the other hand, negative outcomes might arise depending on the partnership, as in any collaborative experience. Having a partner with more game experience can potentially result in a novice not participating (nor cognitively engaging) as much in the learning experience. In this case, even if the two have similar levels of competency in the subject domain, the novice game player might defer to the experienced game player. Yet, pairing two novice game players together might result in increased frustration if neither student is able to master some game mechanic, leading both students to disengage with the computational thinking aspect of the learning experience. Bringing a collaborative approach to game-based learning environments for computer science thus presents a paradox familiar to researchers who study pair programming: How do we create optimal student pairings?

The full study wasn't an experiment that contrasted a single-player condition to the paired gameplay, so the results don't identify the extent to which collaboration impacted the success of the intervention. In fact, a series of pilot studies have shown that all students—female and male—can benefit from playing Engage individually. Rather than establishing the superiority of any one approach to another, the results reported here lay the groundwork for further study of collaboration in game-based learning environments for computer science education.

From observing students who played the educational game in pairs and those who played it individually, we noted several beneficial aspects of pair gameplay, as well as one or two caveats. When students play in pairs, they can provide each other various types of support, although the quality of this support of course depends somewhat on the two individuals. While the learning benefits of collaboration (as measured by a knowledge assessment) might not manifest in the initial session of gameplay, we saw collaborative gameplay lead to equitable learning gains as students continued playing the game over time. Indeed, whereas female students (and students with less gaming experience) used certain key computational thinking strategies less often than their peers during the introductory level of the game, we observed that collaboration led to a sharing of best practices as time went on. Through this "diffusion of innovation"[16] students achieved significant learning gains regardless of their gender or prior gaming experiences.

Future work should investigate how the combination of educational games and collaboration affects students of other underrepresented groups. Although the study had a diverse pool of student participants, this work hasn't yet examined differences based on race or ethnicity, for example. It's important to do so to improve our understanding of how to create game-based learning environments that are equitable for all learners. Additionally, future work should investigate collaboration in games at a finer granularity by looking at game-trace and multimodal data. This will provide deeper insight into the nature of collaboration and what collaborative strategies lead to equitable learning gains for both partners in a paired gameplay scenario. Finally, future work should explore how pedagogical agents can be integrated into human-human collaboration within virtual learning environments to even more fully support a diverse range of learners with different needs. ■

Acknowledgments
This work is supported in part by the National Science Foundation through grants CNS-1138497 and CNS-1453520. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References
1. L.J. Barker, M. O'Neill, and N. Kazim, "Framing Classroom Climate for Student Learning and Retention in Computer Science," *Proc. 45th ACM Tech. Symp. Computer Science Education*, 2014, pp. 319–324.
2. J. Goode and J. Margolis, "Exploring Computer Science," *ACM Trans. Computing Education*, vol. 11, no. 2, 2011, pp. 1–16.
3. O. Astrachan et al., "CS Principles: Piloting a New Course at National Scale," *Proc. Special Interest Group on Computer Science Education Conf.*, 2011, pp. 397–398.

4. K. Falkner, N.J.G. Falkner, and R. Vivian, "Collaborative Learning and Anxiety: A Phenomenographic Study of Collaborative Learning Activities," *Proc. 44th ACM Tech. Symp. Computer Science Education*, 2013, pp. 227–232.
5. P. Lasserre, "Adaptation of Team-Based Learning on a First Term Programming Class," *Proc. 14th Annual Conf. Innovation and Technology in Computer Science Education*, vol. 41, no. 3, 2009, pp. 186–190.
6. J. Liebenberg, E. Mentz, and B. Breed, "Pair Programming and Secondary School Girls' Enjoyment of Programming and the Subject Information Technology (IT)," *Computer Science Educ*ation, vol. 22, no. 3, 2012, pp. 219–236.
7. B. Hanks, "Problems Encountered by Novice Pair Programmers," *Proc. 3rd Annual Int'l Computing Education Research Workshop*, 2007, pp. 159–164.
8. Z. Li and E. Kraemer, "Social Effects of Pair Programming Mitigate Impact of Bounded Rationality," *Proc. 45th ACM Tech. Symp. Computer Science Education*, 2014, pp. 385–390.
9. L. Williams and R. Kessler, *Pair Programming Illuminated*, Addison-Wesley, 2003.
10. J. Strijbos, "Assessment of (Computer-Supported) Collaborative Learning," *IEEE Trans. Learning Technologies*, vol. 4, no. 1, 2011, pp. 59–73.
11. J.M. Wing, "Computational Thinking," *Comm. ACM*, vol. 49, no. 3, 2006, p. 33.
12. P.S. Buffum et al., "A Practical Guide to Developing and Validating Computer Science Knowledge Assessments with Application to Middle School," *Proc. 46th ACM Tech. Symp. Computer Science Education*, 2015, pp. 622–627.
13. D. Franklin et al., "Assessment of Computer Science Learning in a Scratch-Based Outreach Program," *Proc. 44th ACM Tech. Symp. Computer Science Education*, 2013, pp. 371–376.
14. M.H. Frankosky et al., "Spatial Ability and Other Predictors of Gameplay Time: Understanding Barriers to Learning in Game-Based Virtual Environments," *Annual Meeting of the American Education Research Assoc.,* 2015.
15. J. Maloney et al., "The Scratch Programming Language and Environment," *ACM Trans. Computing Education*, vol. 10, 2010, pp. 16:1–16:15.
16. E. Murphy-Hill and G.C. Murphy, "Peer Interaction Effectively, Yet Infrequently, Enables Programmers to Discover New Tools," *Proc. ACM Conf. Computer Supported Cooperative Work*, 2011, p. 405.

Philip Sheridan Buffum is a doctoral student in the Department of Computer Science at North Carolina State University. His research interests include pedagogical agents, computer science education, and exploring the extent to which digital learning environments are equitable. Buffum received an MS in computer science from North Carolina State University. Contact him at psbuffum@ncsu.edu.

Megan Hardy Frankosky is a research psychologist in the Department of Computer Science at North Carolina State University. Her research interests include game-based learning, user experience research, and educational analytics. Frankosky received an MS in human factors psychology and applied cognition from North Carolina State University.

Kristy Elizabeth Boyer is an associate professor in the Department of Computer and Information Science and Engineering at the University of Florida. Her research interests include natural language dialogue, tutorial dialogue systems, and computer science education. Boyer received a PhD in computer science from North Carolina State University.

Eric N. Wiebe is a professor of STEM education at North Carolina State University and a senior research fellow at the Friday Institute for Educational Innovation. His research interests include STEM learning in technology-rich environments, multimodal communication of scientific and technical information, and research-based strategies for helping schools and teachers maximize the potential of new instructional technologies.

Bradford W. Mott is a senior research scientist in the Center for Educational Informatics at North Carolina State University. His research interests include game-based learning environments, intelligent tutoring systems, and computational models of interactive narrative. Mott received a PhD in computer science from North Carolina State University.

James C. Lester is a distinguished professor of computer science at North Carolina State University and the director of the Center for Educational Informatics. His research interests include intelligent tutoring systems, game-based learning environments, and affective computing. Lester received a PhD in computer science from the University of Texas at Austin.