International Journal for the Scholarship of Teaching and Learning

Volume 9 | Number 1 Article 10

January 2015

Showing Up is Half the Battle: Assessing Different Contextualized Learning Tools to Increase the Performance in Introductory Computer Science Courses

Christine Rolka
crolka@hotmail.com

Anja Remshagen
University of West Georgia, anja@westga.edu

Showing Up is Half the Battle: Assessing Different Contextualized Learning Tools to Increase the Performance in Introductory Computer Science Courses

Abstract
Contextualized learning is considered beneficial for student success. In this article, we assess the impact of context-based learning tools on student grade performance in an introductory computer science course. In particular, we investigate two central questions: (1) does the use context-based learning tools, robots and animations, affect student performance? (2) How do age, gender, and ethnicity impact performance? To explore these questions, we compare the impact of educational robots in conjunction with animations against a second group of students who use only animations, while controlling for the effects of gender and ethnicity. We find that the addition of robots did not improve the students' performance in our setting. Instead, our findings support the existing literature stating that gender and ethnicity are important predictors of student success. We also find that attendance is a strong predictor of student success.

Keywords
introductory computer science, contextualized learning, attendance, educational robots, animations

Introduction

For most students at the University of West Georgia, Introduction to Computer Science (CS1300) is their first computing experience beyond computer literacy, and has a high failure rate. Over the past three years the failure rate averaged at about 53.3%. For example, in Spring 2013, only 37% percent of enrolled students passed the course with a letter grade C or higher. Unfortunately, such high failure rates in introductory computer science courses are common in Georgia and beyond (McCracken et al., 2001).

Given the increasing importance of Science, Technology, Engineering and Mathematics (STEM) in the United States, and the emphasis on retention, progression, and graduation in Georgia, a number of innovative practices were introduced at the University of West Georgia in an effort to increase the number of Computer Science (CS) graduates. Those practices include pair programming, which has been shown to increase student performance in CS introductory courses (Braught, G., Eby, L. M., and Wahls, T., 2008), and a context-based tool that teaches games and animations. More recently, educational robots have been developed and used in computer science education.

The use of animations and robotics in an introductory computer science course has the advantage that students can observe the execution of their code in action. This in turn might help them to better understand programming concepts, to test their programs, and to find errors in their code. Animations and robotics bring the abstract to the concrete. We expected that the use of multiple learning tools would appeal to a larger group of students. In Fall 2011 we decided to add a robotics component to CS1300 in addition to animations, which resulted in a significant increase in the passing rate for that semester. The overall passing rate rose eight percentage points from 50% to 58%.

To test if the addition of the robot was a factor in the rising passing rate, we repeated our work in Spring 2013. In this new study, we explore two central questions: (1) does the use of context-based learning tools, robots and animations, affect student performance? We hypothesized that the addition of educational robots helps students make abstract programing concepts more concrete. We also expected that students were more engaged in class and hoped to see an increase in engagement reflected in an increase in attendance. We conjectured that an increase in engagement ultimately improved performance. (2) Female and minority students are underrepresented in Computer Science. For instance, only 14.2% of the bachelor's graduates in computer science in 2012/2013 were female and only 3.8% were black or African American students according to the most recently published Taulbee survey (Zweben, 2014). Thus many interventions to improve retention, progression, and graduation in computer science focus on increasing the participation of these underrepresented groups. Therefore, we test the extent to which demographic factors affect the performance of students that participated in the study. How do age, gender, and ethnicity impact performance?

This paper provides a qualitative and quantitative analysis of the impact of the combined use of robots and animations. We present data to shed light on the relationships among the learning tool, attendance, demographic factors, and students' performance.

Literature Review

Studies show that when the instructor introduces new material and concepts in context, students are more motivated (Savin-Baden, 2003; Savin-Baden & Major, 2004). As a result, many computing educators have experimented with contextualized learning in introductory computer science classes in recent years. In particular, several studies show that games and animations raise student participation, and are therefore frequently used in computer science education to implement a context-based learning environment (Bayliss, 2009; Kölling & Henriksen, 2005; Leutenegger & Edgington, 2007; Schuster, 2010; Sung, Panitz, Wallace, Anderson, & Nordlinger, 2008). These tools provide a framework in which students can write code and immediately observe the execution. In this way students come to understand abstract programming concepts. Students will typically begin by writing small interactive games and animations. Example learning environments for animations

https://doi.org/10.20429/ijsotl.2015.090110

include Alice (Cooper, Dann, & Pausch, 2003), Greenfoot (Kölling, 2010), and Scratch (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010).

Another context-based learning tool that has gained significant attention recently is the use of educational robots. Examples include an adapted version of the Scribbler robots developed at the Georgia Institute of Technology and Bryn Mawr College (Balch et al., 2008); the Finch robot developed at Carnegie Mellon (Lauwers and Nourbakhsh, 2010); and the Lego Mindstorms ("Lego Mindstorms," 2014), used for many years in the classroom at all levels – at middle and high schools, and at the college level (McWhorter and O'Connor, 2009).

Evidence shows that the use of robots in introductory CS programming classes has an impact on student motivation and learning. Furthermore the nature of robots as physical objects enables students to witness the execution of their computer code, and hence robots are considered a useful teaching and learning tool (Balch et al., 2008; Chen & Mahadev, 2012). However, there is controversy as to the impact of robots on student success, particularly because some studies have found a positive impact (Imberman & Klibaner, 2005); others have found a negative impact (Fagin & Merkle, 2003), and even mixed results where robots seem to engage only some students (McWhorter & O'Connor, 2009).

Interventions

CS1300 introduces students to object-oriented programming concepts. Students use an animation tool to implement simple games, animations, and simulations. In the following text we refer to these types of applications as simply animations. Starting in Fall 2011 we included educational robots as a supplement to animation exercises in all CS1300 sections. In the Spring of 2013 we formally assessed the use of educational robots. We describe first how the course was conducted in Spring 2013 followed by a description of the robotics component.

The spring semester 2013 of CS1300 enrolled a total of 87 students among four studio sections. Each studio section had a maximum of 24 students to maximize instructor-student interaction, feedback and support. All sections met once a week for a 50-minute lecture where new concepts were taught. Studios met separately twice a week for 80-minutes each. During the studio session, students engaged with hands-on activities to apply the new concepts taught in lecture. In keeping with best practices, instructors and teaching assistants helped students during the studios and outside class in a tutoring lab. The tutoring lab was staffed with advanced undergraduate and graduate students who were also available online.

Students programmed using Greenfoot. The Greenfoot software was developed at the University of Kent in Canterbury in the United Kingdom for educational purposes allowing novice programmers to implement animations and games (Kölling, 2010). Greenfoot is an Integrated Development Environment (IDE) and enables programmers to easily test and view the results of their code. The IDE enables students to write computer code in a supported environment by highlighting keywords and syntactically important regions, providing documentation so they know what the program can do, and detecting syntax errors.

In Spring 2013, we alternated the use of animations with educational robots. We chose the Finch robot for its relatively low cost, hardware durability, and compatibility with the Greenfoot IDE. The Finch robot was developed at Carnegie Mellon for educational purposes (Lauwers, Nourbakhsh, & Hamner, 2009; Lauwers & Nourbakhsh, 2010). We had also experimented with the use of Scribbler robots before. When introducing the Scribbler robots, students used two different development environments to implement programs, Greenfoot for animations and BlueJ for the Scribbler robots. However, the switch between IDEs within a class offering confused the students significantly. To minimize confusion, in Spring 2013 students used the Greenfoot environment to write all programs – for both the animation and robot studios.

To compare the effect of robotics on student performance, we divided the four studios in Spring 2013 into two groups: two studios used Greenfoot animations only, and the two other studios used the Finch robots in addition to Greenfoot animations. Apart from the studio exercises, the students in both groups were given the same projects, tests, and quizzes. In the robots-and-animation group nine studio meetings were dedicated to animations and five meetings were dedicated to the Finch robots. Students in both groups were encouraged to work in teams of two on their in-class exercises. To further encourage collaboration and teamwork, two students had to share a Finch robot. However, some pair groups decided to work individually on the implementation and just shared the robot for testing purposes.

Typically, students worked on one hands-on exercise during a studio session and most would complete an exercise within two studio periods. Each exercise focused on a particular concept. For example, one of the first exercises in the class focused on writing an algorithm and translating the algorithm into code. The robot sections were asked to create a small dance for their robot to perform while the animation group completed a similar task for an animated object within the Greenfoot world. Each group was given a list of requirements that had to be met in order to successfully complete the exercise: (1) starting and ending points, (2) a minimum length for the program, (3) approval of hand-written design, and (4) dance characteristics, such as changing colors of either the robot's beak or the world background of an animation. The concept was the same for both studios.

Data

For this study, we collected a variety of quantitative and qualitative indicators. Among them: a focus-group interview with students, grade data (tests, studios, quizzes, projects), demographic data (age, gender, and ethnicity), and attendance. Below we discuss in more detail the process of collecting and assessing student performance.

Focus-group Interview

At the end of Spring 2013, two groups of students were interviewed in order to gather information about their attitude towards using the different tools. One group was composed of 10 students of a robot-animation section, and the other group consisted of 6 students of an animation-only section. Questions related to students' perception regarding the helpfulness of the tool(s) to master the course material. An instructor who did not teach any CS1300 studio in Spring 2013 conducted the interview. The interviews were audio recorded.

Grade Data

Throughout the semester we administered four tests, and collected twenty-two non-test items. The final grade was calculated based on a weighted average: 4 tests (50%), 3 projects (20%), 14 studios (10%), 5 quizzes: (10%), and attendance (10%). Our analysis used both the course grade and test grades to measure performance in two ways. First we assessed student performance based on course grades because the course grade determines the retention rate for the course. Our second analysis looks at students' raw test scores to assess student learning. We believe that the test grades reflect student learning more accurately than the course grade because attendance factored into the final grade calculation and due to the level of support provided for non-test items. We did not include non-test items to measure performance because students received significant support to complete studios and quizzes. Although projects were designed to measure a student's ability to solve problems on their own, these were take-home assignments, which afforded students the opportunity to use any resource available to complete them.

Attendance

We are not aware of any literature that details the relationship between attendance and performance in Computer Science courses. We suspected, however, that low levels of attendance might have an impact on student performance because the relationship between attendance and performance has been investigated in other disciplines (Arulampalam, Naylor, & Smith, 2012). Moreover, it is well known that female students are generally more engaged, have higher attendance rates and earn higher grades than male students (Kinzie et al., 2007; Conger & Long, 2010).

The instructor took attendance at each class meeting. There were a total of 27 scheduled class meetings. To account for excused absences, two studio sessions were automatically dropped for all students. In the calculation of students final grade, attendance is the ratio of the total number of times attended to the number of class meetings. In our analysis, cumulative attendance is the count of all class meetings attended by each student. Interval attendance counts the number of times a student was present in class prior to each test.

Results

First we discuss the results of the qualitative data and compare the use of educational robots and animations with the use of animations only.

Before the introduction of robots, studio instructors did not enforce teamwork consistently. Students stayed seated at their individual desktop computer and did not pay attention to their surroundings. Even though students were encouraged to cooperate in teams of two when working on animations, the classroom was often very quiet and students hesitated to speak up.

After introducing robots into our studio sessions, we observed some positive results in the classroom: students were more engaged in their programming exercises than before we introduced the robots, and the classroom atmosphere was more collaborative. Their higher level of engagement can be attributed to several factors: (1) students had to move physically in the classroom to check out their robot and to test their program. (2) they could easily observe the robots of their peers in action, often resulting in comments about each other's solution. (3) the novelty of the application stimulated students further to share their experience with their peers. (4) teamwork was fostered due to limited hardware resources. Once students had found a partner, they tended to collaborate with that student on animation exercises as well.

Two students expressed a positive attitude towards the Finches in the interview (*"The Finches were cool."* and *"I liked working with the Finches"*). But the majority of students in the interviewed group preferred animations.

Students generally liked the fact that they can see their program in action when implementing an animation, game, or robotic program. A student in the interviewed group pointed out: *"Working with [animations] was alright. I mean it was kind of fun sometimes. You know, to be able to see your final [product] once you got done with an assignment or project . . . that you have kind of created."*

The interviewed students did not perceive robots as more difficult than animations. For example, a student commented: *"For me, it was probably about the same."* However, many students complained that, compared with animations, the robotic programs are cumbersome to test. Animations can be tested and observed instantly. In case of the robots, the robot needs to be connected to the computer, the program has to be uploaded from the computer into the robot's memory, and finally executed. Although this process happens within seconds, it introduces several points of possible failure and therefore appears to increase student frustration.

Another source of student frustration was frequent programming environment crashes. Students, for example, complained that: *"The Finches were kind of useless, because the stuff you put into a computer you can just as easily watch move around on the screen … so I feel like it could be just as easily be a picture on the screen than [Finches] move around."* Another student complained that *". . . the Finches are cool like maybe for a class presentation, maybe for a lecture. But for the activities, if it was on screen, like you would type in the code forever and we saw on the computer how it moved, would help us a lot better than having the actual robot and then, even if you put the code correctly, you could still mess up because of the wheels or something like that or the Finch robot would stop working."*

Next we discuss the results of a quantitative analysis of student performance. We explored the effects of the interventions on student retention as measured through the final letter grade. We consider A, B, and C as passing grades; and D, F, W (withdrawal in the first half of the semester), and WF (withdrawal in the second half of the semester) as failing grades. In Spring 2013, 37% of the students passed the class; of that 53% were females.

We conducted an analysis using ordinary least squares (OLS) regression. The central goal of this analysis is to explore "as far as possible with the available data how the conditional distribution of [grades vary] across subpopulations determined by the possible values of [age, gender, ethnicity, and attendance]" (Cook & Weisberg, 1999).

Table 1
Linear regression (OLS) analysis of grades Spring 2013

	Model 1 (Course Grade)	Model 2 (Test Grade)
Age	0.880*	0.922***
	(0.343)	(0.228)
Gender	2.360	6.569**
	(3.897)	(2.244)
Ethnicity	0.286	0.0915
	(0.404)	(0.259)
Intervention	-3.876	-2.594
	(3.772)	(2.236)
Interval Attendance		2.021**
		(0.624)
Cumulative Attendance	3.143***	
	(0.322)	
Intercept	-15.28	40.46***
	(-10.52)	(5.997)
N	66	265
R^2	0.643	0.093

Standard errors in parentheses
* ρ<0.05 ** ρ<0.01 *** ρ<0.001

Table 1 shows the results of two different linear regressions. The dependent variable in Model 1 is the student's course grade and in Model 2 the test grade; and both models test for the influence of variables the literature suggests explain variation in grade performance (Margolis & Fisher, 2001), as well as test for the influences of the intervention type (robot vs. animation) and attendance on performance. In order to see the impact on student's grades, we differentiate attendance by cumulative and interval counts. We measure attendance as a cumulative count for the entire semester in Model 1, and as an interval count before each test on Model 2. Figures 1a and 1b are visual representations of Model 1 and Model 2 respectively. The dashed line indicates the confidence interval in terms of age and cumulative attendance. The red line marks the lowest and highest grade in case of gender, ethnicity, and intervention.

The results from Model 1 (Table 1 and Figure 1a) below show that controlling for other factors, age has a positive and statistically significant impact on course grade ($\beta=0.880$, $\rho<0.05$). It also shows that although gender and ethnicity are positively related to course grade ($\beta=2.360$, $\beta=0.286$ respectively), they are not statistically significant. This is an interesting finding because it contradicts much of the existing literature that finds statistically significant effects of gender (being female) and ethnicity (being Black or White) on performance (U.S. Department of Education, 2013).

One of our central goals in this paper was to explore the impact of the intervention (multiple context-based tools including robots and animations) on student performance. However, we find that the intervention does not affect student performance in either model. Although the intervention appears to have a negative impact on performance, that is sections treated with animations performed slightly better than sections treated with robots-and-animations as shown in Figures 1a and 1b below, those differences are not statistically significant.

In previous semesters we did not maintain careful attendance records, but we had informally observed attendance decrease as the semester progressed. The results from Model 1 indicate that attendance does have a positive and statistically significant impact on course grades ($\beta=3.143$, $\rho<0.001$).

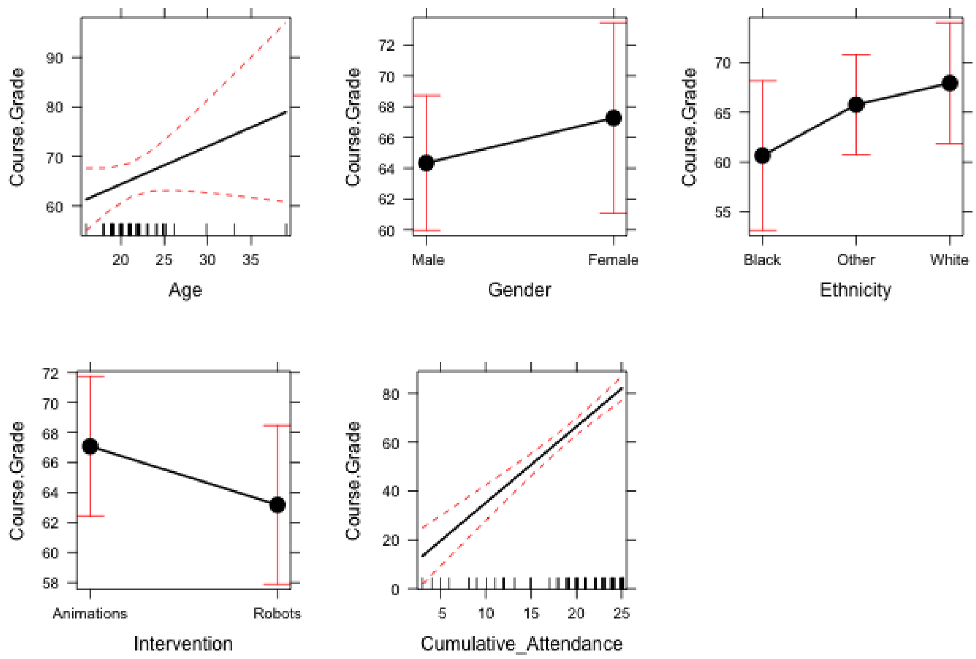

Figure 1a. Plot of all effects on Course Grade (Model 1)

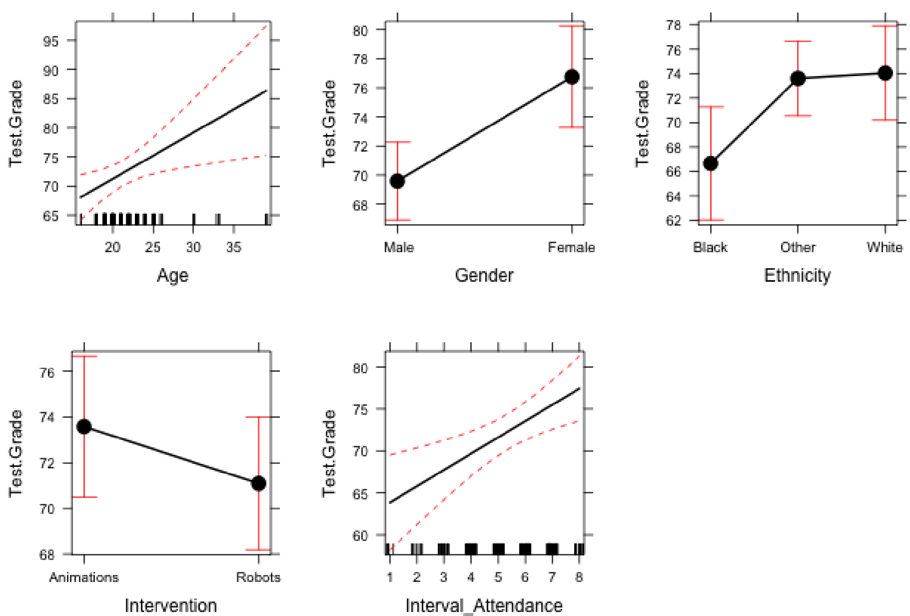

Figure 1b. Plot of all effects on Test Grades (Model 2)

Essentially, this tells us that holding all other variables constant, a one-unit increase in attendance increased a student's course grade by slightly over 3 percentage points. This means that the final grade of a student who attends, for example, 20 class meetings is 6 percentage points higher on average than the grades of those who attend only 18 class meetings. Having perfect attendance did not ensure grade increase, as we will see below, but it provided the opportunity to earn a better grade – hence, showing up is half the battle.

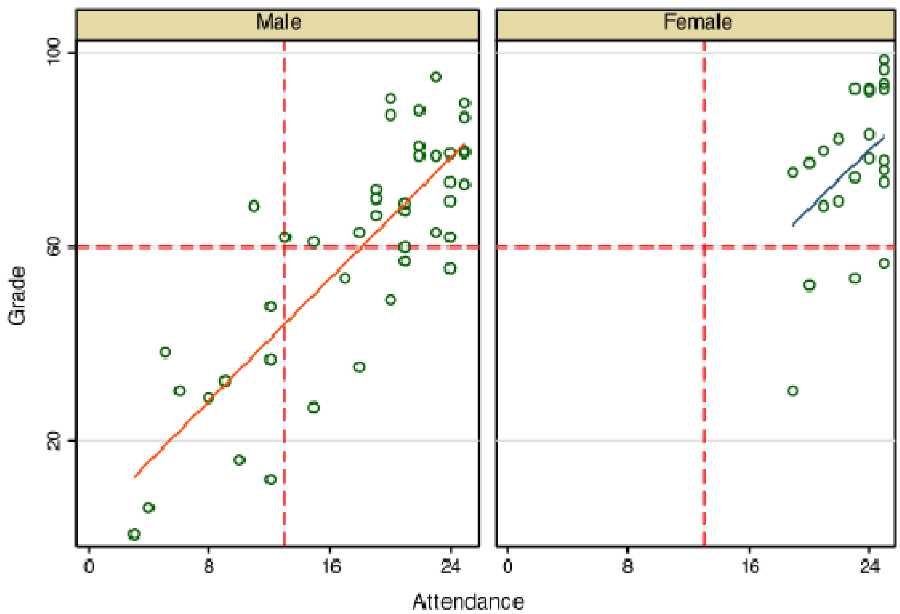

Figure 2. Scatterplot of course grades over attendance by gender.

To better illustrate the impact of gender and attendance on course grade, Figure 2 above represents grade scores plotted against attendance by gender. Circles represent students. Superimposed are: (a) the regression line (solid) for Model 1 above, (b) a horizontal line (dashed) at grade value 60 which represents the minimum threshold to pass the class, and (c) a vertical line (dashed) at attendance value 13 which represents the mean attendance score. Clearly, females attended at higher rates than males, which helps to explain the gender differences in performance. Again, being female did not ensure a higher grade, but because females attended class at higher rates than males they had more opportunities to practice their coding skills, clarify any confusions, ask for help, and ultimately perform better. The results from Model 1 raise a very important question: why did females attend class at higher rates than males? The answer might correlate to the reasons why women are more

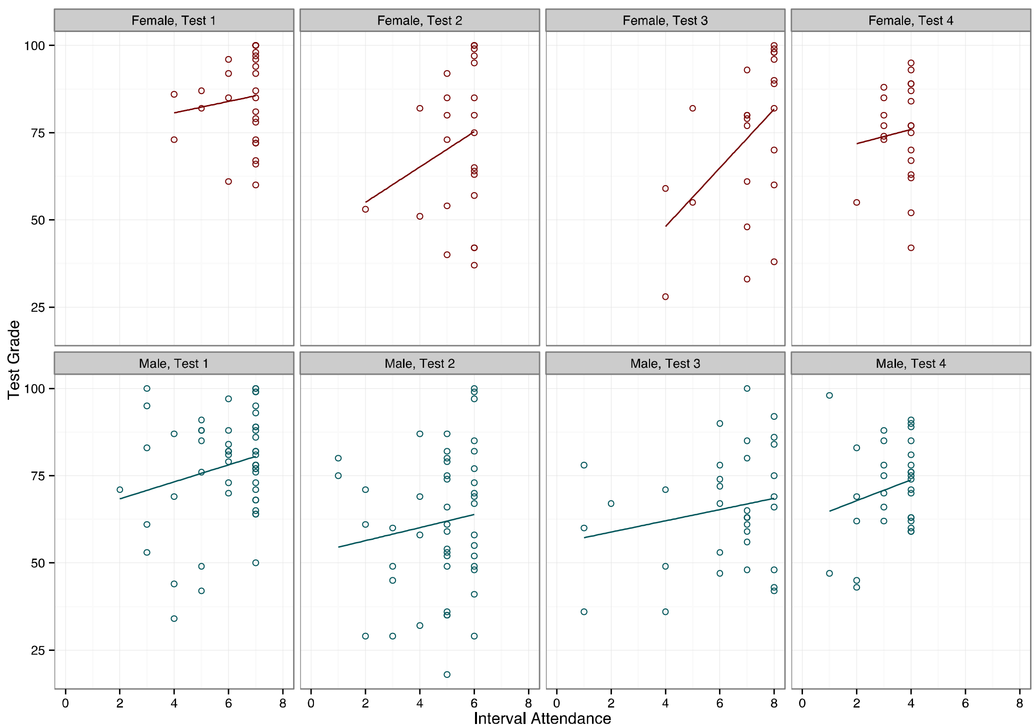

Figure 3. Scatterplot of test grades over attendance by gender

The above results might be affected because attendance is 10% of the course grade. Therefore, we developed a second model that looks at the impact of gender, ethnicity, and attendance on students' test grades. The results from Model 2 indicate that gender, in this case being female, has a positive and statistically significant impact on test grades ($\beta=6.569$, $\rho<0.01$). It also shows that ethnicity, in this case being white or black, has a positive impact on test grades ($\beta=0.0915$) but, as in Model 1, it is also not statistically significant. Again, this is an interesting finding because it contradicts the existing literature.

It is possible that this is an artifact of the small sample (white=24; other/unknown=44; black=19). As in Model 1, Attendance continues to have a positive and statistically significant impact on test grades (β=2.021, ρ<0.01). This is an important finding because regardless of how attendance is measured, it remains influential.

To help us explore why the gender coefficient is larger and significant in Model 2, but not in Model 1, Figure 3 above shows a scatterplot of test grades over attendance by test item and gender. The maximum attendance interval count for each test is 7 for test one; 6 for test two; 8 for test three; and 4 for test four. Again, this plot shows females attended consistently throughout the semester, but not males.

Conclusion

As far as student performance is concerned, our analysis shows that the intervention type used to teach CS to introductory students at the University of West Georgia does not make a difference: students perform just as well with either robots or animations. Students prefer animations to robotics due to the technical difficulties with robots and because animations provide quicker feedback and less opportunities for hardware failure when testing their code. However, the use of robotics did have a positive impact on the classroom atmosphere.

The results show that gender does not have a direct impact on student course grade performance. Instead, gender has an indirect effect on performance through attendance. Hence, it is particularly important to get the students into the classroom. Future work should pay more attention to the interaction between gender, ethnicity, and attendance to determine whether their effects hold across semesters. Of course, showing up to class does not guarantee that a student will pass; but it offers students the opportunity to participate in class exercises and discussions, and to fully engage with the material – and that is half the battle.

References

Arulampalam, W., Naylor, R. A., and Smith, J. (2012). Am I missing something? The effects of absence from class on student performance. *Economics of Education Review*, *31*(4), 363-375.

Balch, T., Summet, J., Blank, D., Kumar, D., Guzdial, M., O'Hara, K., Walker, D., Sweat, M., Gupta, G., Tansley, S., Jackson, J., Gupta, M., Muhammad, M. N., Prashad, S., Eilbert, N., and Gavin, A. (2008). Designing Personal Robots for Education: Hardware, S*oftware, and Curric*ulum. *IEEE Pervasive Computing, 7*(2), 5-9.

Bayliss, J. D. (2009). Using Games in introductory courses: tips from the trenches. *Proceedings of 40th SIGCSE Technical Symposium on Computer Science Education*, 337-341.

Braught, G., Eby, L. M., and Wahls, T. (2008). The Effect of Pair-Programming on Individual Programming Skill. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, 200-204.

Chen, X. and Mahadev, N. V. R. (2012). Enhancing the Undergraduate Teaching and Research Using Robotic Programming. *Journal of Computing Sciences in Colleges*, *28*(2), 57-64.

Conger, D. and Long, M. C. (2010). Why Are Falling Men Behind? Gender Gaps in College Performance and Persistence. *The ANNALS of the American Academy of Political and Social Science 627*(1), 184-214.

Cook, R. D. and Weisberg, S. (1999). *Applied Regression Including Computing and Graphics*. Wiley Series in Probability and Statistics Texts and References Section. New York: Wiley.

Cooper, S., Dann, W., and Pausch, R. (2003). Teaching objects-first in introductory computer science. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, 191-195.

Fagin, B. and Merkle, L. (2003). Measuring the Effectiveness of Robots in Teaching Computer Science. *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education*, 307-311.

Imberman, S. P. and Klibaner, R. (2005). A Robotics Lab For CS1. *Journal of Computing Sciences in Colleges, 21*(2), 131-137.

Kinzie, J., Gonyea, R., Kuh, G. D., Umbach, P., Blaich, C., and Korkmaz, A. (2007). *The relationship between gender and student engagement in college.* Paper presented at the 32nd annual conference of the Association for the Study of Higher Education, Louisville, KY.

Kölling, M. (2010). The Greenfoot Programming Environment. *ACM Transactions on Computing Education, 10*(4), Article 14.

Kölling, M. and Henriksen, P. (2005). Game programming in introductory courses with direct state manipulation. *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*, 59-63.

Lauwers, T., Nourbakhsh, I., and Hamner, E. (2009). CSBots: Design and Deployment of a Robot Designed for the CS1 Classroom. *ACM SIGCSE Bulletin, 41*(1), 428-32.

Lauwers, T. and Nourbakhsh, I. (2010). Designing the Finch: Creating a Robot Aligned to Computer Science Concepts. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1902-1907.

Lego Mindstorms. (2014). Retrieved May 2, 2014, from http://mindstorms.lego.com/.

Leutenegger, S. T. and Edgington, J. (2007). A games first approach to teaching introductory programming.

Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, 115-118.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education, 10*(4), Article 16.

Margolis, J. and Fisher, A. (2001). *Unlocking the Clubhouse: Women in Computing*. The MIT Press.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., Laxer, C., Thomas, L., Utting, I., and Wilusz, T. (2001). ITiCSE 2001 working group reports: A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin, 33*(4), 125-140.

McWhorter, W. I., O'Connor, B. C. (2009). Do LEGO Mindstorms Motivate Students in CS1? *Proceedings of the 40th SIGCSE Technical Symposium on Computer Science Education,* 438-442.

Savin-Baden, M. (2003). *Facilitating Problem-based Learning*. Berkshire, UK: McGraw-Hill.

Savin-Baden, M. and Major, C. H. (2004). *Foundations of Problem-Based Learning*. Berkshire, UK: McGraw-Hill.

Schuster, D. L. (2010). CS1, arcade games and the free java book. *Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education*, 549-553.

Sung, K., Panitz, M., Wallace, S. A., Anderson, R., and Nordlinger, J. (2008). Game-themed programming assignments: the faculty perspective. *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, 300-304.

U.S. Department of Education, National Center for Education Statistics (2013). The Condition of Education. NCES 2013-037. Retrieved from http://nces.ed.gov/pubs2013/2013037.pdf.

Zweben, S. and Bizot, B. (2014). 2013 Taulbee Survey. Computing Research News, 26(5), May 2014.