Computers & Education 82 (2015) 409–420

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

Closing the gender gap in an introductory programming course

Miguel Angel Rubio [a,*], Rocio Romero-Zaliz [a], Carolina Mañoso [b], Angel P. de Madrid [b]

[a] Departamento de Ciencias de la Computación e IA, ETSIIT, University of Granada, C/Periodista Daniel Saucedo Aranda, s/n, E-18071, Granada, Spain
[b] Departamento de Sistemas de Comunicación y Control, ETSI Informática, UNED, C/ Juan del Rosal, 16, E-28040, Madrid, Spain

A R T I C L E I N F O

Article history:
Received 29 July 2014
Received in revised form
1 December 2014
Accepted 2 December 2014
Available online 10 December 2014

Keywords:
Teaching/learning strategies
Programming and programming languages
Gender studies
Improving classroom teaching

A B S T R A C T

Although there is a growing interest in learning to program, the number of women involved in programming remains surprisingly low. We don't understand completely the causes but it has become clear that men and women have different perceptions of programming. The pedagogy of introductory programming courses should take these differences into account. In this study we analyze gender differences in an introductory programming course at the university level. Our results indicate that male and female students have different perceptions and learning outcomes: male students find programming easier, have a higher intention to program in the future and show higher learning outcomes than female students. To reduce these differences we have designed and implemented several learning modules using the principles of physical computing. The physical computing approach aims to take computational concepts out of the screen and into the real world so that students can interact with them. We have applied these modules in a MATLAB introductory programming course in a biology degree. When using these modules both male and female students showed similar results in perceptions and learning outcomes. The use of physical computing principles in combination with the traditional methodology reduced —actually closed— this gender gap.

1. Introduction

In the last years there has been a growing interest in learning and teaching to program (Wortham, 2012). Initiatives like The Hour of Code ("The Hour of Code," n.d.) or Codecademy ("Codecademy: Learn to code," n.d.) have taught programming to thousands of students. Computer science topics have been introduced in the primary-school curriculum in the UK (Brown, Sentance, Crick, & Humphreys, 2014) and New Zealand (Bell, Andreae, & Robins, 2014). New methodological approaches have been developed to help students in these courses. Successful examples are Scratch (Resnick et al., 2009), App Inventor (Wolber, Abelson, Spertus, & Looney, 2011), and Light-Bot ("Light-Bot," n.d.). These learning resources have been shown to improve students outcomes (Goadrich, 2014; Gouws, Bradshaw, & Wentworth, 2013; Guzdial, Ericson, Mcklin, & Engelman, 2014).

There remains one dark spot: the number of women involved in computer science is surprisingly low. In the United States only 0.4 percent of girls entering college intended to major in computer science in 2013 and they made up 14 percent of all computer science graduates, down from 37% in the mid-80s (Alvarado, Dodds, & Libeskind-Hadas, 2012; Patitsas, Craig, & Easterbrook, 2014; Tiku, 2014). Other studies show similarly disproportionate ratios of participation between male and female students in computer science programs (Stoilescu & Egodawatte, 2010). This problem is global: a study conducted on the use of computers and the Internet among fifteen-year olds showed that boys report using computers more often than girls in the vast majority of the 40 countries under investigation (Drabowicz, 2014).

Although we don't understand completely the causes of the differences in participation it has become clear that men and women have different perceptions of programming. Werner, Hanks, and McDowell (2004) analyzed a survey of over 400,000 entering freshman across the US. They found that the gender gap in computer use was almost non-existent but there was a very big confidence gender gap in computer skills. Several authors (Alvarado, Lee, & Gillespie, 2014; Carter & Jenkins, 1999) have found that female students are much less confident in their programming abilities than male students.

* Corresponding author. Tel.: +34 958240466.
E-mail address: marubio@ugr.es (M.A. Rubio).

http://dx.doi.org/10.1016/j.compedu.2014.12.003
0360-1315/

If we want to involve more women in computing, the pedagogy of introductory programming courses needs to change. This is a complex problem and there are no magic bullets but new approaches might help. In 2005 Harvey Mudd College started a three pronged approach: a breadth-first CS1 course with separate tracks according to previous experience, computing research experiences for first-year women and female community building activities. They observed a marked increase of women majoring in computer science (Alvarado et al., 2012). Google has launched a $50 million initiative to teach programming to young girls ("Made with Code," 2014). This initiative includes coding projects, female community building activities and video profiles of women that use programming to solve all kind of problems. In Europe, the European ACM Committee on Women in Computing (ACM-WE) has launched several initiatives to facilitate women participation in computing (Hanson, Ayfer, & Bachmayer, 2014).

One approach that might be effective is contextualized computing. Contextualized computing education is defined as the use of a consistent application or domain area, which effectively covers the core areas of a computer science course (Guzdial, 2010). Examples of contexts for introductory computer science include Media Computation (Guzdial, 2003), traditional manipulatives ("Computer Science Unplugged," n.d.), and robotics (Cuéllar & Pegalajar, 2014).

Students find contextualized approaches to programming very attractive. Instead of writing an abstract program, students can learn about basic programs by programming a robot to exit a maze, animating a story, or creating light symphonies. Rich, Perry, and Guzdial (2004) explored the possibilities of using context teaching to specifically address female students and obtained good results.

One contextualized approach that has attracted increased attention is the physical computing approach. This approach takes the computational concepts "out of the screen" and into the real world so that student can interact with them (Richard, 2008). Several studies have analyzed the feasibility of using physical computing principles in the teaching of computer programming, see for example Ruthmann, Heines, Greher, Laidler, and Saulters (2010).

Male and female students might react differently to physical computing activities. McGill (2012) studied the use of robots in an introductory programming course and found that female students were slightly more intimidated than males by the robots. She also found that female students believed more strongly that using the robots helped them to learn and that it was a pleasure to work with robots.

One alternative approach, used in this study, is to use electronic boards to develop small and simple systems capable of display interesting behaviors (Grasel, Vonnegut, & Dodds, 2010). This approach presents several advantages: the systems are simpler and easier to understand, they are more reliable, show more reproducible behaviors and the overall cost is lower (Hill & Ciccarelli, 2013).

Our aim in this study is to answer the following research questions: is there a gender gap in the traditional introductory programming course? And, if the answer is affirmative, can we use physical computing principles to reduce it? To answer these research questions the following research hypotheses are examined first:

- H_1: Using traditional teaching methods there is no gender difference in the perception on programming.
- H_2: Using traditional teaching methods there is no difference between male and female failure rates.
- H_3: Using physical computing modules there is no gender difference in the perception on programming.
- H_4: Using physical computing modules there is no difference between male and female failure rates.

With this goal in mind we have developed several learning modules based on the physical computing approach. We have used them in an introductory programming course and analyzed the perceptions and learning outcomes of male and female students. As a control we performed the same analysis in another introductory programming course taught with traditional methods.

There have been several studies about gender differences in introductory programming (Murphy et al., 2006; Stoilescu & Egodawatte, 2010) but −to our knowledge− only one study compared the effect of the intervention with a similar group used as control (Sabitzer & Pasterk, 2014).

2. Methods

2.1. Materials

In this study we have developed several learning modules for an introductory programming course at the university level. These modules can be used to teach C/C++, Python or MATLAB covering both compiled languages and interpreted ones. Different course approaches and teaching methodologies might benefit from their use.

One of the first design decisions we had to make was whether to use an electronic board or a robotic platform. Both present several advantages and disadvantages. We decided to work with an electronic board because robotic platforms are more complex and, therefore, their behavior is less reproducible in an educational laboratory (Cuéllar & Pegalajar, 2014). Alvarez and Larranaga (2013) found that factors related to surface friction, battery load or light conditions affected significantly the behavior of the robot and hindered students' work. We heeded McGill's warning that "*potential technical problems should be seriously considered since they can easily negate any potential positive motivational effect*" (McGill, 2012).

We have selected the Arduino microcontroller board (Banzi, 2009) as the development platform. Arduino is an open hardware board that is becoming increasingly common within the teaching community (Grasel et al., 2010; Mellodge & Russell, 2013). One important advantage for our project is that Arduino is a very easy to use board: its first users were artists and designers. Also, thanks to its open-source nature, a wide variety of developers have selected it as a development platform for all kinds of computational systems (Hill & Ciccarelli, 2013).

We designed specific modules for lecture demonstrations and for laboratory sessions (Fig. 1). The contents of the lecture demonstrations and the laboratory sessions are directly related. It is our experience that lecture demonstrations create a desire to learn more about the inner workings of the system shown. We can take advantage of this interest if students find similar activities during the laboratory sessions.

Lecture demonstrations are designed to enhance the traditional teaching methodology, not to replace it. Lecturers will explain a computational concept using the traditional methodology and afterward will reinforce the explanation doing a physical computing demonstration.