Enhancing Student Learning Experiences in Computer Programming Classes using Robotics

Dr. Masoud Naghedolfeizi, Fort Valley State University

Dr.Masoud Naghedolfeizi is a senior professor in the Department of Mathematics and Computer Science at Fort Valley State University. His academic background includes a B.S. in Mechanical Engineering with minor in instrumentation and control, an M.S. in Metallurgical Engineering, and M.S. and Ph.D. in Nuclear Engineering. Dr. Naghedolfeizi's research interests include instrumentation and measurement systems, applied articial intelligence, machine learning, and engineering education. He is the author of numerous research and pedagogical articles in his areas of expertise and has obtained several externally funded projects in the fields of computing and engineering.

Dr. Xiangyan Zeng, Fort Valley State University

Xiangyan Zeng received her Ph.D. in computer science from University of the Ryukyus, Japan. She is currently a professor of computer science at Fort Valley State University. Her research interests include image processing, pattern recognition and machine learning.

Dr. Chunhua Dong, Fort Valley State University

She received the B.E. degree in 2010 from Quanzhou Normal University, China, the M.E. degree in 2012 from Hainan University, Hainan, China, and the D.E. degree in 2016 from Ritsumeikan University, Kyoto, Japan. She is currently a Post-doctoral Associate at Department of Mathematics and Computer Science, Fort Valley State University, United State. Her research interests include image processing and analysis, pattern recognition, medical image registration, medical image segmentation and digital watermarking.

Enhancing Student learning Experiences in Computer Programming Classes using Robotics

Abstract

College level computer programming is known to be challenging and difficult to learn, even after reviewing the subject several times. Research has indicated that learning through application and reflection could enhance student learning of the subject particularly in computer programming. To this end, programmable robots could be utilized to supplement programming activities that encourage and motivate students to apply their creative thinking and programming skills to devise solutions for real-world problems. Since developing a computer program to instruct a robot provides an immediate feedback as whether the program has accomplished its job, it engages students in both learning and reflection processes.

This paper presents the application of an affordable programmable robot in three computer programming classes; namely, Computer Science and Information Systems: An Overview (CS0), Programming I (CS1), and Programming II (CS2). Also, the survey results of student feedback regarding the level of their interest in programming before and after robotic activities, the challenges of programming a robot, and their overall rating of integrating robotic activities in programming classes are presented and discussed.

Introduction

Introductory computer programming is a core subject in the curriculum of computer science major. The subject is frequently taught in three different courses; namely, CS 0, CS 1, and CS 2. The topics covered in CS 0 are often related to various fundamental concepts in computing and computer algorithms. Many computer science programs place a particular emphasis on computer algorithm in CS 0 to familiarize students with programming logic. In CS 1, students learn to write computer programs based on three programming constructs and also study simple data structures such as arrays. In CS 2, the emphasis is on complex data structures and object oriented programming and students are required to write relatively complex programs using software engineering practices.

Computer programming has been always labeled as a hard, abstract, and boring subject by majority of computer science students. There are two main reasons why this subject weighs heavily on students' minds. First, most students have difficulties with understanding programming logic and structuring a program to solve a problem correctly. Second, students are not motivated to practice programming more frequently (beyond the minimum course requirements) and also to meaningfully reflect on their learnings with the aim of improving their programming skills and the quality of their programs.

Introductory programming courses have historically high attrition rates because often students are not sufficiently and dynamically engaged with programming tasks. In fact, failure rates of 30 to 50% have been reported in literature [1]. Beginners frequently feel frustrated and disenchanted with the subject since the challenges of learning programming appear to be too overwhelming to overcome. While laboratory components of these courses help students with getting feedback and encouragement in the face of programming challenges, practice and motivation beyond the lab environment remain uncertain. Therefore, it is necessary to explore interesting methods to engage students in programming and further their creative thinking and problem solving skills.

The success of robotics in stimulating problem solving and critical thinking in both K-12 and college students has been numerously reported and documented [2-4]. The use of robotics in introductory college programming courses could also be fruitful since it provides a practical platform in which programming concepts could be implemented and examined in a more hands-on approach. Since developing a computer program to instruct a robot provides an immediate feedback on whether the program has accomplished its task, it engages students in both learning and reflection processes. In fact, research has indicated that learning through application and reflection could enhance student learning of the subject matter. To this end, programmable robots could be utilized to supplement programming activities that encourage and motivate students to apply their creative thinking and programming skills to devise solutions for real-world problems.

Programmable Robot Choices

There are numerous types of programmable robots in the market. However, finding a type of robot that could be beneficial for introductory programming classes is not an easy task. Ideally, robots should be programmable in the programming language used in the course. Otherwise, teaching two different programming language (one for the course and the other for the robot) could be quite confusing for a beginner. Additionally, the robots should be affordable and have low maintenance cost in order to purchase sufficient number of robots for large programming classes and also keep the sustainability of the program over a long period of time.

The computer science program at Fort Valley State University has recently implemented limited robotic activities in the curricula of introductory programming classes in attempt to motivate students in programming subjects and also reduce the attrition rates in these classes. After evaluating several different robots in summer 2018, the computer science faculty decided to use the Finch robot for all of its introductory programming classes because of a number of interesting features that robot possesses as stated in the next section.

Finch Robot

Under a license from Carnegie Mellon University, a programmable robot named “Finch robot” was launched in 2011 by BirdBrain Technologies [5]. Over the years, the robot has been improved to support different programming languages and platforms. The current version of Finch robot supports 4 different programming languages including Snap, Scratch, Java, and Python. The robot is able to interface with Windows, Mac, Chrome, or Linux based computers.

BirdBrain Technologies also offers suggested curriculum and sample programs for each programming language supported by the Finch robot. The robot includes a number of on board sensors including light, temperature and obstacle sensors as well as a 3-axis accelerometer. Additionally, it is equipped with a buzzer and full color LED lights. The robot does not operate on battery and gets its power through a long USB cable that interfaces with computer. It should be noted that the cable restricts the movement of the robot and could also cause inaccurate movements due to its drag force. Additionally, the lack of rear wheels could cause skidding, especially on smooth surfaces, and imprecise turns/rotations. Figure 1 illustrates a typical Finch robot. The cost of robot is about $100 and educational institutions are eligible for a discount

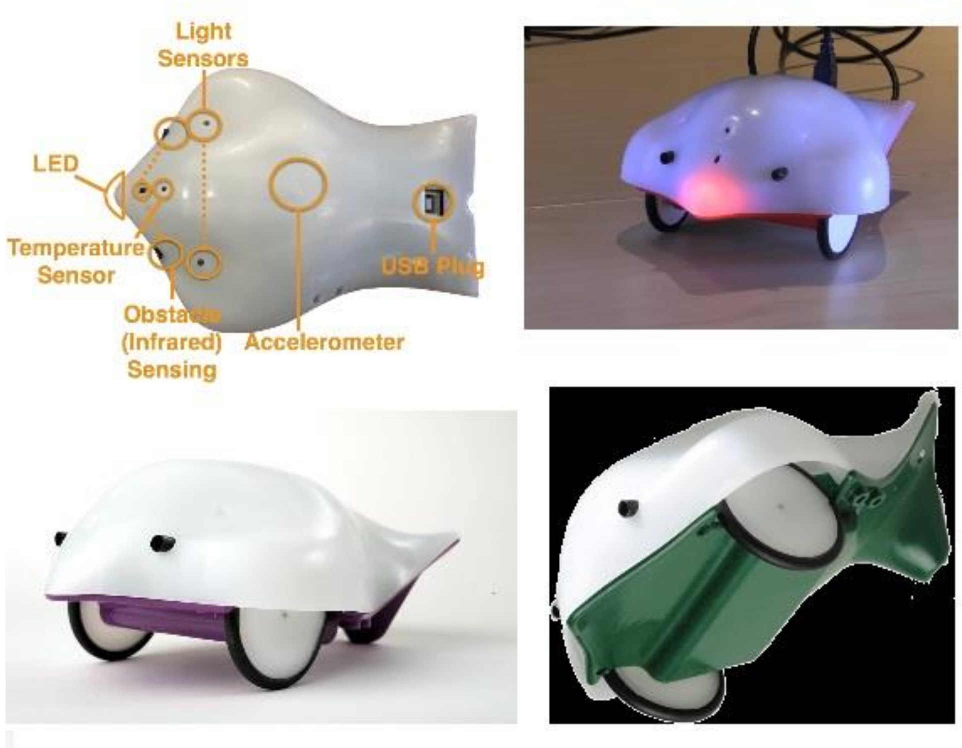

Figure 1: Finch Robot (courtesy of BirdBrain Technologies)

Programming Assignments

During summer 2018, the teaching faculty of introductory programming classes collaboratively identified a number of robotic exercises for the programming classes of fall semester 2018. The robotic assignments were designed and tested and subsequently were given in CS 0, CS 1, and CS 2 classes of fall 2018. The Scratch programming was chosen to program the robot for CS 0 class where the emphasis was on designing computing algorithms. Majority of students in this class have no prior exposure to any kind of programming languages. Therefore, a programming environment like Scratch that is user friendly and does not require students to deal with programming syntaxes would be a natural choice. In CS 1 and CS2 where the programming language is Java, the Finch robot assignments were designed in Java too. In these classes, a robot has been provided to every two students to jointly program it for a specific task.

A short lecture about the field of Robotics and a PowerPoint presentation on robots were delivered to each class before giving any programming assignments. The presentation included basic definitions, categories of robots, major components of a robot, applications, and the role of computer science in robotics. Further, during the lab sessions, Finch robot was introduced to students and a simple interaction with the robot was demonstrated. Additionally, students were provided with a short manual to invoke various drive and sensing commands for the Finch robot.

In CS 0, which does not have any lab component, two class times were allocated to robotic activities in one of our computer labs. Each pair of students working together was given a Finch robot to work jointly on two robot exercises[1]. The first exercise was to write an algorithm and then implement it in Scratch to instruct the Finch robot to go around a rectangular path of 1foot by 2 feet and then return to its starting point. In this exercise, students had to use correct sequence and repetition structures to accomplish the task. They also needed to figure out 90 degree turns and calibrate the distance based on the number of iterations needed for robot movement. This exercise also helped students to become familiarize with the robot and various mechanisms of robot control. In the second exercise students were tasked to modify the previous problem to detect obstacles on its path. The robot was to stop when an obstacle was detected, turn on its red light, sound a buzzer and make an announcement "Obstacle detected. Finch is stopping" through the interfacing computer speaker. The robot would continue its path once the obstacle was removed. In this assignment students used sequence, iteration, and condition structures to complete the assignment.

In CS 1, students developed a Java program that controlled the movement of a Finch robot through use of keyboard input. The Finch robot turned "Left", "Right", "Forward", and stopped based on the keyboard input of "L", "R", "F", "E". The same assignment was given in CS 2, where the robot was controlled through mouse clicks on a simulated control panel which was implemented as a graphical user interface (GUI) in Java. GUI programming is an excellent example of how the object-oriented principle is applied. This exercise was designed to enhance student learning experience of GUI and event-driven programming.

At the conclusion of robotic activities in each class, student feedback and comments were collected using a survey questionnaire. The survey included five (5) multiple choice questions on scale 1 to 5 and one open question to state comments/concerns/suggestions (see Figure 2).

The student responses for each question were tallied and tabulated for each class. The responses of students to the survey questionnaire (shown Tables 1-3) as well as their comments (presented in Figure 3) indicate that majority of students enjoyed having robotic assignments in

---

[1] The positive impact of teamwork on student performance and motivation to learn, particularly in laboratory environments, has been discussed in references [6 &7].

programming classes and gained a more positive attitude toward programming after completing robotic exercises. Admittedly the sample size is small and the survey analysis informal, but nonetheless the initial results are very encouraging.

Student Survey

November 19, 2018

1. On a scale of 1 to 5 where1 is minimally challenging and 5 is highly challenging, how do you rate the degree of difficulty to write a program to instruct/control a robot.
   a. 1
   b. 2
   c. 3
   d. 4
   e. 5
2. On a scale of 1 to 5 where 1 is the lowest interest and 5 is the highest interest, how do you rate your interest in writing codes for controlling and instructing robots?
   a. 1
   b. 2
   c. 3
   d. 4
   e. 5
3. On a scale of 1 to 5 where1 is the minimally increased and 5 is highly increased, how do you rate your interest in computer programming after robotic activities?
   a. 1
   b. 2
   c. 3
   d. 4
   e. 5
4. On a scale of 1 to 5 where1 is the lowest interest and 5 is the highest interest, how do you rate your interest in having more robotic programming assignments in this class?
   a. 1
   b. 2
   c. 3
   d. 4
   e. 5
5. How do you rate the overall robotic activity (activities) conducted in this class.
   a. Not good
   b. Average
   c. Good
   d. Very good
   e. Excellent

6. Please write any comments, concerns, or suggestions that you might have regarding robotic activities in this class

Figure 2: Survey questionnaire for robotic activities in introductory programming classes

Table 1: Survey results for CS 0

Student Survey for Robotic Activities in CSCI 1102							
	1	2	3	4	5	Trend	Average Class rating
Challenging	1	8	1	4	2		58%
Robot coding Interest	0	1	2	5	8		85%
Increased Interest	0	1	2	3	10		88%
More Robotic Assign.	0	2	2	2	10		85%
Overall Rating	**0**	**0**	**2**	**8**	**6**		**85%**

Table 2: Survey results for CS 1

Student Survey for Robotic Activities in CSCI 1301							
	1	2	3	4	5	Trend	Average Class rating
Challenging	0	1	10	2	4		71%
Robot coding Interest	0	0	3	8	6		84%
Increased Interest	0	0	0	12	3		84%
More Robotic Assign.	0	2	6	7	3		72%
Overall Rating	**0**	**1**	**10**	**5**	**1**		**67%**

Table 3: Survey results for CS 2

Student Survey for Robotic Activities in CSCI 1302							
	1	2	3	4	5	Trend	Average Class rating
Challenging	0	1	3	3	0		66%
Robot coding Interest	0	0	2	1	4		86%
Increased Interest	0	0	1	2	4		89%
More Robotic Assign.	0	0	3	2	2		77%
Overall Rating	**0**	**0**	**2**	**4**	**1**		**77%**

Figure 3: Typical student comments

CS0:
"Fun and entertaining"
"I liked it a lot"
"It was very fun and I hope to do more"
"Would be cool to build and program our own robots"

CS 1:
"I enjoyed it. It help students to find what field programming to focus"
"More robotic programming training and more in depth learning"
"It is a good way to see if we really know what we are doing"
"Practice and go over more"

CS 2:
"We should do more robotics"
"I would like to program robots in classes higher than CSCI 1302"

Discussions

During the implementation of assignments, it was observed that most students were able to complete the assignments with no help from the instructor. However, there were a few student groups who needed instructor help with the assignments. Also, there were seven student groups that developed inefficient codes that could complete the task but the codes were not considered good programs. We asked those students to reflect on their solutions and try to develop a more efficient code that accomplishes the task. These students were given extra time beyond the lab hours to complete their programs without any penalty. All except one group of students were able to come up with correct and efficient programs within a week after the assignment due date.

We would also like to emphasize that the robotic activities in these courses were intended to supplement, not to substitute traditional programming assignments. The main idea was to help students to reflect on their solutions and understand the multi-faceted capabilities of programming that includes not only traditional applications for business but also applications for high tech and engineering. The other aspect of these activities was to help student develop a positive attitude toward programming.

We are highly encouraged to continue these activities for future programming classes based on both student feedback and comments as well as the fact that there were a few students in each class who wanted to continue working on robots on their own time beyond the required assignments.

Further, it should be noted that the integration of robotic exercises into introductory programming classes is a time consuming process for the instructors since they need to design and test the assignments and make sure various programming concepts are emphasized in the assignments well in advance. Further, it should be noted that designing object oriented assignments for programming robots are generally more challenging and time consuming than any other types of assignments.

Conclusions

The inclusion of robotic activities in introductory programming classes is seen to increase student interests in programming and also help them become more skillful programmers. The use of an appropriate programmable robot is key to a successful implementation of robotics in programming classes. It is important to use robots that are programmable in the same programming language used in the class. Affordability and low maintenance costs are also important factors to long term sustainability of these activities.

Our initial experience with robotics in programming classes indicate that most students enjoy robotics and they become more motivated to learn programming. We are pleased with the outcome and will continue using robotic activities in our introductory programming classes.

References

1. Bennedsen, J. and Caspersen, M.E.” Failure rates in introductory programming.” SIGCSE Bull. 39, 2 (2007), 32–36
2. Jennifer S. Kay, “Robots in the classroom...and the dorm room,” Journal of Computing Sciences in Colleges, vol. 25, Issue. 3, Jan. 2010, pp. 128–133,.
3. Bers, M., Ponte, I., Juelich, K., Viera, A., Schenker, J.: Teachers as Designers: Integrating Robotics into Early Childhood Education. Information Technology in Childhood Education, 123–145 (2002)
4. Igor Verner, “Characteristics of student engagement in robotics,” in Intelligent Robotics Systems: Inspiring the NEXT , K. Omar, M. Nordin,P. Vadakkepat, A. Prabuwono, S. Abdullah, J. Baltes, S. Amin,W. Hassan, and M. Nasrudin, Eds. Springer Berlin Heidelberg, 2013
5. BirdBrain Technologies, https://www.birdbraintechnologies.com/
6. Korchnoy, E., Verner, I.: Characteristics of Learning Computer-Controlled Mechanisms by Teachers and Students in a Common Laboratory Environment. International Journal of Technology and Design Education 20(2), 217–237 (2010)
7. McDowell, C., Werner, L., Bullock, H.E., and Fernald, J. “Pair programming improves student retention, confidence, and program quality.” Commun. ACM 49, 8 (Aug. 2006), 90–95; DOI: 10.1145/1145287.1145293