ELEMENTARY STUDENTS PROGRAMMING IN C TO MAKE THEIR ROBOTS DO THEIR BIDDING

D. P. MILLER[1,2]

[1] *School of AME, University of Oklahoma,*
Norman, OK 73019, USA
E-mail: dpmiller@ou.edu

R. CLEMENT[2], C. GOODGAME[2] and S. GOODGAME[2]

[2] *KISS Institute for Practical Robotics,*
Norman, OK 73069, USA
www.kipr.org

Thousands of elementary school students participate in Junior Botball Challenge exercises every year. These challenges require students to write programs for their robots and to supplement their basic robots with effectors to carry out the challenge task. This paper presents data gathered from some of the schools that have participated, with a focus on those that did NOT select students based on their interest or ability. It shows that a large percentage of typical elementary school students are able to write working C programs (when given appropriate instruction) that exhibit sequential steps and timing.

Keywords: Computational thinking; coding skills, C programming, engineering design; STEM education.

1. Introduction

Robots have proven to be an effective way of teaching important STEM principles to students.[1–3] The *Botball Challenge Exercises* is a group of robotics software and engineering design exercises designed for students of all ages. The key elements of the program consist of robot equipment, software, curriculum, professional development, robot exercise mats, and of course the challenge exercises themselves. The recommended equipment and training currently use the standard *C* programming language[4] through KIPR's web-based IDE[5] and KIPR's basic robot kit. The challenge exercises are designed to be performed on the mats (see left image of Figure 1) or the floor, with the addition of commonly available items such as soda cans. Materials are reusable, so the costs to the school are low and predictable. Challenge events organized by KIPR and local hosts are available for students who wish to demonstrate their capabilities in a larger, more formal setting. Student teams who successfully complete a challenge exercise at such an event receive a merit button for each team member. The buttons are unique to each particular challenge. The challenges are arranged to provide hands-on opportunity and scaffolding for both programming and general conceptual engineering learning.[6]

This paper presents some of the Junior Botball Challenge exercises as well as data about the success in completing these challenges by primary school students. It is important to note that the students, whose activities are described herein, were selected to participate neither on interest or capability; a contrast to most robotics competitions. Rather, the students talked about in this paper participated in a grade-wide exercise during the normal school day, or were organized in required after-school activities in teams that reflected the diversity and gender makeup of the school. In cases where other selection criteria were used, they are noted when the data is presented. The analysis method is similar to that shown in evaluating the Botball Tournament teams[7] where we use a combination of self-efficacy and statistics about the team performance in official events.

2. Student Challenges and Learning Goals

Challenge 1:

Challenge 1 requires the participant to move the robot in a forward direction for a given distance, stop the robot, and then return it back to the bounded area in which it began. This task must be done autonomously.

Students are given three KIPR library functions to achieve this task, they are as follows:

```
motor(motor,speed);
```

The first argument specifies the port into which a given motor is plugged (0-3). The second argument is the percent (positive or negative) of maximum speed that the specified motor port will be run at. Note that the controller uses back-EMF as a control signal for adjusting power to achieve a specified speed.

```
msleep(numOfMS);// pauses program in current state
                // for numOfMS milliseconds
```

and

```
ao(); // sets speed of all four motors to 0
```

Fig. 1. Some of the participants in the Junior Botball Challenge Events

An example given to students as a starter for making their robot move in the positive direction for 1 second is as follows:

```
int main
{
  motor(0,100); // turn on motors full speed
  motor(3,100);
  msleep(1000); // sleep for 1 sec
  ao();          // turn off all motors
  return 0:
}
```

Obtainable goals for this challenge that are meant to be assessed:

- Student will create a flowchart or pseudo code that will aid them in organizing their structure for how that will achieve the desired task.
- Students ability to successfully follow and use C syntax correctly.
- Students will use pre-made functions to achieve the desired goal.

- Students will go through trial and error or some other computational means[8] to make their robot travel a certain distance in one direction and then move back to their starting point.

Most levels of Blooms taxonomy[9,10] are inherent during the process of this project-based learning activity: Students initial training using the `motor(), msleep()`, and `ao()` functions requires the student to have some **comprehension** of the use of each function. Upon receiving the prompt for Challenge 1, students must **apply** this knowledge to aid them in achieving their stated goal. Continued failure in achieving this goal causes students to spend a significant amount of cognitive function **analyzing** how their robot code could be changed to improve their ability to achieve the stated task and **evaluate** when they are successful.

Challenge 2:

Challenge 2 added turns to the robot's movement in an attempt to go around a can that is in a set position outside of the starting box. Students had to treat the left and right motors of the robot separately in order to have the robot drive around the goal can. In most cases the program was divided into five parts: 1) straight (to the back of the can); 2) turn (to go behind can); 3) straight (go behind can); 4) turn (to face towards start area); 5) straight (to start area). Steps 2 & 4 were often accomplished by running the left motor at an opposite velocity from the right for specific amount of time. Some teams combined steps 2,3, & 4 into a single operation where the the left and right motors were both driven forwards, but at different speeds to have the robot make a wide curve behind the can over a time period, often discovered by trial and error.

Mystery Challenge 24:

Students were able to prepare in advance and practice most of the challenges. In addition to the two normal events being discussed, the students were also presented with a *Mystery Challenge*, i.e., an exercise that was presented to them when they arrived at the event. For both events discussed below, this challenge consisted of having the robot drive across the mat removing a specific set of four cans from their numbered circles (all of which were occupied initially) and leaving the other cans undisturbed in their circles. This challenge of precision driving required the student to design and code a path for the robot that was not only curved, but also required both forward and backward movements.

Table 1. Success Statistics from November 2016 Tulsa event

Challenge	# teams	success	# students
#1 “Tag Your It”	142	1.00	950
#2 “Ring Around the Can”	142	1.00	950
#24 “Mystery Challenge”	132	0.93	883

3. Analysis

Two events are summarized in Tables 1 and 2. In the Tulsa event, 87 of the 142 teams were comprised of four students: two of which were female and two of which were Native American. This reflected the makeup of the school populations. All of these teams successfully completed challenges 1 and 2. While we do not have a specific list of which teams solved the *mystery challenge*, we can calculate that at least 50 of the 87 teams were successful in this challenge. The selection strategy and makeup of the remaining teams was not recorded.

Similarly, the KIPR-DELL event, which was held in Oklahoma City, had 96 teams (consisting of the same 533 students surveyed in Table 3) whose members were drawn from standard 3rd (17%), 4th (37%), 5th 38%) and 6th (7%) grade classes from 16 Oklahoma City Schools. The remaining 1% were the kindergarten and 2nd grade children of some of the teachers. Each school contributed a class of approximately 32 students. Each of the teams had 5 or six students. These were standard classes which did their robotics activities during the normal school day. All of the teams from these classes were able to succeed in the first and second challenges. And at least sixty nine of these teams were successful in the *mystery challenge*. The demographics of the students in these teams included 49.83% Hispanic, 16.84% African American and 5.06% Asian.

A student survey was given at the KIPR-Dell event. The lowest scoring question still has over three-quarters of respondents indicating that they contemplated staying at school longer so they could work more with their robots. More than 80% felt that they could program their robot to do what they wanted and more than 90% enjoyed programming and working on the

Table 2. Success Statistics from December 2016 KIPR-Dell event

Challenge	# teams	success	# students
#1 “Tag Your It”	126	1.00	1100
#2 “Ring Around the Can”	126	1.00	1100
#24 “Mystery Challenge”	114	0.90	995

Table 3. Results of a Student Self-Efficacy Survey. n=533

	Yes	No
I can write code to make my robot move and do what I want it to do	86.31%	13.69%
I am comfortable working on a project with other classmates	93.28%	6.72%
I am more confident sharing my ideas with my classmates	84.07%	15.93%
I am thinking about staying in school longer	77.01%	22.99%
I am interested in learning about jobs that use robots and technology	78.23%	21.77%
By participating I learned new skills and I have improved my ability to code the robot	85.80%	14.20%
I would recommend participating in the Junior Botball robotics program to other kids	91.10%	8.90%
I like writing code and working with the robot	92.28%	7.72%
I would like to keep participating in the Junior Botball Program next year	81.04%	18.96%
I have more fun at school when we get to work with the robots	90.52%	9.48%

robot. Table 3 shows the results from all of the questions on the survey.

4. Conclusions and Future Work

Each of the teams had at least one student who wrote a successful program for the challenge. All of the students were taught the basics of the C programming language, and most of the students felt confident that they could program the robot in the C language. It seems evident from this data, and extensive anecdotal events, that elementary school students are able to handle text-based languages, at least those students in upper elementary grade levels.

While there are numerous graphical programming languages designed for students (e.g., Scratch[11] or Blockly[12]) , there are some clear advantages to teaching students text-based languages. There is a perception amongst older students who have used these non-text-based languages that they are *less authentic, less powerful, slower and more verbose to author.*[13] Our work shows that even typical third grade students, given appropriate scaffolding methods, can program successfully in a standard text-based language, when they are properly motivated.[14] The resulting emphasis on attention to detail, task perseverance, keyboarding and elimination of first language issues[15] might have some real educational advantages.

Numerous other challenges were presented at these events, with the goal of assessing additional programming capabilities and problem solving skills. They will be documented in future publications

Acknowledgements:

Teams were supported, in part, at the Tulsa event by the *Muskogee (Creek) Nation Botball Program*, and at the Oklahoma City event by a grant from *Dell Legacy of Good Youth Learning*. The authors also wish to thank all the staff at KIPR for production of the kits, technical assistance and event production. Thanks also goes out to the students, teachers and volunteers

that help make these events possible.

References

1. E. R. Kazakoff, A. Sullivan and M. U. Bers, *Early Childhood Education Journal* **41**, 245 (2013).
2. S. Magnenat, F. Riedo, M. Bonani and F. Mondada, A programming workshop using the robot "thymio ii": The effect on the understanding by children, in *Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on*, 2012.
3. D. C. Williams, Y. Ma, L. Prejean, M. J. Ford and G. Lai, *Journal of Research on Technology in Education* **40**, 201 (2007).
4. B. W. Kernighan and D. M. Ritchie, *The C programming language*, 2nd ed edn. (Prentice Hall, Englewood Cliffs, N.J., 1988).
5. S. Zeltner and D. P. Miller, Kiss your old kiss goodbye, in *Proceedings of the 2015 Global Conference on Educational Robotics*, (Albuquerque, NM, 2015).
6. L. E. Berk and A. Winsler, *Scaffolding Children's Learning: Vygotsky and Early Childhood Education. NAEYC Research into Practice Series. Volume 7.* (ERIC, 1995).
7. D. P. Miller, S. Goodgame, G. Koppensteiner and M. Yong, Some effects of culture, gender and time on task of student teams participating in the botball educational robotics program, in *Robot Intelligence Technology and Applications 3*, eds. J. H. Kim, W. Yang, J. Jo, P. Sincak and H. MyungAdvances in Intelligent Systems and Computing (Springer, April 2015) pp. 541–557.
8. D. P. Miller and C. Stein, So that's what pi is for! and other educational epiphanies from hands-on robotics, in *Robots for kids: Exploring new technologies for learning experiences*, eds. A. Druin and J. Hendler (Morgan Kaufmann Publishers Inc., 2000) pp. 219–243.
9. B. Bloom, D. Krathwohl and B. Masia, *Bloom taxonomy of educational objectives* (Allyn and Bacon, Boston, MA. Copyright (c) by Pearson Education.¡ http://www. coun. uvic. ca/learn/program/hndouts/bloom. html, 1984).
10. D. R. Krathwohl, *Theory into practice* **41**, 212 (2002).
11. J. Maloney, M. Resnick, N. Rusk, B. Silverman and E. Eastmond, *ACM Transactions on Computing Education (TOCE)* **10**, p. 16 (2010).
12. J. Trower and J. Gray, Blockly language creation and applications: Visual programming for media computation and bluetooth robotics control, in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 2015.
13. D. Weintrop and U. Wilensky, To block or not to block, that is the question: students' perceptions of blocks-based programming, in *Proceedings of the 14th International Conference on Interaction Design and Children*, 2015.
14. C. Kelleher and R. Pausch, *ACM Comput. Surv.* **37**, 83(June 2005).
15. S. S. Brilliant and T. R. Wiseman, *ACM SIGCSE Bulletin* **28**, 338 (1996).