

Learning Code using Lego Robotics

Adam Thomas & George Paravantes

Humber College Institute of Technology & Advanced Learning

Keywords: Engagement, play, robotics, tangible, coding

Article History

Received 26 September 2017

Received in revised form 18 October 2017

Accepted 31 October 2017

Available online 12 March 2018

Introduction

Learning to code has a reputation for being difficult (Gomes & Mendes, 2007; Jenkins, 2002), and requires a variety of skills, such as math and complex problem solving, that are challenging for many students (Foote, 2014; Gomes & Mendes, 2007; Jenkins, 2002)—especially for those students beginning a college program (Oblinger, 2003). Often, students experience high levels of anxiety even before a programming course has started (Jenkins, 2002). This is particularly true for students who are required to take a course in coding, but who do not plan to continue on to a career in this field. Anecdotally, these students find the fundamentals of code difficult, and often end up “hacking” their way through the course.

One approach to addressing this anxiety that has been used with children and youth is to teach code using robotics (Kurebayashi, Kamada, & Kanemune, 2006; McGill, 2012). Learning to code using robotics was found to have many positive effects: a) it allows students to more easily connect individual lines of code to their result (Kurebayashi et al., 2006); b) it

stimulates intrinsic motivation (Kurebayashi et al., 2006; McGill, 2012); and c) it increases overall student grades (McGill, 2012).

However, there is little to no literature on this type of approach with adult learners. Therefore, the purpose of this study was to look at the effects of incorporating robotics in a playful context (Plass, Homer, & Kinzer, 2015), in a series of “Introduction to Coding” courses at the post secondary level. After teaching code at the post secondary level for over ten years, particularly with students who did not necessarily want to or feel the need to learn how to code, our goal was to find a way to re-engage students.

Method

Participants

The students involved in the research included two groups of approximately 60 students (two classes of 30 students each), from two different programs: Web Design and Interactive Media, and Multimedia, Design and Development.

Web Design and Interactive Media

Students enrolled in this program start the program prepared to learn code and have often already had some form of coding education. Students in this program served as the comparison group, and learned to code using traditional methods.

Multimedia, Design and Development

Students in this program are often surprised by the amount of code they are required to learn and usually have no previous coding experience. Students in this program served as the test group. They learned code using traditional methods combined with activities incorporating Lego robotics.

Materials

Teaching Equipment

One of the challenges with this project was to use equipment that did not add to the existing anxiety and/or workload of the students in the test group. With Lego being a relatively familiar childhood toy, we chose the Lego Mindstorms EV3 kits as our hardware (see Figure 1). We hoped this would also add to the

Author Note

Adam Thomas is a professor in the School of Media Studies and Information Technology at Humber College.

George Paravantes is an Interaction Designer and Program Coordinator of the Multimedia Design and Development Program at Humber College.

 This article is published under a [Creative Commons Attribution-Non Commercial-No Derivatives 4.0 International License \(CC BY-NC-ND 4.0\)](https://creativecommons.org/licenses/by-nc-nd/4.0/)

***Innovation Spotlights** are extremely brief contributions that highlight an innovative teaching practice, approach, or tool, and provide accompanying evidence that speaks to the effectiveness of the innovation.

playful aspect of the exercise, which has been shown to support learning (Plass et al., 2015). To keep things simple, and avoid requiring students to learn the proprietary Lego coding language, the Lego Bricks were set up using ev3dev (a Linux-based OS, refined for use with the Lego Mindstorms EV3) and a small PHP API. This made it possible to code the robots using the same code editor, browser, and commands that were used when learning to code using traditional methods.

Research Materials

To assess the impact of the robotics, final average grades were compared between groups, and pre- and post-course surveys were developed. The surveys each consisted of fourteen questions. On the pre-course survey, questions focussed on expectations and coding knowledge prior to the course, and on the post-course survey there were complementary questions focussing on actual experience and coding knowledge after completing the course. See the Appendix for the pre- and post- surveys.

Procedure

In these programs, students traditionally learn code by attending one weekly three-hour class, consisting of a short lecture followed by coding activities and/or group work. In between classes, students are directed to online supports and are given weekly exercises from resources such as Codecademy, Lynda.com, and Treehouse, to complement the content learned in class. This process works well with students who are interested and self-motivated; however, students who are not interested in learning code struggle to remain engaged.

In the present study, the comparison group maintained this “traditional” pattern of teaching and learning. The test group also received this approach for the majority of the course. However, for

a selection of core programming concepts, the test group spent an additional class reinforcing these concepts by programming Lego robotics, immediately following the lesson in which the concept was taught. For example, after students learned to program a control structure (to allow a program to make a decision), they built and programmed a simple autonomous Lego robot. The challenge was to program a robot to navigate around a table using a proximity sensor and control structures to avoid falling off the edge (see Figure 2). The test group received a total of three sessions with the Lego robots.

Students in both classes were informed about the research project at the beginning of the semester. To prevent any actual or perceived coercion, students were recruited to complete the surveys by a third-party research assistant, who did not have any existing relationship with students. Surveys were anonymized, and were optional for students to complete. Surveys took approximately 5-10 minutes to complete, at the beginning and end of the semester, respectively.

Results

Expected vs. Actual Class Enjoyment

On the pre- and post- surveys, students were asked to rank their coding course compared to their other four courses (i.e., Was it their favourite course? Second favourite? etc.). As shown in Figure 3, at the beginning of the semester, approximately one quarter of students in both the comparison (22.5%) and test (25%) groups expected their coding class to rank among their most (1st or 2nd) enjoyable classes. At the end of the semester, while this figure had increased in both groups, the proportion of students in the test group (43.8%) was much larger than the comparison group (28.6%), with 18.9% of students rating it as their very favourite course. Moreover, the proportion of students who ranked the

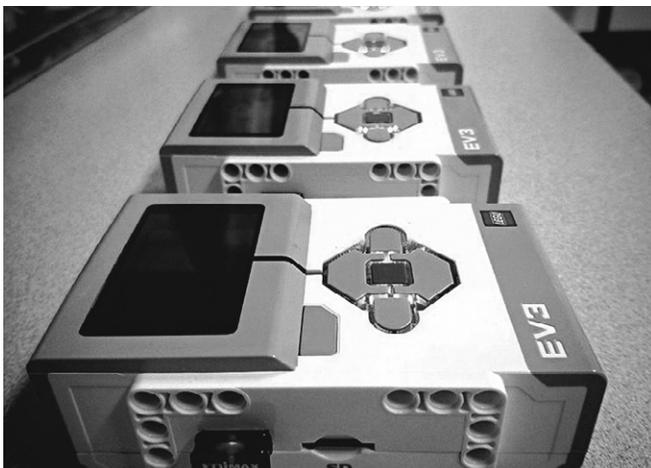


Figure 1. Lego EV3 Kit (Mindstorms EV3, n.d.)

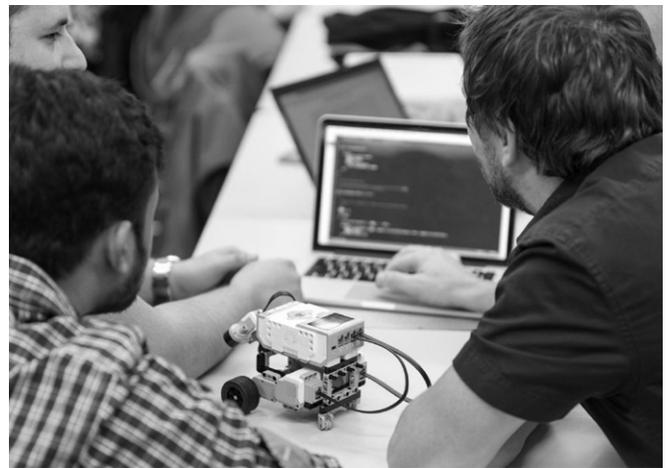


Figure 2. An assembled Lego Robot

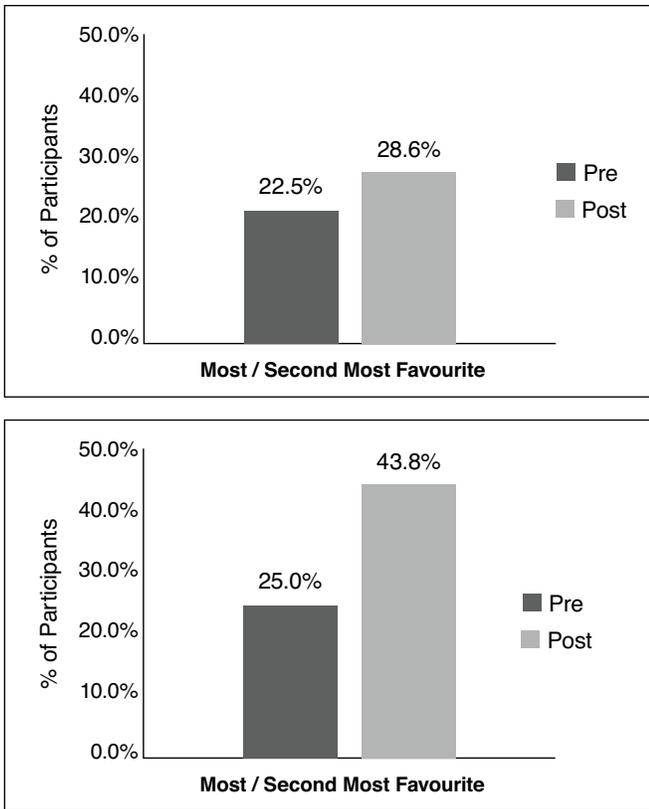


Figure 3. Proportion of students who rated their coding course as their 1st or 2nd favourite course at the beginning (Pre-) and end (Post-) of the semester, for the comparison group (top) and test group (bottom), respectively.

course as their least- or second-least favourite course decreased from 39.2% on the pre-survey to only 25% in the test group. The differences in proportions were not statistically significant between the two groups, $\chi^2(1) = 0.5, p > 0.05$, nor between the pre- and post- frequency distributions of the test group, $\chi^2(4) = 1.55, p > 0.05$, according to chi-square contingency table analyses. However, this is likely at least partly due to the uneven sample size between the Comparison ($n=15$) and Test ($n=48$) groups on the post-test.

Confidence in Coding Ability

Five questions on the pre- and post- surveys asked students to self-report on their confidence levels with executing different concepts in Javascript (the coding language they were learning). Not surprisingly, those in the comparison group (students who were planning to pursue a career in coding) had higher average confidence ratings than the test group, both at the beginning and end of the semester. Still, both groups showed comparable increases in confidence levels over the course of the semester (Figure 4). It should be noted that by the end of the semester, the test group's average confidence levels were similar to those of

the comparison group's pre-semester levels. Due to the relatively smaller sample size of the comparison group on the post-test, an inferential statistical test was not appropriate in this instance.

Knowledge of Coding Concepts

As shown in Figure 5, students in both groups rated their knowledge of course content as approximately equal, both pre- and post-semester. In particular, both groups reported increases in their self-rated knowledge of course content over the course of the semester, with a slightly higher rating for the comparison group at both points in time. Once again, inferential statistical testing was not appropriate due to the uneven sample sizes.

The final average grades for both groups were also similar, with the comparison group at an average of 75.0% and the test group at an average of 73.4%. Interestingly, the previous year, students in that program (Multimedia Design and Development) had a final average grade of 69.4%.

Anecdotal Class Observations

Initially, there was some concern that college students may perceive the idea of learning code using Lego as childish and prefer not to participate. However, before the first class involving the Lego robotics, an email was sent out informing students of the planned activity. Upon arriving to class, the students, who normally would be sitting at their computer stations, had all brought their chairs up to the front of the computer lab and formed a semicircle, in anticipation of the activity. Faculty consistently observed a higher level of participation, enjoyment and engagement in classes that incorporated the robotic activities.

Conclusions

The results showed that students participating in the lego robotics activities showed important changes in terms of confidence and enjoyment of the course. Although there was not a difference between groups with respect to the self-rated knowledge gained, there is preliminary evidence to suggest that final grades may have improved compared to previous cohorts, as has been previously shown in the literature (McGill, 2012). Further research is needed to determine whether this reflects a meaningful improvement or random variation.

Arguably, the increases in confidence and engagement are more important for the group under examination. In particular, the fact that a group of students that are traditionally less interested and prepared to learn code showed improvements in these areas is extremely encouraging, and is consistent with gains in intrinsic motivation that have been demonstrated in children and youth (Kurebayashi et al., 2006; McGill, 2012). Had a similar pre- and post- survey been completed with both programs without any robotics activities, we would expect the test group to gain

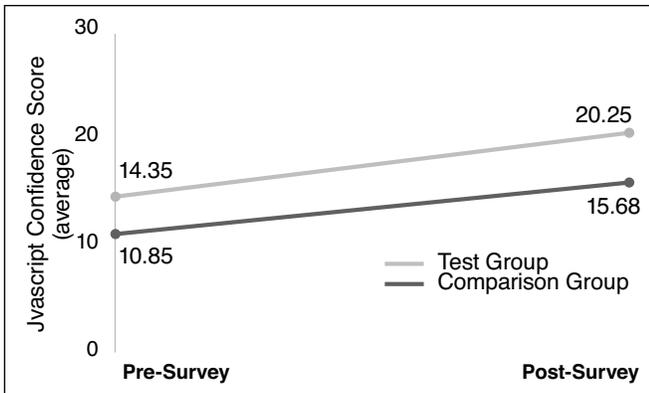


Figure 4. Average confidence index scores for the Comparison and Test groups, respectively, on the Pre- and Post- Surveys. On this scale, the maximum possible score is 25, and the minimum is 0. Higher scores indicate greater self-reported levels of confidence.

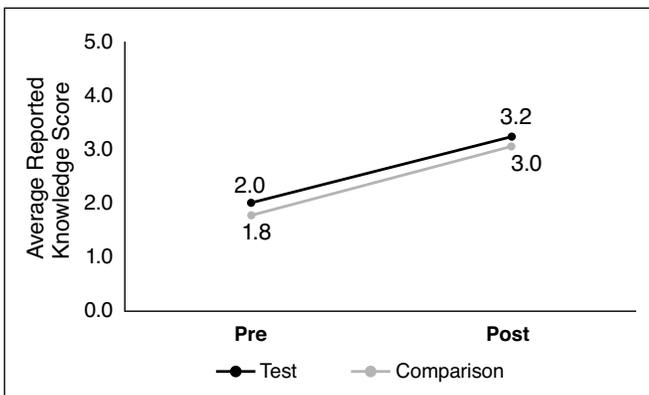


Figure 5. Average self-reported computer programming knowledge score for both Test and Comparison groups from Pre- to Post- survey. Scores are from 1 (Nothing) to 5 (Expert).

relatively less knowledge, and show much lower levels of confidence and enjoyment than the comparison group. Future research could determine whether this is the case, and also whether students in the comparison group (those in Web Design and Interactive Media) show similar gains when they learn with Lego robotics. Given the encouraging findings reported here, we are now in the process of replicating and expanding on these results with improved equipment, more frequent robotics activities, increased methods of quantitative measurement, and a larger sample of students.

Contact

✉ Adam.Thomas@humber.ca

Humber College Institute of Technology & Advanced Learning

✉ George.Paravantes@humber.ca

Humber College Institute of Technology & Advanced Learning

References

- Foote, S. (2014). *Learning to program*. New York, U.S.A.: Addison-Wesley Professional.
- Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. Paper presented at the *International Conference on Engineering Education—ICEE*. Coimbra, Portugal.
- Jenkins, T. (2002). On the difficulty of learning to program. In *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 4, 53-58.
- Kurebayashi S., Kamada T., & Kanemune S. (2006). Learning computer programming with autonomous robots. In: R. T. Mittermeir (Ed.) *Informatics Education – The Bridge between Using and Understanding Computers*. ISSEP 2006. *Lecture Notes in Computer Science*, Vol 4226. Berlin, Heidelberg: Springer.
- McGill, M. M. (2012). Learning to program with personal robots: Influences on student motivation. *ACM Transactions on Computing Education*, 12(1), 4:1-4:32.
- Mindstorms EV3. (n.d.). Retrieved August 07, 2017, from <https://www.lego.com/en-us/mindstorms/products/mindstorms-ev3-31313>
- Oblinger, D. (2003). Boomers gen-xers millennials: Understanding the new students. *Educause Review*, 38(4), 37-47.
- Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist*, 50(4), 258-283.

Appendix

Pre-Course Survey

Please answer the questions by circling the answer that best represents your perspective:

- How would you rank your current knowledge of programming? (1 being nothing and 5 being an expert)
- Based on your current knowledge, how difficult do you expect this introduction to programming course to be? (1 being very easy and 5 being very difficult)
- How enjoyable are you expecting this introduction to programming course to be? (1 being you will NOT enjoy this course and 5 being you WILL enjoy this course)
- Compared to the other courses in this semester how would you rate your expectations for this course? (1 being your most favourite and 5 being your least favourite)
- How many hours a week do you expect to dedicate to this course?
2 or less 3 to 5 6 to 9 10 to 12 More than 12
- How likely will you be to incorporate programming into your employment after you graduate? (1 being NOT very likely and 5 being VERY likely)

7. How much do you think concepts taught in this introduction to programming course will help in your everyday life?
(1 being never and 5 being very often)
8. How would you rate your current understanding of JavaScript?
(1 being none and 5 very skilled)
9. I am confident with creating output and manipulating HTML with JavaScript.
(1 being not confident and 5 being very confident)
10. I am confident with using JavaScript variables to store and manipulate data.
(1 being not confident and 5 being very confident)
11. I am confident with using JavaScript control structures (if statements) to make decisions.
(1 being not confident and 5 being very confident)
12. I am confident with using JavaScript loops (for or while loops) to make repeat code.
(1 being not confident and 5 being very confident)
13. I am confident with using built in JavaScript functions (substr, toUpperCase, to Fixed).
(1 being not confident and 5 being very confident)
14. I am confident with using and creating custom JavaScript functions.
(1 being not confident and 5 being very confident)
6. How likely will you be to incorporate programming into your employment after you graduate?
(1 being NOT very likely and 5 being VERY likely)
7. How much do you think concepts taught in this introduction to programming course will help in your everyday life?
(1 being never and 5 being very often)
8. How would you rate your current understanding of JavaScript?
(1 being none and 5 being very skilled)
9. I am confident with creating output and manipulating HTML with JavaScript.
(1 being not confident and 5 being very confident)
10. I am confident with using JavaScript variables to store and manipulate data.
(1 being not confident and 5 being very confident)
11. I am confident with using JavaScript control structures (if statements) to make decisions.
(1 being not confident and 5 being very confident)
12. I am confident with using JavaScript loops (for or while loops) to make repeat code.
(1 being not confident and 5 being very confident)
13. I am confident with using built in JavaScript functions (substr, toUpperCase, to Fixed).
(1 being not confident and 5 being very confident)
14. I am confident with creating and using custom JavaScript functions.
(1 being not confident and 5 being very confident)

Post-Course Survey

Please answer the questions by circling the answer that best represents your perspective:

1. How would you rank your current knowledge of programming?
(1 being nothing and 5 being an expert)
2. How difficult did you find this introduction to programming course?
(1 being very easy and 5 being very difficult)
3. How enjoyable did you find this introduction to programming course?
(1 being you did NOT enjoy this course and 5 being you DID enjoy this course)
4. Compared to the other courses in this semester how would you rate this course?
(1 being your most favourite and 5 being your least favourite)
5. How many hours a week did you dedicate to this course?
2 or less 3 to 5 6 to 9 10 to 12 More than 12