Journal of Learning and Teaching in Digital Age, 2020, ***6***(1), 1-9
ISSN: 2458-8350 (online)
Research Paper

Overcoming the Lack of Qualified Computer Programming Teachers: a Field Experiment in Belize

Kieran Ryan
(ORCID ID:0000-0003-1259-2595)
University of Belize, Belize
kryan@ub.edu.bz

David Garcia
(ORCID ID:0000-0002-1612-9904)
University of Belize, Belize
dgarcia@ub.edu.bz

Manuel Medina
(ORCID ID:0000-0002-6031-1906)
University of Belize, Belize
mmedina@ub.edu.bz

Dalwin Lewis
(ORCID ID:0000-0001-8484-5200)
University of Belize, Belize
dlewis@ub.edu.bz

Luis Herrera
(ORCID ID:0000-0003-0231-5973)
Sacred Heart Junior College, Belize
luis@shc.edu.bz

Received 22 October 2019, Revised 09 January 2020, Accepted 24 January 2020

ABSTRACT

Several developed countries have initiated compulsory computer programming classes at the primary and secondary school levels with many others aiming to implement them by the end of the decade. A significant issue encountered has been the lack of qualified computer science teachers. If developed countries with abundant educational resources are facing these challenges, how can countries in the developing world with limited resources overcome them? In this research, we pilot a novel solution by utilizing an online portal to deliver a standardized computer programing course utilizing self-regulated and designed-based learning concepts where the teacher and students learn simultaneously. The study utilized a quasi-experimental design that measured learning outcomes before and after a 25-hour course. Five university lecturers (control group) and one middle school teacher (treatment group) taught the course to 178 middle school students using standardized material from the online portal. A computational thinking test was administered pre and post-treatment to measure learning outcomes. The results showed that students of both groups significantly enhanced their computational skills with no significant difference in learning outcomes between groups supporting our hypothesis.

Keywords: *Computational Thinking, computer-programming education, self-regulated learning, designed based learning, learning outcomes*

INTRODUCTION

Countries in the developed world are making a concerted effort to introduce computer programming into the primary and secondary school curriculums, with some already having met the goal and others outlining a roadmap to implement it by the end of the decade.

In Europe, the European Commission has been advocating policy for new skills training, to include coding calling it "the literacy of today" (European Commission, 2016). European countries are at different stages of implementation, some having a strong tradition in computer science in secondary school and are currently extending it to primary schools, some are

***Correspondence to*:** *David Garcia; Lecturer; Department of Mathematics, Physics, and Information Technology; University of Belize; dgarcia@ub.edu.bz; +501-822-1000*

renewing their curriculum to integrate coding, while others are in the planning phases.

Developed countries in East Asia are also in the planning phases where Japan's Education, Culture, Sports, Science and Technology Ministry has decided to make computer programming a compulsory subject at primary schools in the fiscal year 2020, followed by middle schools in fiscal 2021 and high schools in the fiscal year 2022 (Yomiuri Shimbun, 2016). In the United States of America, non-profit organizations such as code.org have led significant efforts in this area.

While developing countries have seen significant pushes to incorporate technology in the classroom, including coding (Brackmann, 2016) due to lack of resources, most find themselves behind in the endeavor for countrywide mandatory coding at the primary and secondary levels.

Education Environment

Belize, a small, medium-income country in Central America, finds itself in a similar situation as many other countries concerning the push for coding in the classroom. In Belize, most secondary schools and many upper primary schools have access to computer labs with internet connections; however, it has no official government policies or plans to incorporate countrywide mandatory coding classes.

The main challenges appear to be twofold. The first is convincing the Ministry of Education that the country can implement these policies in a similar time frame as developed countries, overcoming the mindset that developing countries have to be two to three decades behind developed countries. The second is to overcome the lack of qualified teachers available to teach computer programming, which is a problem that developed countries also face.

Aim of the Study

The ambition of this study is to show that teachers with a limited computer programming background can achieve similar learning outcomes with students as teachers with considerable computer programming background by utilizing standardized online course content. Also, the research results can be utilized by the Ministry of Education to implement policies that integrate computational thinking into schools' curriculum with minimal teacher training and will serve to diminish the two to three decade lag behind developed countries. In this paper, we describe and compare theories originating in the developed world: Self-Regulated Learning (SRL) and Design-Based Learning (DBL), and a theory originating in the developing world: Minimally Invasive Education (MIE). From these theories, we draw a conceptual framework to best combine constructs to create an online portal to improve computer programming learning outcomes of students where there is a lack of qualified computer programming teachers.

We conducted our research at three middle schools in Belize and measured and compared learning outcomes of the courses taught by university computer science professors versus the same courses taught by a teacher with no background in computer programming.

Utilizing a quasi-experimental design, we developed an online learning platform to implement Mitra's (2003) MIE theory combining it with elements of SRL and DBL theory. We then utilized the platform to deliver a computer programming course to middle school students, with students joining either a class with a qualified computer programming teacher or one with a teacher not trained in computer programming. In the second case, the untrained teacher learned the content simultaneously with the students.

Our conceptual framework combined theories found in the literature: two constructed in the developed world and one constructed in the developing world and utilized the framework to implement a platform to solve a commonly encountered problem, the lack of qualified computer programming teachers.

This article organization is as follows — first, a presentation of a review of the literature related to the constructs utilized in the study. Next, the focus is on the framework of combining constructs from two theories, followed by the research model and hypotheses. The research design is then detailed, and then finally, details on how the data was analyzed, followed by the discussion, limitations, and conclusions.

LITERATURE REVIEW

In developed countries, to improve educational outcomes, extensive research has been conducted to establish pedagogical theories on how to best implement computer technologies and the Internet in education. Self-Regulated Learning (SRL) is a prominent theory that has gained much attention (Tsai, Shen, & Fan, 2013) that has been shown to improve learning outcomes of students (Schunk, & Ertmer, 1999; Alavi, & Leidner, 2001; Azevedo, Moos, Greene, Winters, & Cromley, 2008; Wan, Compeau, Haggerty, 2012). A second theory utilized in the developed world is Designed-Based Learning (DBL), which has also been shown to improve learning outcomes in computer programming courses at the primary and secondary levels (Román-González, 2015). Additionally, evaluations of Cisco Network Academy standardized blended online courses have shown that students rate online curriculum and laboratory exercises as the most important resources and that online assessment resources were rated as highly as interaction with the instructor (Sealey, 2011).

In developing countries, researchers have focused on utilizing these technologies to solve the lack of access to the subject matter and the subject experts, which, in the past, has been a more prevalent issue than in developed countries. The solutions for the issue of lack of access, introduced a novel pedagogical theory: Minimally Invasive Education (MIE) that utilizes technology in education, to significantly improve learning outcomes where subject experts are lacking (Mitra, Dangwal, Chatterjee, Jha, Bisht, & Kapur, 2005).

E-Learning Environments and Self-Regulated Learning

Initial experiments to evaluate the effects of SRL in e-learning environments focused on self-evaluation. In these early experiments, students using e-learning systems were able to self-evaluate themselves one or more times (Schunk et al., 1999). Studies that compared the learning outcomes of e-learning environments without self-evaluation and e-learning environments with self-evaluation showed that the latter enhanced student self-efficacy.

More recently, to increase learning outcomes, SRL strategies are included in traditional e-learning environments through the use of pre-training and in-training feedback scripts, and external evaluations (Santhanam, Sasidharan, & Webster, 2008). An extension of this approach incorporates externally-facilitated self-regulated learning by having human tutors facilitate students' e-learning environment self-regulation. This extended approach compared to self-regulated learning environments without human tutors has shown to further enhance learning outcomes (Azevedo et al., 2008).

Design-Based Learning

Project-Based Learning (PBL) is a student-driven, teacher-facilitated approach to learning. Design-Based Learning (DBL) incorporates pedagogical concepts of project-based learning and focuses on the processes of inquiry and reasoning to produce solutions to real-life design problems; it has been shown to have positive results in the instruction of science. (Barrows 1985; Du & Kolmos, 2009).

More recent studies have applied DBL at the primary school level showing significant improvements regarding learning programming concepts, logic, and computational practices with this active approach (Sáez-López, Román-González, & Vázquez-Cano, 2016). The authors describe DBL as an environment as follows:

"In this pedagogical design, students interact and create their own content related to curricular areas with several advantages, such as motivation, fun, commitment, and enthusiasm, showing improvements related to computational thinking and computational practices. Understanding of computational concepts through an active approach, Project Based Learning, usefulness, motivation, and commitment underline the importance and effectiveness of implementing a Visual Programming Language from active methodologies in primary education."

Minimally Invasive Education

The pedagogic theory, Minimally Invasive Education (MIE), is based on experiments conducted in a New Delhi slum in 1993. In the experiments, a computer with an Internet connection was set up for children to use without supervision (Mitra et al., 2005). Results showed that in less than a month, the children learned how to browse, create documents, paint pictures, and play games – essentially, they learned basic computer skills on their own by coming into contact with the technology. This discovery led researchers to define MIE theory and to conduct further research on its implications. Experiments with self-organized learning environments (SOLEs) found that children, following the principles of MIE could improve their English pronunciation on their own (Mitra, Tooley, Inamdar, & Dixon, 2003), and improve their mathematics and science scores in school (Inamdar & Kulkarni, 2007). The implementation of SOLEs is a viable solution in developing countries where geographical, economic, social, and political factors combined with the lack of adequately trained teachers, leads to poor quality public schools (Mitra, Dangwal, & Thadani, 2008).

METHOD

Framework Combining Constructs

For over a decade, to improve e-learning environments, developed countries have employed SRL and DBL strategies. On the other hand, to overcome issues stemming from the lack of adequately trained programming teachers, developing countries have utilized self-organizing environments. However, the literature lacks the concept of combining constructs from both regions. We believe that similar to the introduction of SRL and DBL strategies into traditional e-learning environments; they can be combined with self-organized learning environments to improve learning outcomes further. They may be better suited for this, as SOLEs tend to require more self-regulation than traditional e-learning environments. Figure 1 shows a conceptual framework, based on Piccoli's model (2001), to combine constructs of SRL and SOMEs.

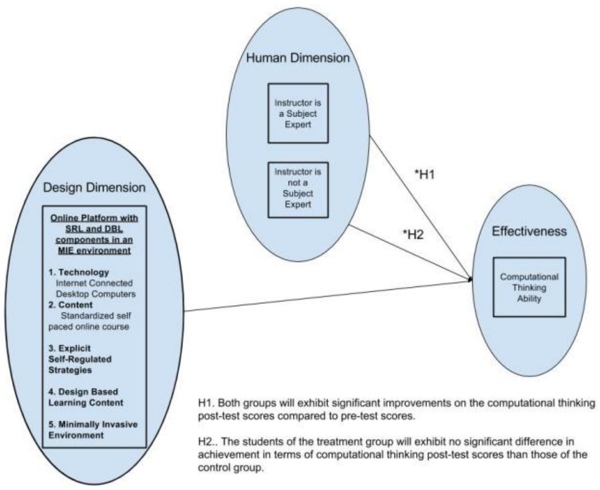

Figure 1. Research Framework Combining Constructs

Research Design

As shown in Figure 2, we conducted a quasi-experiment adopting a two group pre-treatment, post-treatment measurement design, where subject experts taught one group while a non-subject expert taught the other group. We compared the learning outcomes of students participating in a 25-hour computer programming course. The course consisted of one 50 minute class per week for 32 weeks. The control group was taught by university professors (subject experts), while a middle school teacher (non-subject expert) taught the treatment group.

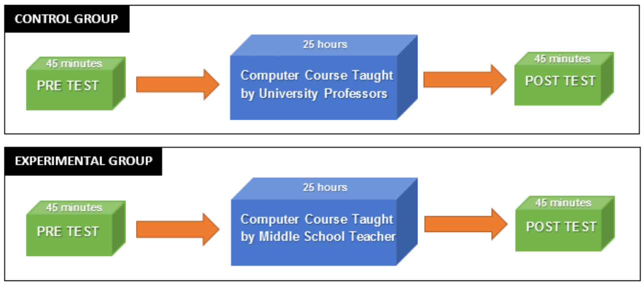

Figure 2. Stages and groups of the project

Minimally Invasive Education theory (MIE) (Mitra et al., 2005), a theory of instructional design that provides a solution for situations where there is an absence of subject experts, provides the foundation for the design of the e-learning platform. Explicit SRL strategies in the form of scripts included in traditional e-learning environments are used to improve learning outcomes (Schunk et al., 1999; Alavi et al., 2001; Wan et al., 2012). Project-Based Learning theory (PBL) has also been shown to improve computer programming learning outcomes (Román-González, Pérez-González, & Jiménez-Fernández, 2017). By combining these environments, we hypothesize:

- Both groups will exhibit significant improvements in the computational thinking post-test scores compared to pre-test scores.
- The students of the treatment group will exhibit no significant difference in achievement in terms of computational thinking post-test scores than those of the control group.

Participants

Initially, four university professors (subject experts) and four middle school teachers (non-subject experts) were expected to participate as instructors in the study. However, due to the realities of conducting a field experiment, the study was actualized with five subject experts teaching one class each and one non-subject expert teaching three classes. Table 1 shows the number of participants and the data collected.

Table 1. Subject Participation by Section

	Projected		Actual		
	Instructors	Students	Instructors	Students	**Usable Data**
Control	4	80	5	128	**123**
Treatment	4	80	1	67	**55**
Total	8	160	6	195	**178**

Students from three urban middle schools in Belize between the ages of 12 and 15, participated in the study. The students included both males and females of similar socioeconomic status. Table 2 shows the ratio of girls to boys. This ratio is roughly the average ratio in the country as boys tend to drop out of school at an earlier age than girls in Belize. All students speak English and are considered native English speakers by the school system of Belize. Some are bilingual, speaking both English and Spanish; students speak Spanish at home with their parents and conduct their studies in English. They are representative of the traditional urban middle school level school students in Belize.

Table 2. Distribution of participants according to gender and intervention group

	Treatment	Control	Total
Boys	24 (43.64%)	49 (39.84%)	73 (41.01%)
Girls	31 (56.36%)	74 (60.16%)	105 (58.99%)
Total	55	123	178

Material

The material utilized for the course is freely available online. One of the tools used to teach the students computer programming was the Scratch visual programming language site from MIT (scratch.mit.edu). Scratch is designed to teach programming to young students by having them design and build games and tell stories, making it an excellent tool for DBL.

The curriculum used, Creative Computing, was developed by Harvard University Graduate School of Education for the general public under a creative commons attribution-ShareAlike 4.0 international license. The daily activities in the curriculum include many SRL tools and strategies such as setting goals, verbalizing and recording content learned, and sharing projects with peers for feedback and improvement of projects. The curriculum includes DBL components throughout, with most sections including a game as a required task. The final project is a group hackathon with all groups designing and demoing a working computer program game or story. Figure 3 shows the Creative Computing Curriculum with the Scratch interface.

A website was designed to present the content of the curriculum in a day by day format, making it easier for the middle school teacher to learn simultaneously with the students.

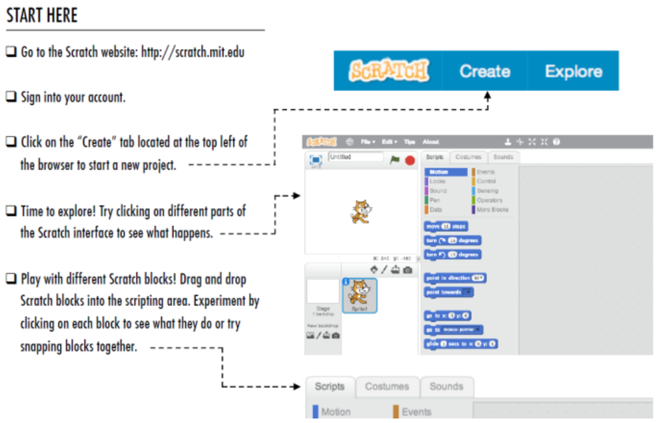

Figure 3. Example Creative Computing Curriculum showing the Scratch interface

Instruments

The Computational Thinking Test (CTt), was designed to measure a set of problem-solving skills that are beneficial to students who must thrive in the modern digital world driven by software. A definition and measurement of CT from a psychometric approach, has shown the test to be valid and reliable (Román-González et al., 2017). Pre and post-test measurements of computational thinking in our study utilized the CTt. Figure 4 shows an example of a CTt question.

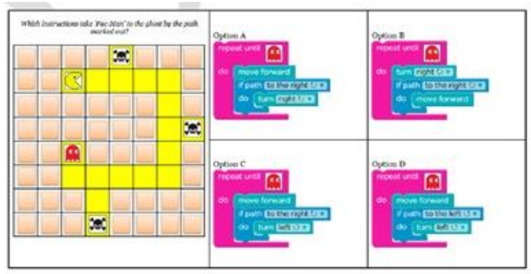

Figure 4. Example question on the CTt

Procedure

Three middle schools were approached and asked if the course could be offered every Friday for one fifty-minute class period to two separate classes. University computer science professors would teach one class, and the regular middle school teacher would teach a separate class. The principals and teachers agreed to the stipulations. However, when the classes started, three out of the four middle school teachers backed out of the study and did not teach coding to their classes. Table 1 shows the resulting participation. One middle school teacher taught three classes, and five university professors taught one class each.

Both the university professors and the middle school teacher utilized the online web portal and instructed the students to click on the appropriate day and follow the instructions provided. The instructors would follow along with the students to complete the tasks. As the university professors could complete the task at a faster pace they found more time to assist the students during the class. The middle school teacher took more time to complete the tasks, with much of the time lost spent making sure the students were moving along the steps correctly. The expectation was that there would be differences in the classes delivered by the subject experts and the non-subject expert, and the primary intention of the study was to observe how much these differences affected student learning outcomes.

Data Analysis

An X^2 test for independence was performed to compare the gender composition of the two groups. This test was used to investigate potential initial gender differences between the treatment and control groups. One-way between-groups ANOVAs, with gender and intervention group serving as the independent variable, was used to explore differences related to pre-test scores.

In each of the CTts (pre-test and post-test), the students' number of correct answers was his or her score in the respective test. Possible scores in each test ranged from 0 to 28 points. A 2 x 2 between-groups analysis of covariance (ANCOVA) was conducted to assess the effectiveness of the interventions on students' computational thinking skills. The independent variables were: (a) the type of intervention, which included two levels (subject expert instructor, non-subject expert instructor), and (b) gender. The dependent variable consisted of scores on the post-test CTt. Students' scores on the pre-test CTt served as a covariate in this analysis to control for eventual pre-existing differences between the groups. One-way between-groups ANOVAs that compared pre-test CTt scores for (a) the students of the treatment group and those of the control group, and (b) boys and girls were used to explore differences. The study's defined significance level was 0.05. All the analyses were performed using the SPSS version 20 statistical package.

RESULTS

Comparison of Group Compositions

As shown in Table 1, out of the 178 participants, 41.01% were male, and 58.99% female. The difference in the ratio of males to females was not significant ($X_{2.227}$, df = 1, p = 0.634). The existence of more females than males in both groups reflects a trend in the country where boys drop out of school at a higher rate than girls.

Comparison of Learning Outcomes by Gender and Intervention

There was no random assignment of an individual student to groups, so an analysis was first conducted to measure the difference in computational thinking before the course. Analysis of CTt pre-test scores showed no statistically significant difference in the performance on the pre-test ($F_{1, 176}$=.786; p=.377) between the students of the Treatment Group (M = 11.80, SD = 3.21) and the students of the Control Group (M = 11.28, SD = 3.74), indicating similar computational thinking abilities prior to the course despite age and gender differences of students, shown in Table 4.

Additional analysis was performed to verify that males and females assigned to the control group had no differences in computational thinking abilities. Analysis of CTt pre-test scores of the control group by gender showed no statistically significant difference in the performance on the pre-test ($F_{1, 121}$=.198; p=.657) between males (M = 11.47, SD = 3.84) and females (M = 11.16, SD = 3.69), indicating similar computational thinking abilities prior to the course. The same is true for the post-test scores ($F_{1, 121}$=.548; p=.460) between males (M = 13.90, SD = 3.93) and females (M = 13.39, SD = 3.56), indicating no significant differences in computational thinking between boys and girls in the control group after the course.

An analysis was also performed to verify that males and females assigned to the treatment group also had no differences in computational thinking skills. Analysis of CTt pre-test scores of the treatment group by gender showed no statistically significant difference in the performance on the pre-test ($F_{1, 53}$=.124; p=.726) between males (M = 11.63, SD = 4.09) and females (M = 11.94, SD = 2.38), indicating similar computational thinking abilities prior to the course. The same is true for the post-test scores ($F_{1, 53}$=.006; p=.941) between males (M = 14.08, SD = 4.87) and females (M = 14.00, SD = 3.43), indicating no significant differences in computational thinking between boys and girls, showing that girls increased their computational thinking skills at a statistically similar rate as boys during the course. Table 3 shows this.

Table 3. Performance Results of the CT Test by gender and group

			Mean	N	SD	Adjusted Means	Student's t	pre-post d	ANCOVA F	Global d
Boys	Control	Pre-Test	11.47	49	3.84				.013	.60
		Post-Test	13.90	49	3.93	13.83	3.688***	0.63		
	Treatment	Pre-Test	11.63	24	4.09					
		Post-Test	14.08	24	4.87	14.21	3.452**	0.55		
Girls	Control	Pre-Test	11.16	74	3.69				.143	0.63
		Post-Test	13.39	74	3.56	13.44	5.159***	0.61		
	Treatment	Pre-Test	11.94	31	2.38					
		Post-Test	14.00	31	3.43	13.90	3.205**	0.70		

*** p-value < .001; ** p-value < .01; * p-value < .05; Adjusted means using pre-test scores as covariate

Comparison of Learning Outcomes by Intervention Only

Upon completion of these analyses, we tested to see if students increased their computational thinking as a whole. When compared, pre-test and post-test scores for both groups showed a statistically significant difference in performance ($F_{19,\ 158}$=2.759; $p < .001$) with the treatment group improving by 2.24 points and the control group improving by 2.31 points supporting our first hypothesis that all students would increase their computational thinking skills. Figure 5 shows the increase in the CTt for both groups.

Finally, we conducted the focal analysis of the study, which was to determine if the two groups would increase their skill by the same amount. Analysis of the CTt post-test scores showed no statistically significant difference in performance on the post-test ($F1,_{176}$=.510; p=.476) between the students of the Treatment Group (M = 14.04, SD = 4.08) and the students of the Control Group (M = 13.59, SD = 3.70) indicating that both groups showed similar improvements in computational thinking ability after the course. Further analysis conducted on the post-test utilizing the pre-test as a covariant also indicated no statistical difference in learning outcomes between groups ($F_{1,\ 175}$=.128; p=.721). These results support our second hypothesis, which was our central hypothesis. Table 4 shows the performance results for both groups.

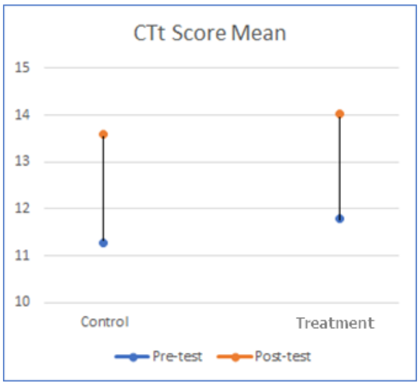

Figure 5. CTt score means pre-test, post-test for control, and experimental groups.

Table 4. Performance Results of the CT Test for the entire sample

		Mean	N	SD	Adjusted Means	Student's t	pre-post d	ANCOVA F	Global d
Control	Pre-Test	11.28	123	3.74	13.67	6.277***	0.62	.128	.61
	Post-Test	13.59	123	3.70					
Treatment	Pre-Test	11.80	55	3.21	13.87	4.716***	0.61		
	Post-Test	14.04	55	4.08					

*** p-value < .001; ** p-value < .01; * p-value < .05; Adjusted means using pre-test scores as covariate

DISCUSSION AND CONCLUSION

The quasi-experiment conducted for the research was carried out at three middle schools in Belize to introduce students to computer programming. Two groups separated the students: a control group, taught by university professors with experience teaching computer programming, and a treatment group, taught by a middle school teacher with no experience teaching computer programming. The course taught to both groups utilized standardized online material that incorporated SRL and DBL methods in its curriculum and delivered its content in an MIE environment. Both groups of students took a computational thinking test before and after the course. The results showed that both groups significantly increased their scores on the post-test as compared to the pre-test. Additionally, there were no significant differences in scores between the group that was taught by the university professors and the group taught by the primary school teacher.

While these findings are not statistically significant, they validate that, under the right circumstances, teachers can learn simultaneously with students when conducting an introduction to computer programming class. Also, the practical significance of the results is crucial for environments (such as developing countries) where there is a deficiency of trained computer science teachers because they demonstrate that similar learning outcomes are achievable for introductory programming and computational thinking courses by teachers with limited or no computer science training as those achieved by teachers with formal computer science training.

Limitations

The study was a quasi-experiment due to students not being randomly assigned to the control or treatment group. However, the assignment of whole classes to either the control or treatment groups was random. Of more significant concern were the changes to the number of instructors participating in the study. Of the proposed four university computer science professors and four middle school teachers, three middle school teachers declined to participate after initially agreeing. This fact poses a bias to the study in that only one out of four middle school teachers was willing to learn simultaneously with the students. It is noteworthy that this issue is also of concern in the developed world where facilitators at code.org refer to teachers leaning computer programing simultaneously with students' as "lead learners" and try to get teachers comfortable with the idea that they will be learning computer science alongside their students.

Future Studies

Of great interest would be to replicate the study while removing the bias of the self-selection of the middle school teacher encountered in this study. Would the learning outcomes be similar if three-fourths of the teachers who participate were not comfortable with the idea of leaning simultaneously with the students?

Conclusion

After presenting the findings of this study to administrators at middle schools in the Cayo District of Belize, ten out of the thirteen agreed to incorporate the computer programming curriculum for the 2017-2018 academic year. A study of this larger-scale pilot project could not only answer these questions but could lead to the curriculum being implemented countrywide by the year 2020, surpassing many developed countries trying to achieve similar goals.

REFERENCES

Alavi, M., & Leidner, D. E. (2001). Research Commentary: Technology-Mediated Learning-A Call for Greater Depth and Breadth of Research. *Information Systems Research*, 12(1), 1-10. https://doi.org/10.1287/isre.12.1.1.9720

Azevedo, R., Moos, D. C., Greene, J. A., Winters, F. I., & Cromley, J. G. (2008). Why is Externally-Facilitated Regulated Learning More Effective than Self-Regulated Learning with Hypermedia? *Educational Technology Research and Development*, 56(1), 45-72. https://doi.org/10.1007/s11423-007-9067-0

Barrows, H. S. (1985). How to design a problem-based curriculum for the preclinical years. *Springer Pub Co*. 8.

Brackmann, C., Barone, D., Casali, A., Boucinha, R., & Munoz-Hernandez, S. (2016). Computational thinking: Panorama of the Americas. *IEEE*, 1–6. https://doi.org/10.1109/SIIE.2016.7751839

Du, X., & Kolmos, A. (2009). Research on PBL practice in engineering education. *Sense Publishers.* https://doi.org/10.1163/9789087909321

European Commission. (2016). A New Skills Agenda for Europe. Working together to strengthen human capital, employability and competitiveness. COM (2016) 381(2). Retrieved from http://https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A52016DC0381

Inamdar, P., & Kulkarni, A. (2007). 'Hole-In-The-Wall' Computer Kiosks Foster Mathematics Achievement - A Comparative Study. *Journal of Educational Technology and Society*, 10 (2), 170-179. Retrieved from http://www.jstor.org/stable/jeductechsoci.10.2.170

Mitra, S., Dangwal, R., & Thadani, L. (2008). Effects of Remoteness on the Quality of Education: A Case Study from North Indian Schools. *Australasian Journal of Educational Technology*, 24 (2), 168-180. https://doi.org/10.14742/ajet.1219

Mitra, S., Dangwal, R., Chatterjee, S., Jha, S., Bisht, R. S., & Kapur, P. (2005). Acquisition of Computing Literacy on Shared Public Computers: Children and the Hole in the Wall. *Australasian Journal of Educational Technology*, 21(3), 407. https://doi.org/10.14742/ajet.1328

Mitra, S., Tooley, J., Inamdar, P., & Dixon, P. (2003). Improving English Pronunciation: An Automated Instructional Approach. *Information Technologies and International Development*, 1(1), 75. https://doi.org/10.1162/itid.2003.1.1.75

Piccoli, G., Ahmad, R., & Ives, B. (2001). Web-Based Virtual Learning Environments: A Research Framework and a Preliminary Assessment of Effectiveness in Basic IT Skills Training. *MIS Quarterly*, 401-426. https://doi.org/10.2307/3250989

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 1(14), 30. https://doi.org/10.1016/j.chb.2016.08.047

Román-González, M. (2015). Computational thinking Test: Design guidelines and content validation. In *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)* 2436-2444. https://doi.org/10.1016/j.chb.2016.08.047

Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Santhanam, R., Sasidharan, S., & Webster, J. (2008) Using Self-Regulatory Learning to Enhance E-Learning-Based Information Technology Training. *Information Systems Research*, 19(1), 26-47. https://doi.org/10.1287/isre.1070.0141

Schunk, D. H., & Ertmer, P. A. (1999) Self-Regulatory Processes During Computer Skill Acquisition: Goal and Self-Evaluative Influences. *Journal of Educational Psychology*, 91(2), pp. 251. https://doi.org/10.1037/0022-0663.91.2.251

Sealey, K. (2011). An Evaluation of Blended Learning Components of the Cisco Network Academy Using a Rasch Model. *ICNS 2011*.

Tsai, C. W., Shen, P. D., & Fan, Y. T. (2013). "Research Trends in Self Regulated Learning Research in Online Learning Environments: A Review of Studies Published in Selected Journals from 2003 to 2012. *British Journal of Educational Technology*, 44(5), 107-110. https://doi.org/10.1111/bjet.12017

Wan, Z., Compeau, D., & Haggerty, N. (2012). "The Effects of Self-Regulated Learning Processes on E-Learning Outcomes in Organizational Settings. *Journal of Management Information Systems*, 29(1), 307-340. https://doi.org/10.2753/MIS0742-1222290109

Yomiuri Shimbun. (2016). Plan to make programming mandatory at schools a step to foster creativity. Translation Retrieved August 10, 2017, from http://www.bgdailynews.com/plan-to-make-programming-mandatory-at-schools-a-step-to/article_ed336f46-1c3e-11e6-ba3e-10604b9f1f4a.html