A Mooc-Based Computer Science Program for Middle School: Results, Challenges, and the Covid-19 Effect

Shai Perach and Giora Alexandron

A MOOC-BASED COMPUTER SCIENCE PROGRAM FOR MIDDLE SCHOOL: RESULTS, CHALLENGES, AND THE COVID-19 EFFECT

Shai Perach
Weizmann Institute of Science
Herzl 234, Rehovot, Israel

Giora Alexandron
Weizmann Institute of Science
Herzl 234, Rehovot, Israel

ABSTRACT

In an attempt to pave the way for more extensive Computer Science Education (CSE) coverage in K-12, this research developed and made a preliminary evaluation of a blended-learning Introduction to CS program based on an academic MOOC. Using an academic MOOC that is pedagogically effective and engaging, such a program may provide teachers with disciplinary scaffolds and allow them to focus their attention on enhancing students' learning experience and nurturing critical 21st-century skills such as self-regulated learning. As we demonstrate, this enabled us to introduce an academic level course to middle-school students.

In this research, we developed the principals and initial version of such a program, targeting ninth-graders in science-track classes who learn CS as part of their standard curriculum. We found that the middle-schoolers who participated in the program achieved academic results on par with undergraduate students taking this MOOC for academic credit. Participating students also developed a more accurate perception of the essence of CS as a scientific discipline.

The unplanned school closure due to the COVID19 pandemic outbreak challenged the research but underlined the advantages of such a MOOC-based blended learning program above classic pedagogy in times of global or local crises that lead to school closure. While most of the science track classes seem to stop learning CS almost entirely, and the end-of-year MoE exam was discarded, the program's classes smoothly moved to remote learning mode, and students continued to study at a pace similar to that experienced before the school shut down.

KEYWORDS

Computer Science Education; Blended Learning; MOOCs; K-12 Science Education.

1. INTRODUCTION

There is a growing realization that K-12 Computer Science Education (CSE) should evolve from a subject for a fortunate few to an opportunity for all [1]. However, attempts to broaden CSE face significant challenges such as acute lack of teachers [2] and equality issues related to gender and socioeconomic differences [3, 4].

In this research, we performed a preliminary evaluation of a suggested solution for this multi-facet challenge: A Blended-Learning (BL) program based on an existing academic "Introduction to Computer Science" Massive Online Open Course (MOOC). Ideally, the academic MOOC will substantiate the required disciplinary depth, enabling the teacher to focus on blending and orchestrating online and offline activities and mentoring and enhancing students' learning. As part of the professional development and training for delivering

the program, the teachers would receive a collection of suggested offline class activities, exams, program delivery planners, and guidance on optimizing the program's facilitation. We refer to this collection as Teacher's Toolkit. By building the program around an appropriate academic MOOC, we provide new-coming teachers with disciplinary scaffolds, creating a scalable introductory CS program while increasing the disciplinary depth. By that, this program aims to address the challenge of the acute lack of CS teachers.

Regarding equality, such a program could spread more easily to schools from low socioeconomic environments, which typically have less access to CS teachers. Additionally, facilitating such a program as an obligatory course in middle-school will also reach an adequate female participation rate. Thus, such a solution has the potential of also mitigating the equality issues mentioned above. Economic-wise, such a program leverages the massive investments in academic MOOCs to the benefit of K-12 education.

This research aimed to be a preliminary evaluation of such a solution. We focused on ninth-graders in science-track classes, chose an appropriate academic MOOC, and then developed the program's principles and initial version. Our main goal was to study whether students who participated in the program can develop a meaningful understanding of core CS ideas and a more accurate perception of the nature of CS. Specifically, our research is guided by the following research questions (RQs):

1. Do students who completed the program achieve the disciplinary learning goals, as defined by the course developers (RQ1)?
2. What is the relationship between participation in the program and changes in students' perceptions of the discipline of computing (RQ2)?

To study these questions, we piloted the program in two classes. The original research design included a control group consisted of ninth-grade classes in this track, which also learn the same MOOC in a BL setting, yet with no pre-designed program (students who learn the MOOC in-class hours with the teacher present for assistance). An additional control group consisted of classes who learn the standard MoE CS curriculum for ninth-grade. However, the transfer to remote learning due to the COVID-19 pandemic severely affected these classes, and eventually, they stopped learning. While this limited our ability to follow a comparative research design as was planned, it emphasized the teacher's critical role in the program and shed light on the importance of developing programs that can flexibly switch between physical and remote learning.

The rest of the paper is organized as follows. Following a short Background in section 2, we describe the program design in section 3. In Section 4, we describe the Research Methodology, and in Section 5, we describe the Results, followed by a Discussion and Conclusions in Sections 6 and 7, respectively.

2. BACKGROUND

MOOC-Based blended learning programs

In many cases, the MOOC used in blended-learning programs is pre-designed to be used as the online part of a blended learning program, sometimes referred to as Blended MOOCs or bMOOCs [6]. For example, Rayyan et al. [7] described such design and development of a blended learning program for introductory physics at Massachusetts Institute of Technology (MIT), where a dedicated MOOC was developed as part of the program development. In the context of K-12, a bMOOC-based program that is closely related to our study is that of Grover, Pea, and Cooper [9]. Grover et al. designed a blended learning program to prepare and motivate middle school learners for future engagement with algorithmic problem-solving. They created a dedicated MOOC as part of the program design and development. The goal was to design a MOOC that encodes both the content and the pedagogical content knowledge (PCK) to enable a non-CS teacher to facilitate blended learning delivery of the program. They reported encouraging results, with students meeting the desired learning goals.

The above examples of MOOC-based BL programs created a *tailored* MOOC as part of the program development. However, designing and producing a dedicated MOOC requires significant resources and is time-consuming [10]. With the explosion of available academic MOOCs, it makes more sense to locate an adequate existing MOOC. Previous research on MOOC-Based blended learning was mainly conducted in the context of higher education [5, 6, 7, 8]. Perez-Sanagustin et al. [11] proposed the H-MOOC framework, which details

indicators to measure the impact of hybrid initiatives at higher education in which locally produced and third-party MOOCs are reused and integrated into traditional courses.

But academic MOOCs usually target undergraduate level and may not be adequate for middle school students. This rationale led HarvardX to adapt its renowned Introduction to CS MOOC, CS50, for high school students [12]. In this research, our goal was to evaluate whether, through a careful design of a blended learning program, an *existing academic* MOOC may be leveraged, as is, for creating a meaningful learning experience for middle school students.

The MOOC

The MOOC we chose for the program is First Steps in Computer Science and Programming in Python [13]. It is an intro to CS elective course for undergraduates who are not majoring in CS. The course was developed by Prof. Benny Schor and Dr. Amir Rubinstein, two highly experienced CS scientists and educators from Tel Aviv University School of Computer Science. The course production and digital pedagogy guidance were done by TAU Online: Tel Aviv University Innovative Learning Center [2].

We chose an academic course because of its disciplinary depth, which is felt throughout the course. Non-academic MOOCs that we considered were more of python coding courses. Additionally, we hoped that the ninth-grade students' successful learning of an actual academic course would provide them with an added benefit of self-efficacy for academic studies.

We chose the specific course because of its content, pedagogy, and production quality. The first third of the course almost entirely overlaps the CS curriculum for eighth & ninth-grade science track classes in Isreal. The second half of the course extends the MoE CS curriculum. It illustrates the versatility of CS by providing an initial hands-on exposure to a selection of four advanced topics in CS: encryption, error detection & correction (communication), image representation & processing, and complexity of computations. Overall, we found that the MOOC content gives a balanced and friendly, yet meaningful, introduction to computer science as a discipline and computational thinking and python programming.

Above all, we gave extra care for choosing a highly engaging course. This MOOC immediately stood out as such.

3. PROGRAM DESIGN

The BL program targets ninth-grade class in a science track and assumes a gross class time of 90 minutes a week. Due to setup time related to getting all students on computers in the lab, the result is a net time of 60-70 minutes a week. The program also assumes additional off-class study (homework) of 30-45 minutes a week. The offline parts, e.g., class activities, discourse, and direct teaching, consume approximately 35% of class time on average. The rest of the class time and the weekly off-class study time are dedicated to self-learning from the MOOC and other digital resources.

Learning Goals

The program inherits the disciplinary learning goals of the academic MOOC and extends them. The MOOC makes an Introduction to CS and Programming Fundamentals with the python programming language. Following successful completion of the MOOC, students should:

- Understand and write short programs in python using the fundamentals of programming such as variables, operators, conditional execution, iterative execution, and functions.
- Understand and use basic data types, strings, lists, and two-dimensional lists.
- Understand the notion of algorithm complexity, error correction, encryption, picture representation, picture manipulation, problems with no efficient solution, computational graphs, and artificial intelligence.

---

[2] The course team, and its university, were not involved in this research, except for providing the end-of-course exam and statistics on the results of the undergraduates who took this course for credit.

In addition to the MOOC's learning goals, the program also defines a goal of constructing a more accurate perception of CS as a problem-solving discipline that uses the computer as a tool to solve real-world problems in diverse fields. This is opposed to conventional naïve notions that are computer-centric (CS is about making/fixing/studying computers) or programming-centric (CS == programming) [9, 14].

The program also aims to develop self-learning skills that leverage up-to-date digital resources. In addition to MOOCs, the program seeks to expose the students to self-learning through engaging popular science podcasts, documentaries, and YouTube channels.

Lastly, the program aims to encourage and develop social/peer learning practices as learning strategies.

Redefining the teacher's and the students' roles

Building the BL program around an academic MOOC, we had the opportunity and need to redefine the teacher's and the students' roles. In a classic school pedagogy, one of the teacher's primary responsibilities is to drive the students' cognitive learning process. This process is usually conducted via a series of actions and reactions to build the motivation for a new concept, concept introduction, connecting to previously learned concepts, concept illustration, exercising the new concept, providing feedback on students' performance, etc. An effective academic MOOC will usually perform such a series of actions (see for example [8]). Specifically, the MOOC chosen for this program is doing an excellent job at it. Our program design urges the teacher to embrace that and delegate a significant part of driving the cognitive learning process to the MOOC. That should enable the teacher to dedicate more attention to more meta-level and social aspects and focus on fostering self-learning, collaboration, and communication –crucial 21st-century skills that typically (at least from our experience) receive less attention in CS classes of this program. Towards that end, we were obliged to define what we believe should be the ideal new roles and class dynamics between the teacher, the students, and the MOOC.

The ideal interplay between the students, the MOOC, and the teacher

The Community of Inquiry (CoI) framework [15] aims to create a meaningful learning experience by developing three interdependent presences – social, cognitive, and teaching. Inspired by this framework, we treated the cognitive presence, teaching presence, and social presence as three pillars for the educational experience; Pillars that construct the ideal roles and interplay between the three entities of the BL program: the students, the MOOC, and the teacher (Fig. 1).
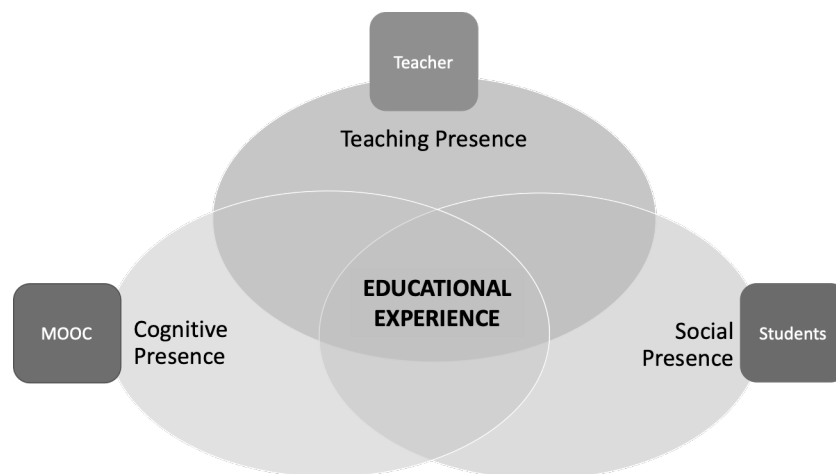

Fig.1: Interplay between the students, the MOOC, and the teacher - Inspired by the CoI framework

As the name 'MOOC-based' suggests, the design assumes that the MOOC is anchoring the Cognitive Presence while the teacher only enhances it. The teacher carries the Teaching Presence pillar. The teacher is also responsible for igniting a prolific Social Presence. However, the teacher should not attempt to carry most of the weight of maintaining a productive social climate. Through the teacher's leadership for setting appropriate class norms, the social presence pillar should eventually be assumed by the students.

We use the terms Pillar to convey that the responsibilities for the presences are not dichotomic. Beyond its assigned presence (teaching, cognitive, or social), each entity (teacher, MOOC, or students) is also responsible for enhancing the other two presences. To that end, each student is a teacher, and the students also enhance the cognitive presence. The teacher has a crucial role in supporting the social climate and a key role in augmenting the cognitive presence. Lastly, the MOOC also organizes the learning process and may even facilitate discussion, for example, by posting a non-trivial open question to be answered on the MOOC's forum.

We envisioned the roles and dynamics portrayed here as the ideal dynamics for the program facilitation. These new roles for the teacher and the students are very different from those assumed in conventional pedagogy. The teacher should not expect or be expected to teach. In this new role, the teacher's primary responsibility is to build and maintain high motivation for learning the subject matter and create learning opportunities for the students. He/she is also responsible for nurturing students' skills that will enable them to cease those learning opportunities. On the same token, students should not turn to the teacher as the first resource for questions on the subject matter. In their new role, students should first discuss with their peers and seek advice there.

It is implausible that these roles and interplay would settle by themselves. The teacher's responsibility is to drive the classroom to this state through purposeful actions and reactions taking place mainly in the first three months of the program. To develop this mindset, teachers will have to go through proper training and be provided with a pre-designed learning program that includes various class activities, tools, and advice for effective facilitation. Alongside this research, we have developed a 'Teacher's Toolkit' for the program consisting of all the offline class activities, other online learning resources such as specific podcast chapters, popular science programs, and YouTube science teaching channels. The toolkit also includes several midterm formative assessment exams, a tool for planning the program facilitation along the year, and advice on optimizing the program's delivery.

4. METHODOLOGY

The research population consists of ninth graders from science track classes. The research sample includes 55 students from two classes located in two public schools from urban cities classified as middle- to high middle-class economic status. The lead researcher taught one of these classes and mentored the other teacher on facilitating the learning program. Fig. 2 illustrates the research design

The summative assessment exam (RQ1)

To assess the achievement of the disciplinary learning goals, we administered the same pencil-and-paper summative exam given by the university to undergraduates who take this MOOC for credit (provided to us by the course team). This exam serves two purposes. First, it enables us to evaluate whether the program students meet the learning goals defined by the course team. We recall that meeting these goals award the undergrads academic credit. Second, it enables us to compare the achievement of the program's students to those of the university students.

Given at the end of each semester, the university's summative exam for the course followed the same structure (but not the same items) across all test dates: 25 multiple-choice questions; three questions per each of the eight units of the MOOC, plus another general question. The questions are designed to measure conceptual understanding of the course material on different levels.
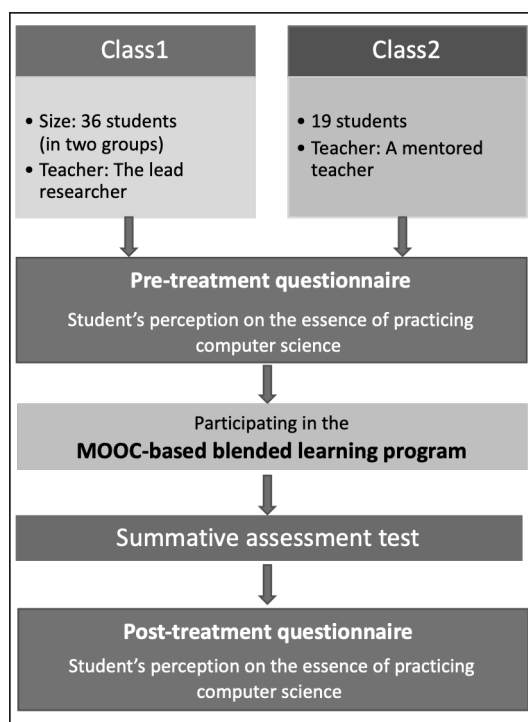

Fig. 2: Graphical illustration of research design

Perceptions of the discipline of computing (RQ2)

To study the influence on student perceptions, the students answered a questionnaire at the beginning of the school year (pre) and again at the end of the school year (post). The part of the questionnaire that refers to students' perception of the discipline of computing includes the following question (translated):

Professor Noga Alon from Tel Aviv University is a world-renowned and one of the country's most respected computer scientists. Out of the list of items below, what can be assumed that professor Alon can do well, even if it is not necessarily what he does? Mark up to 4 items (you may mark less than 4):

1. *Design a computer*
2. *Fix a computer*
3. *Play computer games*
4. ***Code in some computer language (e.g., Java, Python, C#)***
5. *Install the Windows operating system on a computer*
6. *Build mobile apps.*
7. *Build a computer*
8. ***Cooperate with other computer scientists***
9. ***Use the computer as a tool for problem-solving***
10. *Build a website*

This question is a variation of the open-ended question "What does a computer scientist do?" used by Grover et al. [9]. However, we found it more appropriate to use a closed-form rather than an open-ended one for our purposes. Out of the ten choices, three (marked bold) are correct, and seven are distractors. The distractors represent common misperceptions identified by Grover et al. [9]. as common among middle-school students in the US. These are 'CS is about computers' (1,2,7), 'CS is about coding' (6, 10), and 'CS is about IT' (5).

5. FINDINGS

Achievement of disciplinary learning goals (RQ1)

At the end of the blended learning program, the students took the summative test given to Tel-Aviv University's undergraduates upon completing the MOOC. Undergraduates who are non-CS majors can take this MOOC and receive credit after taking the exam on-campus and achieving a passing grade (60). Using this criterion, 83% (44 out of 53) of the participating middle-schoolers met these academic standards defined by the course team.

We also compared the program students' achievements to those of the university undergrad students (Fig. 3). The participating middle-schoolers achieved a median score of 84 and an average score of 79.2 (std. dev: 18.8). In the recent three semesters, this exam was given at the university. Overall, N=221 undergrad students took this test, with a median score of 88 and an average score of 83 (std. dev: 17). To compare this distribution to that of the program participants, we first checked the assumption of equal variances using an F-test. The F-test p.value = .22, so we accepted the null hypothesis that the variance is similar among the groups. We, therefore, may use the appropriate t-test. The t-test results in p.value = .14; hence we accepted the null hypothesis and concluded that it appears as if these two sets of results are coming from the same distribution.

	Program (n=53)	University (n=221)
Median	84.0	88.0
Average	79.2	83.2
Std. Dev	18.8	17.4

Fig. 3:Summative exam grades distribution

To conclude the results regarding our first research question, we found that the vast majority of students who participated in the program met the academic standards defined for the course. Their achievements were on par with the university students taking this course for credit.
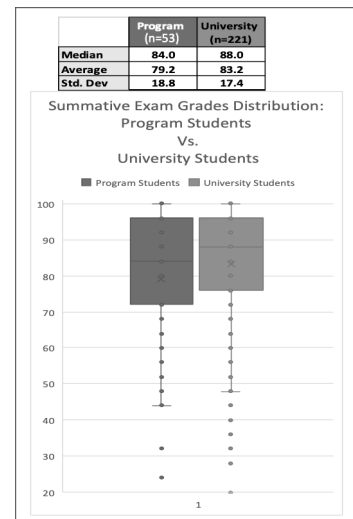
Changes in students' perceptions of the discipline of computing (RQ2)

In the pre/post questionnaire, the students were given a list of ten tasks and were asked to mark up to four that he/she believes that a computer scientist should master. Three tasks in the list reflect a good understanding of CS as a scientific discipline, while the rest are related to common misperceptions of the essence of CS.

To assess the effects on students' perceptions, we calculated the proportion per task. That is the number of students who selected the specific task divided by the total amount of students who answered the questionnaire. A two-sample proportion test per task in the list compared the pre-test results to those in the post and indicated if the change was statistically significant.

Figure 4 illustrates the change in proportions for the three tasks reflecting a good understanding of the essence of practicing CS. All three tasks show an increase in proportions. The task "program in some computer language" was chosen by 96% of the students on the pre-test, so there was very little room for increase. However, the other two tasks, "use the computer as a tool for problem-solving" and "cooperate with other computer scientists", were less trivial for the students before the program. These two tasks show a very significant increase in the post questionnaire. In contrast, all other tasks related to common misperception showed a decrease in the post-test (see Figure 5), though not always significant.

To conclude, regarding our second research question, we found that the students who participated in the program tended to develop a more accurate perception of the essence of CS as a scientific discipline.
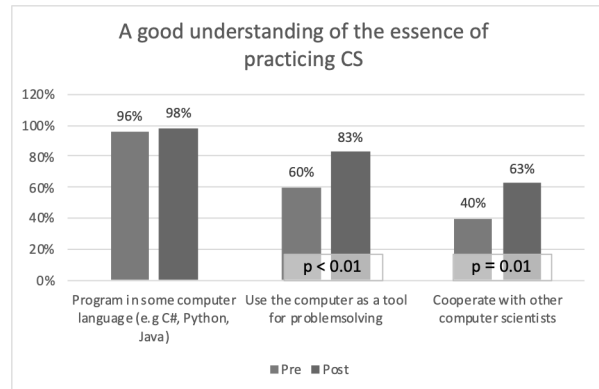

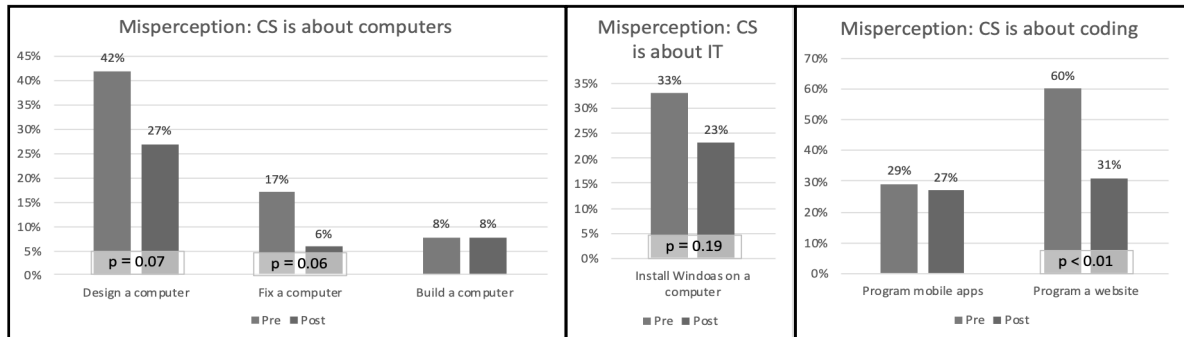Fig. 4: Changes in proportions in tasks reflecting a good understanding of practicing CS

Fig. 5: Proportions change in the various categories of common misperceptions about CS

6. DISCUSSION

This research explored a way of harnessing existing academic MOOCs for the benefit of CSE in K-12, and specifically, on middle school students. To achieve these goals, we selected an appropriate academic MOOC and carefully designed a system of scaffolds around it in the form of a blended-learning program. Scaffolds were planted for the students and the teacher. We piloted the program in two scientific track ninth-grade classes. Our findings demonstrate that ninth-graders who participated in the BL program successfully completed the academic MOOC, achieving summative assessment scores on par with undergrads taking this MOOC for credit and developing a more accurate perception of the essence of CS as a scientific discipline. Following is a discussion on each of these findings.

High achievements of academic goals

The well-validated Community-of-Inquiry framework inspired our systematic approach for re-imagining and designing the teacher's role in such a program. The MOOC was anchoring the cognitive presence while the teacher landed the teaching presence, enhanced the cognitive perspectives, and ignited a prolific social climate.

At the core of the teaching presence, the program gave the teacher tools to organize and drive the learning process. These were manifested through the weekly recommendations on units to learn, followed by opening the next class meeting with a discourse or activities related to these units' topics. Yet, the principal pace-setters milestones were the four pre-scheduled assessments spread throughout the program. Some students abused the liberty of learning at the time and place of their convenience for procrastination. However, an approaching assessment motivated them to close the gap and align with the rest of the class.

The program also provides the teacher with tools and advice to ignite and nurture an open and collaborative social climate. As these social norms evolved, we witnessed interactions where learners progressively render\employ scaffolds to\from their peers. Eventually, these peer scaffolds solved practically every cognitive and technical issue related to the MOOC's academics.

Lastly, at the overlap of the teaching and cognitive presences, the program provided tools and guidelines to instill purpose, inspire, and empower. Class group activities and the discourse around the popular science materials inspired many of the students in the class. We frequently got positive feedback and requests for more of these. We believe that all these played a crucial role in assisting the ninth-graders in completing the academic MOOC and successfully achieving its academic goals.

Student perceptions of Computer Science

As we demonstrated in the findings on student perceptions, the students who participated in the program developed a more accurate perception of the nature of CS as a scientific discipline that its essence is computing (rather than IT-related skills). We believe that the choice of building the program around an academic MOOC, delivered by `real' scientists (as opposed, for example, to a general coding MOOC), served a great deal in supporting students in constructing a better perception of the essence of CS as a discipline. The developers and instructors of the MOOC are two CS scientists who are also deeply involved in CSE. Their disciplinary language and argumentation manifest that CS is about problem-solving through computational thinking, with coding as one part of the process of solving a computational problem using a computer. This difference became evident when we compared this academic MOOC with a Python coding MOOC, developed and served by experienced python developers.

The implications of the COVID-19 shutdown: Change of research plan and new insights

Initially, this research was designed as a comparative study. Aside from the two treatment classes analyzed here, we followed four additional ninth-grade science classes as a control group. One of these four learned the same MOOC in a classroom, yet with no pre-designed program (in a format that can be described as 'online

learning in-class hours with the teacher present for assistance). The other three studied the standard MoE CS ninth-grade CS program.

In the middle of the year, all schools were shut down due to the COVID19 pandemic outbreak. In the two pilot classes, only the program facilitation shifted to Zoom, and the MOOC learning continued asynchronously between the weekly Zoom sessions. However, none of the control group classes succeeded in switching to distant learning. In these classes, CS studies decayed to a complete halt a few weeks into the school closure. The closure thus affected the research design. However, it also underlined the advantage of such a BL program in building student agency and also the flexibility of such a program to operate in different learning conditions.

The classes that followed (pre-shutdown) conventional teaching failed to switch to remote learning simply because the teachers were unprepared, with no digital resources and no experience in teaching CS remotely. Regarding the class that followed the 'online learning in class hours format, we believe that the lack of teaching presence, as defined by our program, was a significant factor explaining this class' failure to switch to remote learning, eventually failing to complete the MOOC (in fact, course data shows that it actually lagged behind the experiment classes also before the school shutdown). We believe that this underlines the value of a pre-designed BL program and the redesign of teacher role to provide meaningful teaching presence in blended learning courses, even if they base most of the cognitive aspects of the content delivery on digital resources.

Limitations

Significant parts of the program were developed as the program's pilot unfolded. That includes many off-MOOC materials and activities, and more importantly, insights into effective program facilitation. As a result, the program facilitation was not uniform throughout the year, nor between the two classes. This is expected, given that this is just the first iteration of development and implementation. However, it also made clear that the program can be improved if we allow it further iterations. Hence, we treat the satisfying results as a lower bound of what such a program could achieve. Also, the results are based on a small sample (two classes) and a non-comparative design.

Future research

Aside from further developing the program, we believe that the main challenge is studying the program's scalability, primarily whether it can be delivered effectively by new-coming CS teachers and even by non-CS teachers.

7. SUMMARY AND CONCLUSION

In an attempt to pave the way for more extensive coverage of meaningful CSE in K-12, this research developed and made a preliminary evaluation of a potential solution – a Blended-Learning Introduction to CS program based on an existing academic MOOC. For this purpose, we developed the principals and initial version of such a program, targeting ninth-graders in science-track classes. Our goals were evaluating the success of this program on two aspects: meaningful learning of core CS ideas and developing an accurate perception of the discipline of computing. The former was evaluated using the same end-of-course exam given to undergraduate students who take the course for credit. The latter was measured using a questionnaire that was developed for this purpose by combining and adapting standard instruments.

The program was piloted in two classes. Overall, we found that the middle-schoolers who participated in the program achieved academic results on par with undergrads taking this MOOC for academic credit. Participating students also developed a more accurate perception of the essence of CS as a discipline.

Scalability was a primary design goal of the program. Yet, testing and studying the program's ability to scale remains a question for further research. The successful transition of the pilot classes to remote learning, especially when contrasted with the control classes who fail to do so, underlined the program's ability to operate on different learning conditions while highlighting the centrality of the teaching presence.

ACKNOWLEDGEMENT

REFERENCES

[1] K-12 Computer Science Framework Steering Committee, " K-12 Computer Science Framework," ACM, New York, NY, 2016.

[2] I. Zur Barguri, B. Haberman, A. Cohen, O. Muller, D. Zohar, D. Levy and R. Hotoveli, "Implementing a new computer science curriculum for middle school in Israel," in *Frontiers in Education*, 2012.

[3] M. Johnson, "Computer science education still has diversity gaps," 2020. [Online]. Available: https://blog.google/outreach-initiatives/education/computer-science-education-research/. [Accessed 11 2 2021].

[4] J. Margolis, Stuck in the shallow end: Education, race, and computing, Cambridge, Mass.: The MIT Press, 2010.

[5] A. Bralic and B. Divjak, "Integrating MOOCs in traditionally taughtcourses: achieving learning outcomes withblended learning," *International Journal of Educational Technology in HigherEducation*, vol. 15, no. 2, 2018.

[6] T. Alghamdi, W. Hall and D. Millard, "A classification of how MOOCs are used for blended learning," in *Proceedings of the 2019 4th International Conference on Information and Education Innovations*, New York, NY, USA, 2019.

[7] S. Rayyan, C. Fredericks, K. Colvin, A. Liu., R. Teodorescu, A. Barrantes, A. Pawl, D. Seaton and D. Pritchard, "A MOOC based on blended pedagogy," *Journal of Computer Assisted Learning,* vol. 32, pp. 190-201, 2016.

[8] K. F. Colvin, J. Champaign, A. R. Liu, D. E. Pritchard, Z. Qian and C. Fredericks, "Learning in an introductory physics MOOC: All cohorts learn equally, including an on-campus class.," *The International Review of Research in Open and Distributed Learning,* vol. 15, no. 4, 2014.

[9] S. Grover, R. Pea and S. Cooper, "Designing for deeper learning in a blended computer science course for middle school students," *Computer Science Education,* vol. 25, no. 2, pp. 199-137, 2015.

[10] P. M. Nissenson and A. C. Shih, "MOOC on a Budget: Development and Implementation of a Low-cost MOOC at a State University Paper presented at 2015 ASEE Annual Conference & Exposition," Seattle, 2015.

[11] M. Perez-Sanagustin, I. Hilliger, C. Alario-Hoyos, C. Delgado Kloos and S. Rayyan, "H-MOOC framework: re-using MOOCs for hybrid education," *Journal of Computing in Higher Education,* vol. 29, no. 1, pp. 47-64, 2017.

[12] E. Carvalho and D. J. Malan, "This was CS50 AP 1617," 28 April 2018. [Online]. Available: https://medium.com/@cs50/this-was-cs50-ap-1617-39f02c275a87.

[13] B. Schor and A. Rubinstein, "First steps in computer science and programming in Python.," 2018. [Online]. Available: IL: https://campus.gov.il/course/course-v1-tau-acd_tau_cs101x.

[14] R. Taub, M. Armoni and M. Ben-Ari , "CS unplugged and middle-school students' views, attitudes, and intentions regarding CS," *ACM Trans. Comput. Educ.,* vol. 12, no. 2, 2012.

[15] R. D. Garrison and N. D. Vaughan, Community of inquiry and blended learning, San Francisco, CA: John Wiley & Sons, Inc., 2008, p. 19.