Journal of Problem Based Learning in Higher Education

VOL. 11, No. 3, 2023 – Page 130-144
10.54337/ojs.jpblhe.v11i3.7390

Using Scratch to Teach Coding in Massive Online Open Courses: A Systemic Analysis

*Joao Alberto Arantes do Amaral **

ABSTRACT

In this case study, we present our findings regarding a massive open online Scratch programming course. The course, which followed a project-based learning approach, was delivered from July 4 to 30, 2022 to 186 students in Brazil. The students were challenged to develop individual coding projects. Our research goal was to investigate teaching and learning course dynamics. We followed a convergent parallel mixed-method approach. We collected quantitative and qualitative data by means of questionnaires. We were able to identify five intertwined feedback loops that drove the educational process. Our main findings are as follows: 1) The development of coding skills was driven by the effort of watching video-lectures, remixing of peers' codes, and by sharing knowledge between the students. 2) The project-based learning approach created opportunities for the students to collaborate and exchange ideas.

Keywords: Massive open online course, Scratch, Systemic analysis

INTRODUCTION

The Federal University of Sao Paulo (thereafter UNIFESP) has an online extension program that aims to provide free educational opportunities to the communities. The course *Introduction to Programming with Scratch* was designed to teach the main concepts of coding using the Scratch programming language. The target audience were children aged from 5 to 14 years old, although the course was open to anyone who wished to enroll.

* Joao Alberto Arantes do Amaral, Federal University of Sao Paulo (Unifesp Osasco), Brazil
Email: joaoalberto.arantes@gmail.com

The course was designed to work in an asynchronous way: it has a virtual learning environment (Google Classroom) and a website with 96 pre-recorded video-lectures (on average, each 10 minutes long).

The course was divided into four modules, each with a set of videos and exercises. We suggested that the students work on one module per week. However, since all modules were available to the students from the first day of te course, they could study at their own pace.

In the first module, we presented the basics of the programming environment and how to work with sprites and backgrounds. We also taught the input and output functions.

In the second module we explained the basics of the programming: loops, conditionals, boolean operators, mathematical operations, and string and list manipulation. In the third module, we taught the concepts of global and local variables, subroutines (blocks), clones, backpacks, and concurrency. We also explained the basic concepts involved in creating games. In the fourth module, we taught how to create animations, geometric shapes, and interactive stories.

In each module we offered practical challenges, requesting the students to work on a given code and remixing it (changing and improving the code by adding new functionalities).

The course lasted four weeks: in the first week of the course the students were asked to choose an individual project theme. The students were free to choose the theme of their own preference: it could be an educational game, an animated card, an interactive story etc. – anything that they were able to create using the Scratch programming language.

In the following weeks the students were asked to develop their projects with the support of the professor and with contributions from other students.

The course had a supportive virtual learning environment (VLE), a Google Classroom. In this platform the students were able to follow each other's projects, interact by making comments and remixing their codes. The students were also able to clarify their questions with the professor.

LITERATURE REVIEW

Scratch is a visual programming language developed by the MIT Media Lab in 2007. Scratch allows the students to create games, animated stories, animated cards, animated dance contests, virtual tours, newsletters, interactive stories, interactive science simulations, tutorials, and music (Brennan, 2011; Maloney et al., 2010). In addition, Scratch also allows the users to import music clips and 2D images (photos and pictures), record voices and sounds, and create graphics and drawings (Resnick & Rusk, 2020).

The MIT Scratch environment has a set of tools that allows the students to program, exchange ideas, work collaboratively, reuse other users' codes (remix) and program external devices such as microcontrollers and robotic kits (Kordaki, 2012; Resnick et al., 2009). The MIT Scratch environment has thousands of users and communities of people (most of them children) from all over the world (Robinson & Resnick, 2017).

Scratch has been used in schools to introduce the concepts of coding, mathematics (Dohn, 2020), science (Yamamori, 2019), music videos (Fields et al., 2015), artificial intelligence (Estevez et al., 2019) and educational robotics applied to STEM (Plaza et al., 2019). Researchers point out that people who learned Scratch as its first programming language have more facility in learning more advanced topics in computer science (Amoni et al., 2015).

Scratch was created with the goal of developing the students’ programming skills and creative learning (Fagerlund et al., 2021; Sefton-Green, 2011; Su et al., 2022). Some authors (Beghetto, 2021; Bustillo & Garaizar, 2016) point out that creative learning is a form of learning where the students have freedom to define whatever they want to create (projects, products, toys) and figure out their own ways of doing so.

The scholars involved in the creation of the MIT Scratch environment (Robinson & Resnick, 2017) suggest that to promote creative learning the students should work on projects that they are truly interested in, collaborating with peers, sharing ideas and knowledge, and doing playful experimentations.

In our course, we created a Google Classroom forum in order to allow the students to share their project ideas, receive feedback from the participants and from the professor. More than that, we encouraged the students to create an account on the MIT Scratch environment. By so doing, they were able to share projects with and to remix projects from people from all over the world.

Researchers point out that project-based learning (thereafter PBL) is an educational approach that is suitable to promote creative learning (Pirker et al., 2016) since it allows students to work collaboratively in projects, sharing knowledge and creating meaningful artifacts (Arantes do Amaral et al., 2015). Nowadays, scholars are using PBL integrated with Scratch in order to teach programming (Wang et al., 2014), to promote mathematical computational thinking (Hadi & Atiqoh, 2021), and to foster creative learning (Husna et al., 2019). Although there is a reasonable amount of research covering the use of Scratch combined with te PBL approach, it seems that there is a lack of information on the dynamics of MOOCs that teach how to program using Scratch. Our research question then became “What are the dynamics of learning present in MOOCs that teach students to program using Scratch?”

METHODS

Research Design

We followed a convergent parallel mixed method approach. In this approach, the quantitative and qualitative data are collected simultaneously (Creswell, 2013; Kennedy & Edmonds, 2016). Each type of data is initially examined separately; after that, they are analyzed together (Kennedy & Edmonds, 2016). By comparing both types of data we were able to have a better comprehension of the teaching and learning dynamics presented in the course.

Participants

Three hundred and forty-five (345) students enrolled for the course. However, 159 never participated, and only 186 participated in the first course activities. Seventy-four (74) students finished the course.

Of the students who finished the course, 55% were female and 45% male. The youngest was 7 years old, the oldest 61 years old. There were 16 children (aged less than 18 years old) and 57 adults. Of the adults, 59.6% have graduate degrees.

Fifty percent (50%) of the students were between 21 and 46 years old, thus the interquartile range was 25 years. The mean age was 33.8 years, the standard deviation was 15.6 years.

The course was opened to students from all over the country, although most of the students were from the State of São Paulo (90.5%). We also had students from seven other States (Rondônia (1.4%), Rio Grande do Norte (1.4%), Rio de Janeiro (1.4%), Pernambuco (1.4%), Alagoas (1.,4%), Sergipe (1.4%)) and from other countries (1.4%)). More than half of the students were teachers (50.7%), 39.3% were K-12 teachers, 9.8% teach at universities and 1.6% teach in NGOs and Corporations.

Instruments to collect data

We collected the data by means of a questionnaire sent to the students at the end of the course. The questionnaire had 21 closed-ended questions and 7 open-ended questions.

The close-ended questions had the goal of collecting demographic information (sex, age, academic background, profession) and measuring the students’ participation, effort, and learning. We defined eight variables: four to quantify the students’ effort and participation (table 1, Appendix 1), four to quantify the students’ learning (table 2, Appendix 1).

The open-ended questions had the objective of gathering information about the students' feelings and perceptions about the course.

Data Analysis Procedures

The quantitative data was analyzed using descriptive statistics, while the qualitative data was analyzed using the language processing method (Graham et al., 1993). We connected the findings of the quantitative data and qualitative data by means of a systemic analysis (Arantes do Amaral, 2019). In descriptive statistics, we used measures of central tendency (mean, median) and measures of dispersion (interquartile range and standard deviation) to analyze the data. We used stacked bar charts in order to visualize data related to students' answers that made use of 3-point Likert scale questions. We used R software to do all calculations.

As previously stated, we used the language processing method to analyze the qualitative data. By using this method, we were able to break the long paragraphs (the answers to the open-ended questions) into small sentences, group similar sentences, and discover the recurrent themes.

RESULTS

Results from quantitative data

In this section we present the stacked bar chart that summarizes the answers of the students to the quantitative Likert scale questions. Figure 1 presents the levels of the five variables (Appendix 1, table 1) used to quantify the students' effort and participation. The levels low, medium, and high are represented by the colors brown, grey and green, respectively. The explanation of the meaning of each variable is described in APPENDIX 1.

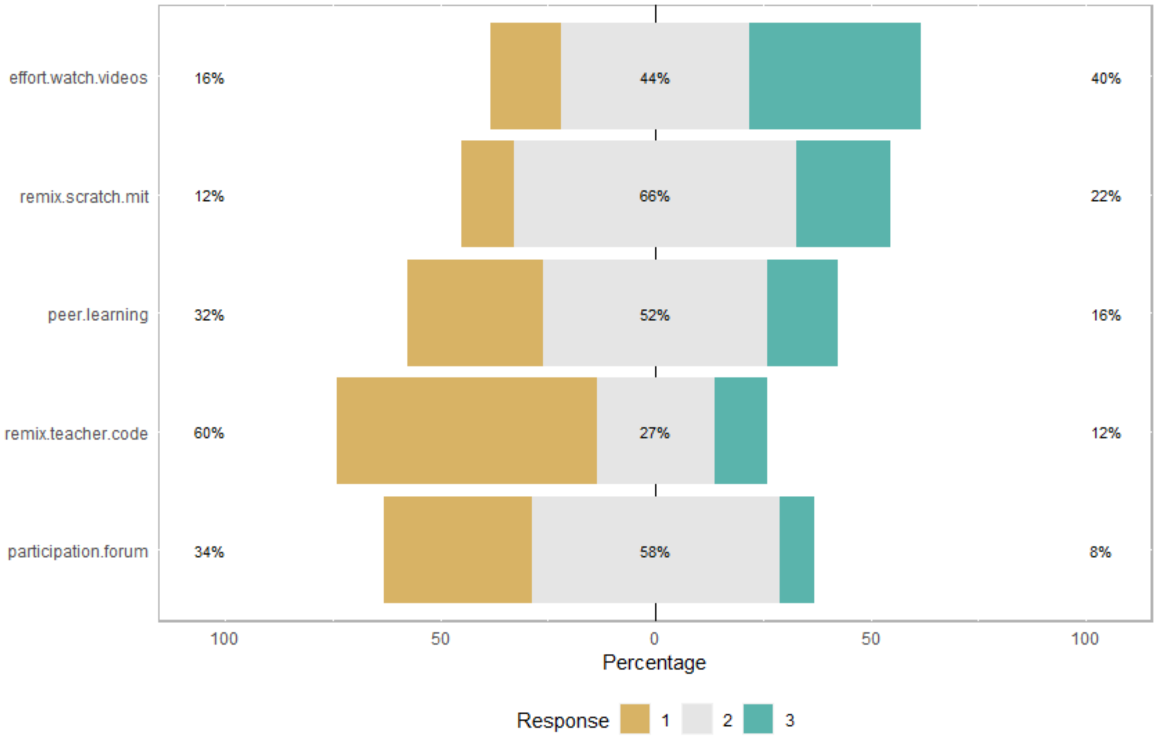

Figure 1. Effort and participation.

Figure 2 presents the levels of the three variables (Appendix 1, table 2) used to quantify the students' learning and satisfaction.

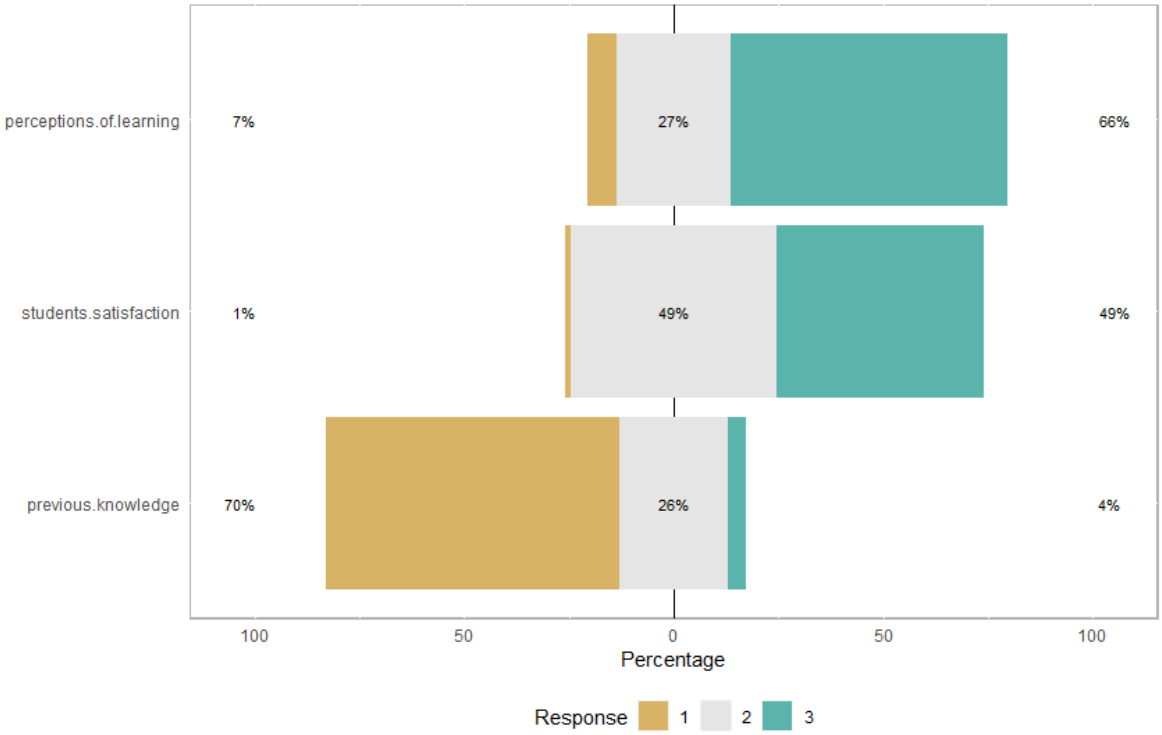

Figure 2. Students' learning.

In the discussion section, we will discuss the impacts of each variable described here, connecting the quantitative data to qualitative data.

Results from qualitative data

Analyzing the qualitative data, we were able to identify five recurrent themes (thereafter, RT).

The recurrent themes below summarize the answers of the students to the open-ended questions.

RT1 – The students learned a significant amount while working playfully on projects that interested them.

RT2 – The professor´s feedback motivated the students to learn, improving their coding skills.

RT3 – The course was well structured, the video-lectures were easy to follow, and the professor’s feedback facilitated the learning.

RT4-Remixing code was fun, helping to develop the students’ coding skills and motivating them to learn.

RT5-The project created opportunities for students to learn how to program, collaborate with other students, and share knowledge.

The recurrent themes show that the course was a joyful experience for the students: they learned by doing, playing, and remixing code. Moreover, the data revealed that the course was well-structured and delivered efficiently. In the discussion section, we will connect each recurrent theme (qualitative results) with each identified variable (quantitative result), performing a systemic analysis.

DISCUSSION

Returning to our research question (What are the dynamics of learning present in MOOCs that teach students to program using Scratch?) we were able to identify five intertwined feedback loops that drove the learning dynamics (Figure 3).

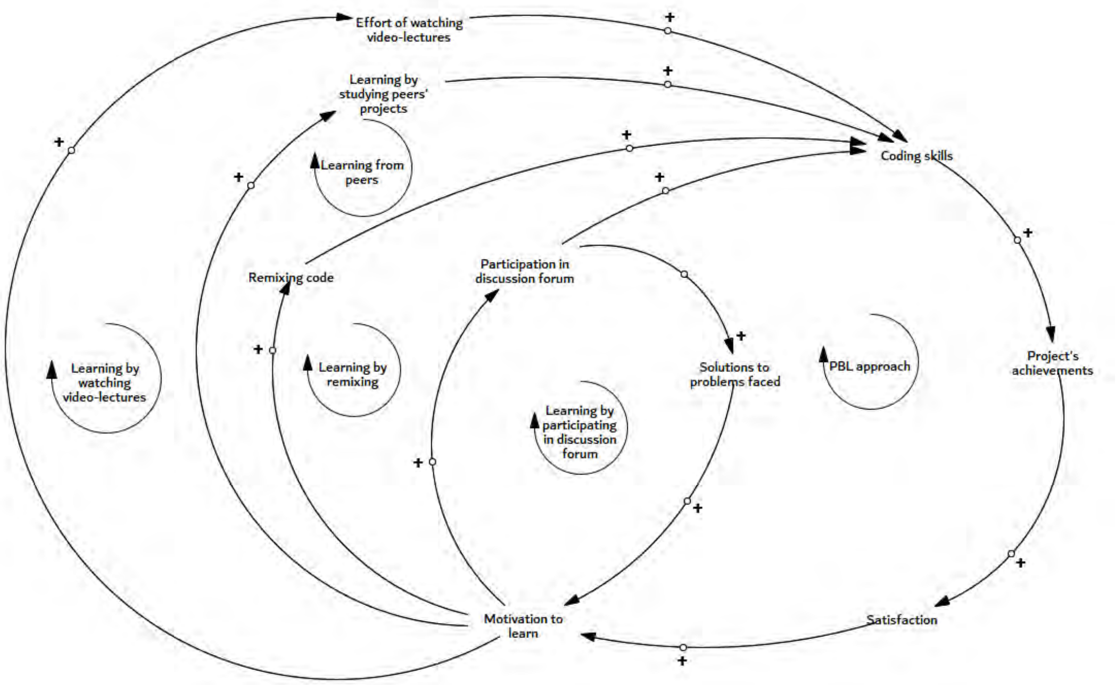

Figure 3. Feedback loops driving the course dynamics.

Feedback loop 1: Learning by participating in discussion forum

More than half of the students (Figure 1, variable "participation.forum") reported that they participated at a medium (58%) to high(8%) level in the discussion forum. In addition, RT2 let us understand that the professor's feedback helped the students learn. Therefore, the more the students participated in the forum, the more they encountered solutions to the problems they were facing, thus increasing their motivation to learn (Figure 3, feedback loop "Learning by participating in discussion forum"). This finding is aligned with the findings of other researchers, who have studied the impacts of the use of discussion forums in learning (Almatrafi & Johri, 2018).

Feedback loop 2: Learning by remixing

The majority of the students (Figure 1, variable "remix.scratch.mit") remixed projects from the MIT Scratch community. For 22% of the students, the remix was high; for 66% it was medium. The students did not have the same interest in remixing the teacher's code (Figure 1, variable "remix.teacher.code"): only 39% of them remixed the code made available. We may speculate that the students prefer to remix projects from the MIT Scratch community because the number of projects was higher there and the themes of the projects were much more diverse. RT4 helped us to understand that the remixing process not only helped the students to learn how to code, but also a fun activity that motivates them to learn more.

Therefore, we may infer that the more the students remixed code, the more they developed their coding skills, increasing the project's achievements and motivation to learn (Figure 3, feedback loop "Learning by remixing"). This finding is aligned with the findings of other scholars (Kang, 2019; Robinson & Resnick, 2017) who point out that remixing code improves learning.

Feedback loop 3: Learning from peers

In relation to the sharing of project ideas, 16% of the students (Figure 1, variable "peer.learning") affirmed that the amount they learned from peers was high, while and 52% stated that the amount was medium. RT5 allowed us to understand that knowledge sharing was present in this course. Therefore, we may infer that the more the students learned by studying projects of their peers, the more they developed their coding skills (Figure 3, feedback loop "Learning from peers"). This finding is in accordance with the findings of other researchers (Pinto & Escudeiro, 2014).

Feedback loop 4: Learning by watching video-lectures

The students let us know that more than 84% of them had watched the video-lectures (Figure 1, variable "effort.watch.videos"). In addition, RT3 let us understand that the video-lectures were easy to understand. Therefore we may consider that watching the video-lectures facilitated the development of coding skills (Figure 3, feedback loop "Learning by watching video-lectures"). Other researchers have pointed out the importance of video-lectures in MOOCs (Diver & Martinez, 2015; Johnston, 2015).

Feedback loop 5: PBL approach

RT1 let us know that the students enjoyed the project-based learning approach. This approach allowed them to learn by doing, by interacting with their peers (Figure 1, variable "peer.learning"), by remixing code (Figure 1, variables "remix.scratch.mit" and "remix.teacher.code"),by participating in the discussion forum (Figure 1, variable "participation.forum") and by watching the video-lectures (Figure 1, variable "effort.watch.videos"). It is interesting to note the students' development of skills in coding: at the beginning of the course, 70% of them had little knowledge of coding (Figure 2, variable "previous.knowledge"), while at the end of the course 66% of the students acknowledged their knowledge level was high (Figure 2, variable "perceptions.of.learning"). More than that, for 49% of the students, the course delivered beyond their expectations (Figure 2. variable "students.satisfaction"). Therefore we may affirm that the PBL approach led to the increase in the number of projects accomplished, increasing the students' satisfaction and motivation to learn (Fig 3, feedback loop, "PBL approach"). This finding is in agreement with the findings of other scholars, who have used PBL to teach how to program with Scratch (Hadi & Atiqoh, 2021; Voinohovska et al., 2019).

Based on the five feedback loops described previously, we came to the following conclusions:

1) The feedback loops 1,2,3 and 4 let us understand that the development of coding skills was driven by their efforts in watching video-lectures, remixing their peers' codes, and by knowledge sharing among the students.
2) The feedback loop 5 lets us understand the importance of the PBL approach, since it creates the opportunities for the students to collaborate and exchange ideas.

FINAL REMARKS

Using MOOC and PBL to teach programming with Scratch was very effective. The MIT Scratch learning environment allowed the students to code online, to easily share their code and to learn with each other students and members of MIT Scratch community. The project-based learning approach encouraged the students to follow their interests, creating projects that they were passionate about.

References

Almatrafi, O., & Johri, A. (2018). Systematic Review of Discussion Forums in Massive Open Online Courses (MOOCs). *IEEE Transactions on Learning Technologies, 12*(3), 413-428. https://doi.org/10.1109/TLT.2018.2859304

Amoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From Scratch to "Real" Programming. *ACM Transactions on Computing Education (TOCE), 14*(4), 1-15. https://doi.org/10.1145/2677087

Arantes do Amaral, J. A. (2019). Combining community-based learning and project-based learning: A qualitative systemic analysis of the experiences and perceptions of students and community partners. *Partnerships: a journal of service-learning and civic engagement, 10*(1), 129-145.

Arantes do Amaral, J. A., Gonçalves, P., & Hess, A. (2015). Creating a Project-based Learning Environment to Improve Project Management Skills of Graduate Students. *Journal of Problem Based Learning in Higher Education, 3*(2), 120-130. https://doi.org/10.5278/ojs.jpblhe.v0i0.1178

Beghetto, R. A. (2021). Creative Learning in Education. In M. L. Kern (Ed.), *The Palgrave Handbook of Positive Education* (pp. 473-491). Springer International Publishing AG. https://doi.org/10.1007/978-3-030-64537-3_19

Bezerra, L. M., & Silva, M. T. (2017). A review of literature on the reasons that cause the high dropout rates in the MOOCS. *Revista Espacios*, *38*(5), 11.

Brennan, K. (2011). *ED-MEDIA 2011--World Conference on Educational Multimedia, Hypermedia & Telecommunications*. AACE.

Bustillo, J., & Garaizar, P. (2016). Using Scratch to foster creativity behind bars: Two positive experiences in jail. *Thinking Skills and Creativity*, *19*, 60-72. https://doi.org/10.1016/j.tsc.2015.08.003

Creswell, J. W. (2013). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, Incorporated.

Diver, P., & Martinez, I. (2015). MOOCs as a massive research laboratory: Opportunities and challenges. *Distance Education*, *36*(1), 5-25. https://doi.org/10.1080/01587919.2015.1019968

Dohn, N. B. (2020). Students' interest in Scratch coding in lower secondary mathematics. *British Journal of Educational Technology*, *51*(1), 71-83. https://doi.org/10.1111/bjet.12759

Estevez, J., Garate, G., & Graña, M. (2019). Gentle Introduction to Artificial Intelligence for High-School Students Using Scratch. *IEEE access*, 179027-179036. https://doi.org/10.1109/ACCESS.2019.2956136

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, *29*(1), 12-28. https://doi.org/10.1002/cae.22255

Fields, D., Vasudevan, V., & Kafai, Y. B. (2015). The programmers' collective: fostering participatory culture by making music videos in a high school Scratch coding workshop. *Interactive Learning Environments*, *23*(5), 613-633. https://doi.org/10.1080/10494820.2015.1065892

Graham, A., Shiba, S., & Walden, D. (1993). *New American TQM*. Taylor & Francis.

Hadi, M. E., & Atiqoh, K. N. (2021). Improving Students' Mathematical Computational Thinking Using Scratch Program through Project Based Learning: A Development Research during Pandemic Covid-19. In *2021 9th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-5). IEEE. https://doi.org/10.1109/CITSM52892.2021.9588856

Husna, A., Cahyono, E., & Fianti, F. (2019). The effect of project based learning model aided scratch media toward learning outcomes and creativity. *Journal of Innovative Science Education*, *8*(1), 1-7.

Johnston, T. C. (2015). Lessons from MOOCS: Video lectures and peer assessment. *Academy of Educational Leadership Journal*, *19*(2), 91-98.

Kang, O. H. (2019). Analysis of the sociality and democratic-citizenship changes from the application of the Scratch remix function in cooperative learning. *Journal of Information Processing Systems*, *15*(2), 320-330.

Kennedy, T. D., & Edmonds, W. A. (2016). *An Applied Guide to Research Designs: Quantitative, Qualitative, and Mixed Methods*. SAGE Publications.

Kordaki, M. (2012). Diverse Categories of Programming Learning Activities could be Performed within Scratch. *Procedia-Social and Behavioral Sciences*, *46*, 1162-1166. https://doi.org/10.1016/j.sbspro.2012.05.267

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), 1-15. https://doi.org/10.1145/1868358.1868363

Pinto, A., & Escudeiro, P. (2014). The use of Scratch for the development of 21st century learning skills in ICT. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-4). IEEE. https://doi.org/10.1109/CISTI.2014.6877061

Pirker, J., Economou, D., & Gütl, C. (2016). *Interdisciplinary and International Game Projects for Creative Learning*. Proceedings of the 2016 ACM conference on innovation and technology in computer science education. https://doi.org/10.1145/2899415.2899448

Plaza, P., Sancristobal, E., Carro, G., Blazquez, G., Garcia-Loro, F., Muñoz, M., & Castro, M. (2019). STEM and Educational Robotics Using Scratch. *In 2019 IEEE Global Engineering Education Conference (EDUCON)*, 330-336. https://doi.org/10.1109/EDUCON.2019.8725028

Resnick, M. (2014). Give P'sa chance: Projects, peers, passion, play. In *Constructionism and creativity: Proceedings of the third international constructionism conference* (pp. 13-20). Austrian computer society.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Brennan, K., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, *52*(11), 60-67. https://doi.org/10.1145/1592761.1592779

Resnick, M., & Rusk, N. (2020). Coding at a crossroads. *Communications of the ACM*, *63*(11), 120-127. https://doi.org/10.1145/3375546

Robinson, K., & Resnick, M. (2017). *Lifelong Kindergarten: Cultivating Creativity Through Projects, Passion, Peers, and Play*. MIT Press. https://doi.org/10.7551/mitpress/11017.001.0001

Sefton-Green, J. (Ed.). (2011). *The Routledge International Handbook of Creative Learning*. Routledge. https://doi.org/10.4324/9780203817568

Su, Y. S., Shao, M., & Zhao, L. (2022). Effect of Mind Mapping on Creative Thinking of Children in Scratch Visual Programming Education. *Journal of Educational*

Computing Research, *60*(4), 906-929. https://doi.org/10.1177/07356331211053383

Voinohovska, V., Tsankov, S., & Goranova, E. (2019). Development of the students' computational thinking skills with project-based learning in scratch programming environment. In *13th International Technology, Education and Development Conference* (pp. 5254-5261). INTED2019. https://doi.org/10.21125/inted.2019.1309

Wang, H. Y., Huang, I., & Hwang, G. J. (2014). Effects of an integrated Scratch and project-based learning approach on the learning achievements of gifted students in computer courses. In *2014 IIAI 3rd International Conference on Advanced Applied Informatics 2014*. IEEE Computer Society. https://doi.org/10.1109/IIAI-AAI.2014.85

Yamamori, K. (2019). Classroom practices of low-cost STEM education using scratch. *Journal of Advanced Research in Social Sciences and Humanities*, *4*(6), 192-198. https://doi.org/10.26500/JARSSH-04-2019-0601

APPENDIX -1

Variables that quantify students' effort and participation		
Variable	**Meaning**	**Levels** **We considered a "reasonable number", a number between five and 10 a "large number" a number higher than 10.*
remix.scratch.mit	it quantifies the effort of remixing code of from people of the MIT Scratch community	low (the students remixed few projects) medium(the students remixed a reasonable number of projects) high (the students remixed a large number of projects)
remix.teacher.code	it quantifies the effort of remixing the professor's code	low (the students remixed few projects) medium(the students remixed a reasonable nuumber of projects) high (the students remixed a large number of projects)
peer.learning	it quantifies the students' effort of learning with the projects of other students	low (the students learned nothing or little with his/her peers) medium (the students have learned in a reasonable way with his/her peers) high (the students learned a significant amount with his/her peers)
participation.forum	it quantifies the students' effort of participating in the discussion forum	low (the students had none or almost no participation) medium(the students participated in a moderate way) high (the students' participation was high)
effort.watch.videos	it quantifies the students' effort of watching video-lectures	low (the students watched few video-lectures) medium (the students watched a reasonable number of video-lectures) high (the students watched all or almost all video-lectures)

Table 1. Variables created to quantify students' effort and participation.

Variables that quantify students' learning		
Variable	**Meaning**	**Levels**
perceptions.of.learning	it quantifies the students' perception of learning at the end of the course	low (the students learned less than they expected to learn) medium (the students learned what they expected to learn) high (the students learned more than they expected to learn)

students.satisfaction	it quantifies the students' satisfaction with the course	low (the course did not meet the students' expectations) medium (the course met the students' expectations) high (the course delivered beyond the students' expectations)
previous.knowledge	it quantifies the students' previous knowledge in programming with Scratch	low(the students had no knowledge or little knowledge of programming with Scratch) medium (the students had some knowledge of programming with Scratch) high (the students had some advanced knowledge of programming with Scratch)

Table 2. Variables created to quantify students' learning.