ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/325485002

Teaching and learning mathematics in primary school through Scratch

Conference Paper · June 2018

DOI: 10.21125/edulearn.2018.1359

CITATIONS
3

READS
2,190

5 authors, including:

Jalal Nouri
Stockholm University
88 PUBLICATIONS **366** CITATIONS
SEE PROFILE

Eva Norén
Stockholm University
42 PUBLICATIONS **124** CITATIONS
SEE PROFILE

Lechen Zhang
Stockholm University
13 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project International Journal of Learning Analytics and Artificial Intelligence for Education (www.i-jai.org) View project

Project Current projects in Mobile learning View project

All content following this page was uploaded by Jalal Nouri on 13 April 2019.

The user has requested enhancement of the downloaded file.

Teaching and learning mathematics in primary school through Scratch

Christer Sjöberg[1], Jalal Nouri[2], Rosmarie Sjöberg[3], Eva Norén[4] & Lechen Zhang[5]

[1,3] *Strandskolan, Tyresö (SWEDEN)*
[2,4,5] *Stockholm University (SWEDEN)*

Abstract

Programming education is a hot topic in many countries around the world. Also in Sweden this topic has received a lot of attention lately due to formal introduction of programming curriculum as of 2018. Mathematics is one of the subjects that is most affected by the curriculum changes as the government in Sweden has decided that teachers of mathematics are to teach programming in compulsory school in order to support problem-solving in mathematics. Albeit there are some previous research investigating questions related to how programming can enhance mathematics education, for instance in form of the seminal work of Papert through the LOGO language, more research is required that scrutinize how new visual block programming language such as Scratch can be used for mathematical learning. It is against such a background we in this paper report on how Scratch has been used in primary school for two years to teach mathematics. We will present four different projects that a teacher has planned and conducted with 68 students that target four different areas of mathematics. As such we describe the didactical strategies that was employed to help students achieve the learning goals, and the associated challenges. We hope that the presentation can be helpful for other teachers and researchers interested in using visual block programming languages for teaching and learning mathematics.

Keywords: programming, mathematics, scratch, coding, computational thinking

1 INTRODUCTION

It is widely acknowledged that computer programming needs to be introduced to young learners [1-2] as a means to develop their computational thinking skills [3-4]. Therefore, many countries have updated their K-12 curriculum to embrace programming. However, besides the increasing interest of studying programming for its own sake, it is might also be worth to investigate programming’s effect on learning other subjects [5]. For example, the new Swedish curriculum for primary school explicitly mentions programming within the frame of the subject of mathematics. As a matter of fact, educational researchers started investigating how computer programming can be used to foster mathematics learning already in the 1960s [6-7]. In fact, mathematical thinking is closely related to computational thinking because “solving a mathematical problem is a process of construction that requires an analytic problem-solving perspective, which is unique and fundamental to computer programming” [11]. Since Seymour Papert created the LOGO programming language in the 1960s, studies were conducted using LOGO to facilitate the learning of, for instance, numerical magnitude and length estimation [11-13].

As the programming tools evolves through time, more programming languages like LOGO are made available for young learners, such as Scratch, Blockly, AppInventor, etc. The design of Scratch, by MIT Media Lab, is especially intended for programmers younger than 16. The “low floor, high ceilings and wide walls” makes

Scratch a popular programming environment [14]. Some previous studies have indicated the potential of Scratch in facilitating mathematics education. The review of [7] showed that during the process of developing CT skills through programming, pupils incidentally or intentionally learned mathematics such as numbers, operation, algebra, functions etc. Another aspect of choosing Scratch is that:

> “The focus of Scratch is on making multimedia products, and sharing them in the large and active online community hosted by the project website. This is intended to enable and develop children’s creativity, but also to introduce them to programming, in a fun way” [8, p. 2]

In Scratch, the program is written by fitting “blocks” together. Thus, the programming language is a visual language [8]. Scratch can also be used to design games, games that can develop young students’ mathematical skills and concepts [9].

It is against such a background we in this paper report on how Scratch has been used in primary school for two years to teach mathematics. We will present four different projects that a teacher has planned and conducted with 68 students that target four different areas of mathematics. The guiding research question has been: What didactical strategies are teachers using to integrate programming into mathematics education?

2 METHODOLOGY

Two of the authors, teachers in primary school, started two years ago to work with programming in several classes at their school. They had very limited personal experience of programming but decided that they would try to learn with their students by choosing Scratch as programming tool.

In this article, we describe how the work has progressed to today. We describe the didactic choices they made, how they gradually discovered new features in Scratch that connect to the existing teaching of mathematics, and the results achieved from the two years of work. Four areas of mathematics have been worked with, namely:

- Multiplication
- The Clock (expressing time)
- Smart mental calculation
- Conversion between temperature scales

2.1 Participants: students and teachers

The teacher has worked with three classes with about 25 students in each. Because some students have left the school, and some have been added during the period, we have chosen to look at the results of the students that have been working with all four projects, which are 68 students.

The participating teachers are Christer Sjöberg, teacher of mathematics, and Rosmarie Sjöberg, ICT manager at Strandskolan. Christer Sjöberg has performed the concrete work with students while Rosmarie Sjöberg has attended parts of the planning, monitoring and analysis.

The students had some minor experiences with Scratch through a former teacher at the end of year 3 and were familiar with the program when they started in year 4 in August 2016. The project described is carried out from September 2016 to April 2018.

3 RESULTS

In the following, four projects will be presented. Each project has started with that students have been given an example of a simple code to start from which have been first explored and discussed collectively with the support of the teacher and through the use of a projector on a white board [8]. After that, students have been given the instruction to reproduce the code with personal additions. Complexity increased with time. In the beginning the students received instructions to create and change graphics, such as switching to an own background. Gradually, whole programs were changed and improved.

3.1 Project 1: Multiplication

Regarding number, a goal for school year 4-6 is to be able to use "*central methods for calculations with natural numbers /…/, main statement and calculations using written methods and calculator. And how methods can be used in different situations*" [10, p. 57]. Learning the multiplication tables in various ways is part of this. The task given to the students early in year 4, was to reproduce code and make additional tests of other numbers in the multiplication tables. As both the teachers and the students were quite unfamiliar with programming, this first project was to make a fairly simple program with a conditional statement. The code that the students could work with looked like this:

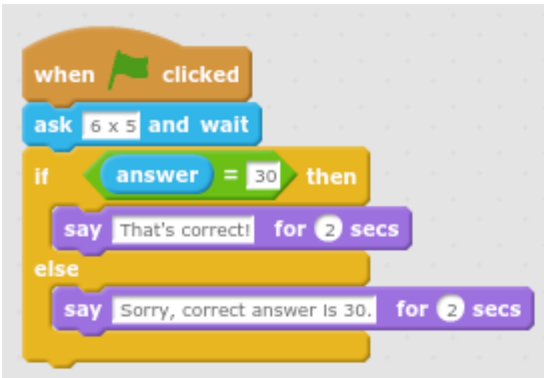

Figure 1. Scratch code for multiplication

Pretty soon students asked for more features because they thought that the program did not meet the requirements for a good program to train multiplication. They suggested for example that the program should specify the number of correct answers and that a timer was added in order to make the game more fun and challenging. In order to address that wish, we were forced to introduce, explore and learn the variable concept together. As a result, a new template was constructed that integrated the variable concept.

In the template, we have also added the feature "costumes" that allows the sprite to change the look using the previously set conditions, i.e. look happy when a right answer is provided (see code and outcome in figure 2).

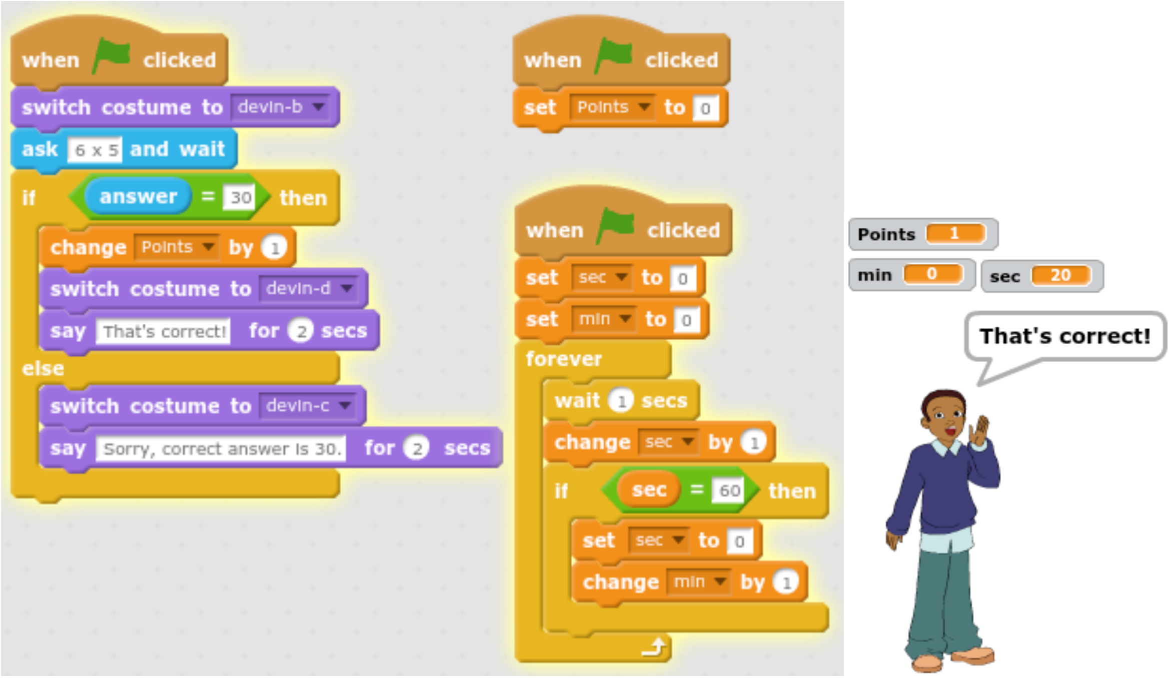

Figure 2. An advanced version of the game for multiplication

A reflection after the first project was that students learned to program incredibly quickly and helped each other when they encountered problems and challenges. Working with multiplication tables was something the teachers would have done with or without programming, but this this was really shown to be just another engaging way to do it.

Another reflection was that many students identified the problems that needed to be solved (for instance the above example of scoring) and the programming activities created the opportunities to practically discover how programming can be used as a tool to solve problems and improve students' digital literacy.

3.2 Project 2: The clock

Being able to express and measure time is a central content of the curriculum in mathematics for grades 4-6. In year 4 many students are still unsure of how to express time, especially in analog ways. Therefore, in this project, students explored digital and analogue clocks through programming.

The activities started with that students received a template as in the previous project. However, in this project students were given different templates based on their prior knowledge of the clock. Those that were confident in reading the clock received more the more challenging task to make applications where time was specified using Roman numerals or with symbols of a 12-hour clock (i.e. am and pm). In the programmed applications the students had to work with the following:

- Figure out the time difference between two analogue time intervals
- Figure out the time difference between two digital time intervals
- Switch between analogue and digital time formats
- Figure out the time difference between two time intervals with Roman numerals
- Enter the time with 12-hour time format

In this project, we tested the Scratch concept messaging. We used the concept to replace clock-faces and to build applications that used multiple clock faces after each other. The template included two clock faces, but the students were encouraged to implement more.

As each clock-face corresponds to a sprite the students needed to program the code of each of these. Below is the code for the sprites and an example of how it looks for the user when the program starts.

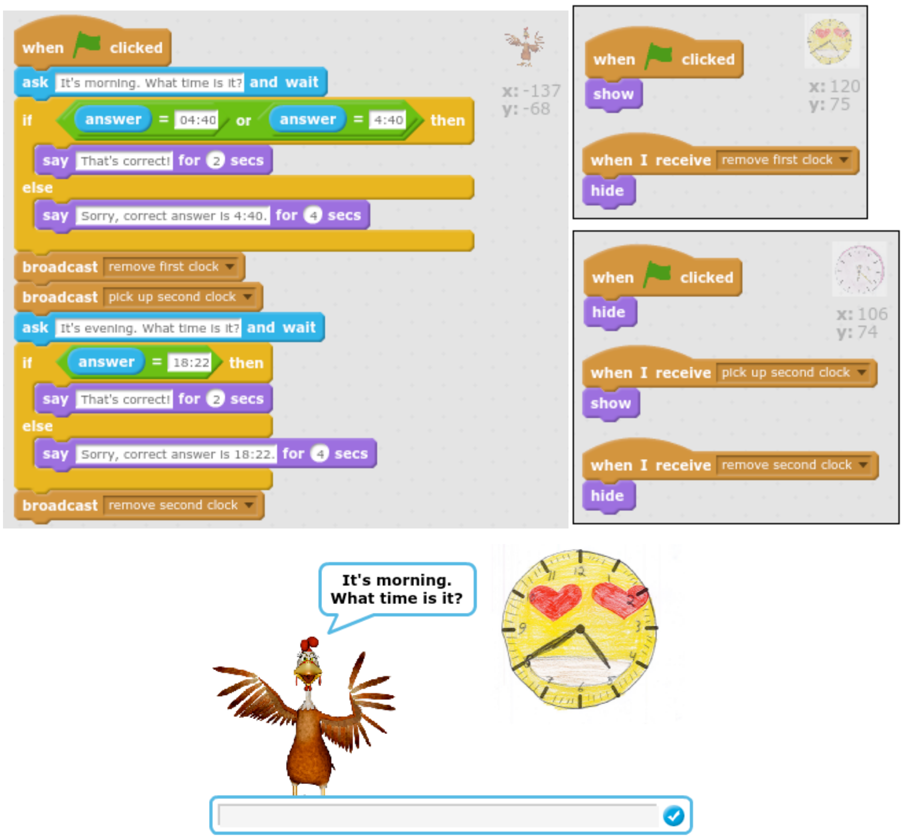

Figure 3. A student application to express time

Although programming and training of the clock were in the focus, it turned out that something else perhaps was the biggest win of this project, namely the concrete work with copyright.

As these particular students have the habit to publish much of their material on a class blog, they have learned to be careful what pictures they use and it proved very difficult to find clock-faces that they were allowed to download and add to their applications. The students solved this by drawing own clock-faces (see figure 3) on paper that were later scanned and digitalized for all to access and use. By doing this, the students were able to put out programs on the blog without violating copyright. A collective discussion was also done about if the students wanted others to be able to use the pictures. In the end, all material on the blog was labelled with a Creative Commons license. This is an example of how a programming activity within the subject of mathematics can entail enhancement of general digital competences.

3.3 Project 3: Mental arithmetic

A curricular goal already in school year 3 was to be able to do mental calculations. This ability is supposed to be developed through the school years. When the students have gained knowledge about multiplication and division tables, it is important to make use of the knowledge in other contexts [10, p. 57]. The following project was called "mental arithmetic," and it can entail being able to take the step from figuring out 3 · 8 = 24 to understand the reverse 24/8 = 3 followed by the challenge to figure out 24 000/8 = 3 000.

Templates were created for a number of mental arithmetic methods and the students were divided into groups (4-5 students in each group). Each group was responsible for a mental calculation method and the task was to make a program in Scratch. Everything was posted on the class blog so students could learn and practice using the other groups programs.

In the following we present an example of the method to calculate the division of large numbers. The following code was provided to the students as a template:

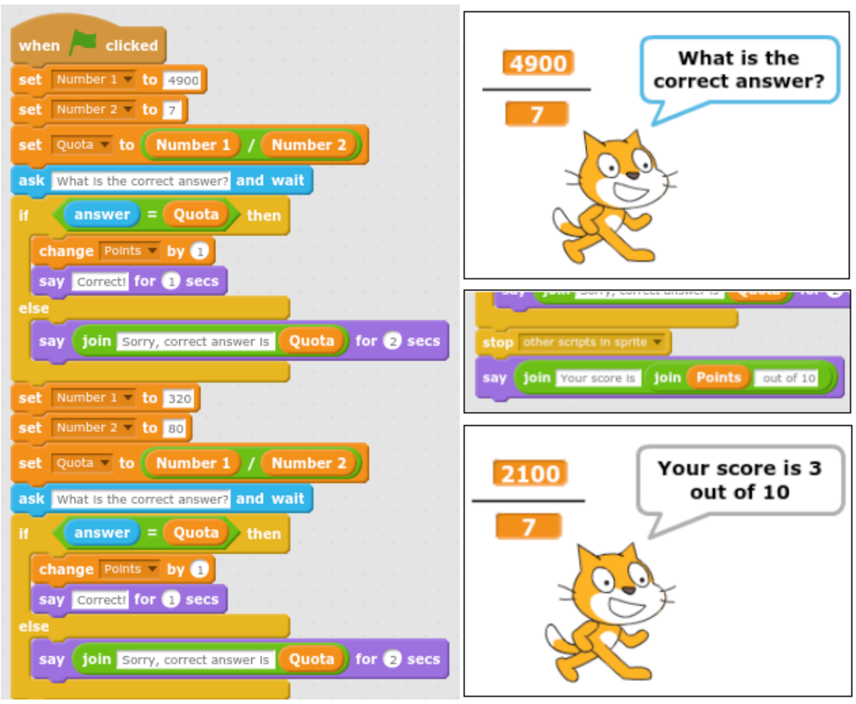

Figure 4. A template for mental calculation of large number division

Although this type of activity helps the students to understand numbers that can be divided easily, a disadvantage of the program as it appears in the template may be that it is always the same divisions and the same order is that the students quickly learn the answers by heart.

In order to solve this challenge, some students started to reflect upon and talk about how to construct a more open program that randomize numbers that can be evenly divided, which from our perspective is a productive mathematical reflection (problem-solving activity). With the support of the teacher and talented parents, a new template was produced (see figure 5).

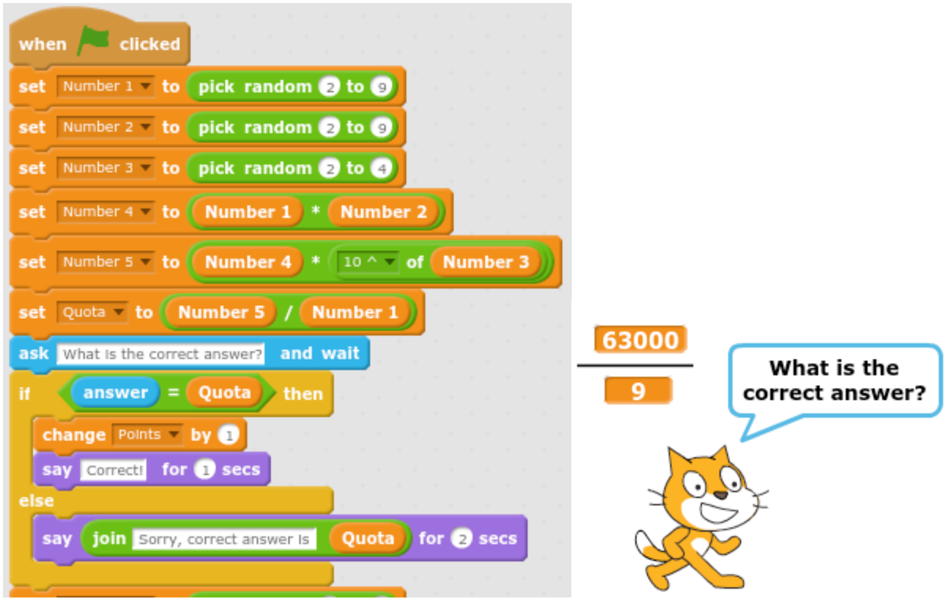

Figure 5. A program for mental calculation that randomize numbers for the division

Most students worked with the first template, but those that worked with the later got an opportunity to work with and create an understanding of random numbers and a more advanced problem-solving approach. This is a good example of how Scratch support students to work individually on different levels.

3.4 Project 4: Temperature conversion

By the end of school year 6 students should be able to use scale in every-day situations [9. p. 58]. The theme of this fourth project was then decided to be on conversion between different temperature scales. As before, students were given different templates in Scratch and they were free to choose any of the following conversions (or combinations of):

- Celsius to Fahrenheit
- Fahrenheit to Celsius
- Celsius to Kelvin
- Kelvin to Celsius

The teacher chose not to have students convert between Fahrenheit and Celsius, because it is rarely useful in Sweden. The different calculation methods were collectively discussed before the programming activity. The different templates looked pretty much the same for the four different programs. In the following we present the program which convert Celsius to Fahrenheit.

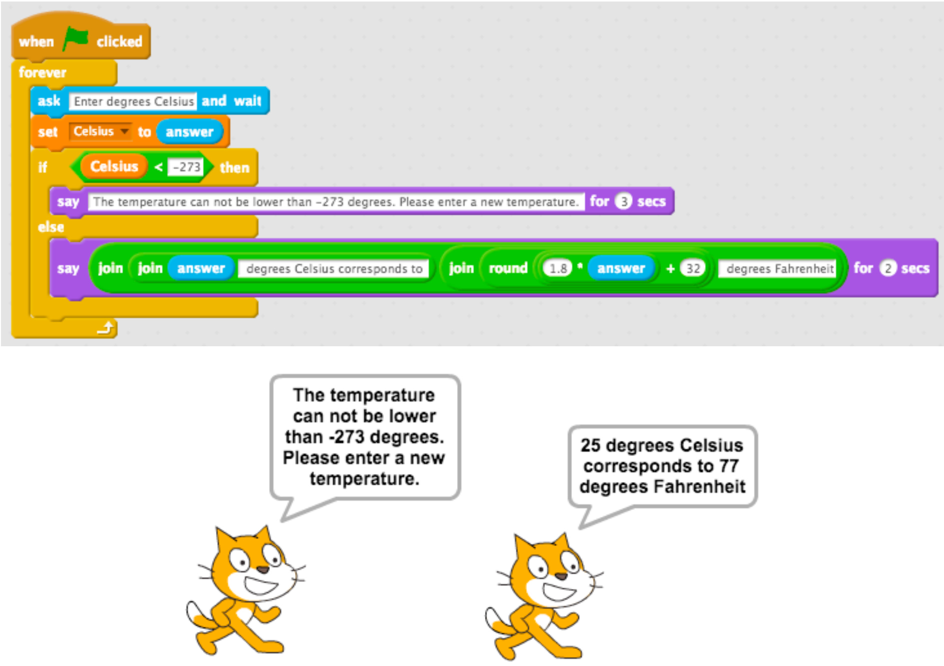

Figure 6. A scratch program for converting Celsius to Fahrenheit

Something that very early caught many students' interest was that it can not actually be colder than -237 ℃. There we took advantage of and created a conditional statement that took into account if you accidentally enter a temperature lower than -237 ℃. For those students who were advanced in their programming skills this program was not challenging enough. Therefore, those students were given the instruction to change the background depending on the chosen temperature (for example a high temperature generates a desert background and a low temperature a winter landscape).

4 DISCUSSION

In the four programming projects described in this paper, the mathematical content could have been elaborated on in other ways. The content, multiplication, expressing time, scale and arithmetic computations are typical curricular objectives, with goals for students this age, described in the national curriculum. We have described the didactical strategies that the teachers employed in the four projects to help students achieve the learning goals, and the associated challenges. The didactical strategies are closely linked to the possibilities of Scratch, like the possibility to *reproduce code*. According to the teachers involved in the projects, this strategy makes different students work on different levels. Some students copy code; others develop their own. This can be seen as a challenge for the teachers, to choose activities on different levels. Up until today the starting points for the students have mostly been the same in, at least these four the projects, but in the second project, on expressing time, the students were given different templates. A future challenge is also to be able to assess the programming skills and digital literacy as well as the mathematical skills the students are developing.

Scratch’s visual language for programming is in earlier studies said to support young students to develop programming skills. One of the main advantages is the enjoyability, in contrast to frustration and anxiety [8]. The four projects presented in this paper have shown the same. Not just to programme, but also an enjoyable and engaging way to exercise multiplication, for example. The strategy here was to design a game. This is common when using Scratch, in this paper the game were constructed to develop young students' multiplication skills and concepts [9]. Another gain with the programming activities was that students beside learning more about how to express time, also learned to be careful with how to handle copy right.

Another didactical strategy used, consciously or unconsciously, is that students can help each other when they encounter problems and challenges. In the third project, there was also a "talented parent" involved. It seems like collaborative work of various kinds occur in the programming situations themselves. The Scratch blog appears to encourage students to do their own findings. To use the blog can be seen as one more didactical strategy.

We believe that the presentation of the four projects and the didactical strategies chosen by the teachers can be helpful for other teachers and researchers interested in using visual block programming languages for teaching and learning mathematics.

REFERENCES

[1] Duncan, C., Bell, T., & Tanimoto, S. (2014, November). Should your 8-year-old learn coding?. In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (pp. 60-69). ACM.

[2] Williams, C., Alafghani, E., Daley, A., Gregory, K., & Rydzewski, M. (2015, October). Teaching programming concepts to elementary students. In Frontiers in Education Conference (FIE), 2015 IEEE (pp. 1-9). IEEE.

[3] Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?. Acm Inroads, 2(1), 48-54.

[4] Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. European Commission, JRC Science for Policy Report.

[5] Benton, L., Hoyles, C., Kalas, I., & Noss, R. (2017). Bridging primary programming and mathematics: Some findings of design research in England. Digital Experiences in Mathematics Education, 3(2), 115-138.

[6] Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with scratch. In Design for teaching and learning in a networked world (pp. 17-27). Springer, Cham.Department for Education, "National curriculum in England: Computing programmes of study," 2013.

[7] Hickmott, D., Prieto-Rodriguez, E., & Holmes, K. (2017). A Scoping Review of Studies on Computational Thinking in K–12 Mathematics Classrooms. Digital Experiences in Mathematics Education, 1-22.

[8] Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. Proceedings of the 22nd Annual Psychology of Programming Interest Group (Universidad Carlos III de Madrid, Leganés, Spain.

[9] Calder, N. (2010). Using Scratch: An Integrated Problem-Solving Approach to Mathematical Thinking. *Australian Primary Mathematics Classroom*, *15*(4), 9-14.

[10] Skolverket [National Agency for Education]. (2011). Läroplan för grundskolan Läroplan för grundskolan, förskoleklassen och fritidshemmet 2011. Revised 2016. Stockholm: Skolverket.

[11] Sung, W., Ahn, J., & Black, J. B. (2017). Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education. Technology, Knowledge and Learning, 22(3), 443-463.

[12] Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 450-467.

[13] Robinson, M. A., Gilley, W. F., & Uhlig, G. E. (1988). The effects of guided discovery LOGO on SAT performance of first grade students. *Education,* 109(2).

[14] Portelance, D. J., Strawhacker, A. L., & Bers, M. U. (2016). Constructing the ScratchJr programming language in the early childhood classroom. International Journal of Technology and Design Education, 26(4), 489-504.

View publication stats