Early Programming Education and Career Orientation: the Effects of Gender, Self-Efficacy, Motivation and Stereotypes

Efthimia Aivaloglou
Open University of the Netherlands
fenia.aivaloglou@ou.nl

Felienne Hermans
Leiden Institute of Advanced Computer Science
f.f.j.hermans@liacs.leidenuniv.nl

ABSTRACT

Programming education currently begins at the elementary school age. In this paper we are exploring what affects the learning performance of young students in programming classes. We present the results collected during an eight-week experimental Scratch programming course run in elementary schools. We emphasize factors that have been found to affect learning performance in adult students, including self-efficacy and motivation, and measure how they affect students of this age group. We further explore the students' view of programming as a career path, and measure the effects of the course, their performance, and the stereotypes that they assume for computer scientists. We find that students' intrinsic and extrinsic motivation and previous programming experience are important factors, being strongly correlated with their self-efficacy and their inclination towards a CS career. For female students only, we also find CS career orientation to be strongly correlated with their self-efficacy.

CCS CONCEPTS

• **Social and professional topics** → *Computing education*; *Gender*;

KEYWORDS

self-efficacy, CS career orientation, programming education, Scratch, K12

1 INTRODUCTION

An increasing number of countries are currently enriching their elementary school education with computing and programming courses [3], using Scratch and other languages specifically designed for early programming education. Moving the introduction of programming from universities to high schools and to elementary schools means that programming education needs to adjust —not only in terms of programming environments, but also in terms of pedagogical approaches.

The literature has suggested several factors that influence learning performance in programming. These include age [8, 16, 18], previous programming experience [2, 15, 20] and self-efficacy [12, 15, 17]. Self-efficacy represents the belief that one can successfully execute behaviours required to produce a desired outcome [4]. In education research, self-efficacy has become one of the most important variables that helps explain the relationship between past performance and future results [12], even in middle-school students [5]. Self-efficacy beliefs have been found to affect career orientation, through impacting the choice of students' college major [10, 17].

In this paper, we are interested in the factors that affect the performance of elementary school students in this new subject. We want to determine if the factors that have been found to apply to university-level students also apply to early programming education. At the same time, we want to examine if the early introduction of programming can affect career orientation, and if elementary school students hold specific stereotypes on computer scientists that have been found to be common [11] and to affect their orientation towards CS studies.

Specifically, we are examining the following research questions:

RQ1 How is the learning performance of elementary school students affected by self-efficacy and their intrinsic and extrinsic motivation within the context of a programming course?

RQ2 What is the effect of those factors, of the course, as well as of established stereotypes on CS scientists, on the selection of programming/CS as a future career path?

RQ3 Does age, gender and previous programming experience affect those factors and relationships?

To answer our research questions, we designed and ran an experimental programming course, teaching Scratch during a series of eight lessons to elementary school students and measuring their performance, self-efficacy and career orientation throughout the course.

The results show that (1) students' intrinsic and extrinsic motivation are strongly correlated with their self-efficacy and, for the case of intrinsic motivation, their inclination towards a CS career, (2) students' previous programming experience is related to their extrinsic motivation, self-efficacy and CS career orientation, (3) the self-efficacy of female students is strongly correlated with their CS career orientation, and (4) course performance and stereotypical beliefs have no significant effect on CS career orientation.

2 BACKGROUND AND RELATED WORK

The relationship between performance and self-efficacy of university level students has been widely studied. In [15] it was found that, in the context of a CS1 course, self-efficacy affected course performance, was influenced by previous programming experience and increased as students progressed through the course. The Motivated Strategies for Learning Questionnaire (MSLQ) was used in a study with 39 university students in their introductory programming course where it was found that, of all motivational and learning strategies scales in the MSLQ, self-efficacy had the strongest correlation with the students' course performance [19]. Examining the relationships between MSLQ scores and academic performance, Credé and Phillips found that the self-efficacy scores had within the highest observed validities for grades in individual classes [6]. Lishinski et al. examined the interaction of self-efficacy and intrinsic and extrinsic goal orientation and their impact on students performance in a CS1 course, and found that the self-efficacy of female students had a different connection to performance than that of their male peers [12]. Examining the benefits of prior programming experience on the performance of students of CS1 courses, Wilcox and Lionelle also found gender differences, with inexperienced students performing similarly, but female students with prior programming experience outperforming their male peers [20].

A number of studies have been carried out on teaching programming concepts to novice programmers with block-based languages in general, and with Scratch in particular. Scratch was taught in middle-school classes containing a total of 46 students in the study presented in [13]. Evaluating the internalization of programming concepts, it was found that students had problems with concepts related to initialization, variables and concurrency. In [21], Wilson et al. presented an eight-week Scratch course given to four primary school classes with a total of 60 students aged eight to eleven, and evaluated it by analysing the projects that the students created. Seiter and Foreman [18] proposed a model for assessing computational thinking in primary school students and applied it on 150 Scratch projects, finding that design patterns requiring understanding of parallelization, conditionals and, especially, variables were under-represented until a certain age. Programming abstractions related to variables and procedures were also found to be underutilized in a dataset of 250 thousand Scratch programs [1]. During an online Scratch course with over two thousand active participants, students over twelve years of age were measured to perform significantly better in questions related to operators and procedures [8].

On the topic of CS career orientation, strong social support and high self-efficacy have been found to be associated with strong orientation of undergraduates toward CS careers [17]. Lewis et al. found that university-level students measure their fit in CS according to their beliefs on stereotypical traits of computer scientists [11]. Discrepancies between self-beliefs and perceptions of professionals in science-related disciplines are known to be associated with lower interest in those disciplines [9]. This indicates that stereotypical beliefs about computer scientists might affect the attractiveness of CS as a career path. Using a draw-a-computer-scientist test, eight and
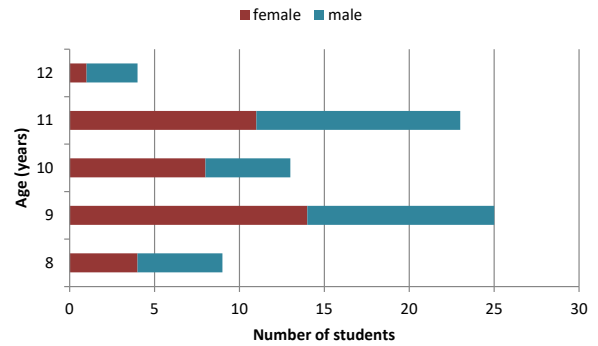

Figure 1: Age and gender of the participants

nine year old students were found to most often draw male computer scientists working alone and performing actions connected to technology in general but not specific to computer science [7].

3 METHODOLOGY

3.1 Participants

The subjects in our study are 74 students attending the three last grades (grades six, seven and eight, aged up to twelve years) of two public elementary schools in the Netherlands. 35 of the students were attending a traditional school, while 39 were attending a bilingual Montessori school. To cater for limitations in the classroom size, in each school the children were randomly separated into two groups, resulting in a total of four groups. The lessons were given during regular school hours and were taught by the authors.

Figure 1 shows the gender and age distribution of the children participating in the experiment. Almost half of the students (38, or 51.35%) are female. The ages vary from eight to twelve years, with an average of 9.84 years.

3.2 Materials and measures

The materials used in the experiment include the lessons materials, a student profile & background questionnaire, where the students reported their age, gender, previous programming experience and whether they know any computer scientists in their social cycle, tests measuring their performance, and questionnaires measuring their self-efficacy, motivation, CS career orientation, beliefs and fit on stereotypes.[1]

3.2.1 Lessons material. The course was composed of eight hour-long lessons, spread in the same number of weeks. It started with one introductory lesson for each group, in which we set up the computers and created Scratch accounts for the students. In addition to that, we showed them a demo of how Scratch works and the basic idea of Scratch, that you can control a sprite on the screen with blocks.

We utilized the Scratch teaching material developed for the online introductory Scratch programming course presented in reference [8], which has been used by over 12,000 children to date.

3.2.2 Learning performance. To gain an understanding on the progress of the students towards learning about programming concepts, we used an interim and a final test at the fifth and the eighth week respectively. The tests included a total of 17 questions, eleven of which were multiple choice and six were free text, which were graded manually. Four of the questions (three multiple choice and one free text) were not taken into account, following common practice to discard questions with less than 25% success rate as too hard to be representative. For the remaining questions, all answers received the same weight towards the final scores. As a result, the test scores in the interim and the final tests are calculated as the total correct answers divided by the number of questions.

3.2.3 Self-efficacy and motivation assessment. To measure the students' self-efficacy beliefs and extrinsic and intrinsic motivation we used the self-efficacy, extrinsic goal orientation and intrinsic goal orientation subscales of the MSLQ [14]. MSLQ consists of fifteen subscales designed from classic social-cognitive learning theories and is widely used as a self-report instrument for measuring student motivation and learning strategies and for subsequently predicting academic performance [6]. The self-efficacy subscale comprises of eight statements which assess both expectancy for success in the course and self-efficacy. They include judgements about the student's abilities to accomplish tasks, as well as about her confidence in her skills to perform those tasks.

Using questionnaires at the beginning, the middle and the end of the course, the students rated themselves on a seven point likert-scale from 'does not apply at all to me' to 'very true of me'. As specified in the MSLQ, students' self-efficacy scores were computed by taking the average of the points given to the eight statements of each questionnaire. The extrinsic and intrinsic motivation scores were similarly calculated, using the corresponding MSLQ subscales.

3.2.4 Stereotypes, student fit and career orientation. We measured the students' belief and fit to four stereotypical traits of computer scientists [11]: *Singularly focused*, indicating that CS requires an obsession with CS, *asocial*, indicating that computer scientists have limited social skills, *competitive*, and *male*. The student profile & background questionnaire included self-assessment statements for each of the singularly focused, asocial and competitive traits, where the students rated their fit on a seven point likert-scale from 'does not apply at all to me' to 'very true of me'. Using the same scale, the students replied to the statement 'I want to become a programmer when I grow up' at the beginning, middle and end of the course.

The students' beliefs regarding the stereotypes were measured using statements like 'Programmers love programming and prefer it over other hobbies' for the singularly focused, asocial, competitive and male traits, where they indicated their beliefs on a seven point likert-scale.

3.3 Data collection and analysis

Table 1 summarizes the timeline of the data collection activities throughout the course. In the first week, the students were given the profile & background questionnaire, which included the self-assessments for stereotypes fit and career orientation. The questionnaires containing the MSLQ subscales for self-efficacy, intrinsic and extrinsic goal orientation were given in the second week, along

Table 1: Lessons and data collection timeline

Week 1	Student profile & background questionnaire
	Stereotypes fit self-assessment
	CS career orientation assessment
	Setup & introductory Scratch lesson
Week 2	Self-efficacy assessment
	Intrinsic & extrinsic goal orientation
	Stereotypes beliefs
	Lesson 1 (from [8])
Week 3	Lesson 2
Week 4	Lesson 3
Week 5	Self-efficacy assessment
	CS career orientation assessment
	Interim test
	Lesson 4
Week 6	Lesson 5
Week 7	Lesson 6
Week 8	Self-efficacy assessment
	CS career orientation assessment
	Final test

with the questions on their beliefs on stereotypes. The students' progress was measured using tests at the middle and at the end of the course, following a repeated self-efficacy and career orientation assessment.

To answer **RQ1**, we analysed the students' test results and looked at individual correlations with their motivation and self-efficacy measurements. For **RQ2**, we calculated the students' fit to each of the stereotypical traits by comparing the scores they gave to their beliefs about computer scientists to their self-assessment scores. We then examined correlations between the fit, the aforementioned factors and the students' self-declared CS career orientation. To answer **RQ3**, we looked at individual correlations and correlation differences between girls and boys, students with and without prior programming experience, and groups of different ages, which were tested for significance. We used the Pearson correlation coefficient for calculating all p-values presented in the results (unless otherwise indicated).

4 RESULTS

Table 2 presents descriptive statistics of the various types of data collected during the experiment. In the following sections, we present the results obtained for each of the research questions.

4.1 RQ1: Learning performance

Learning performance was measured using interim and final tests, the results of which are summarized in Figure 2. Figures 3 and 4 show the results of the self-efficacy and the intrinsic and extrinsic motivation measurements.

Analysing the motivation and self-efficacy measurements in relation to the test results, we found no significant correlation. In other words, the motivation and self-efficacy appear to have no significant effect on the performance of the students on the tests.

Table 2: Descriptive statistics of collected data. Test scores have a maximum value of one, all other items are on a seven point likert-scale.

|	N	Avg	Mean	StdDev
Self-efficacy Initial	74	5.11	5.25	1.09
Self-efficacy Interim	73	5.18	5.25	1.10
Self-efficacy Final	66	5.33	5.56	1.04
Intrinsic Goal Orientation	74	5.01	5.00	1.21
Extrinsic Goal Orientation	74	4.37	4.25	1.37
Test Score Interim	71	0.41	0.33	0.23
Test Score Final	66	0.50	0.43	0.21
CS Career Orientation Initial	73	2.71	2.00	1.59
CS Career Orientation Interim	73	2.70	3.00	1.66
CS Career Orientation Final	66	2.89	3.00	1.91
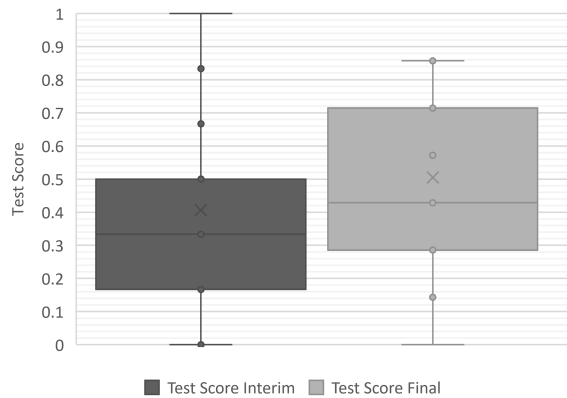
Figure 2: Test results
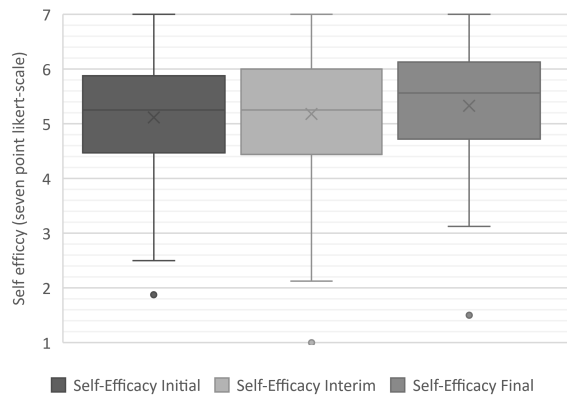
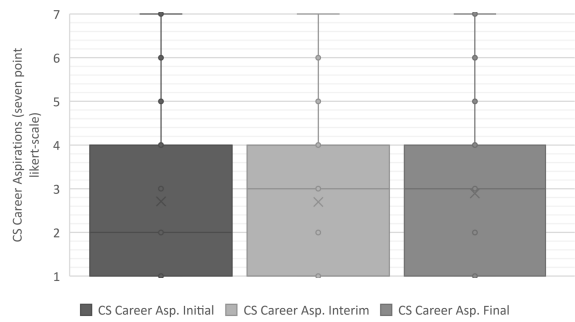
Figure 3: Self-efficacy measurements

Examining the interactions between self-efficacy and motivation, we found strong correlations: extrinsic motivation was significantly correlated with the self-efficacy measurements throughout
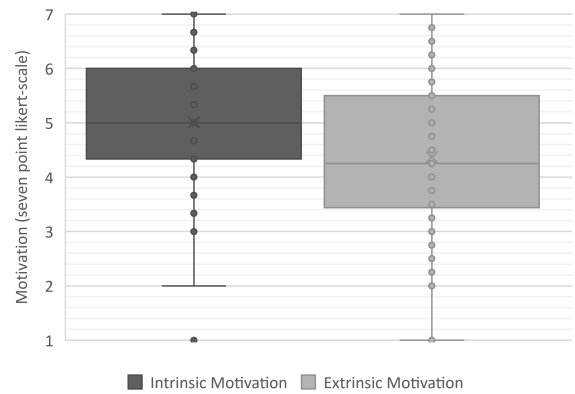
Figure 4: Motivation measurements

Figure 5: Responses to CS career orientation questions

the course (p <0.001 in all three cases). The same applies for intrinsic motivation (p <0.001 for the initial and interim self-efficacy measurements, p = 0.002 for the final one).

> The test performance of the students was not significantly affected by their self-efficacy or motivation. Self-efficacy was measured to be strongly correlated with both intrinsic and extrinsic motivation.

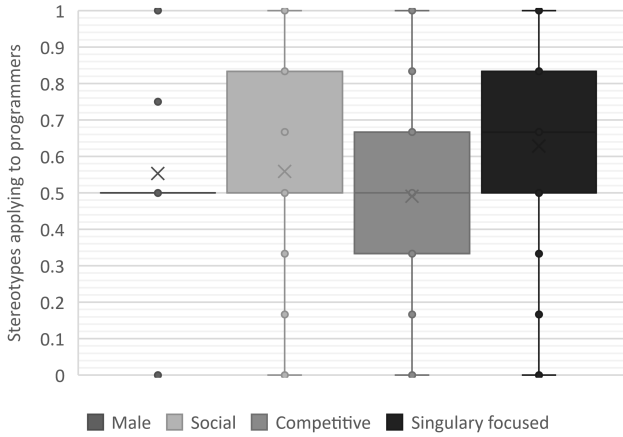## 4.2 RQ2: Programming career orientation

The attractiveness of programming as a future career path was measured using self-assessment questions, the results of which are summarized in Figure 5.

The course did not affect the CS career orientation of the students: there was no significant difference between the repeated CS career orientation measurements during the course. Test performance also appears to have no effect on the measurements. This, however, does not apply for self-efficacy, especially in the middle of the course, when a strong correlation with CS career orientation exists (p = 0.011 for the interim measurements of the two variables). This correlation is attributed only to female students, which we examine separately in Section 4.3.

The strongest factor impacting the selection of programming as a future career path is found to be the students' intrinsic motivation, which is significantly correlated with the initially measured CS

Table 3: Students' perceptions on the stereotypical traits of programmers (converted to 0 to 1 scale)

Stereotypical trait	N	Avg	Mean	StdDev
Male	66	0.55	0.50	0.20
Social	74	0.56	0.50	0.24
Competitive	74	0.49	0.50	0.29
Singulary focused	74	0.63	0.67	0.29

Figure 6: Student perceptions on stereotypical traits of programmers

career orientation (p = 0.029). The same effect was not significant for extrinsic motivation.

Next we examined the students' perceptions on stereotypical traits of computer scientists. Figure 6 and Table 3 summarize the students' responses to the statements described in Section 3.2.4. The results suggest that the students are not inclined towards any particular beliefs about computer scientists. Moreover, the students' fit to the stereotypical traits based on their self-assessment scores was found to have no significant effect on their CS career orientation.

> The students' inclination towards a CS career was strongly correlated with their intrinsic motivation for the course and their self-efficacy in the middle of the course. Their course performance and stereotypical beliefs have no significant effect on CS career orientation.

4.3 RQ3: Age, experience and gender effects

The *age* of the students was found to have no effect on their performance, career orientation, or any other of the examined variables.

In contrast, *previous programming experience* was a factor with strong effects. It was significantly correlated with the extrinsic motivation for the course (p = 0.005), as well as with the CS career orientation (p = 0.025 for the initial measurement and p = 0.031 for the interim one). It was also found to be significantly correlated with the initial and interim self-efficacy (p = 0.001 for both measurements).

Table 4: Gender differences in self-efficacy – test performance correlation (p-values)

Correlation	Females	Males
Self Eff. Initial - Test Score Interim	0.334	0.320
Self Eff. Interim - Test Score Interim	0.162	0.224
Self Eff. Final - Test Score Final	0.909	0.925

We further examined the effect of *gender* in the relationship between test performance and self-efficacy. Table 4 summarizes the results. For female students, the correlation is stronger than for male students, but it does not become significant throughout the course.

Regarding CS career orientation, no significant difference was found between male and female students. However, the factors that affect the attractiveness of programming as a career path were found to be different: For female students, a significant correlation was found between their CS career orientation measurement at the middle of the course and their self-efficacy measurements (p = 0.018 for the interim self-efficacy measurement and p =0.022 for the final one). This correlation was not significant for male students.

Examining students' perceptions on stereotypical traits of computer scientists, female students disagreed about the Male trait. A significant difference (Chi-square test, p = 0.014) was found between the answers of male and female students on the Male trait. In other words, female students replied that programming is a profession more for women, while male students that it is more for men.

> The students' previous programming experience is strongly correlated with their extrinsic motivation, self-efficacy and CS career orientation. For female students only, their CS career orientation is significantly related to their self-efficacy.

5 DISCUSSION

Having previous programming experience was a strong factor in our study: it was found to be significantly correlated with extrinsic motivation, CS career orientation and self-efficacy. This strong effect of prior experience has also be found to apply to CS1 courses [2, 15, 20]. In our experiment, there might be an additional reason for this observation: that programming experience, before our experimental courses, could have been obtained only through home-based or extra-curricular activities (we have confirmed that no previous programming lessons were given to the participants within the school environment). Therefore, having prior programming experience indicates that the students had sought to learn programming themselves, or were encouraged by their environment towards programming, and this proved to give them a significant head start in the course.

The age of the students was found to have no significant effect on their performance. This might be attributed to the fact that the age range was not large, with the majority of the students being from nine to eleven years old. A study on an online Scratch course has found performance differences to be significant after the age of twelve years [8]. Moreover, between elementary school children, differences can become apparent in the context of specific

programming concepts like conditionals or repetitions [16, 18], which were not studied seperately in this experiment.

Beliefs on stereotypical traits of computer scientists have been found to affect how university-level students measure their fit in CS [11]. Fit is important, because discrepancies between self-beliefs and perceptions of professionals in science-related disciplines are associated with lower interest in those disciplines [9]. The effect of students' fit to stereotypical traits was not measured in our experiment, because the students were not measured to hold any of the four stereotypes about computer scientists. This might be attributed to their age; we could assume that perceptions about stereotypical traits of computer scientists have not been developed yet for this age group. Even the Male trait was not assumed, with students favouring neutral or their own gender as typical for the profession. It is therefore to be expected that their CS career orientation was not related to any of the already documented stereotypical traits of computer scientists.

Related to our findings on gender differences on self-efficacy is the work of Lishinski et al. [12]. In their work, they found self-efficacy to be correlated with motivation, but also to have a gender-dependent feedback loop with course performance. Female students of the CS1 course were found to adjust their self-efficacy beliefs based on their performance more efficiently than male students, which could cause them to disengage from CS because they were more prone to internalize early failures in the course.

In our study, we did not observe significant correlations between test performance and self-efficacy. We speculate that this might be due to the tests being too hard for the students to be representative of their course performance, or to the students of this age group not being experienced enough with written tests to produce results indicative of their learning progress. As a result, we believe that the tests grades, depicted in Figure 2, are too low to represent the students' performance distribution or to be correlated with self-efficacy measurements. We did, however, observe a gender difference related to self-efficacy: its correlation with CS career orientation, which was stronger for female students. We believe that, in view of the gender participation gap in CS, the implications of those differences are significant: possible performance failures in early programming courses could lead young girls to prematurely reject a CS career believing, for example, that programming is 'not their thing'. A qualitative study would be useful in further examining this implication.

5.1 Threats to validity

A threat to the validity of our evaluation concerns the participants in our experiment, who might not be representative of all students or even of all students in our country. To mitigate this effect we ran the experiment in four groups from two different public schools. The schools were selected based on factors that are not relevant to the experiment (mainly, their proximity to the university).

Another threat to the validity of our results is the possible influence of the course tutors on the students, on their performance and on their beliefs about the stereotypical traits of computer scientists. While this effect cannot be measured, we compared the results obtained from the students that were taught by each author and found no significant differences in any of the metrics.

With respect to instruments used during the experiments, students might have been too young to understand the statements used for their self-assessment of stereotypical traits. During the courses, students asked for explanations regarding some traits (for example, for the social one), while we had to adjust the explanation of others to make them more understandable. For example, we had to explain the 'singularly focused' with the statement 'Programmers love programming and prefer it over other hobbies'. The same applies for the students' self-assessment of self-efficacy and motivation which was based on MSLQ, where they needed explanations for some of its statements. When explanations were needed, we ensured that we provided them to all four groups of the experiment. However, it was outside the scope of our study to create a verified self-efficacy assessment instrument for elementary school students.

6 CONCLUDING REMARKS

The goal of this paper is to examine what affects the learning performance of school-aged children during an introductory programming course, and whether the course or other factors could affect their CS career orientation. To this end, we taught a series of eight Scratch lessons to elementary school students, measuring their performance, self-efficacy and career orientation throughout the course.

We found the students' intrinsic and extrinsic motivation and previous programming experience to be important factors, being significantly correlated with their self-efficacy and their inclination towards a CS career. The students were not measured to believe to any of the documented stereotypes about computer scientists, which could be attributed to their young age. Our study highlighted gender differences, with the CS career orientation of female students being more related to their self-efficacy. We believe that, in view of the gender participation gap in CS, the implications of those differences are significant and worth exploring further through qualitative studies.

Our research gives rise to other directions for future work. Firstly, we intend to replicate these findings on a larger scale. An intermediate, but necessary, step towards this direction is devising self-assessment instruments for self-efficacy and motivation that are targeted to younger audiences.

REFERENCES

[1] Efthimia Aivaloglou, Felienne Hermans, Jesús Moreno-León, and Gregorio Robles. 2017. A Dataset of Scratch Programs: Scraped, Shaped and Scored. In *Proceedings of the 14th International Conference on Mining Software Repositories (MSR '17)*. IEEE Press, Piscataway, NJ, USA, 511–514. https://doi.org/10.1109/MSR.2017.45

[2] Christine Alvarado, Gustavo Umbelino, and Mia Minnes. 2018. The Persistent Effect of Pre-College Computing Experience on College CS Course Grades. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 876–881.

[3] Computing at School Working Group. 2012. *Computer Science: A Curriculum for Schools*.

[4] A. Bandura. 1977. Self-efficacy: toward a unifying theory of behavioral change. *Psychological review* 2, 84 (1977), 191–215.

[5] Shari L. Britner and Frank Pajares. 2006. Sources of science self-efficacy beliefs of middle school students. *Journal of Research in Science Teaching* 43, 5 (2006), 485–499.

[6] Marcus Credé and L. Alison Phillips. 2011. A meta-analytic review of the Motivated Strategies for Learning Questionnaire. *Learning and Individual Differences* 21, 4 (2011), 337 – 346.

[7] Alexandria K. Hansen, Hilary A. Dwyer, Ashley Iveland, Mia Talesfore, Lacy Wright, Danielle B. Harlow, and Diana Franklin. 2017. Assessing Children's Understanding of the Work of Computer Scientists: The Draw-a-Computer-Scientist Test. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE '17)*. ACM, New York, NY, USA, 279–284. https://doi.org/10.1145/3017680.3017769

[8] Felienne Hermans and Efthimia Aivaloglou. 2017. Teaching Software Engineering Principles to K-12 Students: A MOOC on Scratch. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track*. IEEE Press, Piscataway, NJ, USA, 13–22.

[9] James Daniel Lee. 1998. Which Kids Can "Become" Scientists? Effects of Gender, Self-Concepts, and Perceptions of Scientists. *Social Psychology Quarterly* 61, 3 (1998), 199–219.

[10] Robert W Lent and Gail Hackett. 1987. Career self-efficacy: Empirical status and future directions. *Journal of Vocational Behavior* 30, 3 (1987), 347 – 382.

[11] Colleen M. Lewis, Ruth E. Anderson, and Ken Yasuhara. 2016. "I Don'T Code All Day": Fitting in Computer Science When the Stereotypes Don'T Fit. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ACM, 23–32.

[12] Alex Lishinski, Aman Yadav, Jon Good, and Richard Enbody. 2016. Learning to Program: Gender Differences and Interactive Effects of Students' Motivation, Goals, and Self-Efficacy on Performance. In *Proceedings of the 2016 ACM Conference on International Computing Education Research*. ACM, 211–220.

[13] Orni Meerbaum-Salant, Michal Armoni, and Mordechai (Moti) Ben-Ari. 2010. Learning Computer Science Concepts with Scratch. In *Proceedings of the Sixth International Workshop on Computing Education Research*. ACM, 69–76.

[14] R. Paul, S. Smith, M. L. Genthon, G. G. Martens, C. L. Hauen, G. G. Martens, M. Genthon, and P. Wren. 1991. *Technical Report No. 91-B-004: A Manual for the Use of the Motivated Strategies for Learning Questionnaire (MSLQ)*. Technical Report. The Regents of The University of Michigan.

[15] Vennila Ramalingam, Deborah LaBelle, and Susan Wiedenbeck. 2004. Self-efficacy and Mental Models in Learning to Program. In *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ACM, 171–175.

[16] Kathryn M. Rich, Carla Strickland, T. Andrew Binkowski, Cheryl Moran, and Diana Franklin. 2017. K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17)*. ACM, New York, NY, USA, 182–190.

[17] Mary Beth Rosson, John M. Carroll, and Hansa Sinha. 2011. Orientation of Undergraduates Toward Careers in the Computer and Information Sciences: Gender, Self-Efficacy and Social Support. *Trans. Comput. Educ.* 11, 3, Article 14 (Oct. 2011), 23 pages.

[18] Linda Seiter and Brendan Foreman. 2013. Modeling the Learning Progressions of Computational Thinking of Primary Grade Students. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*. ACM, 59–66.

[19] Christopher Watson, Frederick W.B. Li, and Jamie L. Godwin. 2014. No Tests Required: Comparing Traditional and Dynamic Predictors of Programming Success. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. ACM, 469–474.

[20] Chris Wilcox and Albert Lionelle. 2018. Quantifying the Benefits of Prior Programming Experience in an Introductory Computer Science Course. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*. ACM, New York, NY, USA, 80–85.

[21] Amanda Wilson, Thomas Hainey, and Thomas Connolly. 2012. Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In *European Conference on Games Based Learning*. Academic Conferences International Limited, 549.