



AARHUS UNIVERSITY



Cover sheet

This is the accepted manuscript (post-print version) of the article.

The content in the accepted manuscript version is identical to the final published version, although typography and layout may differ.

How to cite this publication

Please cite the final published version:

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>

Publication metadata

Title:	Students' interest in Scratch coding in lower secondary mathematics.
Author(s):	Niels Bonderup Dohn
Journal:	British Journal of Educational Technology
DOI/Link:	https://doi.org/10.1111/bjet.12759
Document version:	Accepted manuscript (post-print)

General Rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

If the document is published under a Creative Commons license, this applies instead of the general rights.

Students' interest in Scratch coding in lower secondary mathematics

Niels Bonderup Dohn

Niels Bonderup Dohn is an associate professor at the Danish School of Education, Aarhus University, Denmark. His research interests are centred on students' interest and motivation in technology-enhanced STEM education.

Address for correspondence: Niels Bonderup Dohn, Danish School of Education, Aarhus University, Jens Chr. Skous Vej 4, Nobelparken, Building 1483, DK-8000 Aarhus C, Denmark.

Email: dohn@edu.au.dk

ORCID: 0000-0002-4560-556X

Abstract

The ability to code computer programs is considered an important part of literacy in today's society. This paper reports from a case study in two sixth grade classes where Scratch coding was part of six mathematics lessons. The aim of the study was to investigate how Scratch coding affected students' interest development in coding and in mathematics. Data was collected using a convergent parallel mixed methods design. The results show a slight, but nevertheless significant, negative effect on students' average interest in coding, as well as in mathematics. Students attributed this to the level of difficulty and the tedious workflow, indicating that their waning interest was due to the prescriptive nature of tasks that offered neither a sense of accomplishment nor the chance for autonomous input. However, situational interest was triggered in off-task coding situations. These situations were not related to the mathematical coding tasks but to the use of existing Scratch games and animations. The findings point to the importance of design principles that allow students an opportunity to tinker, but also to a need for an increased focus on facilitating the development of design knowledge within teacher professional development.

Keywords: Interest, Scratch, coding, mathematics

Structured practitioner notes

What is already known about this topic

- Computer coding is considered an important part of literacy in today's society.
- Scratch is a visual coding environment that is intended to foster creativity and increase motivation to engage with computers.
- Scratch coding in school settings has been shown to have a positive effect on students' learning.

What this paper adds

- Scratch coding tasks had a significant negative impact on students' interest in coding and mathematics.
- An overly structured lesson design and a lack of autonomy were the main reasons why students' interest declined.

- Boys and girls were equally interested in coding.

Implications for practice and/or policy

- Scratch is a useful tool for coding in school, but coding tasks should be designed carefully.
- Instead of highly structured, abstract tasks, design principles like guided tinkering and fantasy should be considered.

The ability to code computer programs is an important part of literacy in today's society. Coding is part of logical reasoning, which represents one of the key skills of what are now called "21st century skills" (European Schoolnet, 2014). In an attempt to increase interest in coding, much effort has gone into developing tools and activities for young people. One such initiative is Scratch, a visual coding environment developed by the Lifelong Kindergarten group at the MIT Media Lab. Scratch is intended to foster creativity and increase motivation to engage with computers. The Scratch online community's slogan "Imagine, Program, Share" indicates that sharing and the social aspects of creativity are important parts of the philosophy behind Scratch. The term 'Scratching' refers to the fact that existing code can be reused for other purposes and easily combined, shared and adapted to new scenarios – in Scratch, users are encouraged to 'remix', downloading and building upon projects developed and made publicly available by other users (Kafai & Burke, 2014).

A growing number of schools around the world use Scratch as a gateway to coding. Scratch coding has been successfully utilized in subjects including mathematics, science and arts. In mathematics, Lewis and Shah (2012) measured correlation between the results of coding quizzes and grades in maths tests. Ke (2014) found that students showed significantly more positive attitudes towards mathematics after Scratch-based activities where they developed mathematical games. Zavala, Gallardo and García-Ruiz (2013) found improvements in the identification and comparison of numbers after Scratch coding activities. Furthermore, studies have found that students develop their mathematical thinking (e.g. Calao, Moreno-León, Correa, & Robles, 2015) and problem-solving skills (e.g. Su, Yang, Hwang, Huang, & Tern, 2014) while learning to code with Scratch.

All the referenced studies show that Scratch is a useful tool for coding in school settings and that its application has a positive effect on students' learning. Very little is known about younger students' motivation for engaging in Scratch coding in school. There are a few relevant studies based on contemporary motivational theory, however. Feng and Chen (2014), for example, investigated the influence of goal specificity and scaffolding on the learning outcomes of Taiwanese elementary school students. Studying lessons on computer game design, they showed that students provided with problematizing scaffolds demonstrated better self-regulation than those provided with structuring scaffolds. In a study by Ruf, Mühlhling, and Hubwieser (2014), the effects of using one of two different programming environments ("Scratch" and "Karel the Robot") in secondary schools were tested. They found that classes using Scratch had higher intrinsic motivation and performed better than the Karel classes. Nikou and Economides (2014) compared students' motivation to learn coding using Scratch and App Inventor for Android in K-12 educational settings. They found an increase in students' intrinsic goal orientation, task value, control of learning beliefs and self-efficacy when using these two entry-level learning environments for coding. However, even though Scratch was developed with the aim of making coding interesting to young people, interest in Scratch coding has yet to be investigated. Indeed most previous studies have evaluated the interest construct in

vague terms, for example by surveying emotional aspects (enjoying coding in Scratch, having fun with Scratch etc.) without reference to contemporary motivation theory. Enjoyment and fun can occur for many reasons, of which interest is only one. To avoid lack of conceptual precision, it is therefore necessary both to clarify what interest means in a psychological sense and to consider how the impact of interest can be effectively studied.

Concept of interest

Interest refers to the liking of and wilful engagement in an activity. Interest is closely related to intrinsic motivation. When people are freely doing what interests them, they are referred to as intrinsically motivated (Deci, 1992). Thus, interest is a *motive* for intrinsic motivation. Interest is a content-specific motivation variable; i.e. it is always related to specific topics, tasks, or activities. This relationship is both cognitive (involving knowledge and experience) and affective (involving positive feelings and appreciation). Feelings of pleasure, happiness and well-being are typical emotional aspects of interest-based activities (Krapp, 2002).

Two types of interest have been the primary focus of educational research to date: situational and individual interest. Situational interest refers to the psychological state of engagement with content. Individual interest is conceptualized as a relatively stable motivational orientation that develops over time in relation to a particular topic or domain, and which is associated with increased knowledge, value, and positive feelings (Krapp, 2002).

Triggering situational interest involves the immediate affective experiences that individuals associate with their environment, and it appears to be especially important in catching students' attention (Mitchell, 1993). Among the triggers described as promoting students' situational interest are surprise, novelty, hands-on experience, and complexity (Bergin, 1999). Maintained situational interest is a more committed, deeper form of situational interest, in which individuals forge a meaningful connection with the content. Mitchell (1993) proposed that maintained situational interest could best be achieved by involving students in activities which they perceive as personally meaningful. When students do not find activities meaningful, triggered interest can decrease, become dormant, or disappear altogether.

An important question with respect to learning is how to capture students' interest and hold it for an extended period of time in order to stimulate a lasting state of intrinsic motivation. Let us assume that students who have been exposed to an exciting Scratch coding activity in school are stimulated and pay more attention than they did before. For some students, this interest may evaporate as soon as the lesson ends. For others, the interest persists over time and may develop into an individual interest in coding. This would be the case if students start creating colourful games and animated stories with Scratch in their leisure time, for example. Very little is known about this process, even though there are significant educational implications for teachers who strive to promote a long-standing interest in coding. Whereas teachers have little influence on the individual interests (or disinterest) students bring to class, they can influence the development of interests by, for instance, including coding activities to foster situational interest. From an educational point of view, situational interest is the real topic of concern, because it can be a potentially powerful way to motivate students who have little or no pre-existing interest in a topic (Deci, 1992; Hidi & Harackiewicz, 2000).

Students' interest in Scratch coding in lower secondary mathematics

While there is a growing amount of research into students' learning from Scratch coding, research has hitherto not considered how coding affects students' interest. Hence, the purpose of the study was to investigate to how a mathematical coding course affected students' interest development in coding and in mathematics. The research questions were:

RQ 1: *How does Scratch coding in mathematics affect students' interest development?*

RQ 2: *What are the reasons for interest development?*

Based on previous findings on Scratch as a motivating and fun environment, the working hypothesis was that sixth grade students would find Scratch coding in mathematics interesting.

Method

Research design

This study was carried out using a mixed methods design that combines qualitative and quantitative methods. This approach allows combining complementary strengths and non-overlapping weaknesses of quantitative and qualitative methods (Creswell & Plano Clark, 2017). More specifically, a convergent parallel design was used: quantitative and qualitative strands were implemented during the same phase of the research process, the methods were prioritized equally, and the strands were kept independent during analysis before combining the findings in an overall interpretation. This design was used in order to triangulate the methods by directly comparing and contrasting quantitative and qualitative findings for corroboration and validation purposes.

The quantitative measure of interest was conducted using a pre-test – post-test design. The treatment of the study was Scratch coding and the dependent variable was interest. The quantitative data were collected at the end of the first Scratch coding lesson and again at the end of the sixth lesson through a pen and paper-based questionnaire. The qualitative part comprised classroom observation and informal interviews in order to consider sources of students' interest. Participants in informal interviews were selected based on their engagement levels in the course. Both participants with a relatively high level of engagement and participants displaying a lower level of engagement were interviewed to attain a comprehensive picture.

The research design is based on the assumption that in collaborative learning, individual group members represent interdependent agents (cognitive perspective) who at the same time constitute a social entity that creates affordances and constraints for engagement in the activity (situative perspective) (Järvelä, Volet, & Järvenoja, 2010). Based on this framework, interest is seen as an individual phenomenon, mediated by social experiences (Bergin, 2016). The research framework uses a combined analytical unit that captures the individual perspective through questioning (questionnaire, interviews) as well as social mediating aspects through observation of activities. This way, observations constitutes the background for interviews and interpretation of students' interest development. Participants and educational context

The participants selected using convenience sampling were students (N = 44, 22 male and 22 female) aged 12-13 (M = 12.54, SD = .51) from two sixth grade classes at a Danish public school located in an

Students' interest in Scratch coding in lower secondary mathematics

urban area. The participants were familiar with computers and brought their own laptops for the coding course. All participants were novices at coding with no previous experience.






The course comprised six consecutive mathematics lessons over three weeks, each of 45 minutes, utilizing the learning resource *Programming in Maths*.

Programming in Maths (www.programmeringimatematik.dk) is a Scratch-based coding and mathematics-based curriculum for lower secondary students. Its aim is to enable students to engage with important mathematical ideas through learning to code. It incorporates three types of tasks: Scratch lessons (introductory lessons in Scratch), maths lessons (mathematical lessons applying Scratch coding), and coding lessons (for learning to code independent of school subject). The maths lessons comprise eight tasks centred on mathematical functions (variables and coordinates) and seven on geometry (angles, figures, and area). Each task includes additional challenges. One task, for example, was to draw polygons in Scratch, as shown in figure 1. Here, students were asked to insert a start button, make the figure Sprite move to a specific location, get Sprite to draw a line behind itself, to go forward, and to rotate by adding commands and values (Sprites are the objects that perform actions in a project).

Figure 1. Task example: to draw polygons in Scratch (the task is translated from Danish by the author while keeping the visual)

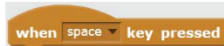









[Insert Figure 1 here]

Geometry – Polygons

Learning goals, you should be able to:	Coding goal, you should be able to:
Draw polygons Draw lines Draw line segments	Insert a start script  Make Sprite move to another place  Make Sprite draw a line  Make Sprite move  Make Sprite turn 

A polygon can be regular or irregular. When a polygon is regular, all sides and angles are equal and when it is irregular all sides and angles are not equal.

Task 4

- 1: Insert 
- 2: Add  and insert e.g. x=0 and y=0
- 3: Add 
- 4: Insert  and type e.g. 130
- 5: Insert  and type e.g. 90
- 6: Add again  and type e.g. 130
- 7: Insert again  and type e.g. 90
- 8: Insert  and type e.g. 130
- 9: Insert again  and type e.g. 90
- 10: Insert the last  and type e.g. 130



Challenge: what is the name of your polygon? What is the sum of the angles? Is the polygon regular or irregular?

Challenge: Make 4 regular polygons with more than 4 angles

Challenge: Make 2 irregular polygons with more than 6 angles

Challenge: Make 2 regular polygons and make Sprite split the polygons into triangles

Programming in Maths provides teacher support in the form of guidelines for implementing the designed curriculum effectively, stipulating learning goals, and presenting exemplary cases. The curriculum targets teachers with no previous coding experience or training.

All curriculum materials were developed by an educational consultant at the Centre of Educational Resources, funded by the Danish Ministry of Education. He had no formal or informal training in coding. His background for designing the curriculum materials was an interest in maths education

and coding in general and Scratch in particular. Despite his lack of training, the tasks in Programming in Maths in point of fact align with tasks previously documented to support the learning of mathematics through coding (Benton, Hoyles, Kalas, & Noss, 2017). Therefore, they form an appropriate outset for investigating how Scratch coding affects interest development.

The educational consultant was also a mathematics teacher at the school in question and a colleague of the teachers in the two classes. The consultant asked these two colleagues to try out the maths lessons coding tasks with their classes. His aim was to user test the tasks before sharing them with mathematics teachers at national level. He was not present in the classrooms during field trials.

The two participating teachers, one female with more than ten years of experience teaching mathematics, the other male with more than twenty years of experience, had no previous experience with Scratch or any other form of computer coding. The teachers prepared for the lessons by going through the support materials and by trying tasks out themselves. Each teacher taught her or his own class. They did not observe each other's lessons, but discussed affordances and constraints with each other and the educational consultant during the period. The role of the author was purely observational to ensure an objective perspective on interest development. Thus, he was neither involved in the design nor the implementation of the curriculum and did not intervene in lessons.

Each lesson was structured similarly: the teacher presented selected maths lesson coding tasks to the students, answered questions and provided support. During lessons, students worked individually or in pairs, solving the tasks on their laptops. The teacher's role was to scaffold students' work by providing help or guidance when needed. In the last two lessons, the majority of the students in both classes showed a lack of commitment while solving tasks. Therefore, the teachers gave the students permission to download games from the Scratch online community, but asked them to play for a maximum of ten minutes before recoding the game and observing how the game changes character. Some students were absorbed in gaming for the entire lesson, whereas others recoded existing games or coded their own project from the ground up.

Data collection

As mentioned, three forms of data collection were employed in the study: questionnaires, classroom observations and informal interviews. Data were collected solely by the author based on the research questions.

The Interest Scale (IS) was used to measure students' interest in mathematics and coding. Items were adapted from various validated instruments (Linnenbrink-Garcia et al., 2010; Mitchell, 1993). Being a five-point Likert scale, it consisted of 20 items on interest. In addition, two items on background variables (gender and age) were included. Since the sample was too small for exploratory factor analysis, the 20 items were treated as two scales on interest based on their thematic content: one on interest in mathematics (10 items) and one on interest in coding (10 items). Each interest scale included five items on the affective component of interest and five items on the value component. Examples are: "Our maths lessons are fun" (affective), "It is important for me to be able to use maths" (value), "It is fun to code" (affective), and "It is important for me to learn to code" (value). In addition, students were asked to describe their positive and negative

experiences with coding in two open-ended items in the post-test questionnaire. These two open questions were included in order to reveal something about the specific reasons for interest (or lack of interest) that cannot be grasped from the analysis of closed items. The questionnaire was produced in a pen-and-paper format and distributed for 15 minutes to all students present at the beginning of the course and again at the end. A total of 44 responses were completed and returned in the pre- and post-test, corresponding to a response rate of 100%.

Classroom observations were "naturalistic observations"; that is, they were embedded in the regular settings of the activities (Patton, 2002). The observations were focused on situations in which students were attentive and emotionally engaged but also on situations where students seemed demotivated or frustrated. These observations were used in the development of questions asked during informal interviews. The informal interview is a method of interview where questions emerge from the immediate context and are asked in the natural course of things. Students were asked how they experienced a given situation, why it was or was not interesting. The informal interviews were short, lasting 2-4 minutes. Students' answers were noted on paper. All students were interviewed at least once during the course. One advantage of informal interviews is that the salience and relevance of questions increases when interviews are built on and emerge from observations; the interview can be matched to individuals and circumstances (Patton, 2002). However, prompted statements, for their part, may interrupt student actions and interactions.

Data analysis

To score the interest scale, the scores for items that are stated in a negative manner were reversed (items 4, 10, 13, and 19). The reliability coefficient (Cronbach's α) of the instrument was calculated for each interest scale. In the pre-test, the reliability coefficient was .91 for interest in mathematics and .82 for interest in coding. In the post-test, the coefficients were .87 and .86, respectively. Gender difference was evaluated with the nonparametric Spearman's rank correlation. Gender was tested against each of the 20 interest items. Correlation coefficients were between -.22 and .28 ($p > .06$) in the pre-test, and between -.22 and .17 ($p > .15$) in the post-test. No significant correlations were found, indicating that boys and girls were equally interested in both mathematics and coding.

The qualitative data sources included answers to the two open items in the IS post-test as well as notes from classroom observations and informal interviews. All data were coded according to reasons for interest development. More specifically, thematic analysis of students' responses was conducted, i.e. themes identified are strongly linked to the data themselves (Patton, 2002). The specific procedure for thematic analysis as defined by Braun and Clarke (2006) was used: 1) prepare and organize the data for analysis, 2) explore and code the data, 3) collate codes into themes, 4) review themes and generate a thematic 'map' of the analysis, 5) define themes, and 6) relate the analysis back to the research question and literature. During the data analysis, answers and notes were read individually and grouped based on reasons for interest development (Table 2). The similarities and differences in views were considered while grouping the data, and explanations were given in line with these findings.

Results

The effects of Scratch coding on students' interest. The questionnaire data revealed that the students found mathematics relatively interesting. The mean score was 3.68 (SD = 1.13) in the pre-test and

Students' interest in Scratch coding in lower secondary mathematics

3.63 (SD = 1.15) in the post-test. Students' average interest in coding was lower, as illustrated in Table 1. In the pre-test, the mean score was 3.16 (SD = 1.24). In the post-test, the mean was 3.06 (SD = 1.18). Students' average interest in mathematics as well as in coding was shown to have decreased following the Scratch coding lessons, as the effect sizes were slightly negative (Cohen's $d = -.05$ and $-.08$, respectively). The decrease was highly significant ($p < .0001$) despite the small sample. Thus, interest was not stimulated during this course as expected; instead, interest levels decreased – both in coding and in mathematics.

Table 1. Students' interest scores towards mathematics and coding

Scale	Scale item	Pre-test		Post-test		Cohen's d
		M (SD)	α	M (SD)	α	
Interest in maths	Maths is exciting	3.00 (1.18)	.91	3.11 (1.15)	.87	-0.05
	Our maths lessons are fun	2.80 (1.21)		2.91 (1.25)		
	I like maths	3.41 (1.19)		3.26 (1.21)		
	Maths is boring (R)	3.18 (1.19)		3.45 (1.19)		
	Maths is interesting	3.23 (1.08)		3.36 (1.14)		
	It is important for me to learn maths	4.48 (0.88)		4.33 (.92)		
	I would like to learn more maths	4.18 (1.13)		3.93 (1.21)		
	I need maths later in my life	4.37 (1.02)		4.19 (1.12)		
	It is important for me to be able to use maths	4.07 (1.09)		3.81 (1.19)		
	I never have a use for what we learn in maths (R)	4.07 (1.30)		3.95 (1.14)		
Interest in coding	Coding is exciting	3.20 (1.17)	.82	3.20 (1.17)	.86	-0.08
	I like coding	3.23 (1.26)		3.05 (1.16)		
	Coding is boring (R)	3.37 (1.29)		3.25 (1.20)		
	It is fun to code	3.30 (1.17)		3.14 (1.21)		
	Coding is interesting	3.20 (1.21)		3.07 (1.12)		
	It is important for me to learn to code	2.91 (1.21)		2.68 (1.20)		
	It is important for me to be able to code	2.64 (1.26)		2.86 (1.13)		
	I would like to learn more coding	3.20 (1.36)		2.89 (1.22)		
	I never have a use for coding (R)	3.35 (1.19)		3.27 (1.17)		
	It is important to know how to code	3.16 (1.29)		3.23 (1.24)		

Table 2. The themes of the open items, ranked by frequency ($N = 44$)

Positive experiences with Scratch coding
Learn to program/can be used later in life (n = 9)
Not just reading books/different from everyday maths instruction/get rid of maths (n = 9)
Something new (n = 7)
No time pressure (n = 5)
Fun (n = 5)
Fantasy/play (n = 3)
Negative experiences with Scratch coding
Tedious in the long run (n = 20)
Difficult/confusing (n = 12)
Monotonous tasks (n = 7)
Malfunction (n = 2)
Pseudo programming (n = 1)

Reasons for interest development. The qualitative data were analysed for reasons for interest development. Based on students' responses, two dominant themes – *task* and *instruction vs. tinkering* - were formulated based on the thematic analysis.

The *task* category includes students' descriptions deriving from the task itself. These responses primarily emphasized difficulties with the tasks and the monotony of the workflow (e.g., "I am not sure how to solve this", "difficult", "too much maths", "too much of the same thing", and "the task is boring"). It seems evident that students' engagement with tasks had a negative influence on their interest. The *task* category also includes positive descriptions, although these descriptions are fewer, as illustrated in table 2. They refer to affective descriptions (e.g., "It is fun"), value-related descriptions (e.g., "I can use coding later on in life"), the appreciation of doing something new or different from everyday mathematics instruction (e.g., "Mega cool to use a computer in mathematics"), and time (e.g., "I can work at my own pace").

The *instruction vs. tinkering* category includes responses that indicate tensions between students' experiences with structured coding tasks on one side and tinkering with game redesign on the other. When students showed lack of commitment during the lessons, the teachers gave them permission to download games from the Scratch online community. Whereas students generally found the coding tasks in mathematics monotonous, difficult, and tedious, most students were engaged in gaming and recoding off-task projects. The positive responses in Table 2 might reflect this, as these activities took place in parallel to the course.

Discussion

The presented study reports on a coding initiative which failed to achieve its aims. Students' interest was not increased as expected. In fact, students' mean interest in coding (as well as in mathematics) declined during the course. Students attributed this to difficulties with the tasks and the tedious workflow. According to the theory of self-determination, three essential needs are important for not only well-being, but for a variety of developmental processes, including interest development, as well as for internalising and integrating externally motivated goals and behaviour with the person's other goals. These three essential needs are competence, autonomy, and relatedness (Ryan & Deci, 2000). Two of these needs are relevant for explaining students' lack of interest (Deci, 1992; Krapp, 2005). Competence refers to the desire to feel efficacious, to have an effect on one's environment, and to be able to attain valued outcomes, whereas autonomy (self-determination) refers to the desire to be self-initiating and to have a sense of acting in accord with one's own sense of self. The need to feel competent was not fulfilled, as illustrated by the many student comments on how difficult and confusing they found the tasks. Neither was the need for autonomy, as students experienced the tasks as being somewhat restrictive. Research has shown that people may engage in strategies to make their performance of tasks more interesting and eventually develop interest in an activity that had been uninteresting (Hidi & Harackiewicz, 2000). Specifically, they can generate and use strategies to make mundane tasks more interesting (such as by turning them into a game). An example from the present study is when two female students spent a lesson personalizing their Sprites, without solving the tasks they had been given. However, learners desire freedom of action only when they believe that they are capable of successfully mastering impending tasks. The pursuit of an optimal level of autonomy is, at the same time, an important prerequisite for fulfilling the need of competence, since the successful mastering of a task can only be experienced when it has been

solved to some degree without the support and detailed instructions of others (Krapp, 2005). This was actually the case for one student, as he did not follow the task instruction but began exploring Scratch's functionalities: "It is more interesting to try it out on my own [than to solve the task]; I want to learn what it can do".

The qualitative data provided nuances and details that the quantitative data did not capture. Situational interest was triggered in some situations. However, these situations were not related to the mathematical coding tasks. When the students lost interest and engagement during the lessons, the teachers allowed them to download existing Scratch projects (e.g. games) and revise them for the rest of the lesson. Most students found these self-directed actions with non-curricular content interesting and motivating, as they were able to design their own games and animations. Here, choice and autonomy was one major source of interest as students had the opportunity to decide for themselves what they wanted to do and how they would do it. Especially those students that were frequent gamers were highly engaged. The self-directed tinkering activities offered them an opportunity to learn to understand computer games in detail. Recoding games appeared effective because activities that are perceived as being personally meaningful to students are a direct way to empower them and thus hold their interest (Mitchell, 1993). Meaningfulness refers to students' perception of the topics under study as meaningful to them in their present lives, i.e. the value component of interest (cf. Schiefele, 1991).

The tension between structured school tasks versus self-directed tinkering on Scratch projects lies in the externality versus internality of their basic goals. Self-directed activities have internal goals; that is, their aims concern the performance of an activity for its inherent satisfaction rather than for some separate consequence. This points to a challenge when designing coding tasks, as the learning objectives of courses within formal education may be incompatible with the principles of self-directed learning. This challenge may be addressed through a learning design which Bjerre and Dohn (2018) term 'guided tinkering'. This approach combines structure and tinkering, allowing for piecemeal tinkering, guided by clearly structured tasks. It builds on Resnick and Rosenbaum's (2013) description of tinkering, self-determination theory (Ryan & Deci, 2000), and the empirically documented need for guided instruction (Kirschner, Sweller, & Clark, 2006). It contains seven design principles: 1) *Teacher pre-defined projects*, 2) *Meaningful, preferably fun, projects*, 3) *Adequate complexity*, 4) *Projects constituted by a series of sub-problems*, 5) *Guided tinkerability*, 6) *Visual or graphical programming language appropriate for the target group*, and 7) *Media integration*.

Even though Scratch coding has been successfully utilized in school subjects like mathematics (Calao et al., 2015; Ke, 2014; Lewis & Shah, 2012; Su et al., 2014; Zavala et al., 2013), it is clear that coding as a structured educational activity did not in itself increase interest in the present study. The reason for this was primarily the controlled structure of the learning tasks. Furthermore, the design principles *Meaningful, preferably fun, projects* and *Guided tinkerability* were almost absent in the tasks, something which the interview data would seem to suggest is a prime reason for students' mean interest decline. It should be noted that the coding tasks are similar to previous Turtle Geometry tasks in LOGO (Abelson & DiSessa, 1980). LOGO coding was often introduced with activities such as generating lists of prime numbers and making simple line drawings that were not connected to students' interests or experiences. Rather than simply presenting relevant problems in abstract form (e.g., drawing polygons), designs should present problems in a more intrinsically interesting fashion, such as by connecting to the students' everyday lives (Wang & Zhou, 2011). It is

important to include aspects such as the use of challenges, curiosity, actual and perceived control, and fantasy in lesson design. In trying to make coding activities interesting, fantasy, in particular, should be taken into consideration. Malone and Lepper (1987) proposed fantasy as a highly important source of intrinsic motivation and interest. Fantasy and make-believe through coding may portray out-of-school situations to which school learning can be applied and thereby reinforce the value of coding.

This study has shown that even though Scratch is a 'fun' environment, as suggested in the literature, Scratch alone is not sufficient to spice up mathematical tasks and make them more interesting. It *does* matter how coding in mathematics is designed if it is to capture and hold students' interest. This points to the significance for an increased focus on facilitating the development of design knowledge within teacher professional development (Goodyear, 2005). Specifically, it shows the need for focusing on interest as a distinct design construct.

Conclusion

This study has shown that coding can have a negative impact on students' interest, even if the coding environment is Scratch. The primary reason for this negative impact was the controlled structure of the learning tasks, limiting student autonomy. According to Papert (1980), the most effective learning occurs when individuals are allowed to construct a product instead of being instructed directly. This could be paraphrased to also apply to interest. Rather than simply presenting tasks in abstract form, designs should present problems in a more intrinsically interesting fashion through the use of tinkering, perceived control and fantasy.

Limitations of the study

This study is apparently the first attempt to investigate students' interest (as distinct from other motivational constructs) in Scratch coding. The study has limitations and further research is needed to support the conclusions.

The results of the convenience sampling cannot be generalized to a broader population because of the potential bias of the sampling technique due to under-representation of subgroups in the sample in comparison to the population of interest. Furthermore, the analyses are based on a limited number of students. However, this study might count as a critical case; the students' declining interest during structured coding tasks could be expected to be valid for a broader range of students (Flyvbjerg, 2006).

Acknowledgements

The author wishes to thank the teachers and students on whom the success of the project was dependent.

Statements on open data, ethics and conflicts of interest

Requests for data may be made to the corresponding author.

The author has abided by all requirements in the Danish Act on Processing Personal Data as outlined by the Danish Protection Agency (www.datatilsynet.dk). Students were given full anonymity. They

have all given their voluntary consent to be part of the investigation after their parents had been informed of its purpose and of the possibility to decline participation.

The author reports no conflicts of interest.

References

- Abelson, H., & DiSessa, A. A. (1980). *Turtle geometry : The computer as a medium for exploring mathematics* (Reprinted ed.). Cambridge, Mass.: MIT.
- 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. doi:10.1007/s40751-017-0028-x
- Bergin, D. A. (1999). Influences on classroom interest. *Educational Psychologist*, 34(2), 87 - 98.
- Bergin, D. A. (2016). Social Influences on Interest. *Educational Psychologist*, 51(1), 1-16. doi:10.1080/00461520.2015.1133306
- Bjerre, A., & Dohn, N. B. (2018). Guided tinkering as a design for learning programming. In N. B. Dohn (Ed.), *Designing for Learning in a Networked World* (pp. 177-196). Oxon: Routledge.
- Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77-101. doi:10.1191/1478088706qp063oa
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing Mathematical Thinking with Scratch. In G. Conole, T. Klobučar, C. Rensing, J. Konert, & É. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings* (pp. 17-27). Cham: Springer International Publishing.
- Creswell, J. W., & Plano Clark, V. L. (2017). *Designing and conducting mixed methods research* (Third edition, international student edition ed.). Los Angeles: SAGE.
- Deci, E. E. (1992). The relation of interest to the motivation of behavior: A self-determination theory perspective. In K. A. Renninger, S. Hidi, & A. Krapp (Eds.), *The role of interest in learning and development* (pp. 43-70). Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc. .
- European Schoolnet. (2014). *Computing our future - Priorities, school curricula and initiatives across Europe*. Retrieved from Brussels: <http://www.eun.org/publications/detail?publicationID=481>
- Feng, C.-Y., & Chen, M.-P. (2014). The effects of goal specificity and scaffolding on programming performance and self-regulation in game design. *British Journal of Educational Technology*, 45(2), 285-302. doi:10.1111/bjet.12022
- Flyvbjerg, B. (2006). Five Misunderstandings About Case-Study Research. *Qualitative Inquiry*, 12(2), 219-245. doi:10.1177/1077800405284363
- Goodyear, P. (2005). Educational design and networked learning: Patterns, pattern languages and design practice. *2005*, 21(1). doi:10.14742/ajet.1344
- Hidi, S., & Harackiewicz, J. M. (2000). Motivating the Academically Unmotivated: A Critical Issue for the 21st Century. *Review of Educational Research*, 70(2), 151-179. doi:10.3102/00346543070002151
- Järvelä, S., Volet, S., & Järvenoja, H. (2010). Research on Motivation in Collaborative Learning: Moving Beyond the Cognitive–Situative Divide and Combining Individual and Social Processes. *Educational Psychologist*, 45(1), 15-27. doi:10.1080/00461520903433539
- Kafai, Y. B., & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. Cambridge, Massachusetts MIT Press.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26-39. doi:<https://doi.org/10.1016/j.compedu.2013.12.010>
- Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential,

- and Inquiry-Based Teaching. *Educational Psychologist*, 41(2), 75-86.
doi:10.1207/s15326985ep4102_1
- Krapp, A. (2002). Structural and dynamic aspects of interest development: theoretical considerations from an ontogenetic perspective. *Learning and Instruction*, 12(4), 383-409.
- Krapp, A. (2005). Basic needs and the development of interest and intrinsic motivational orientations. *Learning and Instruction*, 15(5), 381-395.
- Lewis, C. M., & Shah, N. (2012). *Building upon and enriching grade four mathematics standards with programming curriculum*. Paper presented at the Proceedings of the 43rd ACM technical symposium on Computer Science Education, Raleigh, North Carolina, USA.
- Linnenbrink-Garcia, L., Durik, A. M., Conley, A. M., Barron, K. E., Tauer, J. M., Karabenick, S. A., & Harackiewicz, J. M. (2010). Measuring Situational Interest in Academic Domains. *Educational and Psychological Measurement*, 70(4), 647-671. doi:10.1177/0013164409355699
- Malone, T. W., & Lepper, M. R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning, and instruction* (Vol. III: Conative and affective process analyses, pp. 223-253). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Mitchell, M. (1993). Situational interest: Its multifaceted structure in the secondary school mathematics classroom. *Journal of educational psychology*, 85(3), 424-436.
- Nikou, S. A., & Economides, A. A. (2014, 3-5 April 2014). *Transition in student motivation during a scratch and an app inventor course*. Paper presented at the 2014 IEEE Global Engineering Education Conference (EDUCON).
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Patton, M. Q. (2002). *Qualitative Research & Evaluation Methods*. Newbury Park: Sage.
- Resnick, M., & Rosenbaum, E. (2013). Designing for tinkability. In *Design, make, play: Growing the next generation of STEM innovators* (pp. 163-181). New York: Routledge.
- Ruf, A., Mühling, A., & Hubwieser, P. (2014). *Scratch vs. Karel: impact on learning outcomes and motivation*. Paper presented at the Proceedings of the 9th Workshop in Primary and Secondary Computing Education, Berlin, Germany.
- Ryan, R. M., & Deci, E. L. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology*, 25(1), 54-67.
- Schiefele, U. (1991). Interest, Learning, and Motivation. *Educational Psychologist*, 26(3), 299 - 323.
- Su, A. Y. S., Yang, S. J. H., Hwang, W.-Y., Huang, C. S. J., & Tern, M.-Y. (2014). Investigating the role of computer-supported annotation in problem-solving-based teaching: An empirical study of a Scratch programming pedagogy. *British Journal of Educational Technology*, 45(4), 647-665. doi:10.1111/bjet.12058
- Wang, X., & Zhou, Z. (2011). *The research of situational teaching mode of programming in high school with Scratch*. Paper presented at the 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, Chongqing, China.
- Zavala, L. A., Gallardo, S. C. H., & Garcia-Ruiz, M. Á. (2013). *Designing interactive activities within Scratch 2.0 for improving abilities to identify numerical sequences*. Paper presented at the Proceedings of the 12th International Conference on Interaction Design and Children, New York, New York, USA.