TYPE Original Research
PUBLISHED 15 February 2024
DOI 10.3389/feduc.2024.1291070

OPEN ACCESS

EDITED BY
Russell Tytler,
Deakin University, Australia

REVIEWED BY
Joseph Ferguson,
Deakin University, Australia
Alberto Bellocchi,
Queensland University of Technology,
Australia

*CORRESPONDENCE
Maria Andrée
✉ maria.andree@su.se

RECEIVED 08 September 2023
ACCEPTED 29 January 2024
PUBLISHED 15 February 2024

CITATION
Andrée M, Anderhag P, Björnhammer S and Salomonsson N (2024) Aesthetic experience in technology education – the role of aesthetics for learning in lower secondary school robotic programming.
Front. Educ. 9:1291070.
doi: 10.3389/feduc.2024.1291070

Aesthetic experience in technology education – the role of aesthetics for learning in lower secondary school robotic programming

Maria Andrée[1,2]*, Per Anderhag[1,2,3], Sebastian Björnhammer[1,2,4] and Niklas Salomonsson[2,3]

[1]Department of Teaching and Learning, Stockholm University, Stockholm, Sweden, [2]Stockholm Teaching and Learning Studies, Stockholm, Sweden, [3]City of Stockholm, Stockholm, Sweden, [4]Kunskapsskolan, Stockholm, Sweden

Introduction: Within the technology education research field, aesthetics has primarily been treated as either related to artifacts, design processes and innovation, or as related to students' enjoyment, appreciation, and participation in technology and technology education. This study focuses on the role of aesthetics in technology learning more specifically the learning of programming. Previous research has pointed to aesthetics as important for the learning of programming, e.g., that programming activities in higher education typically involve experiences of frustration. While previous research is primarily based on student reports, there is a need for further exploration of processes of learning to program. The aim of this study is to explore the role of aesthetics for student learning to program in and what these processes may mean in relation to a disciplinary aesthetics of the technology subject.

Methods: The study was part of a design-based study with the overall purpose to develop the teaching of programming in lower secondary school. Data was collected from a programming task designed and implemented in school-year 9 (the students were aged 15–16) in Technology in two lower secondary classes. In total, three teachers participated in the implementation. The students pair-programmed Lego robots that should perform specific movements, such as following a curved line. Each group recorded their coding process along with audio, resulting in videos that documented the gradual evolution of their programs. These videos, capturing the real-time programming and associated student and teacher conversations, serve as the data for this study. In order to analyze the role of aesthetics in classroom conversations a Practical Epistemology Analysis was applied.

Results: The results show that aesthetic judgments were important for orienting learning toward (1) the movement of the robot and (2) the ways to be in the programming activity. During the programming activity, the students expressed feelings of frustration but also joy and humor.

Discussion: The findings concur with previous research and contribute to further understanding the role of negative and positive aesthetic experiences in the teaching and learning of programming. The importance of the objects of aesthetic experience found in this study are discussed as part of a disciplinary aesthetic of programming.

KEYWORDS

aesthetic experience, programming, technology education, robot programming, disciplinary aesthetics

Introduction

This study delves into the significance of aesthetic experiences within the realm of technology education, building upon a line of research on disciplinary aesthetics in education. In this context, disciplinary aesthetics is conceptualized as connected to and emerging in specific school subject practices (Wickman et al., 2021). We draw inspiration from the investigations of aesthetic experiences in science and science learning by Wickman (2006), who demonstrated that aesthetic experiences play an important role in the learning of science (see also, e.g., Jakobson and Wickman, 2008). By examining how people talk about what they do in technology education practices – what they express as interesting-uninteresting, nice-disgusting, cute-ugly and so on – we seek to gain insights into what characterizes a school-subject-specific aesthetics of technology and what role a disciplinary aesthetics may have for teaching and learning technology.

Aesthetics and technology education

Typically two different but interrelated denotations of aesthetics are discerned; aesthetics as (I) a set of design and art practices, and (II) aesthetics as expressions of affect, emotion and taste (Wickman, 2006; see also Prain, 2020). In the realm of technology education, aesthetics has been discussed in relation to both denotations: that is, both as a foundational element in design practices, and as cultivation of appreciation of aesthetical qualities in technological artifacts and technology education, which are in turn intertwined with personal identity and lifestyle (DeVries, 2016).

In previous research on aesthetics in technology education related to practices of design and innovation of technical solutions there are some main lines of reasoning. First, aesthetics is seen as related to the quality of design and disciplinary content knowledge required for analyzing and constructing designs. For example, Haupt and Blignaut (2008) have investigated what aspects of aesthetic design theory that can be taught in technology education, such as visual language and design principles. In this line of reasoning aesthetics is considered a domain-specific construct that students need to develop through explicit teaching. However, aesthetics has also been framed as a critical curricular dimension that goes beyond design and innovation in that it may provide opportunities for students to "step outside of conventional reasoning processes imposed by the rest of the curriculum" Lewis (2005, p. 36). In this line of reasoning aesthetics is framed as complementary to design in engineering processes and as a means to expand the notion of design and problem solving in technology education to encompass the creative potential of design teaching (Lewis, 2009).

DeVries (2016) argues that aesthetics in the sense of appreciation of technological artifacts and qualities, is connected to personal identity and the ways we experience the artifacts surrounding us. Moreover, DeVries (2016) argues that aesthetics plays various roles in different technological domains. Notably, the field of architecture is related to different logic or visions of aesthetics such as modernism, art deco, or brutalism. These aesthetics follow the same rules of logic as for other types of reasoning making it possible to discuss qualities in the architecture related to different aesthetic visions. Likewise, in industrial design, appearance of a product is intertwined with functionality – shapes, colors and so on connected with what the design is supposed to achieve. Consequently, this form of aesthetical reasoning aims at conclusions regarding appreciations of experiences (e.g., Haupt and Blignaut, 2008). There is also previous research on appreciation of (or taste for, *cf.* Anderhag et al., 2015) technology and technology education which is mainly motivated by a recognition of the significant relationship between student interest and learning outcomes (Witherspoon et al., 2018; del Olmo-Muñoz et al., 2022; *cf.*, Potvin and Hasni, 2014). Gender differences in attitudes toward technology have also been a subject of investigation (e.g., Virtanen et al., 2015; Witherspoon et al., 2016; Svenningsson et al., 2018), as well as the implications of these attitudes related to the need of a qualified workforce, and societies need of technological literate citizens (Ardies et al., 2013, 2015; Witherspoon et al., 2016). Most of this research on student attitudes and motivation toward different aspects of technology builds on Likert-type questionnaires (Potvin and Hasni, 2014). An implication of this is that the knowledge to date on the role of aesthetics experiences for student learning technology is largely based on students' recollections of their experiences of technology class rather than of investigations into classroom practices.

Technology education and programming in school

In this study, we zoom into the role of aesthetic experiences in teaching programming as part of technology education and the ways in which aesthetic experiences contribute to student learning.

In contrast to other areas of technology education, such as construction work (building of bridges, towers, cars etc.), the aesthetics is not so much related to design features or artistic expressions of a produced artifact but more with the processes of designing programming solutions. During such processes different disciplinary aesthetics may be constituted, potentially orienting student learning in unexpected or unwanted directions. For example, in a study of programming activities for novice learners in primary education, Sparf et al. (2022) caution against prioritizing aesthetics in the form of artistic expressions in programming as it may overshadow the technical aspects. Sparf et al. (2022) analyzed students' informal conversations during programming lessons at three Swedish science centers. They identified five different student approaches to programming: mathematical, trial and error, step-by-step, routine and aesthetic. Sparf et al. (2022) argue that when students approached programming aesthetically, they tended not to focus on the functional solutions to the assignment and, thus, the students missed out on opportunities to discuss and reflect upon the main purpose of the programming. They argued that although aesthetics may be an important part of programming, a focus on aesthetics as artistic expressions of the product may result in the technical task of programming becoming subordinate. It has thus been demonstrated that student engagement in programming education is both personal and situational including dimensions of cognitive, behavioral and emotional engagement (Sparf, 2021). The findings presented by Sparf et al. (2022) diverge from the insights in a seminal paper on programming education authored by Turkle and Papert (1990). In their work, Turkle and Papert (1990) challenge prevailing assumptions and expectations associated with programming as primarily a rational pursuit. Their study, which was based on observations and interviews with students in primary school and university, reveals diverse student approaches to programming intricately linked to who the students are, think, and how they act as individuals. When observing these students in action, Turkle and Papert (1990) observed a spectrum of approaches, including formal and abstract methods. However, they also noted that the interactions of highly successful programmers with their material "more reminiscent of a painter than a logician" (p. 128). Students who did not conform to the conventional view of programming as a rational-logical process expressed a sense of pressure to transform themselves in order to align with the prevailing programming culture they were a part of.

In von Hausswolff and Weurlander's (2020) study on an introductory programming course in a five- year engineering program at a Swedish university, the students described the process of programming as an emotional roller-coaster. The students experienced both frustration and inadequacy when encountering obstacles, as well as joy and relief when making progress. In particular, the results of the study show how frustration was widely recognized as part of programming practices among the students. Based on the results, von Hausswolff and Weurlander (2020) emphasize the significance of considering the social and emotional dimensions in programming education. A similar argument is put forward by Ford and Parnin (2015) based on an analysis of professional programming practices where they identified situations of frustration when software developers are programming (e.g., in the process of identifying what portion of a code was causing problems). In the analyses of the social dimensions of programming education von Hausswolff and Weurlander (2020) also noticed that aesthetic values were frequently expressed by the students. These results indicate that aesthetics can play a role in students' engagement with technology and that aesthetics form a vital part of programming education in tertiary education. Interestingly, the aesthetic quality of the code was not necessarily consistent with its functionality. Instead, the students primarily emphasized readability and abstraction.

In a systematic literature review on introductory programming Luxton-Reilly et al. (2018) conclude that several studies have explored student engagement in introductory programming encompassing papers focused on time on task, encouragement of self-regulated learning, and the issues surrounding disengagement. Much research focuses on internal characteristics of students and issues of self-regulated learning. Luxton-Reilly et al. (2018) also conclude that affect and emotion is a topic which has received attention (*cf.* a systematic literature review on the role of anxiety when learning to program by Nolan and Bergin, 2016). For example, there are a few studies that have analyzed students' emotional experiences when students work with introductory programming tasks. Based on an interpretative phenomenological analysis of student interviews, Huff and Clement (2017) concluded that experiences of frustration were connected to identity formation, experiences of shame, and maladaptive help. In addition, Kinnunen and Simon (2010) show that the way students talk about their experiences with programming assignments was dominated by emotional expressions. Based on a survey of 388 undergraduate introductory programming students' emotional reactions, Lishinski et al. (2017) also concluded that frustration is the most important emotional reaction and that there is a correlation between students' emotional reactions and their performance on programming projects (*cf.* Martin et al., 2017). Robins et al. (2003) discuss the strains and challenges with learning programming as associated with the abstract nature of programming concepts, the difficulty in understanding algorithms and problem-solving strategies, and the need for students to develop programming skills along with their understanding of programming principles.

In summary, previous research on programming education have emphasized the importance of designing educational programming environments that enhance student engagement, motivation and learning (*cf.* Martin et al., 2017). Existing literature, primarily focused on tertiary-level programming education, indicates that programming activities encompass a spectrum of emotions, ranging from frustration to joy. Students in these contexts frequently express aesthetic values. Notably, at the tertiary level, there seems to be a distinct disciplinary aesthetics of programming education. However, when it comes to programming within compulsory technology education, the knowledge about the role of aesthetic experiences and the ways in which they mediate learning in classroom practices is more limited. Additionally, numerous studies rely on students' self-reported experiences of strains and frustrations. It is not unlikely that aesthetic experiences in compulsory technology classrooms may manifest differently where the students have not chosen programming as a professional career. More generally, the research on aesthetics in technology education points to the importance of fostering appreciation of technology and technology education. An exploration into how such appreciation is cultivated in classroom practices would provide a more nuanced understanding of the role of aesthetics in and in what ways the aesthetics of programming and technology education may be understood as a disciplinary aesthetics of programming.

Aim and research questions

The aim of the study is to explore the role of aesthetics in programming activities as part of technology education. The research questions asked are:

- What are the objects and phenomena aesthetically evaluated when students are programming in technology class?
- What role do aesthetic evaluations have for student learning in programming?

Theoretical framework

As mentioned in the introduction, two types of meanings are usually associated with aesthetics, where one specifically deals with art and design processes, and the other concerns what people experience. The latter meaning, which since Kant (1790/1987) is linked to people's feelings of pleasure and displeasure, and what they find beautiful or ugly, is the focus of the current paper. The study takes a pragmatic perspective on learning, drawing on previous empirical studies grounded in John Dewey's works on aesthetics. We primarily take our starting point in the tradition that stems from Dewey's problematization of aesthetics solely linked to the realm of art, and that emotion, practice, and facts usually has been treated as separated when people learn. Instead, aesthetic experiences are viewed as connected to learning in general and thus having significance for how people understand and act in the world. To Dewey (1997), aesthetic experience goes beyond mere sensory pleasure or the recognition of artistic value. Rather, it entails an emotional engagement with an object, event, or circumstance, resulting in an awareness of perception, response, and intellectual engagement, in which anticipation is consistently present regarding what lies ahead. Therefore, within the process of learning, individuals continually form connections toward completion and closure. According to Dewey, who saw human conduct as a result of an evolutionary process of adapting to the world, aesthetic, and cognition is intertwined and so constituting processes that by necessity, as we in some way or another always are "in the world", is continuous with previous experiences. These processes entail a rhythm of anticipation and potential fulfillment, and the interplay is shaped by actions in which aesthetics both impact and facilitate the rhythm's continuity (Dewey, 1997).

Continuity is closely related to purposes and to Dewey, individuals naturally engage in activities driven by a purpose, objective, or a goal. Through these purposeful activities, people develop habits of coping with the world, so constructing their understanding and learning new things and skills (Rorty, 1991). Purpose and continuity thus intertwine as purpose connects different experiences and activities and when an individual engages in an activity with a purpose, it becomes a part of the broader scope of their experiences (Dewey, 1997). Therefore, interpreting and comprehending aesthetic expressions within a classroom necessitates an understanding of what is going on in the situation where a certain object is aesthetically evaluated (Wickman and Östman, 2002). For example, when students describe worms as "cute" during practical work in science class, it does not necessarily imply a desire to keep worms as pets at home (Jakobson and Wickman, 2008). In this context, the aesthetic object pertains to worms within the science classroom and the purposes transacted by the students (such as daring to hold the worm when examining it), rather than worms in general.

Aesthetic experience and student learning

The role of aesthetic experiences for student learning and especially regarding what may characterize a specific disciplinary aesthetics have recently been studied within the field of science education (e.g., Caiman and Jakobson, 2022; Ferguson et al., 2022; Hannigan et al., 2022). Common to these studies are their interest in the specific purposes, processes, and objects that students and teachers discern as interesting, beautiful, ugly, disgusting, etc., and the significance these distinctions have for the meanings about the subject (and themselves) that are constituted in the classroom. A person's aesthetic experience is evident through the verbal aesthetic judgments s/he makes, and such judgments have been shown to orient student talk and action toward (or away from) the purposes of the classroom activity (Wickman, 2006). Aesthetic judgments thus have a key role for what route learning take in the classroom, irrespectively whether it is students learning about marine animals and insects at the university level (Wickman, 2006), electrical circuits in secondary school (Anderhag et al., 2015), earthworms (Jakobson and Wickman, 2008), or ecology (Caiman and Jakobson, 2022) in primary school.

Being expressions of preferences of like/dislike, or perceptions of beauty/ugliness, aesthetic judgments are directed toward something, such as for example that a code is described as 'nice' by students in the technology classroom. In so doing they simultaneously express something about their feelings while labeling the subjective qualities of the object (Wickman, 2006). These qualities, however, are continuously negotiated as part of the social situation in which they are transacted, potentially leading to agreement among participants ("yes, it is a nice code"). In this way, the participants may (or may not) jointly construct a shared understanding of what and how objects are aesthetically valued in the technology classroom. For example, a nice code may be easy to read, short, or having certain functional segments. Besides being closely connected with the learning of the procedures and facts of a specific school subject (such as what characterizes a nice code and how to produce such a code), aesthetic experience is however also closely connected to different ways-of-being in the classroom. For example, who is smart or nice or who can distinguish what is smart or nice (such as a nice code). Such distinction of taste is central to how people navigate socially, distinguishing themselves and others as belonging or not. Developing an understanding of the disciplinary aesthetic is thus important for students' opportunity to learn cognitively about the subject but also to successfully participate in settings where the subject is at stake (Anderhag et al., 2015). As the disciplinary aesthetic can hinder or enable students' socialization and identification as participants, teaching is of great importance for supporting students in negotiating aesthetic objects within the classroom culture. The immediacy of communication is facilitated by aesthetic judgment, especially when a student lacks familiarity with certain cultural concepts (Wickman, 2006). This sometimes leads to the use of more practice-specific content of communication and sometimes functioning as a natural part of the practice (*cf.* Knorr-Cetina, 1999).

In this study we focus on the students' verbal aesthetic judgments during a programming lesson in technology class, with an awareness that this constitutes a specific aspect of the broader communication taking place within the classroom. We analyze these judgments in relation to the aesthetic objects to which they are directed, with the intention of gaining a deeper understanding of the role of aesthetics in students' learning of programming within the technology classroom.

Study design

The present study is part of a collaborative project between teachers and researchers with the overarching aim of developing teaching in programming within the context of the Swedish primary school's technology subject.

Programming in Swedish compulsory school technology education

In Swedish compulsory school, technology education is taught throughout compulsory school as a subject aiming to foster technological literacy. The Swedish school subject Technology (*Teknik*) aims at providing students with opportunities to develop an understanding of the role of technology in society; to develop technical awareness and vocabulary. As a school subject Teknik represents technical knowledge traditions, which are different from the knowledge traditions of science, and related to the specific contexts and practices in which the technical and technology knowledge becomes meaningful (Björkholm et al., 2016). In particular, reflection, analysis and construction of technical solutions is emphasized

(Swedish National Agency of Education, 2022). In the Swedish technology syllabus, programming is part of the core content *methods for developing technological solutions* and in years 1–3 (age 7–9) the students are supposed to learn to control objects, such as a robot, using programming. In years 4–6 (age 10–12) the students should learn to control their own constructions or other objects by using programming, and in years 7–9 (age 13–15), the students are supposed to use programming for controlling and regulating their own constructions. Programming is thus primarily a tool for controlling objects and a progression in terms of knowing how to program is not formulated in the technology syllabus.

Programming was a relatively new content in the technology curriculum for Swedish compulsory school when the project started, and there was generally a lack of experience both among teachers and educational researchers in Sweden regarding what characterizes programming education within the scope of the technology subject (Vinnervik, 2022). What content, concepts, procedures should be emphasized, and how can it be connected to the broader objectives of the technology subject? Within the project, there was, therefore, a shared interest in developing teaching in programming.

Study setting

The study was conducted within the research environment Stockholm Teaching & Learning Studies (STLS) (Andrée and Eriksson, 2019). Within STLS, researchers and teachers collaborate in designing and conducting small research and development projects that address challenges and questions that have been identified in the teaching of different school subjects. The teachers' participation (10% of full-time-employment) is funded by their respective school authorities and the activities, lessons, and series of lessons that are jointly developed are tested in the participating teachers' classrooms. As part of participating in STLS the teachers implemented lesson designs, which had collectively been developed, with their students in their respective classrooms. The sample of the participating students in the study thus depends on the teachers' participation in STLS.

The collected data comes from two lower secondary technology classrooms (School A and B) in Stockholm, Sweden. Niklasson was the teacher of one of the student groups participating. The students (year 9, ages 15–16) were working with a task of programming Lego robots that should perform specified movements, such as following a curved line. The lessons were designed so that students were given the opportunity to develop skills and understanding specified in the curriculum of the technology subject. In school A this task was part of a broader unit that we had jointly designed that centered around sustainability and technology. During this unit, both the teacher and students collaborated to discuss and create a model of an automated recycling station. Conversely, in School B, students solely focused on the curved line task. Throughout the lessons, the teachers actively moved around the student groups, assessing their programming progress. They interacted with the students by posing and answering questions, and also encouraged discussions that allowed students to articulate their programming intentions. The students utilized the Lego Spike package on their iPads while engaging in pair programming their robots.

The study adheres to the ethical guidelines of the Swedish Research Council (2017). The students and their guardians of the participating classes were provided information about the purpose and the design of the project. The participating students and their guardians all signed a letter of consent. The collected data of the study is handled according to the General Data Protection Regulation (GDPR).

Data collection

The lesson was part of the regular teaching and aimed to support the students' understanding of programming. The organization of the lesson followed the teachers' usual approach when programming with the students. The teachers divided the students into groups of two or three, where each group had access to an iPad and a robot. The teachers explained the aim of the lesson and asked them to start screen recording when they started with the tasks.

Each student group screen recorded their coding process along with audio, resulting in videos on their Ipads that documented the gradual evolution of their programs. A total of 7 screen-recorded videos, with 4 from School A and 3 from School B, were transcribed verbatim. The length of these videos varied from 30 to 60 min. These videos, capturing the real-time programming and associated student conversations, serve as the data for this study.

The data was initially analyzed in order to explore what strategies the students use when they were programming a robot to perform a specific movement, which we have reported in a previous study (Anderhag et al., 2023). During this process we noted that the students recurrently used aesthetic judgments while they were programming. The presence of aesthetic judgments were not something we had expected or planned for when we designed the lessons together with the teachers, nor was it something that the teachers consciously acted upon or considered when they were teaching the students. Thus, the focus on aesthetics of the present study emerged through the processes of designing, implementing and reflecting upon programming classroom practice rather than as a consequence of planning for it. In the current study, we use the same data as in the previous study but with the analytical focus on aesthetic judgments used by the participants.

Data analysis

The transcribed films were initially analyzed to identify aesthetic situations, primarily evident when students verbally made aesthetic judgments while they were programming. Aesthetic judgments are evident as the students and teachers make judgments of inclusion and exclusion on language use (language in a multi-representational sense), procedures (ways-to-act), and ways to be. We then conducted a categorization of the types of objects and phenomena toward which students directed their aesthetic evaluations. The categories were developed, integrated and reorganized until agreement was achieved between the co-authors (*cf.* Glaser and Strauss, 1967). Potentially, objects and phenomena could be physical items, such as a robot, iPad, or a piece of (digital) code, but they can also encompass actions (e.g., ways of programming) and concepts (e.g., loops). We deliberately chose to use this rather inclusive heuristic to encompass a wide range of situations. These situations were then analyzed using Practical Epistemology Analysis (PEA) (Wickman and Östman, 2002). PEA is grounded in a pragmatic perspective where learning is operationalized as discourse changes as part of an activity (Kelly et al., 2012). We primarily used three of the analytical concepts of PEA, *stand fast*, *gap* and *relation*, to identify the role of aesthetics for student learning. What stands fast in a situation are things, phenomena, actions, words that the interlocutors do not question in talk or action. As an activity proceeds, such as when students are programming, gaps emerge as a result of encounters between persons and artifacts (*what does this code do?*) or phenomena (*why did the robot stop moving?*). In order for the activity to proceed, the gap needs to be filled with a relation to what stands fast, that is things that are intelligible in the situation.

The following transcript exemplifies the concepts further. The students had programmed their robot to stop at a red line after it had followed a curved line. They had debugged their code and in turn 1 they decided to run the program to see if adjustments worked as intended. In turns 3–5 a gap is noticed, that is, the robot did not stop when it reached the red line. In order to proceed with the activity, the gap "how should we stop the robot at red?" needed to be filled with a relation to what stands fast. Several things in this situation stand fast, such as for example they do not question or discuss that the robot should stop at red, or that something is wrong with the code. The gap is filled in turns 6–7 as S2 noted that they had forgotten to insert a stop command after the sensor command (wait until sensor B registered red). A relation is established "how should we stop the robot at red – by inserting a stop command."

1. S1: We can try just first.
2. [testing the new code].
3. S2: Oops! [The robot crosses the red line]
4. S1/S2: [laughter].
5. S2: It's supposed to stop?
6. S2: When B. When the color is B. Damn, we need to stop there, we forgot [giggle]. Damn. It's fighting! [the robot keeps moving].
7. S1: Yes, stop.
8. S2: Should I try now?
9. [testing].
10. S1: Now it might be good
11. E2: Yes.
12. S1: Here it comes [the robot moving along the line].
13. S2: Damn [giggles], what, what movements it makes!
14. S1: Perfect movement.
15. S2: Check it out! Check it out! Stop then! Like that! Yeeeees.
16. S1: Yeeeees.
17. S2: Yeeeees.
18. S1: And then, then we must have, then we make a new one like this. Sensor A.

The transcript exemplifies the flow between anticipation and fulfillment of an aesthetic experience. After the students had inserted the stop command and executed the program again, the changes were aesthetically evaluated as potentially good, so anticipating that the robot would now move as intended. While observing they made positive consummatory judgments on its movement and the situation is summed up in turns 16–17, as they happily concluded that the changes to the code had resulted in a stop at red preceded by *perfect* movements. In turn 18, they started with the next task. In this example, hence, the 'movement of the robot' was an *object* of students' aesthetic evaluations.

In summary, the first step of the analysis aimed to generate results to answer research question 1. That is, what is evaluated aesthetically when students are programming. We were thus interested in what objects and phenomena the students distinguished. In the next step, PEA was used to gain a deeper understanding of what was extracted in step 1, potentially producing results on what identified objects and phenomena may mean in relation to learning to program. In other words, what consequences do these aesthetic objects and phenomena have for the meanings students construct while programming. This step aimed to address research question two.

Results

Throughout the programming activities aesthetic judgments were used by the students and their teachers for evaluating actions conducive to the purposes of the activity, that is, programming the robot to move in a specific way. Such actions were primarily dealing with the emerging code and how it could be altered. Although rarely articulated explicitly by the participants, the students were thus engaged in an activity where they were developing, testing, and evaluating a technical system where the construction (the code) was adjusted to improve its functionality (movement). In the programming activity, primarily two kinds of aesthetic objects were evaluated: *The robot movement* and *Ways to be a programmer*.

In what follows, we will present our findings in relation to the two research questions of the study, we will do so by presenting the two aesthetic objects identified under separate headings. The second research question, what role aesthetic evaluations had for student learning, will be handled throughout the paper and summarized at the end of the section.

The robot movement as an aesthetic object

Disgust and frustration when the robot takes an unwanted direction

Although the students understood the task to program the robot to make it follow a curved line, they were not sure how to use the example code to make the robot move in alignment with the purpose. The uncertainty about how to proceed resulted in frustration among the students and the analysis revealed several instances where they were using negative aesthetic judgments while they were programming their robots. These judgments were directed toward different aspects of the programming activity, primarily toward the unexpected and unwanted ways the robots moved but also for example the losing of Bluetooth connection between the Ipads and the robots or the depletion of the batteries of the robots. Negative aesthetic judgments were thus used in situations where the activity stopped or took an unwanted direction. In Example 1 a student expresses frustration as the wheels of the robot did not move as expected.

Example 1: *This was incredibly disgusting*

1. Student: This was incredibly disgusting. Because, you see, the problem before was that they were driving backwards, and now they are driving forward. When you turn. When you turn them

around, then it starts moving forward. But then, they do not go at the same time. Which is quite strange. I mean.

The student felt disgusted by the fact that the wheels moved in unexpected ways and the irregularity of the movement of the robot was judged as quite strange. In the example the situation was evaluated as disgusting; sometimes the students' frustration was directed explicitly toward the robot– the performer of the unexpected movement. The next example illustrates a similar situation: the students had been working with their program for 40 min when the robot unexpectedly no longer stopped at the red line.

Example 2: *I could easily kill you*

19. S1: What happened now?
20. S2: I'm going to start crying soon.
21. S1: It missed the red one?
22. S2: I'm going to start crying soon. I'm going to start crying soon.
23. S1: [Laughs].
24. S2: Do you [the robot] want to die? Because I can easily kill you.
25. S1: Are you threatening it [giggles]?
26. S2: It's just a matter of breaking you apart. Do you know how easy that is? I have the power here, not you.

In turn 1 S1 noticed a gap, the robot did not stop at red. The students did not expect this to happen as they had inserted, tested, and debugged a code that previously had made the robot stop at the red line. S2, being frustrated, said that she is about to start crying (turn 2, 4). Later S2, still frustrated, humorously asked the robot if it wanted to die and that she could easily kill it and take it apart (turns 7, 9). The joke was well perceived by S1 who giggled as she questioned S2 threatening the robot (turn 8). The two examples above showed how students were engaged in the programming activity and express frustration but also humor as their actions did not result in the anticipated outcomes. If the students did not establish relations that filled the respective gaps (they did) the only possible scenario would be that the activity ended. That is, they could not continue with the task. Although jokingly, S2 seemed to consider the possibility of putting an end to the activity by simply breaking the robot.

Celebrating desired robot movements

Examples 1 and 2 showed instances where students evaluated situations and artifacts that did not comply with their expectations of what route the activity should take. The teachers were attentive to situations where students faced challenges, and provided support and guidance to help students navigate their way forward. The next example showed such a situation. The students had been working for some time coding and debugging but still not yet been able to program the robot to perform the task. In turn 1, the teacher asked how things were going:

Example 3: *I feel like I do not want to continue, honestly*

1. Teacher: How are things going for you?
2. S1: Well, uh... It's not going well at all. It's going insanely bad today. I feel like I do not want to continue [in Swedish *tappar lusten*], honestly.
3. Teacher: Well, we... we'll spice it up [in Swedish *lustar upp det*]. We'll fix it.
4. Teacher: Okay, press that... uh...um. No, let us do it this way. I'll give you some stuff.
5. Teacher: Um... [introduces a new variable "lowPower" and sets it to 20].
6. Teacher: Then it stops all the other blocks, this one and also stops the wheels.
7. S1: Yeah, that's awesome.
8. Teacher: So, now you have that, and then you can start figuring out how to make it drive around in here.
9. S1: Yeah. That's great. So now this should work elegantly.

Things were going "insanely bad" and the student was about to lose his will to continue. In turn 3, the teacher told the students that they together would "spice it up" and that they would "fix it." He then made some changes to the code and inserted a new variable, while doing that he described what the changes would do for the movement of the robot. These changes, and the anticipated outcome of these procedural alterations, were distinguished and aesthetically evaluated by student 1: "Yeah, that's *awesome*" (turn 7), "Yes. That's *great*. So now this should work quite *elegantly*" (turn 9). In the example, positive anticipatory aesthetic judgments were thus used for evaluating procedural distinctions of inclusion that the student felt would lead the activity toward the preferred outcome. Such examples were of course not only connected to previous situations of frustration, as shown in Examples 1–3, but were used throughout the activity for evaluating how procedures were conducive to anticipated outcome. Examples 4 and 5 exemplifies such instances. In Example 4, the students had tested the code and wanted to fine tune the movement of the robot.

Example 4: *It's kind of nicest*

1. S3: No.
2. S2: Yes!
3. S2: It's kind of the nicest.
4. S3: [laughter].
5. S2: Wait, let us try exactly like that so it ends up like that again...
6. S3: If we manage to get it like that again...
7. S1: But, I mean, fifty-five cents, it goes fifty-five centimeters, that's maybe a little bit too little, a lot.
8. S2: No, so, now we go again.

In turns 1–2 students S2 and S3 commented on the robot after they had made some small changes to the code. S2 then made a distinction, "It's kind of *nicest*," making a consummatory evaluation of the movement they just had observed. As evident from the following turns, "nicest" stands fast in the situation and the students were eager to have the robot repeat the (nice) movement and discussed the possibility of making some minor adjustments to the code (turns 5–8). In example 5, below S2 describes to the teacher what he wants to accomplish with the code he is working with, namely having the robot make a continuous turn without stopping first.

Example 5: *No, you want a neat turn*

1. S2: So that's what I need help with. How can it turn like that? Because right now, it just stops and then turns sixty degrees immediately.

2. Teacher: Exactly. In that case, you need to rotate both wheels, maybe like this. Either you rotate sixty degrees, and what you are actually saying is that one wheel rotates while the other stays still.
3. S2: Yes. But actually, I do not want it to drive straight and then turn.
4. Teacher: No, you want a neat turn.
5. S2: Yes.
6. Teacher: Yes, and that's the thing with wheels, it's this...
7. S2: ...needs to move slower.
8. Teacher: Needs to roll further than the other.
9. S2: Yes.
10. Teacher: Exactly. So that it continues rolling forward but also starts to turn slightly.
11. S2: Yes.
12. Teacher: So you need to divide it.
13. S2: Does it have to do with power again?
14. Teacher: Yes, exactly. And then, you simply assign different percentage values to them.
15. S2: I see.
16. Teacher: So that one goes a little faster than the other. If you think about different radii, then one must be much faster than the other.
17. S2: Yes.
18. Teacher: To make a neat turn.
19. S2: Yes.

In turn 1 a gap was addressed by S2: how do I make the robot make a turn without stopping? A relation was put forward as the teacher suggested that the student should alter the movement of the wheels for making the turn. These suggestions did not address what the student wanted the robot to do and in turn 3 he clarified that he does not want the robot to move straight and then make a turn (describing a L-shaped movement). The teacher responded by saying "No, you want a *neat* turn" so making a distinction of exclusion regarding a sharp robot movement. In the situation "neat" stood fast and in the following turns the teacher and the student discussed how the wheels should move in order to perform an arch-shaped, and so neat, turn without stopping first.

Also, the code, rather than the robot or its movements, could be evaluated aesthetically. Usually, it was the teachers who made such judgments. In one instance the teacher asked a student group "Could you make this [a part of the program controlling the wheels] *nice* by adding another variable?". The teacher thus addressed a normative aspect of the code; it could be more or less nice and there were ways to produce a nicer looking code. In this particular situation, the students created a new variable controlling the power of the wheels that they named "Hjulben." The choice of name was a sort of playful joke, in Swedish "Hjulben" (literally translating to wheel legs) means bow-legs.

Finally, consummatory aesthetic judgments were also used in situations of fulfillment when processes came to an end. The next example illustrates how the teacher evaluated how one student group had accomplished the task of programming the robot to perform a specific movement. This is the same group that were joking about killing the robot and before Example 6 they had tried and had made several changes to the code in order to have the robot move as intended. The robot was following a curved line by using light sensors, stopped at a red line where it picked up a Lego-brick, and finally backed up and stopped when its sensor registered a green line. In turns 1–5 the teacher and the two students were observing and commenting on the movement of the robot.

Example 6: *It's so beautiful. It's so beautiful. Fantastic!*
1. S1: Ah!
2. Teacher: Ah! Now! There!
3. S1: Nooo!
4. Teacher: Well, well, but still...
5. S1: But now it will not find... (inaudible)... green.
6. Teacher: Fantastic. No, exactly, let us redo it.
7. S1/S2: Mm.
8. Teacher: Maybe we should film it because that was brilliant.
9. [They run the program].
10. Teacher: It's so beautiful. It's so beautiful. Fantastic.
11. S1: But then, it does not touch the green.
12. S2: No, that's it. We need a different command then.
13. S1: Yes.
14. S2: Otherwise, it works.
15. Teacher: But that was it, now it's adjusted, now it will work.
16. S2: Yes.

The teacher and students discussed the movements of the robot and evaluated how well it performed what it was supposed to do. Due to a slight deviation of the movement of the robot, its sensor missed to register the green line, which the students noted (turns 5 and 11). At the same time the teacher summarized their work as they had managed to program the robot to perform the first, rather complex, steps of the task. The movement, and indirectly the code, was evaluated as *fantastic*, *brilliant*, and *beautiful*.

Ways to be a programmer as an aesthetic object

Celebrating oneself as a programmer

The students did not only distinguish and aesthetically evaluate conducive or preferred procedures when programming their robots, they also made distinctions on ways to be in the classroom. For example, they referred to each other as "smart" or as being "Mr technician", thus distinguishing themselves and/or their classmates as persons who know how to solve a programming problem. The following example showed a situation where the students succeeded in programming the robot to perform the wanted movement. The students had worked for a long time altering and debugging their code when they in turn 2 ran the program.

Example 7: *What a freaking genius!*
1. S1: Go. Try it.
2. [the students test their program].
3. Both: Aaaaaah!
4. S2: Who's as smart as me!
5. S1: It works! Oh my god! [clapping her hands].
6. S2: Who's as smart as me!
7. S1: [Name of S2], what a freaking genius!

As with the previous examples, the aesthetic judgments are expressed in evaluating a situation that came to fulfillment – the robot did as anticipated. Here, however, it was not only the movement of the robot that was evaluated but also the person who programmed the robot. In turn the students screamed in joy as they saw the robot do as expected, in turns 20 and 22 Student 2 made a distinction on herself, saying *Who's smart as me?* That Student 2 is smart was also supported by Student 1, saying in turn *What a freaking genius!*.

It's okay to feel that programming is difficult

As described earlier, the teachers were very careful in supporting student reasoning on how the problems, i.e., the movement of the robots, could be addressed through changes in the code. In such instances they regularly acknowledged the student contributions and clarified how it tallied with what they wanted to accomplish. That the tasks were complex and that potential problems were connected to this complexity, rather than the students' understanding or competence, were made explicit by the teachers throughout the activity. In example 8, the teacher and the student, who had been debugging her code for some time, cheerfully agreed that programming was tough and it was okay to hate it.

> Example 8: *I hate programming, but it's fun.*
> 1. S1: It's not working, I hate programming [laughter].
> 2. Teacher: Me too, but it's still fun [laughter], it's really tough.
> 3. S1: I hate programming, but it's fun.
> 4. Teacher: It's fantastic when it works.
> 5. S1: Yes.
> 6. Teacher: But it's really tough when it does not work.
> 7. S1: Yes [happily]!

Programming could thus be a tough activity and several aspects of it could be difficult to grasp, which the teachers continuously acknowledged. During the lesson, the teacher had explained to the whole class how variables worked and how they could be used. In Example 9 below, the teacher made the students in one of the groups aware of the variable "Power" after having discussed how they could program the robot to follow a curved line by using its light sensors.

> Example 9: *Did it feel completely intuitive or was it very strange to understand it?*
> 1. Teacher: Do you see that there's something called "Power" here? What is it?
> 2. S1: Yes, then it can copy the actual speed to the next one.
> 3. Teacher: Ah, well said. This is called a variable. Did you feel like it was completely intuitive or did you find it very strange to grasp?
> 4. S1: No, I understood it quite quickly.
> 5. Teacher: Great. It's just like X or Y in math.

Learning to program and potentially also to develop a sense of belonging in the practice of programming, thus entails using certain procedures and objects, such as variables. In this process, the function of parts of the code could seem strange or difficult to understand and that was okay. Making students aware of how their doings comply with purposes are likely to be important for how they perceive themselves as participants in the programming activity. For example, as exemplified in the previous example of the teacher making a distinction of inclusion on the student explanation of the variable Power. Another example of how the teachers made students aware of how their doings adhered to the practice of programming was when a student with the support of the teacher reasoned her way to how they should proceed. In Example 10 a student describes her problem out loud to herself and then found the solution on her own, which, according to the teacher, is something that programmers often do.

> Example 10: *This is a well-known phenomenon in programming, it's called rubber ducking*
> 1. S1: I understood, thank you very much.
> 2. Teacher: Do you know what you just did? This is a well-known phenomenon in programming, it's called rubber ducking. Let me explain. Many programmers have a small plastic duck on their programming desk.
> 3. S1: Okay.
> 4. Teacher: And when they have a problem that they do not understand how to solve.
> 5. S1: Mm.
> 6. Teacher: They explain the problem to their plastic duck, and just by explaining the problem, they often find the solution.
> 7. S2: Exactly. Okay, but then you'll [S1] have to explain it to me.

In the data, there were few situations when the students explicitly made distinctions on ways to be in terms of exclusion, that is, saying that they could not program or that they are not a programming or technology person. However, there were instances where they joked or used irony in humorous ways to distinguish themselves or their classmates as not being great in what they were doing. For example, by jokingly saying that they would get an A in technology when they got stuck or executed a code that did not result in the movement they had expected.

Summary of results

Both students and teachers used aesthetic judgments to assess (1) the movement of the robot and (2) ways to be in the programming activity. In such situations, the interlocutors evaluated whether anticipated or observed procedures tallied with the purpose of the activities, that is, having the robot perform specific movements. Even if judgments only rarely explicitly distinguished aspects of the code, such as being a nice or ugly code, the observed and evaluated function oriented the students' exploration and adjustment of the evolving code. For example, in the case of having the robot perform a neat turn, the teacher and the student first discussed what parts of the robot that should be altered for making a neat turn (different power on the wheels), and after that they discussed what such an alteration amounted to in code. Aesthetics was thus shown to have an important role in orienting student learning, with some exceptions all student groups succeeded in producing code that made the robot perform the wanted movement. Moreover, the analysis showed that students expressed feelings of frustration during the programming activity. If students recurrently experience technology class or programming as a practice where their actions rarely or in arbitrary ways lead to the expected outcome and feelings of fulfillment, it is likely that they will develop negative attitudes toward the subject. As with distinctions in procedures, distinctions on ways to be are likely to be of importance for how the individual views him/herself as competent in the technology classroom. It is thus likely that a student who continuously is distinguished as not belonging, either by her/himself or others, is likely to turn away from the subject. However, the teachers were very attentive to situations where the students got stuck and, in such instances, they (1) acknowledged the difficulties of the tasks, (2) scaffolded them in reasoning how to proceed, and (3) made them aware of how specific code segments could be used to solve encountered problems.

Discussion

Within the technology education research field, aesthetics has primarily been treated as either related to artifacts, design processes, and innovation (e.g., Haupt and Blignaut, 2008; DeVries, 2016), or as related to students' enjoyment, appreciation, and participation in technology and technology education (e.g., Potvin and Hasni, 2014). Within the research specifically investigating the learning of programming, affect and emotion are generally noted as important for student learning (Kinnunen and Simon, 2010; Luxton-Reilly et al., 2018). Our study can be said to traverse these three areas as it has sought to explore what objects and phenomena the participants appreciate (or not) and what consequences such aesthetic experiences have for learning to program in technology class. In line with previous studies on disciplinary aesthetics and taste in various school subjects, learning technology does not solely mean to learn what is the case in terms of concepts, facts, and phenomena but also to learn what is and what is not valued, enjoyed, and appreciated in the technology classroom. Consequently, learning a disciplinary aesthetic or a taste also means learning who you are or could be in relation to the subject.

Aesthetic judgments as a means to orient student learning in the context of technology education practice

So, how are our findings related to previous research on aesthetics and learning technology and especially learning to program in technology class? The study contributes to the notion of aesthetics being a central element in the understanding, design, and evaluation of technological artifacts in that it presents empirical evidence on how such processes may unfold in classroom action. In line with the work of Wickman (2006) and Jakobson and Wickman (2008), this study has demonstrated the role of aesthetic judgments for orienting learning processes toward the purposes of the activities. In these processes, especially the movement of the robot and the student as programmer were seen to be the focus of the distinctions made, possibly constituting important aspects of what Hannigan et al. (2022) refer to as experiential, subject-based aesthetics which "entails participants' feelings in engaging with the purposes, objects, instruments and inquiry strategies of a subject" (p. 798).

The aesthetics did thus not primarily revolve around the technological artifact that the students were designing (*cf.* DeVries, 2016), namely the code, but rather its functionality as it was expressed through the movement of the robot. Even if it is not surprising that the students aesthetically evaluated the outcome of the code rather than the code itself, the finding is nevertheless interesting in relation to learning disciplinary aesthetics as it may exemplify an everyday taste of the novice programmer and that there may be different aesthetics at stake in the activity (*cf.* Hannigan et al., 2022). There are ample examples from other subject disciplines demonstrating what the experienced connoisseur finds aesthetically pleasing (or not), such as; what is nice in an experimental setup in science (Wickman, 2006), what is interesting in a certain way of presenting data in math class (Ferguson et al., 2022), or what beauty there is in mathematical inquiry (Sinclair, 2006). We can only speculate, and more research is needed, but it is likely that the more experienced programmer successively develops an appreciation of certain aesthetic qualities of the code that to the novice may seem elusive (as in this study). In the study of von Hausswolff and Weurlander (2020) at the university level, for example, a good-looking code amounted to a shared, although implicit, understanding of readability and level of abstraction. Also, the teachers in our study occasionally made distinctions on the code, making the students aware of aesthetically pleasing ways of adjusting their program that would better accomplish what it was supposed to do. The findings thus imply possible instances where positive (or negative) aesthetic experiences may be made continuous between on the one hand the outcome of the program (movement/function), and on the other hand the program itself (code/construction). Again, these aspects need to be studied further but it is likely that such scaffolding is important for students' opportunity to develop a disciplinary aesthetics that is current and recognized also in other programming settings.

Frustrations as opportunities for productive struggle

The programming tasks in this study were a kind of semi-open inquiry in that it had a clear goal - programming the robot to perform a specific movement - while at the same time allowing for various alternative ways of reaching this goal. It turned out that some of these open tasks were difficult for many of the students, and they dedicated a lot of time to writing, testing, and modifying the code required to accomplish the specific movement. This process resulted in a great deal of frustration, a well-known and widely recognized feeling within the field of programming (Lishinski et al., 2017).

If students repeatedly perceive programming as an activity where their efforts seldom result in the intended outcome and feelings of fulfillment, it is probable that they will develop negative attitudes toward the subject. Negative aesthetic experiences do however not necessarily have to be problematic; the important thing is that the students want to take part in the activity and that negative aesthetic experiences are handled and transformed in the long run (Wickman, 2006). The study of Björnhammer et al. (2023) demonstrated how aesthetic experience shifted between positive peaks and negative lows during an inquiry activity in science, where one student group commenced with resignation but ultimately solidified their commitment, while another group, by way of comparison, embarked with excitement but eventually found themselves mired in frustration. One might posit that tasks, such as the programming activity in this study, could be structured to minimize the occurrence of negative turns, however, Björnhammer et al. (2023) cogently argue that even though reducing the degree of freedom within an activity might limit the risk of failure, it would also inevitably transform an open inquiry into a more constrained activity. Hence a precarious balance exists, determining when modifications would render an entirely different activity. However, our conclusion is not to exclude or protect the students from negative aesthetic experiences.

The teachers in our study were very attentive to situations when the students got stuck and, in those situations, the teachers acknowledged the difficulty of the tasks, and affirmed that it was okay to lose motivation and not know what to do. Through scaffolding student reasoning, they usually conclude together how to proceed purposefully with the activity and with some exception all student groups succeeded in bringing processes to closure. These strategies, to acknowledge difficulties of tasks and encouraging and support students' reasoning on productive ways of moving through difficulties and toward purposes have previously been suggested to be important characteristics of practices where taste may develop (Anderhag et al., 2015). We may see this as means to creating spaces and opportunities for students to engage in a 'productive struggle' (*cf.* Warshauer et al., 2021) with programming.

Distinguishing oneself as a competent participant

In addition, the students also aesthetically evaluated themselves as participants in the programming activity. As with distinctions on procedures, also distinctions on ways to be are important for how the students view themselves as competent in the technology classroom and it has been argued that feelings of frustration while programming is intimately connected to identity formation (Huff and Clement, 2017). Although not specifically exploring emotions or aesthetics, Turkle and Papert (1990) demonstrated that students who approach programming in 'artistic' and unconventional ways, may perceive that their ways of being do not comply with the norms and values reproduced in the programming courses. Turkle and Papert (1990) therefore argue for 'epistemological pluralism', allowing for different ways of being as a programming student. It is thus likely that a student who is continuously distinguished as not belonging, either by her/himself or by others, is likely to turn away from the subject. This was however not the case in the programming activity studied here, besides playful jokes that were well-perceived, the students did not make negative aesthetic judgments regarding themselves or their classmates. Although this is likely to be of importance for how the students perceive themselves in relation to the technology subject and the disciplinary aesthetic emerging, we cannot say whether this was a result of the activity, the teaching, or reflecting a generally positive classroom environment. Whatever the cause was for the good-humored persistence of the participants, we can conclude that becoming a programmer can be a hard and frustrating journey where moments of challenge and adversity may, eventually, provide an important contrast that magnifies the sense of consummation and joy when students finally overcome obstacles. In such situations of fulfillment where normative and cognitive relationship are summed up into aesthetical wholes, the novice programmer may actually turn out to be "a freaking genius!"

Data availability statement

The datasets presented in this article are not readily available although the principle of public access to information applies since, for the processing of personal data in research, there are special requirements. Requests to access the datasets should be directed to dso@su.se.

Ethics statement

Ethical approval was not required for the study involving human samples in accordance with the local legislation and institutional requirements because the collected data do not include sensitive personal data. Written informed consent for participation in this study was provided by the students' legal guardians. Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

Author contributions

MA: Writing – original draft, Writing – review & editing. PA: Writing – original draft, Writing – review & editing. SB: Writing – original draft, Writing – review & editing. NS: Investigation, Resources, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This research was funded as part of Stockholm Teaching & Learning Studies, a joint initiative from Stockholm University and the education administrations of the municipalities of Botkyrka, Nacka, Stockholm, and Södertälje, and the free-standing school owners Kunskapsskolan and Fryshuset.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Anderhag, P., Hamza, K., and Wickman, P.-O. (2015). What can a teacher do to support students' interest in science? A study of the constitution of taste in a science classroom. *Res. Sci. Educ.* 45, 749–784. doi: 10.1007/s11165-014-9448-4

Anderhag, P., Salomonsson, N., Bürgers, A., Estay Espinola, C., Fahrman, B., Seifeddine Ehdwall, D., et al. (2023). What strategies do students use when they are programming a robot to follow a curved line? *Int. J. Technol. Des. Educ.*, 1–20. doi: 10.1007/s10798-023-09841-x [Online ahead of print].

Andrée, M., and Eriksson, I. (2019). A research environment for teacher-driven research–some demands and possibilities. *Int. J. Less. Learn. Stud.s* 9, 67–77. doi: 10.1108/IJLLS-02-2019-0015

Ardies, J., De Maeyer, S., and Gijbels, D. (2013). Reconstructing the pupils attitude towards technology-survey. *Design Technol. Educ. Int. J.* 18, 8–19.

Ardies, J., De Maeyer, S., Gijbels, D., and van Keulen, H. (2015). Students' attitudes towards technology. *Int. J. Technol. Des. Educ.* 25, 43–65. doi: 10.1007/s10798-014-9268-x

Björkholm, E., Andrée, M., and Carlgren, I. (2016). Exploring technical knowledge in the primary technology classroom. Australasian. *J. Technol. Educ.* 3, 1–16. doi: 10.15663/ajte.v3i1.23

Björnhammer, S., Lundegård, I., and Gyllenpalm, J. (2023). Students' meaning making in an open inquiry - two paths. [Manuscript submitted for publication]. Department of Teaching and Learning, Stockholm University.

Caiman, C., and Jakobson, B. (2022). Aesthetic experience and imagination in early elementary school science – a growth of 'science-art-language-game. *Int. J. Sci. Educ.* 44, 833–853. doi: 10.1080/09500693.2021.1976435

del Olmo-Muñoz, J., Cózar-Gutiérrez, R., and González-Calero, J. A. (2022). Promoting second graders' attitudes towards technology through computational thinking instruction. *Int. J. Technol. Design Educ.* 32, 2019–2037. doi: 10.1007/s10798-021-09679-1

DeVries, M.J. (2016). *Teaching about technology an introduction to the philosophy of Technology for non-philosophers*. (*2nd Edn.*). New York: Springer.

Dewey, J. (1997). *Experience and education*. (New ed.). New York: Simon & Schuster.

Ferguson, J. P., Tytler, R., and White, P. (2022). The role of aesthetics in the teaching and learning of data modelling. *Int. J. Sci. Educ.* 44, 753–774. doi: 10.1080/09500693.2021.1875514

Ford, D., and Parnin, C. (2015). "Exploring causes of frustration for software developers," IEEE/ACM 8th International Workshop on Cooperative and Human aspects of Software Engineering, 115–116, Institute of Electrical and Electronics Engineers.

Glaser, B. G., and Strauss, A. L. (1967). *The discovery of grounded theory: strategies for qualitative research*. Chicago, IL: Aldine De Gruyter.

Hannigan, S., Wickman, P.-O., Ferguson, J. P., Prain, V., and Tytler, R. (2022). The role of aesthetics in learning science in an art-science lesson. *Int. J. Sci. Educ.* 44, 797–814. doi: 10.1080/09500693.2021.1909773

Haupt, G., and Blignaut, S. (2008). Uncovering learning outcomes: explicating obscurity in learning of aesthetics in design and technology education. *Int. J. Technol. Des. Educ.* 18, 361–374. doi: 10.1007/s10798-007-9029-1

Huff, J., and Clement, H (2017). The hidden person within the frustrated student: an interpretative phenomenological analysis of a Student's experience in a programming course. In Proceedings Conference: 2017 ASEE Annual Conference and Exposition, (Paper ID #19136).

Jakobson, B., and Wickman, P.-O. (2008). The roles of aesthetic experience in elementary school science. *Research* 38, 45–65. doi: 10.1007/s11165-007-9039-8

Kant, I. (1790/1987). *Critique of judgement*. Indianapolis: Hacket.

Kelly, G. J., McDonald, S., and Wickman, P.-O. (2012). "Science learning and epistemology" in *Second international handbook of science education*. eds. K. Tobin, B. J. Fraser and C. J. McRobbie (Dordrecht: Springer, Netherlands)

Kinnunen, P., and Simon, B. (2010). Experiencing programming assignments in CS1: the emotional toll. In Proceedings of the Sixth International Workshop on Computing Education Research (ICER'10). Association for Computing Machinery, New York, NY, USA.

Knorr-Cetina, K. (1999). *Epistemic cultures: How the sciences make knowledge*. Cambridge: Harvard University Press.

Lewis, T. (2005). Creativity—a framework for the design/problem solving discourse in technology education. *J. Technol. Educ.* 17, 35–52. doi: 10.21061/jte.v17i1.a.3

Lewis, T. (2009). Creativity in technology education: providing children with glimpses of their inventive potential. *Int. J. Technol. Des. Educ.* 19, 255–268. doi: 10.1007/s10798-008-9051-y

Lishinski, A., Yadav, A., and Enbody, R. (2017). Students' emotional reactions to programming projects in introduction to programming: measurement approach and influence on learning outcomes. In Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17). Association for Computing Machinery, New York, NY, USA

Luxton-Reilly, A., Ibrahim Albluwi, S., Becker, B.A., Giannakos, M., Kumar, A.N., Ott, L., et al. (2018). Introductory programming: a systematic literature review. In Proceedings companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018 companion). Association for Computing Machinery, New York, NY, USA

Martin, C., Hughes, J., and Richards, J. (2017). Learning experiences in programming: the motivating effect of a physical Interface. In Proceedings of the 9th International Conference on Computer Supported Education, SCITEPRESS (162–172).

Nolan, K., and Bergin, S. (2016). The role of anxiety when learning to program: a systematic review of the literature. In Proceedings of the 16th Koli calling International Conference on Computing Education Research (Koli calling '16). Association for Computing Machinery, New York, NY, USA

Potvin, P., and Hasni, A. (2014). Interest, motivation and attitude towards science and technology at K-12 levels: a systematic review of 12 years of educational research. *Stud. Sci. Educ.* 50, 85–129. doi: 10.1080/03057267.2014.881626

Prain, V. (2020). "Methodological challenges in researching students' aesthetic responses in science inquiry" in *Methodological approaches to STEM education research, 2019*. eds. P. White, R. Tytler, J. C. Clark and J. Ferguson (Newcastle upon Tyne: Cambridge Scholars)

Robins, A., Rountree, J., and Rountree, N. (2003). Learning and teaching programming: a review and discussion. *Comput. Sci. Educ.* 13, 137–172. doi: 10.1076/csed.13.2.137.14200

Rorty, R. (1991). *Objectivity, relativism, and truth*. Cambridge: Cambridge University Press.

Sinclair, N. (2006). *Mathematics and beauty: Aesthetic approaches to teaching children teachers'*. Joplin: College Press.

Sparf, M. (2021). "I am magic!": pupils' engagement when designing in learning programming. *Designs Learn.* 13, 35–43. doi: 10.16993/dfl.168

Sparf, M., Löfgren, H., and Kreitz-Sandberg, S. (2022). Design for learning programming. Approaches taken by novice learners. Nordic. *Stud. Sci. Educ.* 18, 6–22. doi: 10.5617/nordina.8251

Svenningsson, J., Hultén, M., and Hallström, J. (2018). Understanding attitude measurement: exploring meaning and use of the PATT short questionnaire. *Int. J. Technol. Des. Educ.* 28, 67–83. doi: 10.1007/s10798-016-9392-x

Swedish National Agency of Education (2022). Läroplan för grundskolan samt för förskoleklassen och fritidshemmet [curriculum for the compulsory school, preschool class and school-age educare]. Swedish National Agency of Education. Available at: https://www.skolverket.se/undervisning/grundskolan/laroplan-och-kursplaner-for-grundskolan/laroplan-lgr22-for-grundskolan-samt-for-forskoleklassen-och-fritidshemmet

Swedish Research Council (2017). *God forskningssed [good research practice]*. Stockholm: Vetenskapsrådet [Swedish Research Council].

Turkle, S., and Papert, S. (1990). Epistemological pluralism: styles and voices within the computer culture. *Signs* 16, 128–157. doi: 10.1086/494648

Vinnervik, P. (2022). Implementing programming in school mathematics and technology: teachers' intrinsic and extrinsic challenges. *Int. J. Technol. Des. Educ.* 32, 213–242. doi: 10.1007/s10798-020-09602-0

Virtanen, S., Räikkönen, E., and Ikonen, P. (2015). Gender-based motivational differences in technology education. *Int. J. Technol. Des. Educ.* 25, 197–211. doi: 10.1007/s10798-014-9278-8

von Hausswolff, K., and Weurlander, M. (2020). Social dimensions in the lab session when novices learn to program. IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden

Warshauer, H. K., Starkey, C., Herrera, C. A., and Smith, S. (2021). Developing prospective teachers' noticing and notions of productive struggle with video analysis in a mathematics content course. *J. Math. Teach. Educ.* 24, 89–121. doi: 10.1007/s10857-019-09451-2

Wickman, P.-O. (2006). *Aesthetic experience in science education: Learning and meaning-making as situated talk and action*. Mahwah: Lawrence Erlbaum Associates.

Wickman, P.-O., and Östman, L. (2002). Learning as discourse change: a sociocultural mechanism. *Sci. Educ.* 86, 601–623. doi: 10.1002/sce.10036

Wickman, P.-O., Prain, V., and Tytler, R. (2021). Aesthetics, affect, and making meaning in science education: an introduction. *Int. J. Sci. Educ.* 44, 717–734. doi: 10.1080/09500693.2021.1912434

Witherspoon, E. B., Schunn, C. D., Higashi, R. M., and Baehr, E. C. (2016). Gender, interest, and prior experience shape opportunities to learn programming in robotics competitions. *Int. J. STEM Educ.* 3, 1–12. doi: 10.1186/s40594-016-0052-1

Witherspoon, E. B., Schunn, C. D., Higashi, R. M., and Shoop, R. (2018). Attending to structural programming features predicts differences in learning and motivation. *J. Comp. Assis. Learn.* 34, 115–128. doi: 10.1111/jcal.12219