A Culture Perspective on Students' Programming in Mathematics

Rune Herheim
Western Norway University of Applied Sciences
Rune.Herheim@hvl.no

Marit Johnsen-Høines
Western Norway University of Applied Sciences
Marit.Johnsen-Hoines@hvl.no

Abstract

Digital technology has become a key part of the development of society and influences people's ways of being, thinking, and communicating. Young people's digital cultures are continuously developing, and this article investigates how two students' productive struggle when programming in mathematics can be analyzed and understood from a cultural perspective. The data consists of two seventh-grade students who share a computer and program a pentagon. The task proves to be challenging and the students face several kinds of struggles. However, the students are persistent, apply communication qualities, and make continuous refinements in ways that create an interesting interweaving of mathematics and programming. The students' communication qualities are used to describe characteristics of a productive struggle in the mathematics classroom and how such a culture can be supported.

Keywords: Culture, community of communication, programming, productive struggle, mathematics, middle school, student pairs.

Introduction

This article investigates a pair of seventh-grade students struggling to program a pentagon. Our intention is to contribute to a better understanding of how programming and struggling can become an important part of a productive learning culture in which students' mathematical language and learning are supported and challenged. Important inspirations are this motto of a first-grade mathematics class: "If you are not struggling, you are not learning" (Carter 2008, p. 136) and the focus on how students engage with and express important mathematical ideas through programming in the ScratchMaths project in the UK (e.g. Benton et al., 2018). Facilitating struggling, programming, and expressing ideas can be described as developing a culture for learning. Our approach to culture is to a small extent based on traditions and history, rather it is positioned in the present, aimed at building a culture for learning.

Two Perspectives on Culture

Eriksen (2001) defined two perspectives for understanding culture – a *historical perspective* and a *current perspective*. From a historical perspective, culture can be understood as the practices, values, and ways of being that are transferred, while changing, to the following generations. This transfer of traditions influences how people think, communicate, and learn, and plays a role in how societies are built and how new generations are educated. The perspective emphasizes the historical roots and corresponds with perspectives from ethnomathematics (d'Ambrosio 1987; Bishop 1991; Fyhn & Nystad, 2014; Gerdes 2014). The historically rooted culture perspective is supported by traditions and experiences – and as culture is transferred to new generations, culture changes. Cultural learning is about people becoming part of and also influence on culture through cultural practices.

Within the current perspective, the focus is on how people socially build culture, on the present and the possibilities for mutual understanding (Eriksen, 2001, p. 60). Eriksen emphasized that culture is what makes communication possible. He discussed how the two perspectives influence the relationship between groups, between "we" and "the others". While the historical perspective can stabilize the relation between e.g. ethnic groups, the current perspective describes culture as created dynamically and continuously with an element of spontaneity; culture changes and is directed towards the future more than the present. Eriksen argued that the past is insufficient as a guideline for action because every period requires its diagnoses and its solutions. Internal connections do not exist in a particular cultural universe but in the whole created by each individual (p. 61). According to Eriksen, both the historical and the current perspective must be taken into consideration to see the whole picture of cultural processes. The current perspective on culture corresponds with Skovsmose's (2014) emphasis on foreground. He discussed ethnomathematical perspectives and showed how young people's intentions are directed towards the future, but at the same time have historical roots. This implies, in line with Eriksen, that they search for opportunities for their lives in an extended community, where communication is possible. They build cultures. Facilitating such communication requires the development of a culture of participation in which students' open sharing of ideas are valued, respected, and expected, Bennett (2014) argued.

Based on his discussions of the two perspectives on culture, Eriksen (2001) developed three key concepts. He described how different ethnic groups build *communities of interest* through their cultural peculiarities, and as part of larger communities. Identifying and developing joint interests and intentions are vital for building culture. He argued that culture can be regarded as a *community of communication*. Developing certain ways of talking and understanding builds culture. The experiences position people in a field of intersection between past and future and between individuality and group community. Eriksen emphasized that *cultural common denominators* make communication possible and are important for building communities of communication.

Students' digital learning actualizes the current perspective on culture. Children and young people's digital activities can be described by their continuous exploration of possibilities. They are individually active, and they search and develop local and global communities of interest as well as communities of communication. They build on their own and other's experiences, they struggle to understand, evaluate, and make choices and decisions, they take part in and develop a culture for learning. They gain skills and insights by trying out things, and quite often, they take a systematic trial and error approach. Sometimes they fail, but the mistakes help them generate insights that can be used in their next attempt. Like the second author's grandson (15 years) explains about ill-defined games:

> You must struggle more to understand … then I have to try something and if things do not work out, then I have to go back and think about where I did some choices. See if I could have done something differently and then try again.

Students' initiative, interests, and intentions are driving forces, and cultural common denominators create conditions for these processes. When their digital activities and digital learning take place outside school, they take ownership and are in control. As students, they are part of the school culture where learning is structured and organized by the curricula and the teachers. The progression in school might limit students' options and their possibilities to communicate with others, which in turn, can influence their ownership. Eriksen's (2001) current culture perspective actualizes a discussion about how young people can build a digital learning culture where they find digital learning in school relevant to their own digital learning culture. This can imply the identification of relevant cultural common denominators between activities they take part in at school and outside school.

Programming and Productive Struggle

Digital literacy is the ability to use technology to develop 21st-century knowledge and skills such as critical thinking, communication, and collaboration (Dede, 2010). The focus on digital literacy has re-orientated to include *programming*, and during the last decade, programming is integrated into mathematics (and other subjects) in many countries (Balanskat & Engelhardt, 2015). In the future, Balanskat and Engelhardt argued, students need to not only be consumers but also take part in developing technology. Bishop (1991) emphasized, from a cultural view on mathematics education, that the increased use of computers in mathematics education calls for a critical awareness of how and when to use mathematical techniques, and "this requires much greater thought, but also a different kind of thinking and therefore it requires a very different approach to the curriculum" (p. 8). Programming has become an important 21st-century skill, and this emphasis is becoming increasingly more evident in national policy documents (Bocconi et al., 2018). However, Forsström and Kaufmann (2018) and Popat and Starkey (2019) have reviewed research on programming in mathematics education and found a lack of convincing evidence for the educational potential of programming. They call for more research on collective learning through programming in mathematics, particularly because programming transforms education – it creates new ways of learning and communicating in the mathematics classroom.

We argue that one of these ways concerns the development of a community of communication in which struggling, trial and error, is systematized and made productive. The concept of *productive struggle* concerns students' "effort to make sense of mathematics, to figure something out that is not immediately apparent" and "solving problems that are within reach and grappling with key mathematical ideas that are comprehendible but not yet well formed" (Hiebert & Grouws, 2007, p. 387). It is not about creating needless frustration or give challenges that are beyond reach for the students, rather it is about facilitating a culture that provides challenges that make sense and facilitates understanding in progress. The choices and understanding during the process of working towards a solution are as important as solving the challenge. Such processes are dependent on the establishment of a joint community of interest for learning between the students. The National Council of Teachers of Mathematics (2014) highlighted that supporting students' productive struggle in learning mathematics is an important teaching practice. Like Mason, Burton, and Stacey (2010) argued: "being stuck is an honourable state and an essential part of improving thinking" (p. viii). Boaler (2016) supports the idea of struggle as valuable because that is when "the brain sparks and grows" (p. 11). Mason et al. (2019) and Boaler (2016) argued for a classroom culture where asking good questions, discussing and reasoning about complex problems, are more emphasized than calculating quickly. The students should be encouraged to take risks, struggle, fail and feel good about working on hard problems. It is about what Lee and Johnston-Wilder (2013) termed as mathematical resilience: "to develop a positive adaptive stance to mathematics such that it will allow them to continue learning despite barriers and difficulties" (p. 164).

Little research is done on the link between programming and struggling. One exception is Kim, Yuan, Vasconcelos, Shin, & Hill (2018) who found that processes of debugging when doing block-based programming promotes the understanding of programming and can facilitate productive struggle. Kapur (e.g. 2014) developed the concept of productive failure based on the value of letting students solve problems before being taught the concepts and procedures, even if it leads to failure. Granberg (2016) built on Kapur's (2014) work when she focused on how students' problem solving can generate productive struggles when they use GeoGebra to solve linear function tasks. The following recurring student comment illustrates the significance of students developing their methods and making errors: "We did not actually think until we discovered that we failed" (p. 46). GeoGebra provides instant feedback and visualizations, and this helped the students recall and reconstruct prior knowledge, which was vital for making their struggles productive.

Productive struggles can be linked with programming through computational thinking. Computational thinking is an approach that involves systematic steps to solve problems and find solutions, and programming is often required to execute these solutions (Morris et al., 2017). Several countries in Europe have integrated computational thinking and programming as a key competence to be acquired in their national curriculums (Balanskat & Engelhardt, 2015). Two components of computational thinking are debugging (finding and fixing errors) and persevering (keep going, be resilient), and these are core aspects within productive struggles.

Warshauer (2015) developed a framework for students' productive struggle based on how teachers' responses can "work to build community understanding" (p. 379). She described a community of communication where teachers do not funnel students towards quick answers and the easy correct method and do not deprive students of the opportunity to think and struggle. In her framework, Warshauer identified four kinds of struggles. The first is to *get started* and concerns confusion about what the task asks for and how to approach it. The second is to *carry out a process* and involves slow progress due to problems with carrying out procedures and making errors. The third is *uncertainty in explaining and sense-making* and concerns challenges in explaining the work and justifying strategies. The fourth is to *become aware of* and *express misconceptions and errors*. Warshauer underlined the importance of offering tasks that challenge students to do mathematics that goes beyond memorization and procedures and how maintaining this level of cognitive demand is dependent on communication qualities. Such communication qualities can be regarded as cultural common denominators and important for building a community of communication (cf. Eriksen, 2001). Important examples are teacher responses that provide information and link students' thinking with prior understanding. According to Warshauer, this involves careful listening, revoicing, and questioning that give direction and demand intellectual work. It is about developing a learning culture in which mistakes are made into potential sources for learning and sense-making.

Alrø and Skovsmose's (2002) concept of *dialogue* can be used to elaborate on cultural common denominators like Warshauer's communication qualities. Taking an Alrø and Skovmose perspective, the productive struggle is a collaborative activity where processes and answers are not given beforehand. This implies elements of unpredictability and risk and requires an open, equal, inquiring, and learning-oriented dialogue. According to Alrø and Skovsmose (2002), the concept of *intention in learning* plays an important role in a learning-

Journal of Mathematics and Culture
December 2020 14(2)
ISSN 1558-5336

oriented dialogue. This is closely related to Eriksen's (2001) concept of the community of interest because intention means having an interest and voluntarily be striving for something. Intentions cannot be forced upon someone else, because "intentional orientation must be performed by the person himself or herself" (Skovsmose, 1994, p. 184).

We find that Eriksen's (2001) current perspective on culture, the three concepts community of communication, community of interest, and cultural common denominators, Warshauer's framework for productive struggle, and Alrø and Skovsmoses' (2002) concept of dialogue offer a rationale and an approach for investigating students' collaboration when they strive and struggle. Being able to describe and analyze what productive struggle looks like, can provide insights into how to facilitate and develop a teaching and learning culture that supports rather than hinders such processes.

The Research Question

The perspectives presented above suggest that striving and struggling to solve mathematical problems is necessary for engaging with and expressing mathematical ideas and that developing a culture, a community of communication with particular qualities, can facilitate this. However, little research exists on students' mathematical struggles in programming settings, and on how such struggles can be productive and promote students' mathematical communication. The research question addressed in this article is, therefore: *what characterizes a culture, a community of communication, that facilitates students' struggle when programming a pentagon*? We investigate how the students take part in and start developing a culture for learning. The aim is to offer insights into what characterizes students' struggle, to identify qualities in what they say and do when they are challenged to program a pentagon and to gain a better understanding of how productive struggle can be facilitated.

Methodology

Eight seventh-grade students divided into four pairs participated in the study. Each pair shared a laptop and did block programming with Scratch. The students had little experience with programming, but Scratch does not require knowledge about programming language/syntax and the interface is quite transparent and user friendly. The students were challenged to use Scratch and program a quadrilateral, pentagon, a circle, a star, and a house. Two Master of Education students conducted the activity and the data collection, and they aimed to facilitate a mathematical focus by making tasks of appropriate complexity so that how to get started and to carry out the process was not immediately apparent for the students. The master's students wanted to give room for the students' learning culture by providing few to no guidelines for how to approach the tasks. To mitigate the chances for the struggles to be counterproductive, the master's students did a pilot study and prepared the tasks and the use of Scratch together with the teacher and their supervisor (the first author of this article).

This article concentrates on Ida and Knut's effort to program a pentagon. Their work was video recorded perpendicularly from the side, and an external microphone ensured good sound quality. A screen-recording app documented what the students did on the computer, and the recordings were merged into a picture-in-picture film, see Figure 1. This provided multimodal and informative documentation of their collaboration; on how they discussed, used the computer, and reacted to what happened on the screen. The four excerpts are chosen because they are informative and represent well the students' struggles.

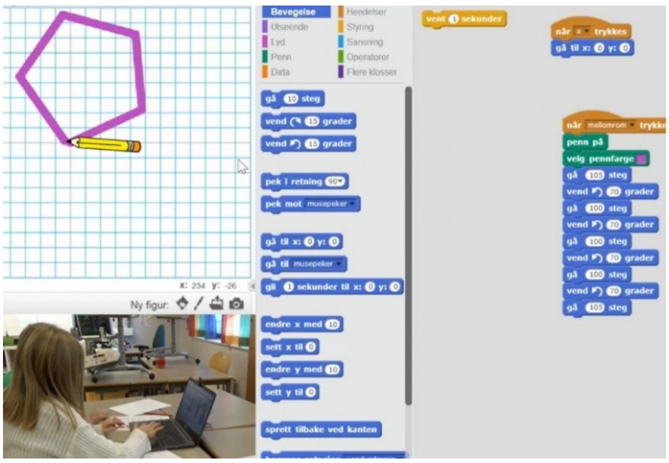

Figure 1: Screen dump from the PIP film

Eriksen's (2001) approach to culture is used as an analytical framework to gain insights into what characterizes a culture that facilitates productive struggle. When investigating aspects concerning the community of interest, we searched for traces concerning students' intention in learning. When investigating cultural common denominators, we identified and analyzed the communication qualities emphasized by Warshauer (2015) as part of her four kinds of struggles and Alrø and Skovsmose's (2002) concept of dialogue. This gave grounds for identifying key characteristics of a community of communication and gave insights into what characterizes a culture where students' productive struggles, their joint development of interests and cultural common denominators are facilitated.

Analysis: Programming a Pentagon – Managing Different Struggles

The following excerpts show, chronologically, how Ida and Knut collaborate to program a pentagon. They immediately face challenges with deciding the number of steps for the side lengths and how many degrees to turn for making suitable angles. Before this, they had programmed a square.

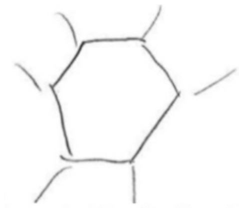

Figure 2: Drawing

Ida and Knut struggle to get started because they must discuss what a pentagon looks like. Knut takes a sheet of paper and draws a test shape (Figure 2). He counts the sides, becomes uncertain, and asks Ida if she thinks it is a pentagon. Ida first says yes, then takes a closer look and counts six sides: "That's a hexagon", she says. Both are trying to draw a pentagon by revising Knut's initial drawing but get confused: "This looks like a house?" "Then we make a house and a pentagon?" "No, it cannot be correct […] because the angles have to be identical." "Do they?" Their tone of voice is questioning and dwelling, they make multiple drawings, and question, listen, point at the drafts, and invite each other to participate, they are developing a community of communication. Ida and Knut show uncertainty and are struggling to understand

Journal of Mathematics and Culture
December 2020 14(2)
ISSN 1558-5336

– together. They dare to tell each other that they do not know and that they are uncertain. Both try different drawings and are aware of the possibility of making errors. It is risky, but they can operate with uncertainty, and it becomes clear that they trust each other. The question about identical angles makes them halt. They look at each other, turn and ask one of the master's students, but she encourages them to decide the criteria themselves. Ida and Knut carry on, agree on five sides as a key property and revise Knut's drawing to make a sketch similar to a regular pentagon. They have struggled to interpret the task mathematically and have not yet talked about programming. They have however made a strategic first step in preparing for solving the challenge of making a pentagon and started to establish a community of interest. A joint intention in learning becomes a cultural denominator through a learning-oriented dialogue and helps them managing uncertainty and the potential struggles that lie ahead of them. Their struggle now turns into a discussion about the size of the angles, and they start using the computer:

Knut: How can we do it? Wait, how many degrees are a pentagon?

Ida: I have no idea. A quadrilateral is 90°. A pentagon, I have no idea. It is at least more than 90°. […] I think we need more than 90°. It has to be more than 90°, kind of, because all of them are obtuse.

Knut: 120?

Ida: Yes, maybe. I don't know.

Knut: Let's just try, then.

Knut shows uncertainty and is inviting when he asks, "How can we do it?" He appears to have the sum of interior angles in mind when he asks how many degrees there are in a pentagon. Ida

Journal of Mathematics and Culture
December 2020 14(2)
ISSN 1558-5336

responds, and she as well expresses uncertainty: “I have no idea.” They continue to take the risk of telling the other that they do not know and strengthen their trust in each other. Ida follows up, refers to the 90° angles in a quadrilateral, and seems to have a regular polygon in mind. Ida goes on to argue that the angles must be more than 90° by calmly saying, “because all of them are obtuse”. Knut suggests trying 120°. Ida is not sure, but they decide to start coding, see Figure 3.

Figure 3: The first code

The program stops functioning temporarily, and when it starts working again, the students decide to try 160° instead of 120° and get the angle in Figure 4. When the angle is drawn, they say “What?” with a confused tone of voice. They see that 160° is an error and agree that it makes sense to try a smaller angle like 60° and get the angle in Figure 5. The *turn* block gives the turning angle α and not the anticipated interior angle β (α and β are supplementary angles), and this adds to their sense-making struggle of finding an appropriate angle. However, their systematic approach to test angles less than and more than 90°, and the way they support each other to address the issue of the number of sides, give them a better understanding of how the program works and make them ready to start extending the code to make a pentagon by inserting more of the *move* and *turn* blocks.

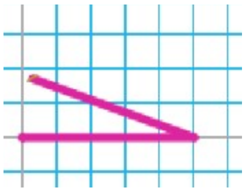
Figure 4: Test angle

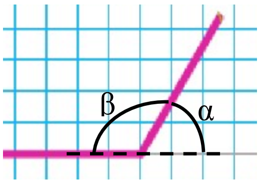

Figure 5: Test angle

Ten minutes earlier, Ida and Knut programmed 90° angles when making a square and that is probably a reference for them when they test angles larger and smaller than 90°. Programming the square was quite unproblematic because both knew the geometry of the square, and the interior angle was the same as the turning angle. When challenged to program a pentagon, it becomes evident that they are unsure about what a pentagon looks like. They are told to decide the criteria themselves. Even though both are confused, they make and investigate new drafts collaboratively, and at the end, they agree to make a pentagon with five sides and the angles should be of equal size. They go on to investigate the programming part, remembering that the square was made by moving a number of steps and turning 90°, but they struggle to become familiar with how the *turn* block works. Although the Scratch interface is quite user-friendly, it is not straight forward to understand that the *turn* block draws turning angles and not interior angles. The programming contains a mathematical problem, and the geometry contains a programming language problem, and the students are challenged to enter both languages simultaneously.

This initial phase has productive qualities for at least two reasons: the students become aware of what the task asks for and how to approach it, and they start developing a community of communication through how they apply dialogic communication qualities. Both are engaged, make drafts, express insecurity, and listen to each other's ideas. Their communication shows respect, trust and equality. Ida and Knut run a risk when they show insecurity, make proposals, and investigate a problem to which they do not know the answer. These characteristics, together with their persistence and mathematical resilience, the willingness to succeed, and how they manage to make a basis for a carry out process, describe their joint community of interest and learning-oriented intention.

Ida and Knut take a systematic trial and error approach, demanding and vulnerable, involving Warshauer's (2015) four kind of struggles and the establishment of cultural common denominators with the communication qualities included in Alrø and Skovsmose's (2002) dialogue concept. The dialogical approach continues when the students carry on and revise the code by changing the *turn* blocks to 60°, see Figure 6. They are both focused and lean and point at the screen. The communication is intense, energetic, they almost speak at the same time, but they manage to reply and continue each other's utterances. They end up with the shape in Figure 7, and the confusion and struggle continue:

Knut: We lack one. Look!

Ida: Yes, I don't quite get it.

Knut: No …

(They click space again, and get the same shape)

Ida: Then we have to take *move* 100.

Knut: No, it makes a hexagon.

Ida: Ok, but then we can't use 60°. Then we must have less. No, more.

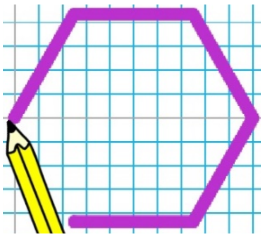

Figure 7: Hexagon?

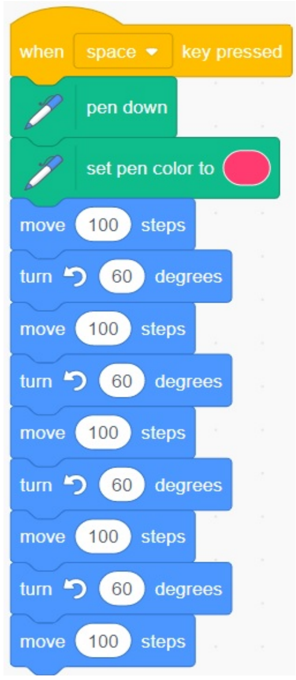

Figure 6: The code

The shape lacks one side to be closed. Ida and Knut get confused and uncertain, click the spacebar one more time but get the same shape again. They realize that 60° gives a hexagonish shape and try a different angle. Ida suggests that "we must have less" and then corrects it quickly to "No, more". They continue the inquiring process, try 80°, and get the shape in Figure 8 where the last side intersects the first. Based on this shape, they quickly decide to try 70°:

Knut: Wait! Isn't that a hexagon? … A pentagon.

Ida: Yes, that is what we want. 70° then. 70 is probably ok since 60 isn't … isn't, let us try 70, he-he.

Knut: Yes, it might be … if it is 70, then I'll be happy!

Ida: Yes, me too … okay. (Clicks the green flag)

Ida: Yees! No, what is that? (Points to the gap)

Knut: 1, 2, 3, 4, 5 (points to the sides), yes, then we got it!

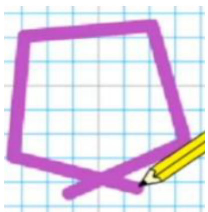

Figure 8: 80° angles

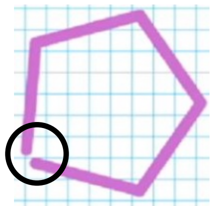

Figure 9: 70° angles

Ida: But it is not completely … closed … (points to the gap again)

Knut: It is, it is, it is (energetic tone of voice), they can't see that.

When the *turn* block is changed to 70°, they get the shape in Figure 9. It looks like a pentagon, but the last side stops just before it gets back to the starting point, the shape is not closed. Knut wants to accept it, and says, "They can't see that". Ida asks the master's students to approve it, but the master's students challenge them to explain why there is a small gap and give a hint by asking if it could be something with the numbers. Knut acknowledges the room for improvement, and they find yet again energy to revise their code:

Knut: Can try 102, no 105, on the first and the second. No, on the first and the last.

Ida: But maybe it is the degrees?

Knut: No, it is not something with the degrees, we have tried that too much.

Ida: But maybe some that are longer than others?

The students are aware of the gap error and adjust the first and the fifth *move* block from 100 steps to 105, click the green flag and get what looks like a closed pentagon (it kind of works because they have pen size 5). The gap is between the first and the fifth side, so it makes a lot of sense to increase the length of these two sides to get a closed polygon. They give a sly smile; point to the screen and comment that the sides are not equal.

Discussion and Concluding Comments

What characterizes a culture that facilitates students' mathematical struggle when programming a pentagon? There must be a mathematical challenge that is complex enough so that developing a community of communication, a community of interest, and cultural common denominators are required. Ida and Knut had to inquire what a pentagon looks like because they needed an understanding of its geometrical properties to get started with the programming. This understanding is used as a basis and is further challenged when they face some of the affordances and constraints of Scratch. When they start coding, the code blocks in Scratch facilitate a focus on the length and number of sides and the size and number of angles. Scratch features like the code blocks become something shared between the students, the code blocks become part of the cultural common denominators that are important for developing a community of communication.

This study supports and adds to the findings by Granberg (2016) and Kim et al. (2018) that programming can promote productive mathematical struggles. Ida and Knut grappled with the pentagon and the programming, both of which were ideas that were comprehendible but not yet well formed for them. Through the process, in an investigative mode, they dealt with several mathematical ideas about properties of pentagons such as the role of angles, the size of the angles and different types of angles, regularity, distinguishing between types of polygons according to the size of angles, and key questions like: can sides intersect and do polygons have to be closed? Making continuous adjustments to the angle size in the code involved mathematical understanding about the turning angle and the supplementary interior angle to make sensible adjustments. It was a process that can make them open to take on a more mathematical stringent approach later. For instance, spending time discussing regularity and becoming aware of the fact that their polygon is not regular can have increased their readiness to learn more about regularity and angle sizes. Ida and Knut struggled, but their working process was supported by how the shapes that represented each code were drawn instantly and provided immediate feedback. In this respect, the computer screen can be regarded as a third participant in their community of communication. It was a multimodal context in which mathematical ideas were represented with several different representations: the students' drawing with pencil and paper, the programming part with the codes and the drawings of the shapes, and the student's gestures and oral explanations. Thus, their struggle to program a pentagon can be characterized as a systematic trial and error approach that facilitated and required an interweaving of programming and mathematics. From a teacher perspective, being able to identify and understand what productive struggle looks like can be important in order to be able to acknowledge and support such student approaches.

Mason et al. (2010) argued that “a great deal more can be learned from an unsuccessful attempt than from a question which is quickly resolved” (p. ix). The struggle to program a pentagon could easily have been a missed opportunity if the teacher or a peer had given Ida and Knut a quick solution for how to program a pentagon: Use a turn block of 72° and a move block of x steps together with a repeat block and voila: a pentagon is drawn. The master’s students, however, did something quite different. The students were encouraged to use Scratch to program certain geometrical shapes, interpret the tasks, get familiar with Scratch, and decide the criteria for the pentagon – the students’ independence was called for. They had to struggle to investigate the mathematics and the programming language, and Ida and Knut faced all four kinds of Warshauer’s struggles. To *get started* they had to discuss the number of sides and if the sides had to be of equal length, and then if the angles had to be equal. Their struggles continued when *carrying out the process* of programming by first understanding how the turn function worked and thereafter with different angles, narrowing down towards an appropriate angle. On several occasions, the students faced *uncertainty in explaining and sense-making*. They started at 120°, then tried out 160° and got confused. Trying 60° gave them a shape that looked like a hexagon but lacked one side. They went on to try 80° and got a shape that both were about to accept as a pentagon, but the last side intersected with the first. Trying 70° gave them a shape that was almost a pentagon, but it had a small gap and was not closed. Knut argued that it was good enough and the task was solved, but the idea of changing some of the side lengths gave renewed energy. Finally, they managed to make what appeared to be a closed polygon with equal angles but different side lengths. It was not a regular pentagon, but the task never required that, and it was, mathematically speaking, not closed.

Despite their struggles, Ida and Knut kept going. They showed mathematical resilience and focused on the investigations and procedures they discussed and agreed to carry out. They managed to *become aware of and express errors* and evaluate them to make new, adjusted efforts. Their struggling processes can be described as loops: discuss–make new attempts–get negative/positive result–evaluate. The moments when a revised code was ready to be tested appeared to be critical: The students straightened up, looked at the screen, held their breaths, and then clicked the green flag. When becoming aware of an error, they brought forth new energy to start a new loop. Even after the fourth loop, when Knut had to acknowledge the tiny gap, they saw new possibilities for revising their strategy. Through their revisions, they showed a more and more refined understanding of the problem. A key characteristic of their struggle to program a pentagon was how it triggered communication qualities: they found strategies, tried them out, learnt from them to get new ideas for the next attempt, and they expressed ideas and listened to each other. Ida and Knut applied an open, inquiring, and learning-oriented dialogue and communicated in ways that made uncertainty and mistakes into sources for learning and sense-making. They strived to understand and master the mathematics and programming activity, they showed intention in learning. These qualities made their communication possible and correspond with the students' digital approaches outside school. The students showed that they can build a culture for learning that makes such communication possible, and the ways they communicated contribute to developing this culture.

Acknowledgement

This research is part of the LATACME project funded by the Research Council of Norway. Two Graduate students, Oda Pettersen and Tonje Lindberg did the data collection autumn 2018.

References

Alrø, H., & Skovsmose, O. (2002). *Dialogue and learning in mathematics education. Intention, reflection, critique* (Vol. 29). Dordrecht, the Netherlands: Kluwer Academic Publishers.

D'Ambrosio, U. (1987). *Socio-cultural bases for mathematics education*. Campinas, Brasil: Unicamp.

Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding – Priorities, school curricula and initiatives across Europe*. Brussels, Belgium: European Schoolnet.

Bennett, C. A. (2014). Creating cultures of participation to promote mathematical discourse. *Middle School Journal*, *46*(2), 20–25.

Benton, L., Saunders, P., Kalas, I., Hoyles, C., & Noss, R. (2018). Designing for learning mathematics through programming: A case study of pupils engaging with place value. *International Journal of Child-Computer Interaction*, *16*, 68–76.

Bishop, A. J. (1991). *Mathematical enculturation. A cultural perspective on mathematics education*. Dordrecht: Kluwer Academic Publishers.

Boaler, J. (2016). *Mathematical mindsets*. San Francisco, CA: Jossey-Bass.

Bocconi, S., Chioccariello, A., & Earp, J. (2018). *The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group*. https://doi.org/10.17471/54007

Carter, S. (2008). Disequilibrium and questioning in the primary classroom: Establishing routines that help students learn. *Teaching Children Mathematics*, *15*(3), 134–37.

Dede, C. (2010). Comparing frameworks for 21st century skills? In J. Bellenca & R. Brandt (Eds.), *21st century skills. Rethinking how students learn* (pp. 51–75). Bloomington, In: Solution Tree Press.

Eriksen, T. H. (2001). Kultur, kommunikasjon og makt [Culture, communication, and power]. In T. Hylland Eriksen (Ed.), *Flerkulturell forståelse* [Multicultural understanding] (pp. 57–72). Oslo: Tano Aschehoug.

Forsström, S. E., & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. *International Journal of Learning, Teaching and Educational Research*, *17*(12), 18–32.

Fyhn, A., & Nystad, K. I M. (2014). Sami culture and values: A study of the national mathematics exam for the compulsory school in Norway. *Interchange*, *44*(3–4), 349–367.

Gerdes, P. (2014). *Ethnomathematics and education in Africa*. Boane, Mozambique: Instituto Superior de Tecnologias e Gestão (ISTEG).

Granberg, C. (2016). Discovering and addressing errors during mathematics problem-solving – a productive struggle? *Journal of Mathematical Behavior*, *42*, 33–48.

Hiebert, J., & Grouws, D. A. (2007). The effects of classroom mathematics teaching on students' learning. In J. Frank & K. Lester (Eds.), *Second handbook of research on mathematics teaching and learning* (pp. 371–404). Charlotte: Information Age.

Kapur, M. (2014). Productive failure in learning math. *Cognitive science*, *38*(5), 1008–1022.

Kim, C., Yuan, J., Vasconcelos, L., Shin, M., & Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science*, *46*, 767–787.

Lee, C. & Johnston-Wilder, S. (2013). Learning mathematics – letting the pupils have their say. *Educational Studies in Mathematics*, *83*(2), 163–180. DOI 10.1007/s10649-012-9445-3

Mason, J., Burton, L., & Stacey, K. (2019). *Thinking mathematically* (2nd ed.). Essex, England: Pearson Education Limited.

Morris, D., Uppal, G., & Wells, D. (2017). *Teaching computational thinking and coding in primary schools*. London: Sage.

National Council of Teachers of Mathematics (NCTM) (2014). *Principles to actions: Ensuring mathematical success for all*. Reston, VA: NCTM.

Popat, S. & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, *128*, 365–376.

Skovsmose, O. (1994). *Towards a philosophy of mathematics education*. Dordrecht: Kluwer Academic Publishers.

Skovsmose, O. (2014). *Foregrounds. Opaque Stories about Learning*. Rotterdam: Sense Publishers.

Warshauer, H. K. (2015). Productive struggle in middle school mathematics classrooms. *Journal of Mathematics Teacher Education*, *18*(4), 375–400.