Parents' and children's learning when collaborating on inquiry-based mathematics and computational thinking tasks

Møller, Anders Kalsgaard; Kaup, Camilla Finsterbach

Published in:
Journal of Pedagogical Research

DOI (link to publication from Publisher):
10.33902/JPR.202319033

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Møller, A. K., & Kaup, C. F. (2023). Parents' and children's learning when collaborating on inquiry-based mathematics and computational thinking tasks. *Journal of Pedagogical Research*, *7*(2), 108-126. https://doi.org/10.33902/JPR.202319033

Journal of Pedagogical Research
Volume 7, Issue 2, 2023
https://doi.org/10.33902/JPR.202319033

Research Article

Parents' and children's learning when collaborating on inquiry-based mathematics and computational thinking tasks

Anders Kalsgaard Møller[1] and Camilla Finsterbach Kaup[2]

[1]*Aalborg University, Denmark (ORCID: 0000-0003-4581-4094)*
[2]*University College of Northern Denmark & Aalborg University, Denmark (ORCID: 0000-0001-7442-9174)*

In this paper we study how children aged 12-15 years learn together with their parents while solving a series of playful inquiry-based tasks with an educational robot. The purpose of the study is to understand how children and their parents learn mathematics and computational thinking in non-formal out-of-school learning activities. For the study we designed tasks that included mathematic problem solving and programming. The tasks were designed based on input from three mathematics teachers who participated in individual workshops. Over a period of approximately six weeks, three families worked together on the tasks. In the process, they were told to self-record and self-assess the process. The families video-recorded the process and after each task they completed they answered a few questions. At the end of the intervention, we interviewed the families about the process. The results showed examples of how the families worked with mathematics and programming within different practices. In general, the children were more challenged when it came to understanding the problems, the abstraction process and problem solving. In this part they received guidance from the parents. Conversely, the children were not particularly challenged by programming and in applying mathematics in the solutions..

Keywords: Robots; Mathematics; Collaborative learning; Computational thinking; Programming; Cultural probe

Article History: Submitted 20 October 2022; Revised 10 March 2023; Published online 10 June 2023

1. Introduction

Science, technology, engineering, and mathematics (STEM) competencies are in high demand, and many experiments have been conducted to introduce programming and computational thinking (CT) to children in primary school. While most teaching activities typically occur during regular school hours, there is significant potential for engaging students in STEM-related activities outside of school (Stevens & Bransford, 2007). According to Eshach (2007), to fully understand children's science learning, it is essential to also look at learning that takes place out-of-school. Eshach (2007) identifies three types of learning: formal, non-formal, and informal. Formal learning occurs in school as planned activities. Non-formal learning occurs in prepared but adaptable institutions and situations in an out-of-school context. Although non-formal learning has similar characteristics to formal education, the motivation for learning may be entirely internal for the individual.

Address of Corresponding Author

Anders Kalsgaard Møller, PhD, Aalborg University, Fredrik Bajers Vej 7K, 9220 Aalborg East.

ankm@learning.aau.dk

How to cite: Møller, A. K. & Kaup, C. F. (2023). Parents' and children's learning when collaborating on inquiry-based mathematics and computational thinking tasks. *Journal of Pedagogical Research, 7*(2), 108-126. https://doi.org/10.33902/JPR.202319033

Informal learning refers to situations that arise spontaneously, such as among friends and families. The family environment has a significant influence on children's development (García-Valcárcel-Muñoz-Repiso & Caballero-González, 2019). Sheehan and colleagues (2019) found that parents who engage in programming activities benefit their children's learning, and from Vygotsky (1978), we know parental involvement plays an essential role in children's learning. In this study, we apply a non-traditional approach to parental involvement. The non-traditional approach is more culturally and socially inclusive. It recognizes the importance of home and school involvement by supporting parents to help with modeling, encouragement, and communication. To support children, we are exploring how parents can use scaffolding (Wood et al., 1976). Scaffolding occurs within the zone of proximal development. The zone represents a space between learners' actual and potential levels of development, and it is a dynamic concept that helps learners move forward (Vygotsky, 1978). In scaffolding, the learner moves continuously toward the next level of development. Initially, scaffolding may appear strongly, disappear occasionally, and reappear when necessary (Wood et al., 1976).

In this paper we study how children aged 12-15 years learn about mathematics and computational thinking together with their parents. This, with the purpose of answering the following research question:

How do children and parents learn and support each other when engaging in non-formal inquiry-based learning activities involving mathematical- and computational thinking?

2. Literature Review

2.1. Computational Thinking

In 1980, Seymour Papert coined the term computational thinking (CT), and he believed that CT could be used to teach children to think in new ways and develop their creative and problem-solving abilities. Jeanette Wing later defined it as: "*the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent*" (Wing, 2011 p. 1).

Grover and Pea (2013) recommend that students learn programming and CT using a programming tool that is easy for beginners to use and powerful enough for advanced programmers. To provide a low floor, graphical programming environments can be used (e.g., block-based programming), which allows the user to focus more on the design and content creation rather than the syntax. Scratch is one of the most widely used graphical programming environments (Zhang & Nouri, 2019; Brennan & Resnick, 2012). Scratch is a block-based programming language developed at MIT for children. With Scratch the user can create interactive stories by combining blocks of codes. Based on Scratch programming, Brennan and Resnick (2012) developed a framework for studying and assessing CT development. This framework was divided into computational concepts that students use when programming (such as sequences, loops, and conditions) and practices that students develop when they apply the concept (abstracting and modularizing, testing, and debugging, reusing and remixing). Finally, computational thinking perspectives about who they are, how they interact with others, and how the technological world works (p. 1). When working with technologies, different disciplines are often intertwined and borrows methods from one another (Denning & Freeman, 2009). Keeping disciplines intertwined when teaching can be beneficial since, for example, CT and mathematics build a reciprocal relationship for learning between the two domains and are more closely aligned to professional practices (Weintrop et al., 2016).

In the same way, educational robots can promote learning across all STEM disciplines as children engage in activities that involve constructing, programming, and interacting with robots and, in the process, learn about gears, actuators, sensors, and programming. Furthermore, working with robots can also support collaboration, creativity, self-confidence, and leadership (González, & Muñoz-Repiso 2018; Bers, 2012)

Researchers have also demonstrated positive effects on children when parents and children collaborate, such as when they co-view (Strouse et al., 2017), co-read (Lauricella et al., 2014), or program (Sheehan et al., 2019). Collaborative problem-solving allows people to acquire more abstract knowledge compared to individual problem-solving (Scwartz, 1995) and competencies such as abstraction, algorithms, and procedures, debugging, problem decomposition, parallelization, pattern recognition, and simulation can be enacted while engaged in engineering design tasks with their families (Ehsan & Cardella, 2017).

2.2. Computational Thinking and Mathematics

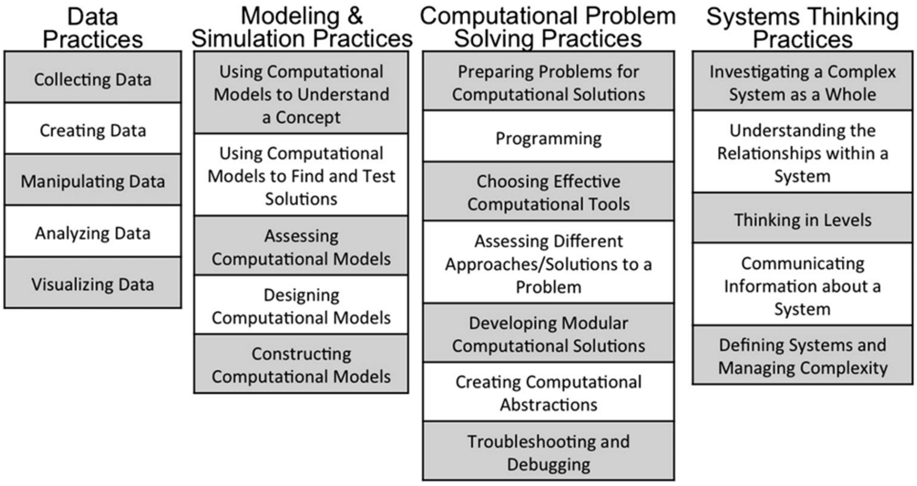
For K-12 students, robots can be an appropriate way to learn programming and problem-solving. Programming goes beyond simply coding; it reinforces abstraction and decomposition concepts and teaches children how to solve problems (Lye & Koh, 2014). This concept aligns with Sung and Black's (2021) view that CT involves solving problems systematically, sequencing the solution steps, and expressing solutions in a language an information-processing agent could understand. Problem-solving is highlighted as the common ground between CT and mathematics (Kallia et al., 2021). The core competencies of CT for mathematics are abstraction, decomposing, pattern recognition, generalization, modeling, automatization, analytical thinking, generalization, and evaluation (Kallia et al., 2021). Families can gain insight into prior problem-solving approaches as they work with those aspects. One of the things that link CT and mathematics together is problem-solving. According to Kallia et al. (2021), problem-solving should be a central goal of mathematics education According to Sung and Black (2021), CT can be used in mathematics to provide a better perspective on the field and support children's understanding of mathematics. Schoenfeld (1992) emphasized that mathematics is a subject that aims to understand patterns that surround us and are present in our minds. Mathematical language is based on rules, which children need to learn, but they must be motivated to move beyond rules to express mathematical activities. The goal of learning to think mathematically entails, learning a mathematical perspective, applying mathematization and abstract processes, and developing competencies with mathematical tools (Schoenfeld, 1992). The overlap between these disciplines has also been described by Pei et al. (2018). Their view is that CT and mathematics are distinct areas of study that support one another and may also be instructional related: Although they are closely related, they only sometimes appear together in instruction.

According to Lockwood et al. (2016), CT strengthens a child's understanding and learning of mathematics and ability to solve mathematical problems. For children to understand a mathematical problem, CT must be facilitated and provide them with conceptualizing factual and conceptual knowledge (Lockwood et al., 2016). A task can facilitate scaffolding that supports a child's CT and mathematical thinking. It emphasizes the importance of more knowledgeable peers of parents knowing when children are ready to advance to the next level of problem-solving (Lockwood et al., 2016). The more knowledgeable peers is aligned with Vygotsky's concept of ZPD. Using CT and mathematics can help children better understand how a problem-solving process works. Children who work with CT in mathematics develop procedural and strategic knowledge, which helps them plan and solve mathematical problems (Lockwood et al., 2016).

To gain a deeper understanding of mathematical phenomena, families can use computational problem-solving strategies, such as programming, algorithmic thinking, and creating computational abstractions (Weintrop et al., 2016). According to Weintrop et al. (2016), CT for mathematics and science can be framed and defined from a more theoretical perspective using a taxonomy with four categories. An example of where the taxonomy is applied is for a group of physics students who are attempting to calculate the gravitational constant and conservation of energy and momentum from a series of data observations. In the process, they go through several practices such as Data Collection, Data Visualization, Data Analysis, and Using Computational Models to Find/Test Solutions. An inquiry-based approach to mathematics can be enhanced by using this taxonomy to analyze and integrate CT into math curricula. This paper will focus on the categories of computational problem- practices. This category was also used when we created the task used by the families. Figure 1 illustrates the taxonomy in its entirety.

Figure 1

Computational thinking in mathematics and science taxonomy (Weintrop et al., 2016 p. 135)



The taxonomy will be used to analyze how families develop CT skills when solving tasks with the robot RoboMaster. The taxonomy presents opportunities for looking at how families develop math practices within each category when working with robots. Although most children are exposed to mathematical concepts in their early years, developing connections between computational skills and practical science methods is rare, especially in the out-of-school context.

2.3. Meaning-making within Families

According to Sheehan et al. (2019) parents can promote children's STEM learning by coding together. Through this, the parents can support the children through spatial talk, question-asking, task-relevant talk, and responsiveness, creating meaning about the task together. Taking part in social contexts involves two processes of meaning-making. According to Wenger (2010), participating in social life is directed at engaging in activities, conversations, and reflections. Similarly, we create physical and conceptual artifacts, such as words, tools, concepts, methods, stories, documents, links to resources, and other reification forms that reflect our shared experience and organize our participation. The interplay between participation and reification is essential for meaningful learning in social contexts. It is meaningless to produce artifacts without participation, and it is pointless to achieve implementation without artifacts. However, it is essential to note that participation and reification are not inseparable. When people participate and use reification, the memory of those two is intertwined but distinct. As they interact, a social history of learning combines individual and collective learning. The participants of this history define a 'regime of competence,' criteria, and expectations they rely on to recognize their membership in the community.

The competence can be seen as: 1) Being able to understand what matters, what the community's enterprise is, and how it offers a lens through which to view the world, 2) Having the ability (and permission) to engage in productive community interaction, and 3) Making appropriate use of the repertoire of resources that the community has accumulated over time (Wenger, 2010, p. 180)

We analyze the communication of the families and observe how they negotiate and make sense by working on tasks with the robot.

2.4. Inquiry-based Scaffolding

We offered families opportunities to engage with complex tasks through scaffolded inquiry-based tasks. Providing scaffolding creates a more tractable learning environment for families by making complex and challenging tasks available, manageable, and within the zone of proximal development (ZPD) of the families (Hmelo-Silver et al., 2007; Vygotsky, 1978, Wood et al., 1976). A collaborative inquiry-based environment fosters content knowledge development and skills for inquiry-based learning, often seen in scientific disciplines. When children/families are given structure and guidance by mentors who scaffold them through task structuring and hints without explicitly giving them the final answer, scaffolding can be seen as an integral part of a learning environment where they become increasingly competent problem-solvers. As families gain experience, scaffolds gradually fade as they can accomplish learning goals without the scaffold (Lajoie, 2005). In the zone of proximal development of the families, scaffolding is essential for assisting them in comprehending aspects of a task they cannot solve on their own (Vygotsky, 1978).

According to Saye and Brush (2002), scaffolds can be divided into two processes. Hard scaffolds are "static supports that can anticipate and plan for student difficulties with a task in advance" (p. 81). In addition to hard scaffolds, there are also 'soft scaffolds,' which are dynamic and situational. To provide soft scaffolds, teachers should continuously diagnose their learners' understandings and provide timely support to their responses (Saye & Brush, 2002, p. 82). To support families in trying inquiry-based learning, we developed tasks that act as hard scaffolding. With these scaffolds, families can move forward with the subsequent phases of inquiry and engage in self-directed learning. In our study, we investigated how families used the hard scaffolds and how they worked together to provide soft scaffolds to enable them to understand the inquiry process better and the use of RoboMaster.

3. Research Design

In this paper, we study how children and their parents engage in inquiry-based learning outside of school, which meant that the study had to take place in the families' own homes. Due to practical and privacy reasons, it is difficult to carry out studies where you enter people's homes. To overcome these challenges, we apply a method inspired by the cultural probe approach (Gaver et al., 1999). Cultural probe is a method for collecting information about a particular culture, group, or context, in which a probe containing various artifacts is sent to the user, encouraging them to share different things about their daily life. In our study selected families receive a package with a series of playful educational tasks with the DJI RoboMaster S1 educational robot https://www.dji.com/dk/robomaster-s1. Using this approach, we instructed the families to self-record and self-assess the process of solving the tasks. As an alternative to more time-consuming and intrusive ways of collecting data, the cultural probe concept is explored here for use with families for gathering contextually sensitive information (Wyeth & Diercke, 2006). By asking the families to self-record themselves we gave them control of what they wanted to document. This way respecting the privacy of their household while still gaining access to information.

3.1. Participants

Three Danish families participated in the study. In the following we present a short description of each family and their opinion and experience with math and programming.

Family 1: A mother and her two sons aged 14 and 15. The two boys think that math in school is okay but sometimes challenging. Besides, the youngest boy had no prior experience with programming. The oldest boy has a little experience with building and programming a small robot. The mother had a little experience with block-based programming.

Family 2: A mother and her son aged 13. The son is decent at math but more interested in other science courses. He also had a little experience with Scratch programming. The mother has no prior experience with programming.

Family 3. A father and his daughter aged 12. She likes math and have tried to program a micro:bit in school and have done some Scratch Junior projects with her father. The father has some experience with programming.

3.2. Data Collection

For a period of six weeks the families were invited to experiment with the robot and try to solve the tasks from the booklet together. We instructed the families to self-assess the process of solving the tasks. During each task, the families used a screencast on the iPad that recorded data about how the families programmed RoboMaster. The families were also told to use the camera to document the process. After each task they completed there was an envelope with five questions about their work with RoboMaster. They were e.g., asked to review the task and what challenges they had. In the process they were asked to discuss both mathematic, CT, and robotics-related content and how they collaborated.

The families received information about the purpose and process of the study, how data was collected, used, and stored. To ensure voluntariness we asked both children and parents to sign a consent form after introducing them to the study. The collection and storage of the data was done in accordance with the existing GDPR legislation.

After completing the test period each family was interviewed about the process. The interviews were held as semi-structured interviews (Ravitch & Carl, 2021). The interview contained questions about their experience of working with the robot, how they had worked and collaborated, what they thought about the specific tasks and mathematics and programming activities.

3.3. Materials and Tasks

For each family we put together a package with the robot RoboMaster S1, an iPad, a video camera, a booklet with tasks and different materials needed for the tasks such as cones, tape, practice targets and a folding rule (Figure 2).

RoboMaster S1 is an educational robot (Figure 3). The robot is built with four omnidirectional wheels that enable it to drive in all directions. It has a cannon that can shoot with laser and water beads. It can be remotely controlled and programmed with Scratch using an app for IOS and Android devices. The robot has a camera mounted under the cannon which provides a first-person view from the tank when controlling it. The robot has several built-in LED's that can be programmed to change color. The robot also comes with build-in sensors that enables it to respond to touch and sound. Furthermore, the camera uses image-recognition to detect different objects such as practice targets, people, hand gestures and marked lines.

We developed eight tasks for the booklet. The tasks were developed based on input from individual sessions with three mathematic teachers who all had experience with working with CT. The teachers participated in sessions to help us create age-appropriate tasks for the RoboMaster that included mathematic and CT content. In the sessions we first introduced RoboMaster. We proceeded to ask questions concerning their students' learning goals and different approaches for teaching, including specific activities, didactics, and pedagogical considerations. Finally, we asked about specific activities with RoboMaster to meet these learning objectives. The input was adapted, and subtasks added to ensure a proper scaffolding of the content. The booklet was afterwards sent to one of the teachers who gave feedback. The feedback included suggestions for different wordings and minor changes to the content. We focus on the category of computational problem-solving developed by Weintrop et al. (2016). They define this category of practices and strategies from computer science to capture its contributions to contemporary mathematics and science (p. 139). Using the category, we scaffolded the task so families could learn different mathematics practices to solve different problems step-by-step. An overview of the tasks can be seen in Table 1 and an excerpt of task 1 can be found in the appendix.

Figure 2
The package with the Robomaster robot, iPad, video-camera, booklet and other materials

Figure 3
Picture of the RoboMaster S1 robot

3.4. Data Analysis

We have followed a thematic analysis (Braun & Clarke, 2006) using interviews and video recordings data. According to Humble and Mozelius (2022), thematic analysis supports an analysis that can contribute to a deeper understanding of what is being studied. The thematic analysis was chosen to create a deeper understanding of how children and parents learn together. In the analysis of the data we carefully followed Braun and Clarke's six steps of thematic analysis (2006) where themes have continuously been developed, adapted, and merged/divided. The six steps are: familiarization with the data, generating initial codes, searching for themes, reviewing themes, defining and naming themes, and producing the report. The selected quotes in the results and analysis section were translated from Danish to English and reproduced to capture their meaning rather than a literal translation.

Table 1
Overview of the tasks in the booklet

Task	*CT component*	*Programming*	*Math*	*Taxonomy*	*Scaffolding Task*
1	Evaluation. Evaluate the solutions	Introduce the programming environment and block-based programming. Copy the block from the booklet.	The purpose of this task is to introduce mathematical concepts such as coordinate systems, translation, and rotation.	Computer Programming	Copy the solution from the tasks and program RoboMaster Explain what happens when you run the code Try by yourself to create a code to move RoboMaster backward.
2	Pattern Recognition. Identifying parts of a problem that repeat. Loops are introduced to change the color of the RoboMaster LEDs.	Introduction of loops, new blocks, and parameters.		Computer Programming Choosing effective computational tools	Building on knowledge learned from task 1. Changing the color. Try it out yourself by combining the colors in other combinations. Try to understand what happens if changing some of the parameters.
3	Decomposition: The process of breaking down a problem into smaller pieces. Building the algorithm for a square in small steps and then using loops to create a square.	Using knowledge from previous tasks. Combining more blocks. Repeat.	Geometric shapes (squares and rectangles)	Computer Programming Choosing effective computational tools	Building on knowledge learned from previous tasks.
4	Algorithms: A sequence of steps that can be followed to achieve the desired result.	Introducing of different sensors and events e.g., changing the color of the LEDs when the robot drives into a wall, when it registers the sound of a clap or when the camera detects a person.		Computer Programming Choosing effective computational tools	Building on knowledge learned from previous tasks.

Table 1 continued

Task	*CT component*	*Programming*	*Math*	*Taxonomy*	*Scaffolding Task*
5	Abstraction: Creating useful generalizations by identifying and representing only the relevant parts of a problem.	Using yaw and pitch degrees to turn the cannon.	Rotations about different axis.	Computer Programming Choosing effective computational tools Assessing different approaches/solutions to a problem	Using the cannon. Manual control of RoboMaster to shoot targets.
6	Algorithms: A sequence of steps that can be followed to achieve the desired result.	Manually control of RoboMaster and then program to control it automatically. Different coding concepts introduced such as conditions.		Computer Programming Choosing effective computational tools Assessing different approaches/solutions to a problem	Driving RoboMaster through a slalom track. First manually controlled and afterwards programmed to automatically complete the track.
7	Abstraction: Creating useful generalizations by identifying and representing only the relevant parts of a problem.	The players shift to program RoboMaster to a square. In this task variables and coordinate systems are introduced.	Building a coordinate system on the floor. Algebra	Computer Programming Choosing effective computational tools	Inspired by the game battleship. The programming is scaffolded by showing how to solve the solution. Try to explain what happens when using conditions.
8	Abstraction: Creating useful generalizations by identifying and representing only the relevant parts of a problem.			Computer Programming Choosing effective computational tools Assessing different approaches/solutions to a problem	Open ended task. The families use their previous knowledge to program RoboMaster.

In the process we first transcribed the interviews and made observation notes from the video recordings. This way we familiarized ourselves with the data and began labeling it with codes (Braun & Clarke, 2006). In this process, we also anonymized the data.

In the following sections, we will use the notation F1, F2, and F3 when we refer to each family and F1C1 as the youngest child of family 1, and F1P is the parent in family 1.

4. Results and Analysis

In the following section we first provide an example of how the families worked with the tasks through a vignette of F3's interaction while solving task three. The vignette shows how they work together and discussion of CT and mathematical concepts and the use of different Computational problem-solving practices. Subsequently, we present an analysis of the results divided into two overarching themes. First, we analyse the distribution of roles between children and parents and how they supported each other. In the next part, we analyse how they learned and developed an understanding of Computational Thinking and Mathematics.

4.1. Vignette

In the vignette, F3 is working on task three where they use tape to create different geometric shapes and afterwards program Robomaster to move in the given shape.

The father reads the task, and the daughter programs on the iPad during the activities. With tape, the daughter draws a one-meter line on the floor. The daughter calculates how many seconds the robots should be moving if the chassis speed is 0,5 m/s. She says 2 seconds, and then they try it with the robot.

After that, the father asks if she can get RoboMaster to move backward (this was not a part of the task).

> F3C. *"Yes, instead of 0 degrees, we have to change it to 180 degrees."*
> F3P: Confident young lady.
> F3C: No, I just had mathematics.
> FCP: Why is this mathematics?
> F3C: It's about degrees, and I have had about angles.

With her father's help, she can see how she uses mathematics to solve the problem. It can also be related to Weintrop et al. (2016)'s taxonomy, where she used computational problem-solving practices to test and explain how to solve the problem.

Her next step was to place another 0,5-meter tape perpendicular to the tape on the floor. Then she tries to make the following algorithm on the iPad. By pointing to the iPad, her father assists her in choosing the right block. They are having problems finding the correct block to use when the robot has to turn 90 degrees. In addition, they are trying a variety of solutions. After some tries, they found the right solution.

Now they must build a rectangle. In the booklet, they are now presented with loops. The daughter starts to make the first part of the algorithm from the beginning, with help from her father.

Next, her father asks if she wants to repeat the process or use a loop.

Her father explains that she must build half of the rectangle.

> F3C: puts a loop on?
> F3P: yes, you can ask it to do it all again. The same movements must be repeated.
> F3C: Okay, let us make a loop. How do we do that?

They identify that the first part of the program that needs to be repeated two times to make the rectangle. By then, they are using the CT practices: pattern recognition. To accomplish this, they developed an algorithm that repeats half of the rectangle twice. They are using the CT practices: algorithm.

After completing the task, the father and daughter decided to set up the speed to 1 m/s (This was not a part of the task). They then discuss how to determine and manipulate the measurement to solve problems about the distance and then change the chassis to translate at 0 degrees for 1 second. Furthermore, the family realized that by doubling the speed while keeping the duration fixed, the RoboMaster would travel half as far, demonstrating the family's problem-solving abilities.

4.2. Roles and Family Support

In F1, all three family members worked together on the tasks. Occasionally, only the mother and F1C2 worked on the tasks as the youngest had lost patience. In general, the mother would read the tasks, and the boys would do the programming. However, the mother would also pitch in when it became challenging or if the boys gave up.

> "So you had to do some strange things on the iPad sometimes. Me and (F1C2) had tried something and spent 5-10 minutes programming, then you had to go back in and delete it" (F1C1)

The above quote shows how the mother occasionally tried to experiment with programming but at one point, she ended up deleting the codes to the boys' frustration as she found programming challenging.

In F2 it was primarily the boy who has worked on the project and asked for help when he needed it. He received help from both his older brother (aged 17), his father and his mother. The help he received was mainly to understand the tasks, and practical issues. After completing the tasks, he has often shown the tasks to the other family members.

> "It was a lot about understanding the tasks.... So, he (the father) got like... Then he got to tell me what to do, because he is not.... He's not that good at how to code them and then when he explained it to me, I could figure out the code afterwards." (F2C1)

The quotes show a clear distribution of roles. The boy did the programming but needed help to understand the task. The father did not understand the code but helped with explaining the task.

In F3, the robot and the tasks were considered a fun activity to do together. It was mainly the father who read the tasks and the daughter who programmed; however, the roles have also changed a bit along the way and the father has sometimes helped if there were difficulties.

> "yes, I've touched it a few times (The iPad), when I get too enthusiastic, and then I'm reminded of what the roles are, and I go back to my role"(F3P)

This shows how the two are both equally enthusiastic and involved in the tasks. The father explained that the daughter could do most of the programming but sometimes needed help with translations because the interface was in English.

All three families were very busy, and it was difficult to find time to work with the robot. The various commitments such as school, work, cooking and cleaning and various hobbies meant it was often during the evenings or in the weekends that there was time to do the tasks.

All the children expressed that the robot was fun, cool and had a lot of functions. Overall, the impression seems to be that the progress has been fun, however with some frustrations and all have enjoyed having a shared activity.

4.3. Computational Thinking and Mathematics

All children expressed that it was relatively easy to program the robot. They showed understanding of computational concepts (Brennan and Resnick, 2012 such as repeat (loops), variables and if/else statements (conditions). Implicitly, they also talked about parameters and sequences.

F1C1, F1C2 and F2C1 believed they had an easier time understanding the codes and programming the robot than their parents. The biggest obstacle seemed to be to understand the task and afterward figuring out how to solve it.

> "what kind of codes do you set up… That has not been the most difficult (thing). It is about figuring out what to do, i.e. figuring out how long it will run for? How do you get it to spin and stuff? The code itself makes sense, but information and then how to put it into the codes (is harder)." (F2C)

F2C points out that the process from understanding the task or problem and finding a way to solve it and translate that into programming was the hardest part. This shows an abstraction process, a computational practice (Brennan and Resnick, 2012). In addition, the described process includes several of the categories in Weintrop's (2016) Computational problem-solving practices.

The children also applied different debugging strategies such as experimenting with changing the code. Another strategy was used by (F3C) who knew that a segment of the code from the last exercise had worked before, thus he deduced that the error must be somewhere else. This approach shows that (F3P) understand that code can be divided into modules and that you can reuse and remix codes. Both approaches are also descripted as computational practices by Brennan and Resnick (2012) and as Developing Modular Computational Solutions by (Weintrop, 2016).

While the parents were not as quick at picking up programming as the children, they also learned something about programming in the process by collaborating with the children and from the children presenting the solutions as is evident in the following quote:

> "He (the father) was very interested in how it worked. I think he also learned some of it along the way when I showed it to him."(F2C)

As with the programming it seemed that children found it easy to understand the mathematics involved in the tasks. They recognized the mathematics and could apply their knowledge to solve the problems.

> "Yes, a coordinate system, when you know the coordinate system, it was fairly easy." (F2C)

> "there was the equation of how X works, so that also made sense." (F2C)

However, there were also times when the mathematics caused a bit of trouble.

> "the thing with calculating exactly where it is, then having to shoot at it with what's it called? yaw and pitch I think it was called. That's the thing with calculating how many degrees it would have to shoot to hit the things, I think that too... that video is considerably long." (F2C)

Although (F2C) understood how to calculate in degrees, it was new and challenging for him to have to do it in two dimensions, namely with yaw and pitch.

5. Discussion

The results show families engaging in mathematical thinking and CT practices. Using CT practices within the context of a task, enables mathematical reasoning to be supported and enacted. We created the booklet to support the families through scaffolded inquiry-based tasks. The booklet contains tasks that support the taxonomy developed by Weintrop et al. (2016). As we have chosen an inquiry-based approach most of the practices used and articulated were within the category of Computational problem-solving practices (Weintrop 2016). The booklet is a hard scaffold that gives families a plan for completing the tasks (Saye & Brush, 2002). In a collaborative inquiry-based environment, families provide themselves with soft scaffolds through which the child or parent explains CT concepts, mathematical concepts, or CT practices to each other. The hard and soft scaffold is shown in our vignette of task three. They supported each other by explaining mathematical concepts, such as degrees and angles, when they needed to rotate the robot, or CT concepts, such as loops, when they needed to repeat an algorithm. It was highlighted that the children, in two of the families had challenges with understanding the problems and finding a way to solve it. The parents often assisted with this part. Which indicate a superior understanding of the abstraction process (Schoenfeld, 1992) that enabled the parents to solve problems with the robot despite a lack of experience with programming. Which is consistent with the findings in (Ehsan & Cardella, 2017) and (Kallia et al., 2021). Once the children understood the solution, they could easily program the solution. This shows that within computational problem-solving practices, the children were particularly good at computer programming, whereas the parents had a greater understanding of practices such as: preparing problems for computational solutions and computational abstractions. This could be due to the parents' greater experience with general problem solving and reading text descriptions.

We saw the families work with troubleshooting and debugging practices. We saw F2C was able to break down the code into smaller parts, understanding that the part he used in the previous task should be working the same way. Thus, he showed an understanding of the practice Developing Modular Computational Solutions. As the skills within this practice increase it becomes easier to incrementally construct solutions, test components and be able to reuse them (Weintrop, 2016 p. 139).

The vignette, presented in this paper, shows that the task provided the family with a hard scaffold. Having completed the task efficiently, the family began assessing different solutions to the same problem and creating a new task. Their understanding of mathematics is helping them solve problems by creating CT practices. This is in line with Lockwood et al. (2016), which states that CT can strengthen a child's understanding and learning of mathematics. The interviews, evaluations, and video observations show that families use mathematical language to explain what happens when, for example, they have programmed RoboMaster to create a square (Schoenfeld, 1992). It was motivating for the families to work with RoboMaster. The families express that they enjoy working with the robot but also emphasize the connection to mathematics in their evaluations. As families work on tasks and use RoboMaster, they engage in activities and conversations. Through RoboMaster, they produce meaning and investigate CT and mathematical concepts. Negotiation occurs when they solve problems together. When working with RoboMaster, the families collaborate and create a community and history of working with the robot. For the families, reification is getting into meaningful processes to work with the robot, understand CT, and use mathematics when programming (Wenger, 2010). When using RoboMaster as an artifact for reification, they can help and support each other. By working with the robot, they develop new competencies in CT and programming and use mathematics to understand CT practices. Over time, they accumulate a repertoire of resources.

With our research approach of using cultural probes, we have collected data about interactions and conversations that are typically difficult to access without intervening and disrupting the environment in which they occur. Additionally, the sporadic occurrence of the events makes it challenging to gather data. The recorded data has been analyzed using thematic analysis. One risk of using thematic analysis is the potential for subjective bias when collecting and analyzing data (Humble & Mozelius, 2022). Since the families themselves have been responsible for the data collection, they have determined what was recorded and when. This means that they have largely been able to decide what was important to document. The disadvantage is that we have not been able to make these selections ourselves and may not necessarily have the complete picture. The research design may have provided more ecological settings, however with the cost of control of the scenario.

The findings in our study are based on results from three families where we see great diversity in the families' approach and their prerequisites for working with the tasks. We therefore expect similar variation in other families.

Thus, more research is needed to investigate this area further. Similar, the study is carried out with selected tasks primarily within the category that Weintrop (2016) defines as computational problem-solving practices. The parent-children relationship might unfold completely differently if the focus was on one of the other categories.

6. Conclusion

Engaging in non-formal inquiry-based learning activities focused on mathematical and computational thinking can help families develop new competencies and support each other's learning. Through collaboration on tasks, families can investigate CT and mathematical concepts, create meaning, and improve their CT and programming skills. The study found that the use of a booklet as a hard scaffold helped families complete tasks, while soft scaffolds facilitated collaborative inquiry-based learning and enabled family members to explain mathematical and CT concepts to each other. Children may develop proficiency in computer programming and using mathematical concepts, while parents may gain a greater understanding of how to create computational abstractions and prepare problems for computational solutions. Parents can also provide support to children, indicating mutual support between them in the learning process. Further research in this area is important, as there may be variation in how families approach tasks, and different CT practice categories may lead to different outcomes in parent-child learning.

Author contributions: All authors are agreed with the results and conclusions.

Declaration of interest: None of the authors have a conflict of interest in this project.

Ethics declaration: The research conducted in this paper does not require an Ethics Committee Approval according to the Danish legislation.

Funding: No funding source is reported for this study.

References

Bers, M. U. (2012). *Designing digital experiences for positive youth development: From playpen to playground*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199757022.001.0001

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, *3*, 77–101. https://doi.org/10.1191/1478088706qp063oa

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 annual meeting of the American educational research association*, *1*, 1-25. http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Denning, P. J., & Freeman, P. A. (2009). The profession of IT computing's paradigm. *Communications of the ACM*, *52*(12), 28-30. https://doi.org/10.1145/1610252.1610265

Ehsan, H., & Cardella, M. E. (2017). Capturing the computational thinking of families with young children. *Paper presented at 2017 ASEE annual conference and exposition*. Columbus, Ohio. http://doi.org/10.18260/1-2--28010

Eshach, H. (2007). Bridging in-school and out-of-school learning: Formal, non-formal, and informal education. *Journal of Science Education and Technology, 16*, 171–190. https://doi.org/10.1007/s10956-006-9027-1

García-Valcárcel-Muñoz-Repiso, A., & Caballero-González, Y. A. (2019). Robotics to develop computational thinking in early Childhood Education. *Comunicar. Media Education Research Journal*, *27*(1), 63-72. http://doi.org/10.3916/C59-2019-06

Gaver, B., Dunne, T., & Pacenti, E. (1999). *Design: cultural probes. interactions*, *6*(1), 21-29. https://doi.org/10.1145/291224.291235

González, Y. A. C., & Muñoz-Repiso, A. G. V. (2018). A robotics-based approach to foster programming skills and computational thinking: Pilot experience in the classroom of early childhood education. In *Proceedings of the sixth international conference on technological ecosystems for enhancing multiculturality* (pp. 41-45). https://doi.org/10.1145/3284179.3284188

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational researcher*, *42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and achievement in problem-based and inquiry learning: a response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, *42*(2), 99-107. https://doi.org/10.1080/00461520701263368

Humble, N., & Mozelius, P. (2022). Content analysis or thematic analysis: Similarities, differences and applications in qualitative research. In M. A. Y. Oliveira & C. Costa (Eds.), *Proceedings of the European Conference on Research Methodology for Business and Management Studies* (vol. 21, pp. 76-81). Academic Conferences International Limited.

Kallia, M., Borkulo, S., Drijvers, P., Barendsen, E. & Tolboom, J. (2021) Characterising computational thinking in mathematics education: A literature-informed Delphi study. *Research in Mathematics Education*, *23*(2), 159-187. https://doi.org/10.1080/14794802.2020.1852104

Lajoie, S. P. (2005). Extending the scaffolding metaphor. *Instructional Science*, *33*(5), 541-557. https://doi.org/10.1007/s11251-005-1279-2

Lauricella, A. R., Barr, R., & Calvert, S. L. (2014). Parent–child interactions during traditional and computer storybook reading for children's comprehension: Implications for electronic storybook design. *International Journal of Child-Computer Interaction*, *2*(1), 17-25. https://doi.org/10.1016/j.ijcci.2014.07.001

Lockwood, E., DeJarnette, A. F., Asay, A., & Thomas, M. (2016). Algorithmic thinking: An initial characterization of computational thinking in mathematics. In M. B. Wood, E. E. Turner, M. Civil & J. A. Eli, (Eds.), *North American Chapter of the International Group for the Psychology of Mathematics Education* (pp.1588-1595). The University of Arizona.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, *41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Papert, S. (1980). *Mindstorms: children*. Computers and powerful ideas, New York Basic Books Inc.

Pei, C., Weintrop, D. & Wilensky, U. (2018). Cultivating computational thinking practices and mathematical habits of mind in lattice land. *Mathematical Thinking and Learning, 20*, 75-89. https://doi.org/10.1080/10986065.2018.1403543

Ravitch, S. M., & Carl, N. M. (2021). *Qualitative research: Bridging the conceptual, theoretical, and methodological.* Sage Publications.

Saye, J. W., & Brush, T. (2002). Scaffolding critical reasoning about history and social issues in multimedia-supported learning environments. *Educational Technology Research and Development*, *50*(3), 77-96. https://doi.org/10.1007/BF02505026

Schwartz, D. L. (1995). The emergence of abstract representations in dyad problem solving. *The Journal of the Learning Sciences, 4*(3), 321-354. https://doi.org/10.1207/s15327809jls0403_3

Sheehan, K. J., Pila, S., Lauricella, A. R., & Wartella, E. A. (2019). Parent-child interaction and children's learning from a coding application. *Computers & Education, 140*, 103601. https://doi.org/10.1016/j.compedu.2019.103601

Schoenfeld, A. H. (1992). Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. In *NCTM Handbook of research on mathematics teaching and* learning (pp. 334-370). NCTM. https://doi.org/10.1177/002205741619600202

Stevens, R., & Bransford, J. (2007). The LIFE Center's lifelong and lifewide diagram. In J. A. Banks (Ed.), *Learning in and out of school in diverse environments: Life-long, life-wide, life-deep*. University of Washington Center for Multicultural Education.

Strouse, G. A., Troseth, G. L., O'Doherty, K. D., & Saylor, M. M. (2018). Co-viewing supports toddlers' word learning from contingent and noncontingent video. *Journal of Experimental Child Psychology*, *166*, 310-326. https://doi.org/10.1016/j.jecp.2017.09.005.

Sung, W. & Black, J. (2021). Factors to consider when designing effective learning: Infusing computational thinking in mathematics to support thinking-doing. *Journal of Research on Technology in Education, 53*(4), 404-426. https://doi.org/10.1080/15391523.2020.1784066

Vygotsky, L. S., & Cole, M. (1978). *Mind in society: Development of higher psychological processes*. Harvard University Press. https://doi.org/10.2307/j.ctvjf9vz4

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147. https://doi.org/10.1007/s10956-015-9581-5

Wenger, E. (2010). Communities of practice and social learning systems: the career of a concept. In: C. Blackmore (Eds.), *Social Learning Systems and Communities of Practice* (pp. 179-198). Springer. https://doi.org/10.1007/978-1-84996-133-2_11

Wing, J. (2011). Research notebook: Computational thinking—What and why. *The link magazine*, *6*, 20-23.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry, 17*(2), 89-100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Wyeth, P., & Diercke, C. (2006). Designing cultural probes for children. In *Proceedings of the 18th Australia* conference on *Computer-Human Interaction: Design: Activities, Artefacts and Environments* (pp. 385-388). https://doi.org/10.1145/1228175.1228252

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education, 141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Appendix. In this appendix we present an excerpt of the booklet. The excerpt is a translated version of task 1 from Danish to English.

Task 1

It is time for you to create your first program.

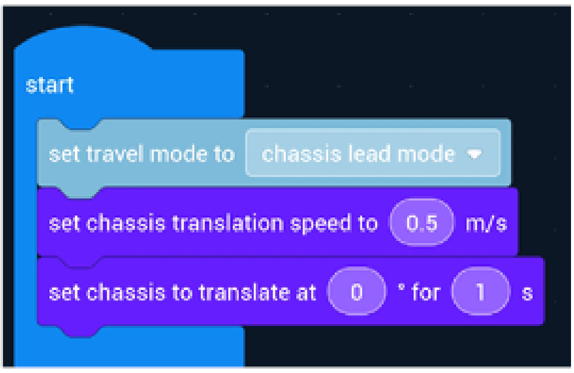
Remember to start the videocamera and press Go Record on the iPad.

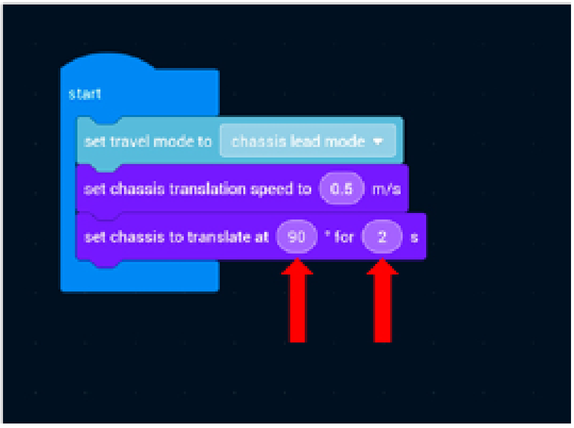
First copy the example from the picture to the right by pulling the code blocks into the start block. you find the codeblocks under the tab named Chasis

Pres the play-button in up in the right corner. If you have copied the code correct Robomaster should move half a meter forward.

Take some time on discussing the code you just entered before moving on to the next step.

Now try to change the two parameters as shown on the picture to the right. Do you understand what is going on? Experiment with the code and try different values.





Let us walk through the code together. on line 1 we set the gimpal to follow the chassis. You can also choose to let the chassis follow the gimpel (gimbal lead) or you can choose to let them move independent of each other (free mode). For now we want the gimpal to follow the chasis.

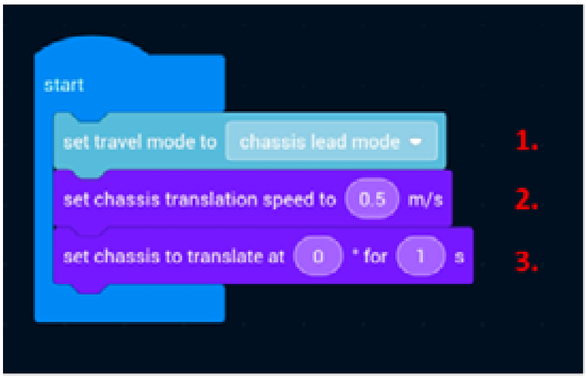
In line 2 we set the speed of the Robomaster. Here we choose 0.5 m/s.

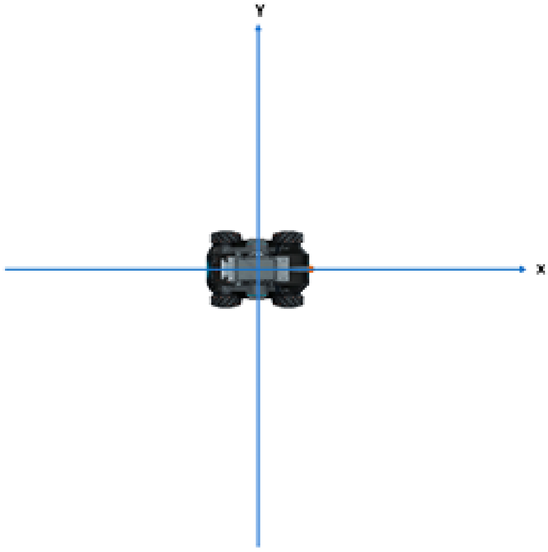
In line 3 we command Robomaster to move in a given direction for a given duration using degrees and seconds as input.

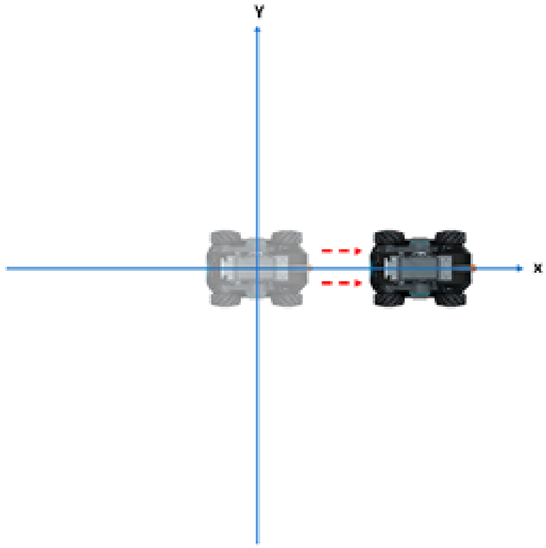
Perhaps you worked or heard about translation in math?

To understand translation we have placed Robomaster in a coordinate system where the gimpal and the front of robomaster follow the X-axis.

When we type the command in line 3 we tell Robomaster to translate along the x-axis (0º) for 1 second with the speed we determined in line 2.







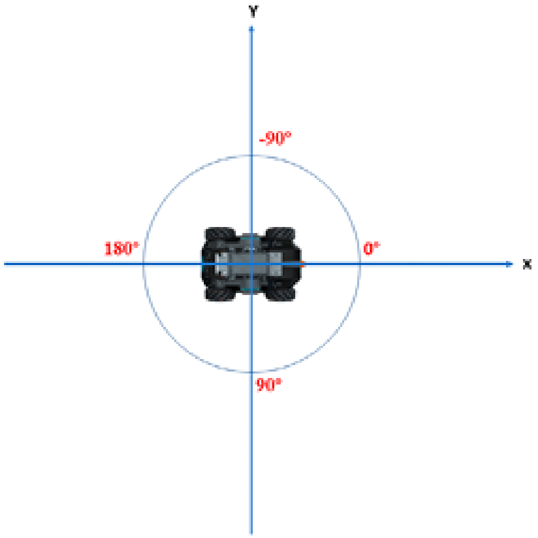
4

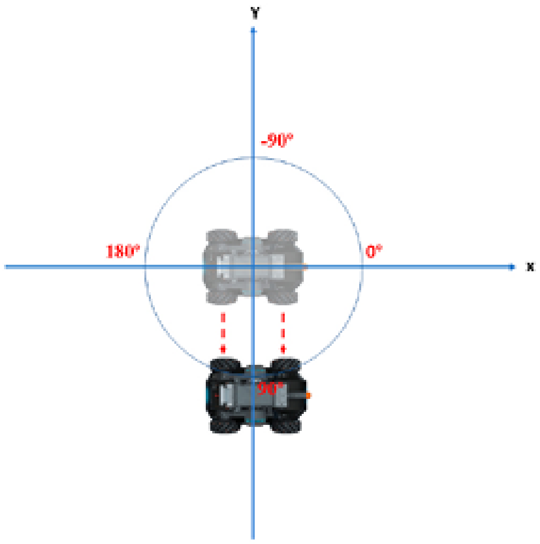
We can command Robomaster to move in any direction by changing the degrees. As you may be aware a circle is 360º. If you turn 180º, you have turned half of a circle an you now fase in the oposite direction. When programming Robomaster, you work with two semicircles. If you want Robomaster to move to the right you use 90º as the direction parameter as indicated on picture 2.

Try it yourself.

Now you can try to make it move backwards and to the left.

Can you guess what direction it will move if you tell it to translate in the direction - 135 º?



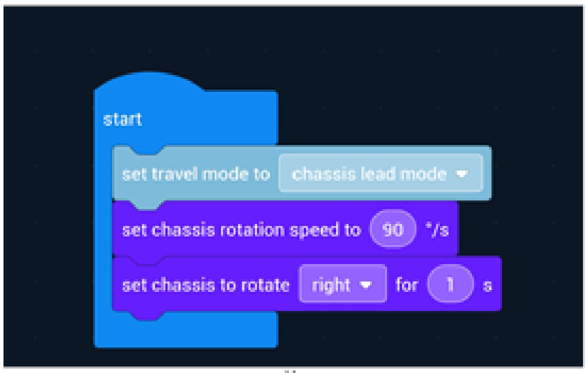


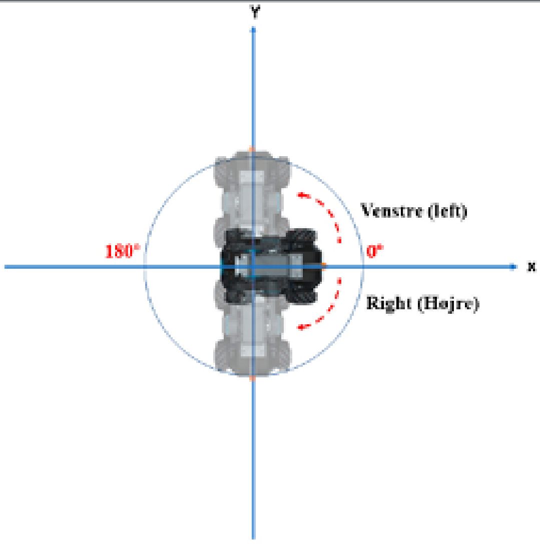
Similar to translate you can also rotate the chassis.

take a look at the example to the right. As with the previous example to make the chassis rotate you first have to input the rotation speed (this time in º/s) and next if it should rotate left or right and for how long.

In this example it rotate to the right for 1 second. As it rotate with a speed 90º/s for 1 second it rotate 90º

The program is again illustrated in a coordinate system, indicating how Robomaster can rotate either to the right or left, where 180º corresponds to a half revolution.





Try to create the example above yourself. Then, try changing the values.

As the final step in task 1, try to make Robomaster first rotate and then move forward. Experiment by combining the code blocks from the previous examples

If you need help there is an example of how to solve the task on the next page.

When you are done please open the envelope and answer the questions when you are done with task 1.