Exploring programming didactics in primary school - a gender perspective

Anna Åkerfeldt
Department of Teaching and Learning
Stockholm university
Stockholm Sweden
anna.akerfeldt@su.se

Susanne Kjällander
Department of child and youth studies
Stockholm university
Stockholm, Sweden
susanne.kjallander@buv.su.se

Linda Mannila
Department of Computer and Information Science
Linköping university
Linköping, Sweden
linda.mannila@liu.se

Fredrik Heintz
Department of Computer and Information Science
Linköping university
Linköping, Sweden
fredrik.heintz@liu.se

***Abstract*— This Research Full Paper explore inclusion in programming in primary school. Education plays a crucial role in engaging a diverse group of students with different social backgrounds and interests. Therefore, this study aims to shed light upon inclusion in programming in primary school, focusing on gender to increase the knowledge regarding inclusion in programming didactics. The following research questions have guided the study: How are programming activities designed in primary school? How do pupils approach the programming tasks given? Can any gender differences be observed, and what are the consequences for the teaching practice? The theoretical framework used to analyse the empirical material is at the intersection between multimodal social semiotics [1] and a design-oriented perspective [2]. The empirical material consists of classroom video observations. Programming lessons in grades 4-8 have been observed and videos was recorded during 2019-2020. The pupils have worked on eight different programming tasks during the lessons. Analysis of these programming activities (tasks, instructions and resources used) focusing on gender has been made. Findings show two aspects 1) interest and position and 2) representations of knowledge. Regarding interest and position, the study of programming activities shows both similarities and differences between girls' and boys' approach to the task. Similarities are shown regarding the learning activities. No differences in coding strategies or creativity are observed if the task has an open design. The differences are shown in the guided tasks, where boys tend to engage in the tasks from their interests rather than following instructions and girls tend to follow the instructions given by the teacher. From a gender perspective, the boys might find programming more creative and fun, and the girls might feel less engaged as their interest falls into the background. Secondly, knowledge representations might affect who is seen as an expert within the CS field. For example, in grades 4 and 5, a male voice was represented in the video clips and a guest teacher used when presenting programming activities. The resources used in the lessons can be seen as representations of knowledge. In this case, they are always connected to a social and cultural domain [3], an environment foremost represented by males in this case.**

Keywords—education, primary school, programming, didactic, gender

I. Introduction

In today's society, we use digital technologies daily to communicate with friends and family, do our work, buy groceries, and do our banking. Even applying for a visa or passport can be done through digital systems. These applications are often designed and programmed by a homogenous group of people, resulting in people with different cultural backgrounds and experiences not being represented and can impact the design choices made in the production of the application. For example, Buolamwini and Gebru [4] investigated commercial facial recognition applications. They found that females with darker skin were the most misclassified group with an error rate of 34,7%, while for lighter-skinned males it was 0,8%. Furthermore, AI systems, such as health care used to detect skin cancer, are being used and are nearly as good as an expert in the field. The dataset used to train the machine to recognise skin variation that can be a sign of skin cancer lacks a dataset with labels for various skin types, such as hairy skin, thickness, and colour [4]. The examples mentioned above need to be addressed from different aspects to secure and work towards making systems that consider a diverse group of people. One way to address these issues is to interest a diverse group of individuals to apply for computer science. In 2014 England introduced 'computing' as mandatory in their national curriculum. The subject computing includes; computer science, information technology and digital literacy. Sentance [5] argues that the new curriculum and its requirements "... challenge stereotypes around who can study computing." (p.1). Education plays a crucial role to engage a diversity of pupils to become interested in and proceed with a career in CS. This study aims to explore and shed light upon inclusion in programming didactics in primary school to increase the knowledge regarding inclusion. The following research questions have guided the study: How are programming activities designed in primary school? How do the students approach the programming tasks given? Can any gender differences be observed, and what are the consequences for the practice?

II. Literature Review

In the U.S. culture, females are negatively stereotyped as having lower abilities in mathematics and science than males [6]. Females also have a feeling of not belonging in a field that males dominate because of, for example, norms, values, and structures [6]. In a masculine culture, women have difficulties seeing themselves and taking place in such a culture and therefore choose other fields to proceed with their careers [6]. Master and Meltzoff [7] state that "… two stereotypes are intertwined: (a) a "cultural fit" stereotype (the belief that "math is for boys") and (b) an "ability" stereotype (the belief that boys have more ability to do STEM problem-solving than girls)". [7] p. 216. Therefore, there might be different social expectations for girls and boys regarding CS [7]. Sultan et al. [8] investigated the relationship between technologies and women in the research literature. Findings showed that a common narrative in the research literature when describing the relation was adjectives such as "reluctant", "insufficient", "discouraged" and "lacking" [8] p. 236. In the broader sense, programming, and the interest in STEM subjects (science, technology, engineering, and mathematics) have traditionally been dominated by males. Male domination has changed during the last decades in science, especially in biology. However, in technology and especially in computer science, females are still underrepresented [6] [9] [10]. A report from Catalyst shows that women earned 18,7 % of bachelor's degrees in the USA during the years 2015-2016. Further, a recent study by Wang, Stanovsky Weihs and Etzioni [11] concerning gender trends in the computer science literature shows a similar pattern. An analysis of 2.87 million papers from conferences and journals in computer science showed that the gender gap if the same pace continued and was projected in the study, will be closed in the year 2137 [11]. Overall, this means that technology, especially computer science, is dominated by males in higher education, in the workforce, and the academic field.

A. *Primary school – Gender and programming*

Many countries have recently revised the curriculums to meet new demands of the fast development and use of digital technologies in society. The research field about teaching and learning programming in primary school is an emerging field. However, there are still few studies that address issues related to gender and teaching and learning programming. This section will present previous studies addressing gender issues and programming. Kalelioglu [12] studied 10-year-olds and their reflective thinking skills towards problem-solving in educational robotics. Their focus was to study gender differences. Findings showed no differences in females' and males' problem-solving reflecting skills. Females completed programming activities more quickly and accurately than male students [12]. Atmatzidou and Demetriadis [13] have done a similar study focusing on two factors, age and gender. They explored the impact of computational thinking activities and if these factors impacted the development of computational thinking skills. Findings show no differences in developing computational thinking skills regarding age or gender. Women in the study expressed themselves better than their male peers when describing algorithms. Previous research shows that girls tend to be more interested in programming if the programming activity involves storytelling [14] [15] [12]. Females are often negatively stereotyped as having low abilities and interest in technology; therefore, there is a gender gap in the STEM areas of science and mathematics. However, this can be changed through female role models and education. Previous research shows that: "The treatment experience significantly increased girls' technology-related motivation (as measured by their interest and self-efficacy) and eliminated the gender gap with boys' motivation—suggesting that providing girls with early experiences can boost their motivation in STEM. These results are encouraging because they suggest that girls' interest in STEM is not set in stone but is malleable and can be changed through interventions." [7] p. 223.

Korucu et al. [16] have studied and compared internet and mobile usage and class and gender to students' computational thinking skills and if skills differ meaningfully. One hundred sixty students participated in the study (aged 11-15). Findings show that computational thinking skills differ from educational level, class level, and mobile technologies possession durations (in years). In terms of the other factors, no differences were observed. Interest [17] and cultural capital [18] have been investigated to understand and broaden participation and change attitudes towards learning programming. Lacheny [18] suggests two strategies to bridge the gap between formal and informal learning settings, community representation and computational integration. Community representation "... enrols cultural capital to highlight students' social identities, histories, and community networks in programming activities…" (p. 176) and computational integration "... locates computing not only in technologies but also in culturally situated practices and local sources of wealth generation, prompting computational tasks to be driven by culture." (p. 176). These two strategies might broaden the participation in CS activities in educational settings, according to Lachney [18].

III. Theoretical Framework

The theoretical framework used to analyse the empirical material is at the crossroad between multimodal social semiotics [1] and a design-oriented perspective [2]. From these perspectives, students' learning processes are continuously situated in a social and cultural context. Learning is seen as a design process of semiotic transformation and formation by the students in different activities [2]. By highlighting and bringing attention to learning as a design process, the attention shifts from learning as a "… context-free and mental collection of 'facts'" [2, p.17] to learning as an act of transforming and forming activities in a social world. Further, a central assumption is that the social world is not solely shaped by verbal or written language but also by a multitude of resources such as images, colours, sounds, etc. In the design process, such as programming activities, the students give shape to their interests by using different resources that have been assigned to be used through teachers' instructions. These resources are representations of knowledge, and they are always connected to a social and cultural domain [19] [2]. In this case, resources that are used for programming are carriers of a set of regulations and constructions of the social world. Different representations of knowledge have various semiotic potentials, leading to the result that various aspects of the content will be highlighted and communicated to the students [3] [20].

IV. Method

A. *Context of the study*

Define The data analysed in this study are part of a larger longitudinal research project - Programming didactics in Sweden. Two Swedish schools (in the article referred to as School S and School B) were selected to participate in the project. Both schools are based in the suburbs of a large city, are organised by the municipality and are so-called unit schools, meaning that students attend the same school from preschool up to ninth grade (age 6-15). School S has approximately 600 students, and School B has approximately 1000 students. The project schools are hence classified as large schools as most schools organised by the municipality in Sweden have around 100-200 students.

The schools were selected as they are at the forefront of implementing and using digital resources for teaching and learning. For instance, all schools have their own digital device, and the schools have been early adopters both considering software and hardware. Since 2011 School S has had one-to-one device. From grade 1, they have iPads, and from grade 7, they are using Chrome Books. Programming and digital competence has also been considered necessary in these schools. For example, in school S, the teachers had begun to teach programming before the revised curriculum came into force. In School B, the school leaders emphasised the importance of teachers' adequate digital competence and provided appropriate professional development activities. Further, School S has also established a group of pupils called Digital learners, where pupils from all grades can apply to be part of the group. The group is seen as a specialist when using technologies and is often used by the teacher as assistants. For example, the digital learners can join in for lessons to help younger classmates learn functions, for example, in Scratch.

In each school, three classes were selected as "project groups" followed for three years. We have followed students starting grade 1 at the beginning of the project throughout grades 1-3, students starting grade 4 through grades 4-6 and those starting grade 7 through grades 7-9. As programming is explicitly mentioned in mathematics and technology in the Swedish curriculum, we have focused our project group's specific study on those subjects. Due to the Covid-19 situation in 2020 and 2021, we had limited possibilities to visit and observe the project classes. Therefore, the project data have been collected from grades 1-2, 4-5 and 7-8. Gender differences become more prominent in the later grades than among younger pupils [21]; therefore, grades 4-8 was chosen (Table I). Fourteen girls and eighteen boys were observed (Table II).

B. *Ethical considerations*

The Swedish Ethical Review Authority has approved the research project. Information about the research project was held during an annual guardian meeting at the beginning of the project, where they had the opportunity to ask questions. Pupils, their guardians, and teachers who participated in the study were informed verbally and written and asked about participation through written informed consent. The pupils could decline to participate in the study when the researcher visited the lesson even if the guardian had consented to the child's participation.

C. *Empirical material*

This study is part of a larger research project called Programming didactics to study teaching and learning programming in Swedish primary schools. A qualitative method was used to get insight into programming lessons and the design of the programming tasks. Video classroom observations have been used to explore inclusion in programming in primary school, focusing on gender. The collection of the empirical material was framed according to the Learning Design Sequence-model (LDS, ”Fig. 1”) [22], [23]. The model is grounded in a design for learning perspective [2]; [24] and is used both to guide the data collection and to guide the analysis to explore teaching and learning in an educational setting, in this case, programming. Lengthwise an activity was one or several lessons: the longest one lasted eight lessons. The focus of the video observations was to follow a programming activity from start to end, from the point where the teacher introduced the theme or activity until it was finished. This article focuses on the primary transformation unit, particularly the pupils' interest and positioning when working with the tasks given. During 2019 - 2020 fourteen Learning Design Sequence (LDS) with students aged 7-14 years were observed. Eight of these LDS will be included in this study to shed light on how the tasks are designed and how pupils approach the task, e.g. design processes. During the lesson, the focus was on one or a group of pupils sitting with each other, working with the tasks given. During the lessons, the researchers followed pupils and walked around and observed pupils' work during the lessons.

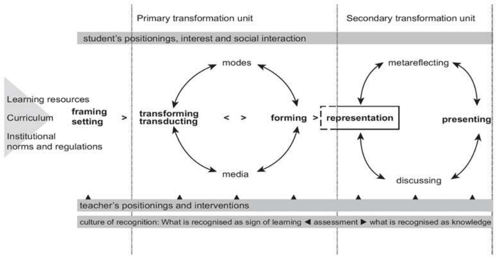

Fig. 1. Learning Design Sequence model, which was used to frame and analyse the videoobservations.

D. *Anlaysis*

The analysis focused on how the pupils approached the programming tasks during the first lesson and how these could be described to illuminate teaching and learning programming challenges from a gender perspective. The pupils' first approach to the task was analysed to shed light on if there were differences in girls' and boys' first encounters and engagement in the task and how they designed their learning paths.

TABLE I. EMPIRICAL MATERIAL

[a] Number of pupils that were observed and talked to during the lesson.

	Task	*Resources used*	*Teacher*	*Cat.*
4:1	Micro:bit	Youtube-clip, presentations (male speaker).	Female	Guided
4:2	Secret message	Whiteboard, instructions on paper	Female and male	Open
5:1	Coordinate system – creating a game	Scratch, website instructions	Male	Guided
5:2	Creating a game	Whiteboard, Tynker	Female	Open
5:3	Creating a math game	Scratch, website instructions	Female	Guided
7:1	Buildings	Instructions in google-docs, www.koda.nu	Female	Open
7:2	Number sequence	Whiteboard, movie-clip (male speaker), Zifro	Female	Guided
8:1	Casino game	Instructions in google-docs Scratch	Male	Open

TABLE II. NUMBERS OF OBSERVED GIRLS AND BOYS IN EACH GRADE

Table Column Head		
Grade	*Girls*	*Boys*
4	4	6
5	4	7
7	6	4
8	0	1
Total:	14	18

V. TASK DESIGN AND THE PUPILS' DESIGN PROCESSES

The eight analysed tasks are divided into 1) guided and 2) open design. Four of the tasks were categorised as closed and four as open (see table 1). Below the tasks and the pupil's design process will be described.

A. Guided design

The tasks that were categorised as guided were designed with clear frames. The frames consisted of step-by-step instructions given by the teacher through a video clip (task 4:1) or learning resources (5:1, 5:3, 7:2). The purpose was often to learn a specific function, software or hardware. It could also be a combination of the above mentioned. For example, in the Micro:bit task, the purpose was to learn how to program a Micro:bit in an application and transfer the program to a Micro:bit.

The tasks Programming a Micro:bit (4:1) and Creating a game with the coordinate system (5:1) in School S had a guided design with limited possibilities for the pupils to improvise and figure out solutions on their own. In both tasks, the pupils followed a step-by-step guide. In task 4:1, the instructions were provided through a video clip, and in 5:1 through a website. In task 4:1 the instruction was provided through a video clip from youtube, and the pupils followed the instructions given via the Micro:bit programming environment on their iPads. The teacher is passive while the clip is playing. The first instruction was to program a smiley, followed by instructions on how to program a text and a flashing heart. During the introduction, the teacher gave a strict order to the pupils that they were not allowed to improvise and do something on their own. A boy and a girl were followed more closely during the lesson, and they both followed the instructions and got the task done quickly. When they had finished the tasks, the boy and the girl started adding and creating patterns to the flashing heart in line with the given task. Another boy (Boy 3) who called for the researchers' attention wanted to show the program he had made. He had programmed a mini story. The following text was displayed on the Micro:bit in the application: "Do not forget to bring an Umbrella" and a symbol for an umbrella and animated raindrops. Boy 3 had redesigned the task and started to combine the instruction by using both the text functions and flashing symbols, in this case, as an animation. He engaged and moved forward from his interest and designed his learning path. The boy and the girl had also redesigned the task by adding patterns to the flashing heart but not as much as Boy 3.

In grade 7, the task (7:2) is done with the application Zifro. The pupils are training pattern recognition and doing this by programming number sequences in Zero. The pupils follow the application's instructions and work their way through. The two girls that are followed work their way through the tasks. An example of the task is: Create a program that writes all odd numbers from 0 to 20. When they are done with the material, they start to work in their math books with the same tasks. In the math game task in grade 5 (5:3), the pupils are instructed by the teacher to focus on adding scores to the math games they are working with. Instructions on how to do that are given through a website that consists of five lessons. Lesson number three contains instructions on how to add a score. Two boys were followed during the lesson. None of the boys that were followed was adding scores to the game. They were occupied working on adding other interactive elements involved in the coming lessons stated on the website instructions. They might have added score functions in a previous stage or in a previous lesson but are not shown during the observations. Towards the end of the lesson, one of the boys adds a score function but does not get it to work. He asked his friend and together they solved the problem but another one arose, and they had no time to finish as the lesson ended.

The Coordinate-system (5:1) task was similar to the micro:bit task. The differences were that Scratch's block-based programming tool and text-based instructions from a website were used instead of video clips. The pupils worked individually with iPads and followed the instructions from the website to complete the assignment. The first task given on the website was to animate a robot that should eat fruits placed on the scene, “Fig. 2”. One girl was followed, and she completed the task given and then started to design a game on her own. The game was to dress a girl. Clothes were placed on the scene and when the user clicked on a garment or an accessory it was placed on a person “Fig. 3”. To program the game, she used the same skills as she had learned through the task, e.g., placing objects on the screen using a coordinate system. Three boys were followed while they created a football game. During the lesson, they got stuck and couldn’t solve how to place an object, in this case, a football, in a specific position on the scene. It is unclear if they had followed the instructions in the first task to animate the robot.

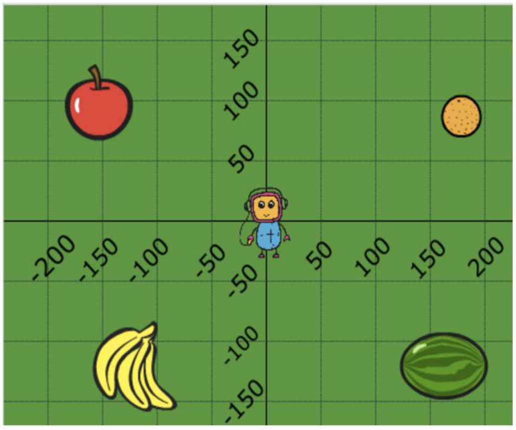

Fig. 2. Screen shot on the coordinate system were the pupils should program the robot to eat the fruits. https://www.kodboken.se/start/kodaiskolan/aventyrxy/uppgifter-i-scratch/folj-frukten?chpt=0

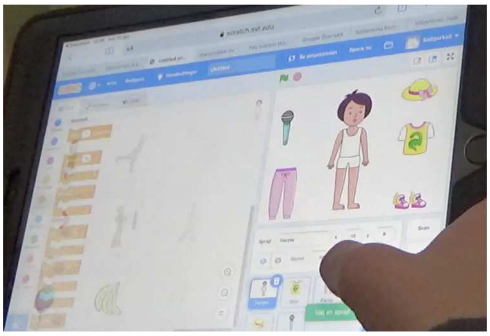

Fig. 3. Screen shot of the cut out doll that one of the girls did after she had finished the Coordinate-system task (5.1).

In both the Micro:bit, Coordinate-system and math game tasks, where boys were observed during the lesson, they tended to engage in the task from their interest and try things out to a greater extent than the girls despite that they were told not to (especially in the Micro:bit task). The girls observed (tasks 4:1 & 5:1) followed the instructions, and after they had completed these, they tried things out and added their interest to the task. In the task Number sequence, the girls that were observed did the task and did not add anything beyond the task given.

To conclude, the guided design left limited possibilities for the pupils to add or engage in the task from their perspective. However, the boys tend to explore and engage in the tasks from their interest to a higher extent despite the limitations and instruction given by the teacher or the resource used.

B. *Open tasks*

Four tasks were categorised as open. In contrast to the guided task, these tasks were designed so the pupils could add their style, touch, and personal interest to their work. The purpose was similar to the guided task, to learn specific functions, software or hardware. The tasks that have been categorised as open were; Secret message (4:2), Creating a game (5:2), Buildings (7:1) and Casino game (8:1). The tasks were communicated through google documents (7:1 & 8:1), teacher instructions during the lesson and paper instructions (4:2). The instructions for task 5:2 was communicated through teachers' instructions and written notes on the whiteboards.

Task 4:2 in grade 4, School B, was to create a secret message using binary code. The task was done with pen and paper, and the instructions were to create a message for a friend. The task was open because the pupils could engage in the task from their interests and write a meaningful message from their perspective. For example, a boy wrote a message directed to a previous shared experience with his friend about a person they had met. Another boy had written "Fortnite" as a secret message, which was a popular online game and had just released a season update.

In grade 5, School B, the task was to create a game in software called Tynker (5:2). The pupils had previously done a game with the software where the user controls a dragon that collects diamonds. The frames that are given by the teacher were that the game must include two figures and animation and if they got time over they can include sound effects. The girl that is followed creates animation and describes the task as "making a movie". Three minutes into the task she adds sound. She adds several characters and changes the graphics and skins of the character. During the lesson, she doesn't add any game elements. Two girls and two boys show their games after the lesson for the researcher. The games had different designs when it comes to gaming elements. One of the girls has done two games with the same concept where the user is prompted to find objects on the screen. The other girl has done a game where an avatar is running from a ghost. If the ghost catches the avatar the game is over. If the avatar manages to reach a point on the screen a next level would appear. One of the boys' games was designed to touch an aeroplane with a finger. The plan was flying across the screen and if it was not touched the game was over. The other boy's game was to avoid meteors by jumping up and down with an avatar. The avatar was designed as a Pokemon which was popular at the time.

In the tasks Buildings (7:1) and Casino game (8:1) the pupils had the opportunity to add and use their own interests in the design of their program. In for example the task Buildings the pupils had the possibility to design their own house and create objects from their perspectives. The task was to write a program that was drawing out geometric forms and use these to design a house. The pupils that were observed used this opportunity to add a personal touch to their buildings as they created for example houses with different styles and colours, they added curtains, round windows, and details such as flowers, trees, animals and pools in the gardens surrounding the house. There were also different styles of the houses such as a barn , "Fig. 4" and a forte "Fig. 5". During the lesson two of the boys were working and experimented with animations and on their screens, no buildings were shown. One was working on trying to get a circle to move back and forth on the screen and the other programmed a word to move randomly across the screen.

The task Casino game was also categorised as an open task although the task was a continuous work on a previous task they had done. The pupils had in a previous work designed a prototype for a casino game to calculate and work with probability theory. The programming task was to redesign and create the game in Scratch. The design of the game was open and the pupils had the possibility to add details based on their personal interests. The boy that was followed used approximately half of the lesson to find graphics for his game. He was doing a fortune wheel and had difficulties finding graphics that he wanted to use.

Fig. 4. Example of a building created by a girl.

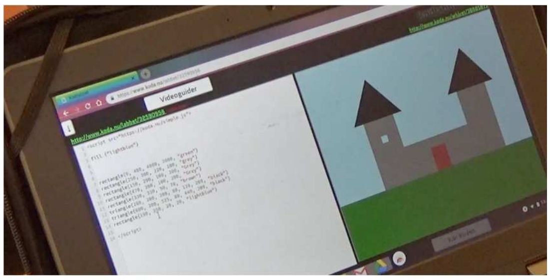

Fig. 5. Examples of a building created by a boy.

VI. Discussion

From the analysis of the eight programming tasks, two aspects are raised and discussed. 1) Interest and positioning 2) Representations of knowledge. Firstly, regarding interest and positioning, it could be interpreted that the boys were more engaged in the sense that they approached the task from their interest even if the task was guided. They designed their learning paths more closely to their interest in both the guided and open design of the task. In the open tasks, no gender differences could be observed. The girls and the boys used their own experiences and interests and incorporated their social world into their programming activities [17]; [18]. Previous research has shown that girls are more engaged if the programming activity involves storytelling [14] [15] [12]. Findings from this study show similar results. Even though the tasks did not explicitly have storytelling elements, the girls tended to frame the task as storytelling. For example, in task 5:1, the girl has framed the task as making a movie. In the same task, a girl created a game where a ghost chased an avatar. Finding in this study shows that an important aspect is to enrol the pupils' interest and their identities and community network in the programming tasks as well as in situated practice, which is put forward by Lachney [18] as strategies to bridge the gap between formal and informal learning settings, community representation and computational integration.

The girls positioned themselves as knowledgeable in both the open and guided tasks. However, the boys are more eager to move forward with the tasks given, and it could be observed that they are positioning themselves as more knowledgeable as they often skip steps in the instructions given. Previous research has shown no gender differences regarding skills; however, according to Korucu et al. [16], differences in educational level, possession of mobile technologies and class level affected programming skills. Pupils' programming skills have not been investigated; however, previous studies show no differences [12; 13], which entails that the girls' motivation and self-confidence in the fields need to be addressed further in education [7].

The starting point and how girls and boys design their learning processes differ. In the guided tasks, the girls that were observed were following the instructions and were not engaging in the task from their interests to the same extent as the boys. The girls follow the instruction linearly, while the boys tend to have a more non-linear way through the instructions given by the teacher or the learning resource. There is one exception: the boy in task 4.1 is following the instruction in a linear matter. From a gender perspective, the risk is that the boys might find programming more creative and fun. The girls might find programming tasks less engaging as their interest comes into play first after completing the task, for example, in the Coordinate task where the girl started to make her game design (the cutout doll) after she had finished the task. The boys started to design a football game straight away.

When pupils worked with the open tasks, these differences did not show. What could be noted is that when the design of the tasks is more open, both girls and boys tend to work more with the graphical design, for example, choosing the right colours or choosing and positioning objects at the scene. It could lead to learning programming skills falling into the background as the pupils are focused on other aspects of their designs instead of training their skills. In the open tasks, it could also be more difficult for the teacher to support the pupils if they get stuck as the pupils are working on a range of different activities. For example, in the casino game task, the pupils designed a variation of casino games where the teacher had less control over the lesson and what could unfold. Therefore it might be more challenging to teach when the task has an open design. However, it tends to engage the girls to a greater extent.

The tasks that are categorised as open had elements of boundaries. One example is the task "Secret message", where the tasks are done together in the class. The pupils cannot move ahead and complete the task on an individual level before their classmates are ready, leading to demotivation towards the task. The pupils in the observed situation (four boys) are patiently waiting for the next step sitting still and awaiting new instructions. Most of the guided and open tasks were done individually, even though the pupils helped each other during the lessons. Another example is the task Casino game which was based on a previous task where the pupils had worked with probability theory through the design of a casino game. However, in the tasks that were categorised as open, there was a possibility for the pupils to add and design their work from their interests.

Findings in the study show that the resources used are foremost represented by males (through video clips used and the male guest teacher) even though the teachers are mostly females. Males represented the movie clip that was shown in the classroom. The guest teacher in grade 4 was also a male, while the pupils' ordinary teacher was a female (see table 1). The resources that are used in the lessons can be seen as representations of knowledge, and they are always connected to a social and cultural domain [3]. By using the video clip in the way that was done, the teacher resigns from being a knowledgeable person within the field, which will not be, from a gender perspective, fruitful if we want to broaden who has access to a career in computer science. Suggesting that providing girls with early experiences can boost their motivation in STEM. Female role models need to be present in programming activities both in class and in resources used, as previous studies show that it could increase girls' technology-related motivation [7].

A. *Limitations of the study*

This study is a qualitative study and has observed a limited number of pupils. Findings and conclusions need to be seen in this light. The gender differences that have been highlighted could be related to individual differences rather than gender differences which needs to be further investigated. However, findings from this study are similar to previous studies [7; 17;18], and the issue is an urgent matter for education and society to address if we are striving for a diverse group of people in the area of computer science. Recent research by Master et al. [25] demonstrates that girls are less interested than boys in computer science as early as six years old, which may contribute to gender disparities. In turn, this entails that more research needs to be done in the lower grades to increase the knowledge about gender disparities and bridge the gap.

B. *Conclusions and implications for practice*

This study contributes to primary education and how to plan and design tasks with a focus on gender inclusion. The teacher has an essential role in encouraging, positioning and engaging girls to believe that they have a future career as computer scientists. Findings from this study indicated that boys approach the tasks from their interest and position themselves as programmers regardless of the design of the task. On the other hand, girls follow the instructions and approach the guided task linearly and do not incorporate their interests and seem to position themselves as less engaged. However, when the tasks are designed as open, the girls approach their tasks from their interests and seem more engaged. According to Selander and Kress [1; 2; 3; 23], pupils' learning processes are continuously situated in a social and cultural context. In this study, learning is seen as a design process of semiotic transformation and formation by the pupils in different activities [2]. Designing the tasks as open where the pupils learning as seen as an act of transforming and forming activities in a social world seems to be a way forward for girls to be more engaged, and in turn, it might lead to an increased interest for girls in the CS field.

ACKNOWLEDGMENT

The research project funded by Marcus and Amalia Wallenberg Foundation.

REFERENCES

[1] Kress, G., *Multimodality: A Social Semiotic Approach to Contemporary Communication*, Taylor & Francis, 2010.

[2] Selander, S. Designs for Learning. *Designs for Learning*. 1, 1, 10–22, 2008.

[3] Kress, G. and Selander, S. "Multimodal design, learning and cultures of recognition," *Internet Higher. Education*. 15, 4, 265–268, 2012.

[4] Buolamwini, J. and Gebru, T. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," *Proc. Mach. Learn. Res*. 81, 1889–1896. 2018.

[5] Sentance, S. "Moving to mainstream: Developing computing for all," *ACM Int. Conf. Proceeding Ser.*, 1–2. 2019.

[6] Cheryan, S., Ziegler, S.A., Montoya, A.K. and Jiang, L. "Why are some STEM fields more gender-balanced than others?" Psychol. Bull. 143, 1, 1–35. 2017.

[7] Master A. and Meltzoff, A.N. 2016. "Building bridges between psychological science and education: Cultural stereotypes, STEM, and equity," *Prospects 46*, 2, 215–234. 2016.

[8] Sultan, U., Axell, C., Hallström, J. Seery, N., Buckley, J., Canty, D. and Phelan J. "Girls' Engagement in Technology Education: A Systematic Review of the Literature," Retrieved December 12, 2018 from http://urn.kb.se/, 2018.

[9] Hill, C., Corbett, C. and Rose, A. "Why So Few? Women in Science, Technology, *Engineering, and Mathematics*," Retrieved April 24, 2020 from www.aauw.org, 2020.

[10] Catalyst. "Women in Science, Technology, Engineering, and Mathematics (STEM): Quick Take", Catalyst. Retrieved April 24, 2020 from https://www.catalyst.org/research/women-in-science-technology-engineering-and-mathematics-stem/, 2019

[11] Wang, L.L., Stanovsky, G., Weihs, L. and Etzioni, O. "Gender trends in computer science authorship." *Communication of the ACM, 64, 3, 78-84*, 2021.

[12] Kalelioglu, F. 2015. "A new way of teaching programming skills to K-12 students," *Code.org. Comput. Hum. Behav. J.* 52, 200–210, 2015

[13] Atmatzidou, S. and Demetriadis, S. "Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences," *Rob. Auton. Syst.* 75, 661–670, 2016.

[14] Kelleher, C. and Pausch, R. "Using Storytelling to Motivate Programming," *Commun. ACM* 50, 7 (2007), 59–64. Retrieved from http://cacm.acm.org/magazines/2007/7/5615-using-storytelling-to-motivate-programming/fulltext, 2007.

[15] Denner, J., Werner, L. and Ortiz, E. 2012. "Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?" *Comput. Educ*. 58, 1, 240–249, 2012

[16] Korucu, A.T, Gencturk, A.T. and Gundogdu, M.M. "Examination of the Computational Thinking Skills of Students," *Journal of Learning and Teaching in Digital Age*, 2017.

[17] Moyer, L., Klopfer, M., and Ernst, J. V. "Bridging the arts and computer science: Engaging at-risk students through the integration of music," *Technology and Engineering Teacher,* 77(6), 8-12, 2018.

[18] Lachney. M. "Computational communities: African-American cultural capital in computer science education," *Computer Science Education*, 27, 4, 175–196, 2017

[19] Bezemer, J. and Kress, G. "Writing in multimodal texts: A social semiotic account of designs for learning," *Writ. Commun*. 25, 2, 2008.

[20] Svärdemo Åberg, E. and Åkerfeldt, A. 2017. "Design and recognition of multimodal texts: selection of digital tools and modes on the basis of social and material premises?" *J. Comput. Educ*. 4, 3, 283–306, 2017.

[21] Sullivan, A. and Bers, M. U. "The impact of teacher gender on girls' performance on programming tasks in early elementary school," *Journal of Information Technology Education. Innovations in Practice.* 17, (2018), 153–162, 2018.

[22] Selander, S. *Didactics after Vygotskij. Designs for Learning*. [Didaktiken efter Vygotskij. Design för lärande], Liber. 2017.

[23] Selander, S. "Designs in and for learning—a theoretical framework," *Designs for Research, Teaching and Learning. A framework for future education*. Björklund Boistrup, L. & Selander, S. (Eds.). Routledge. https://library.oapen.org/handle/20.500.12657/51655, 2022.

[24] Åkerfeldt, A. *Didactic design with digital resources: A study of representations of knowledge in a digitalised school.* [Didaktisk design med digitala resurser En studie av kunskapsrepresentationer i en digitaliserad skola] Diss. Stockholm : Stockholms universitet, Stockholm, 2014.

[25] Master, A., Meltzoff, A.N. and Cheryan, S. "Gender stereotypes about interests start early and cause gender disparities in computer science and engineering," *Proceedings of the National Academy of Sciences,* vol. 118, no. 48, p. e2100030118, 2021.