University of Bergen

Department of Linguistic, Literary and Aesthetic Studies

DIKULT350

Master's Thesis in Digital Culture

Fall 2017

Programming in School

An insight to the Norwegian programming pilot and the inclusion/exclusion of girls in computer programming education

Fay Pedersen Tveranger

Sammendrag

Pilotprosjektet i programmering som ble igangsatt høsten 2016, og inkluderte i starten 146 norske skoler. Det nye prosjektet har blant annet ledet til debatter om temaer som lærerkompetanse, inkludering av jenter og prosjektets egentlige relevans. En intensjon med dette prosjektet er å ta et dypdykk i programmerings piloten for å undersøke hva elever og lærere tenker om prosjektet, og programmering som et fag i den norske skolen. En annen intensjon er å eksaminere jenters posisjon i forbindelse med pilotprosjektet, og programmering og databruk.

Studien fokuserer på å knytte pilotprosjektet og inkludering av jenter opp mot historiske kontekster og teori innen innføring av programmering i skolesystemet og kjønnede posisjoner innen databruk, og er en kvalitativ studie med deltakere involvert i programmeringspiloten. Resultatene viser til en positiv holdning fra både lærere og elever i forbindelse med implementering av programmering i undervisningen. I kontekst til ulikheter i kjønnsfordeling, viser studien til at det ikke tyder på at elevene har noen betydelig formening om at biologisk kjønn er en direkte årsak til ujevn kjønnsfordeling, mens det heller er individuelle interesser som ofte er det som danner grunnlaget for at programmering ikke blir valgt av jenter.

Acknowledgements

I would like to express sincere gratitude to all the supportive and insightful people that have participated in the development of this project. First, I would like to thank my supervisor Hilde G. Corneliussen (Associate Professor in Digital Culture at UiB and Senior Researcher at Vestlandsforsking), and co-supervisors Mary Flanagan (Fairchild Distinguished Professor of Digital Humanities at Dartmouth College) and Jill Walker Rettberg (Associate Professor in Digital Culture at UiB). Thank you all for giving me useful and professional guidance, for bearing with me, and for keeping my spirits high across multiple time zones.

Finally, I must thank everybody else that has contributed in some way or another in finishing this thesis; Ingar Mikal Halse, Vibeke Pedersen, Camilla Kottum Elmar and Tjerand Silde. Sincerely, thank you.

Bergen, September 2017
Fay Tveranger

Table of Content

1 Introduction

1.1 Project background: The programming pilot and the inclusion of girls

From the fall semester of 2016, a number of Norwegian secondary schools added computer programming as a course to their curriculum. The course is not mandatory, but a part of a pilot project that will run for three years, and is also open to every school who wishes to implement the elective. Programming in Norwegian schools has been debated, and one heavily weighed argument *for*, voiced by *Digitutvalget*, is that there is a real concern for the population's lack of digital competence and schools not sufficiently facilitate for education on how digital technology works (NOU2013:2). The minister of education in Norway, Thorbjørn Røe Isaksen, claims ”an important goal for the implementation of coding as an elective is to increase the students' interest in technology, natural sciences and mathematics" ("Koding blir valgfag på 146 skoler" 2016). However, there is little to no mention of measures to facilitate for inclusion of girls, a heavily underrepresented group in ICTs, in any of the official reports issued by the Ministry of Education and Research themselves. An instance that however does seem concerned by gender disparities is Senter for IKT i Utdanningen, a sub department of the Ministry of Education and Research. In a news article from June 2016, Jon Haavie, Roger Antonsen and Torgeir Waterhouse, supporters of code clubs and programming in school, ask the minister of education to "Look to UK, Røe Isaksen", where they already have taken such measures, and, according to the authors, much higher ambitions concerning programming than Norway (Haavie, Antonsen, and Waterhouse 2016).

Through the three year-period 146 schools distributed amongst 53 counties are participating in the pilot. The Norwegian government has set aside 15 million kroner to the project, but not every school participating gets financial support, nor do the schools that are implementing the elective outside of the pilot (ibid). In addition to not funding all of the schools that have the programming elective, there are not any supported research projects to study the effect of the programming pilot.

This study has, in my opinion, both a scientific, personal and a socially useful purpose. The pilot project is scientifically interesting because it is, so far, untapped territory. There have been conducted research studies on programming in other countries, but in Norway these are however limited. The project is socially useful because it concerns a new part of the national curriculum, and the primary education of children. The personal stakes lies close to the societal ones, in the notion that I, as a female in ICT's want to make inquiries about the current state of women and girls, in particular, in ICT education.

This thesis has thus two main objectives: to examine the newly implemented programming elective and how the teacher and the pupils have embraced the new subject. Some of the pupils had experience with programming, using the tool *Scratch*, but most, including the teacher, was not familiar with programming prior to enrolling into the elective course. The second objective is to investigate which methods are used to involve more girls in the programming elective. The gender distribution is here, as in many sectors of ICTs – both professional and educational, very uneven, and this thesis aim to investigate the current status of gender disparities, and present measures to inclusion.

Shortly after the pilot project was implemented, I contacted one of the schools participating in the Bergen area. I have been following and observing the class of eight graders as they have been introduced to the world of programming. Since the programming pilot's implementation, different discussions have surfaced, ranging from a call for better education for the teachers teaching the elective, to the pedagogy and what to be taught in these classes. There have however been few calls to make inquiries about, or even out, gender disparities.

The elective has been an independent subject, but Røe Isakesen claims it will be tightly related to another ongoing project on strengthening competence in STEM

(Science, Technology, Engineering and Mathematics)-fields: “The project concerning programming as an elective has to be seen in the context of the new STEM-strategy [..]. It is an underlying strategy for mathematics, natural sciences and technology in schools and kindergartens from 2015-2019” (“Koding blir valgfag på ungdomsskolen” 2015). The strategy, however, contains no mentions of programming at all. Still, there are some mentions of technology, but mostly in the context of mathematics and natural sciences (Kunnskapsdepartementet 2015). This is contributing to a view of programming as a subject that is dependent on being a part of the existing curriculum, and not as a stand-alone subject, and thus not as important as other subjects.

1.2 Digital competence

There is a need for defining the term digital competence. The term has been used to describe the practical use of digital tools and critical evaluation of digital texts (NOU2015:8). However, there is no mention of *production* or *design* when defining the term (ibid), and the term seems to be confused by another term, *digital literacy*.

Digital competence is understood to be a central concept in education, but there are still calls for enhancing the overall digital competence in the Norwegian population: “If digital competence is to be enhanced in the population, the foundation has to be lain in school” (Corneliussen and Prøitz 2015, p. 23). In the community of education politics in Norway, digital competence is juxtaposed with expressing oneself orally, reading, writing, and doing mathematics, and is characterized as ‘the 5th basic skill’ by The Norwegian Directorate for Education and Training (“Rammeverk for grunnleggende ferdigheter” 2015).

There are a lot of intersection concepts when it comes to digital learning. Two key concepts are *digital literacy* and *digital competence*. Digital literacy involves accessing digital media and ICT and critical evaluation of digital media and media content, while digital competence involves use and employment:

> [..] involves the confident and critical use of ICT for employment, learning,

> self-development and participation in society. This broad definition of digital competence provides the necessary context (i.e. the knowledge, skills and attitudes) for working, living and learning in the knowledge society (Ala-Mutka, Punie, and Redecker 2008, p. 4).

The need for digital competence is essential in most areas in day-to-day life and in the work place. Hence, understanding how the digital tools that we use on an every-day basis actually works, in addition to learning the basic principles in programming, is the focus of the pilot project. This can range from how Big Data is collected and used in commercial settings, or recommendation services in applications such as Netflix or Spotify, or how our choices, search terms, location and settings affect the output result in social media or search engines. As argued by José van Dijck, as software increasingly structures the world: "it also withdraws, and it becomes harder and harder for us to focus on it as it is embedded, hidden, off-shored and merely forgotten about" (van Dijck 2013, p. 29; Berry 2011). This notion suggests that the consumers must be more aware of the hidden layers in web applications, games, and other digital technology, thus learning to code is a way in this direction.

Today, most of us are consumers of digital technology, therefore, it is essential that we start to facilitate for ICT education as early as possible, according to The Norwegian Directorate for Education and Training ("Forsøkslæreplan i valgfag programmering," n.d.).

On these premises, The Norwegian Directorate for Education and Training shaped a tentative plan for the purposes of the programming elective with the following goals divided by two categories, *modelling* and *coding* (ibid):

Modelling

The goal for the education is that the pupil should be able to:

- know how computers and computer programs work, including a selection of widely used programming languages and their use

- make tasks into concrete sub-tasks, evaluate which sub-tasks are solvable digitally, and create solutions for these
- document and explain code by writing appropriate comments and by presenting their own and other's code

Coding

The goal for the education is that the pupil should be able to:

- use multiple programming languages where at least one is text-based
- use basic principles in programming, such as loops, tests, variables, functions and simple user interaction
- develop and de-bug programs that solve defined tasks, including mathematical problems and the controlling or simulation of physical objects
- transfer solutions to new tasks by generalizing and modifying existing code and algorithms

By moving away from the conception of digital competence as a tool exclusively purposed for the use of digital media and analysing digital texts, room is made for programming and otherwise the *production* of digital texts to be included as well. This thesis is dependent on this understanding of the term, and in turn supports the arguments that the programming pilot is an interesting and important project for Norway and the Norwegian schools.

1.3 Research questions

My research takes a multi-disciplinary course; as it is an analysis of the interviews conducted and observations from the classroom, theories from gender studies, digital culture and technology history. I use these approaches and methodologies to answer the following the research questions:

- ✓ **How have the pupils and teachers appropriated the programming elective?**

One of the purposes of this thesis is to study the Norwegian programming pilot as a whole, with a focus on the pupils and the teachers involved. So now, when the programming pilot is running, how do the teachers and the pupils cope with the new subject, taking into account that some of the participants did not have any experience with programming beforehand?

- ✓ **In which ways does the Norwegian programming pilot include or exclude girls from the programming elective?**

I also intend to investigate which means have been put to work to include an underrepresented group in programming, and ICTs in general, girls, and why it is problematic for the programming pilot that ICTs are perceived as *gendered*. Perhaps implementing programming at the earliest stage as possible in school can prevent challenging preconceptions to form?

With my background in new media studies and digital culture, I intend to approach these topics with a constructivist approach in relation to both gender- and technology studies. I will also look to historical context on gender in relation to technology, and to the situation of programming education in European countries, but my main goal is to investigate the Norwegian initiatives, and in which ways they deem digital competence, and more central in this project, programming an important part of education.

1.4 Structure

In the second chapter, I will present the theories that are applied to the analysis of data and how I take gender and technology into account. The chapter will examine theories of social constructivism in general, and in the context of gender and technology. The third chapter looks into women in early computer history and current problematic pop cultural phenomena. The fourth chapter will investigate previous research related to topics on gender in relation to technology, programming in school, and pedagogy in the Norwegian school system. The fifth chapter focuses on my methodology and how I have approached my own research. I will elaborate on how and why I have chosen the participants, which methods used and ethical concerns when conducting research. Here, I will also study the main source for my data: the semi-structured interviews, observations and e-mail interviews. In the sixth chapter, I will analyse the interviews and observations and discuss the data. The discussion and analysis of the interviews is divided into two parts: insights from the pupils, and then insights from the teacher. In both parts, the statements are categorized by topics or *codes*, which is a strategy used in grounded theory. The seventh chapter will provide with key findings and recommendations for future research.

2 Theoretical aspects

The theories presented here will provide a foundation for the analysis of the programming pilot and inclusion of girls in programming education. The analysis will utilize the research questions as points of references, and the theories chosen will make up the foundation of how the collected data is investigated.

A key aspect of this thesis is to investigate how girls are included in programming education in the Norwegian pilot project. Therefore, there is a need to take a closer look into what gender is and what it means in the context of computers. The approach to gender and technology chosen in this study has roots in social constructivism, where gender and technology both are seen as social phenomena, and in turn, this particular theoretical lens was chosen as a way to demonstrate how cultural symbols saturate gender and technology, and gender in relation to technology.

2.1 Social constructivism

Argued by Søndegaard, social constructivism bears much similarity with socio-cultural thinking as they both operate in the field of the individual and culture, and the mutual development that occurs between the two of them (Søndergaard, 2006, p. 35). While socio-culturalism works to define some main traits in the development between the two, social constructivism points to language as a mediator in the process (ibid). The idea behind social constructivism was developed as a sociological tool by, amongst others, sociologists Peter Berger and Thomas Luckmann (ibid). The views on social construction by Berger and Luckmann have been adopted in many fields of the humanities and social sciences, including *Science, Technology and Society* (STS), an interdisciplinary field (Lynch 2016, p. 101-102) which will be elaborated on later in this chapter. Their fundamental view is that

> [..] stable social institutions emerge from highly flexible possibilities at the individual and interactional level, which become externalized and objectivated, eventually being taken for granted for realities (ibid).

According to social constructivist thinking, social understandings and categories (e.g. sex and gender, emotionality, rationality, and identity) are all social products, and the way the world is understood is then a social product. It is therefore, according to social constructivists, interesting for scientific research also to study the underlying social, moral, political and economic institutions that work as backdrops for and by the assumptions we make of human behaviour (Søndergaard 2006, p. 36).

2.1.1 Gender is social

First, a look at what is implied by gender in this thesis. The English language operates with two categories of gender, where *sex* refers to biology, while *gender* is often seen as the social construction of sex (Corneliussen, 2002, p. 57). Whereas in the Norwegian language we have only one word that condenses the terms sex and gender – 'kjønn', which in turn require a specification whether one refers to 'kjønn' in a social or biological context.

As cited by Joshua Meyrowitz, Simone de Beuvoir suggests that the initial difference in behaviour between genders starts when boys are thrust into the outside world of men, while the girls are left at home (Meyrowitz 1986, p. 202). In other words, as boys grow older, they get a tough start in life, while girls are allowed to stay behind, in the comfort of the home, leading to boys rejecting all "feminine" aspects of themselves, at the same time at girls are shielded from any potential "masculine" traits (ibid, p. 204).

Gayle Rubin, for example, utilizes psychoanalysis to offer a theory on how gender identity is formed, while Joan W. Scott proposes a broader view that is combining Rubin's theory and three other aspects (Scott 1986, p. 1067). This view involves four elements: cultural symbols, normative concepts, kinship and subjective identity, which conclude that gender is a product of power relations:

> Gender is a constitutive element of social relationships based on perceived differences between sexes, and gender is a primary way of signifying relationships of power (ibid).

Søndergaard asks these questions about gender:

> Is gender something that is inside us or outside us or between us? Is it something stable that changes its expression, or perhaps is it an expression, which stabilizes us? (Søndergaard 2006, p. 9).

The need for labelling the world has led to many aspects of life to be *gendered*. Fields of work and education, such as ICTs, has bore the cultural sign of masculinity, while work and education in the health care-sector, *femininity*. Gender labels and gendered institutions appear to be important to stabilize society as a way to keep it organized, and as Scott suggests, a way to signify relationships of power (ibid).

As a counterpart to the constructivist approach, there are deterministic views that lean toward a biological approach in the context of gender differences. This is often called evolutionary theories. In these approaches, gender differences are approached with men and women's reproductive natures in mind (Lippa 2010, p. 1099). In a study done by Richard A. Lippa, professor of psychology, on differences in personality and interest between men and women, women are categorized as more 'people-oriented', while men are more 'thing-oriented', and the differences vary due to difference in cultures (ibid).

> Gender differences in personality tend to be larger in gender-egalitarian societies than in gender-inegalitarian societies, a finding that contradicts social role theory but is consistent with evolutionary, attributional, and social comparison theories. In contrast, gender differences in interests appear to be consistent across cultures and over time, a finding that suggests possible biologic influences (ibid).

While being careful not to disregard biological gender differences entirely, a social constructivist approach benefits the argument that the absence of women in ICTs is not biological, but a product of social structure. Women and girls have not always have been absent in the field, but as the culture changes, so does discourse and social symbols.

2.1.2 And so are science and technology

Gender in this thesis is treated as a social construction, and the same applies to science and technology. According to Wenda Bauchpies, Jennifer Croissant and Sal Restivo humans are fundamentally social, therefore everything human-made is a part of the social construction we live in, ‘a web of social relationships’ (Bauchpies, Croissant, and Restivo 2006, p. 2). As Bauchpies et al. strive to learn what purpose, and symbolic meaning science and technology have in our society, I strive in this thesis to study the same, but in the context of the programming pilot.

As described by Bauchpies et al. in *Science, Technology, and Society* (2006), STS gathers concepts of history, philosophy and sociology of science and technology and makes up a hybrid discipline, aiming to understand the products of science and technology (Bauchpies, Croissant, and Restivo, 2006, vii). The field has been attempted concretized, but always ends up as an interdisciplinary field with a central dogma: technosciences are social and cultural phenomena (ibid). The main idea is that science and technology are products of social and cultural constructions, and use tools from both social sciences as well as humanities to analyse and understand these fields. Technosciences are the grey areas, the ‘messiness’, and the ‘ambiguities’, and when cleared away, the ‘clean’ product is revealed; science and technology (ibid, p. 7-8). To reach the objective and pureness of science and technology, one must erase instances such as the people involved in developing the science and technology, and hence, instances such as gender, class and ethnicity is made invisible.

As software studies experts have continually emphasized the social and cultural importance of coding technologies, José van Dijck contends to the notion of technology as social construct, and presents this explanation of the term “platform”, which can be understood in the context of games, social media, and other digitally mediated platforms:

> Platforms are computational and architectural concepts, but can also be understood figuratively, in a sociocultural and political sense, as political stages and performative infrastructures (van Dijck 2013, p. 29; Gillespie 2010).

An opposing theory to this constructivist way of thinking is technological determinism. Explained by Sally Wyatt, technological determinism is a way to understand technological inventions, where they are perceived to act as both the determinants and stepping stones of human development (Wyatt 1996, p. 169). “Technological determinism is imbued with the notion that technological progress equals social progress” (ibid, p. 168). In other words, according to this notion, we must simply adapt to the technology and what it require from us. This way of viewing technology indeed mirrors the call for more education in digital literacy and –technology, and the call for more producers instead of passive consumers, or at Wyatt puts it:

> [..] technological determinism [..] leaves no space for human choice or intervention and, moreover, absolves us from responsibility for the technologies we make and use (ibid, p. 169).

A critique of this particular view of science as an object unaffected by social construction, and gender in particular, comes from feminist thinker Sandra Harding (Cornelissen 2011; Harding 1986). Harding, as other feminist thinkers, uses gender symbolism and construction of individual gender in her critique on how science is conceptualized.

As Bauchpies et al. , Wyatt and Harding, I will apply a constructivist way of looking at science and technology to understand how the pupils appropriate coding and programming and the exclusion and inclusion of girls in this context. By studying technology as a social construction, I aim to open ”[..] up the pathways to new ways of looking and to understanding how knowledge and difference are constructed, applied, and maintained” (Bauchpies, Croissant, and

Restivo 2006, p. 32). As Bauchpies et al. aim to be careful about blindly accepting who designs, who controls, who uses, and who benefits from production, distribution, and consumption of technologies (ibid, p. 10-11), I will be careful about accepting the state of digital literacy in Norwegian schools, the pilot project, and the inclusion of girls.

3 Women and computers

To get an overview of the state of women in ICTs today, I believe it is necessary to investigate earlier computer history. In this chapter I will first present some of the most important women in computer history, and lastly, I will investigate current pop-cultural phenomena that illustrate a shift in how gender is perceived in relation to the computer. This last part dips into the relationship between STEM-fields (science, technology, engineering and math) and computer gaming, but this relationship will be further explained in the next chapter.

3.1 Women in early computing

Women in early computer science history are not frequently mentioned in historic texts, which lead to the expectation that women never played an important role in computer technology development from its early days (Gürer 1995, p. 175). In her text about *Pioneer Women in Computer Science* (1995), Denise Gürer explains that when women in early computer programming were asked how they were treated by their peers, most replied that the treatment and respect they received were the same as for men (Gürer 1995, p. 176). Women in those days were actually categorized as ideal programmers because it required traits as being patient, persistent and having an eye for detail (ibid). Kathleen McNulty, one of the first programmers of the Electronic Numerical Integrator (ENIAC), stated:

> The girls were told that only men could get professional ratings. The time came later in World War II when no more men were available, and women were pushed into supervisory positions. Finally, in November 1946, many of the women received professional ratings (Gürer 1995, p. 177).

It wasn't until years later that the field of computer science became less than ideal for women. Karen A. Frenkel, science and technology journalist, painted a not so pretty picture of an American graduate school from the early 1980s:

“women describe experiences of invisibility, patronizing behaviour, doubted qualifications, and so on” (Frenkel 1990, p. 37). A cause of the paradigm shift in how the field was perceived, Gürer claims, was when the male hierarchy business structure of companies grew in size (Gürer 1995, p. 177). Another theory, pointed out by programmer Judy Clapp, suggests that

> It had all to do with expectations. At that time, working women were expected to be nurses or schoolteachers. Thus, to be given the chance to work in a technical field was a great opportunity. However, upon closer inspection, almost all the leaders and managers were men (Gürer 1995, p. 177).

3.1.1 Role models

There have in fact been some strong female role models in computing. In 1843, mathematician Ada Lovelace published a descriptive article on the first account of a prototype computer, *Sketch of the Analytical Engine Invented by Charles Babbage* (Holmes 2015). Though the computer was never built, her project made a significant impact to the modern notion of computing (Montfort et al. 2013, p. 129). The works of Ada Lovelace has been met with praise, but are rarely mentioned in history. According to Shortt, there are still some that claim that Lovelace did not write her mathematical publishing (Shortt 1998; Coyle 1996). In writings about Lovelace there is a noticeable trend of describing her by characteristics like “The Enchantress of numbers” or “The Enchantress of Abstraction”. As for other women in early computer sciences, they are often reduced to their gender or that they are housewives. This can also be reflected to Alan Turing, who was at some point reduced to his sexual orientation.

After the unveiling of ENIAC in 1946, designed by Presper Eckert and John Mauchly, a group of six women were appointed to program the computer (Gürer, 1995, p. 177). The “computers”, Kathleen McNulty, Frances Bilas, Elizabeth Jean Jennings, Frances Elizabeth Snyder, Ruth Lichterman and Marilyn Wescoff, hence became the body and brains behind programming the world’s first electronic general-purpose computer.

Admiral Grace Hopper, of the US Navy, is also a noteworthy person in computing history to be acquainted with. In the post-war years Hopper was, amongst other significant developments, known for developing the FLOW-MATIC programming language, as the only implemented business data processing language at that time (Gürer 1995, p. 176). In her own words, she was "the third programmer on the first large-scale digital computer" (ibid).

3.2 Problematic pop-culture – Barbie and GamerGate

As will be elaborated on in the next chapter, there has been a call for diversity in the technology industry. Recently, Google was caught in a controversy regarding views of gender in relation to software engineering, making gender in relation to the computer an on-going debate (Hossenfelder 2017). "[..] the current representation of women, underprivileged and disabled people, and other minorities, is smaller than it would be in an ideal world, which we don't live in" (ibid). In the after-math of the controversy, many voiced their opinions on related matters, and Mary Flanagan, professor in Digital Humanities at Dartmouth College, wrote this:

> I started my own publishing house when game publishers—even of old-fashioned board games—wouldn't publish my game, because it was too "feminine" and "activist"—assumptions not based on playing the game itself, but talking to the inventor. Women leaders in the games and tech space are often forced outside established venues and do it on their own. Heck, it was even suggested that I change my name to a man's name to be more competitive on paper (Flanagan 2017).

So what can we do about it? Sabine Hossenfelder, a woman working in theoretical physics, argue that educating people about biases, removing obstacles to education, and the changing of societal gender images is a means to even out gender disparities in ICTs, however slowly.

Technologies that are developed by a consistently similar demographic has made it difficult for women and other underrepresented groups to find their voice in the commercial sphere (Flanagan 2013, p. 224). With (perhaps) noble intentions, concepts such as *pink* and *casual games*, and the cultural phenomena GamerGate have proved to be part of a problematic pop-culture.

Mattel, the company behind many girls' favourite childhood toy, has been contributing to both reinforce misconception about women in ICTs and to improve a more diverse understanding of women in the field. In 1996, *Barbie Fashion Designer*, a so-called *pink* game, became the most successful game of that year, and proved the existence of a market tailored for girls (Kafai et al. 2008, p. xi). However, most of the games of this time all promoted the traditional values of femininity, and played on girls' interest in their appearance (ibid, p. xv).

As we are well aware of, Barbie has had a *lot* of jobs. She is, amongst other occupations, a veterinarian, a chef, a fitness instructor, and in 2010 she also became a certified computer engineer. The book *Barbie: I can be a Computer Engineer*, supplementing the new release, however, proved otherwise. Barbie has a new project coming up, a computer game. As it turns out, Barbie does not know what she is doing, since she really is in charge of the design ideas, and ends up giving her little sister Skipper a computer virus and has to ask her two male friends to help out with her mess (Romano 2014). As a response to the patronizing portrayal of Barbie, Kathleen Tuite launched the website *Feminist Hacker Barbie*, where the users can re-write the story.

To retaliate for their 2010-misstep, Mattel released in 2016 *Game Developer Barbie*. Contrary to *Computer Engineer Barbie*, whose computer was bright pink, matching her shoes and glasses, and was coding in binary code, *Game Developer Barbie* has now more realistic and appropriate equipment. She now has neutral coloured clothes, bright red hair and a computer that shows javascript.

Fig. 1: Casey Fiesler, *Computer Engineer & Game Developer Barbies*. 2016. Digital Image. Available from Flickr, https://www.flickr.com/photos/cfiesler/27426208252/ (accessed August 16, 2017)

Advocates for a more diverse gaming industry have been under fire on several occasions, and especially the GamerGate controversy has sparked discussions around the roles and representation of women in the industry (Kafai et al. 2016, p. 13). GamerGate is a harassment-movement targeting female developers and people that are vocal and in supporting of a more diverse game development industry (Nakamura 2016, p. 35). The movement is the re-surfacing of an incident that took place in 2012, where media critic Anita Sarkeesian became the victim of cyber harassment and an attempted DDOS attack on her website after the launch of a Kickstarter campaign for her video series (Kafai et al. 2016, p. 23). The movement was first recognized as GamerGate in 2014 after allegations against game developer Zoe Quinn, accusing her of trading sexual favours for positive press for her recently released game *Depression Quest* (ibid).

This kind of movement impose an unhealthy view of the game development industry, and maintain or increase the lack of diversity in gaming and game development, technology- and computer studies and jobs, when the message received is that there is no room for girls or people that does not “fit” in the culture imposed by the GamerGaters. Therefore, it is important to look to the history of the computer when women were more prominent in computer sciences, even though not very visible at the time.

Being aware of the important women’s impact on computer history might lighten the conception of programming being inherently a masculine trade. “Computing historians have suggested that programming started out, if not as feminine work, then at least as ambivalent and clearly open to women in the 1950s and early 1960s ” (Corneliussen and Prøitz 2016, p. 99; Light 1999). A lack of role models, or relatable people, might enhance the feeling of ‘ambient belonging’, leaving girls feeling out of place (Corneliussen and Prøitz 2015, p. 22). Debated by Corneliussen and Prøitz: if we consider programming an essential skill, a way to cope with the gender disparities, the solution is perhaps bringing programming into the schools (ibid, p. 21). This argument enhances the pilot project’s position of importance in dealing with the uneven gender distribution. However, none of the official documents, such as the proposed curriculum (Utdanningsdirektoratet, n.d.), or the official statements from minister of education, Thorbjørn Røe Isaksen, make any mention of the importance of inclusion. Reading the newly issued strategy for ICT education in Norwegian schools, some mentions are made regarding inclusion of minority pupils, however in the context of language, and not gender (Kunnskapsdepartementet 2017).

4 Research on gender in relation to the computer, and pedagogy and programming in school

In the previous chapter, I demonstrated historic lines as to illustrate how women participating in early computing were regarded, and then compared history to the more recent cultural signs and stereotypes that has emerged around the computer context. The focuses in this thesis are to make inquiries about the gender imbalance in computing and the adaption of programming in school. To conduct this study, and be able to make any statements about the areas of focus, I have had to take several approaches. These include gender research, mostly in regard to gender perception and the perception of women in technological contexts, and children and programming in school and tools for learning programming.

Even though there is a wide array of literature concerning these topics, I have chosen to narrow it down to what I think will enhance the analysis, discussion and points of views presented in this project. I will start with gender in relation to the computer, and then move towards programming in school in Europe, and then to the Norwegian context and pedagogy used in the Norwegian school system. Finally, I will take look at how the programming pilot project is incorporated into the existing curriculum.

4.1 Gender in relation to the computer – why are the girls missing?

As visited in the theory chapter, gender has been used as a mean to categorize people, actors and entities we surround ourselves with. Søndergaard points to language as a factor in the mutual development that occurs between the individual and the culture (Søndergaard, 2006, p. 35), and this can be connected to why the institution of the computer, and computer activities often are perceived masculine. Science- and otherwise objective language has been defined as a masculine language (Turkle & Papert, 1990, p. 150). In computation and programming terminology, aggressive terms are used, like “the computer has crashed” or to “kill”, “abort” or “execute” a task (ibid). In 1990, Sherry Turkle and Seymour Papert discussed, amongst other things, how men and women approach computers and programming. They argue that while women feel more comfortable with a more relational, interactive, and connected approach to computers, men prefer a more distanced style (ibid p. 151). The cultural construction of the computer is connoted to aggression, domination and competition, which lead to a conflict on how women appropriate technology (ibid p. 150). Turkle and Papert thus argue that women already in the 1970s and 1980s were faced to bargain with the cultural associations of computer technology or with the cultural constructions of being a woman (ibid p. 151). In other words, women were, and still are faced with 'ambient belonging', a term used in environments where conceptions or stereotypes tied maleness or masculinity lead to women or girls feeling out of place (Corneliussen and Prøitz 2015, p. 30; Cheryan et al. 2009).

The perspectives on male and female computer users which illustrate that women prefer a more 'organic' approach, while men a more clinical, 'soft' vs. 'hard', Turkle ties to the upbringing of boys and girls (Corneliussen 2002, p. 26). As cited by Corneliussen, Turkle argue, seemingly based on a general acceptance in regards to gender that:

> In our culture girls are taught the characteristics of soft mastery – negotiation, compromise, give-and-take – as psychological virtues, while models of male behaviour stress decisiveness and the imposition of will (ibid).

This approach to the computer does however not resonate with the perception of the stereotypical hacker. The hacker is positioned as a strong, masculine character in computer culture and has an intimate or more personal relationship to his or her computer and code (ibid, p. 27). Argued by Wendy Faulkner, identifying with the hacker has also been seen as off-putting to women, considering the perceived asocial nature, when being social is believed to be a fundamental element of being a woman (Faulkner 2009, p. 172).

In a study done on after school code clubs in rural villages in Norway, Hilde

Corneliussen and Lin Prøitz ask if these clubs can be an arena to increase girls' interest in computers and digital competence, or another boy's club where girls will feel left out (Corneliussen and Prøitz 2016). Examining literature on the history of women in computing, Corneliussen and Prøitz could state that there "is no obvious, natural or biological necessity in women being a minority in computing or in computing being associated with masculinity." (ibid, p. 7). Contending to this, Faulkner argues that:

> [..] there are frequently mismatches between such stereotyped images and actual people and practices. When we look at actual people and practices, much of the apparent non/congruence between gender and engineering identities disappears: we find that engineering practice is profoundly heterogeneous (simultaneously 'social' and 'technical'), for example, and that men and women engineers alike have reasonable people skills (Faulkner 2009, p. 172)

Instead, Corneliussen and Prøitz encourage a social constructivist view on gender and computing, and discourage the discourse that has made the computer, and thus programming, gender-specific. So, why are there still a lack of women in computing?

4.2 Girls, games, and science and technology

Even though women have indeed made significant contributions in computer history, they are rarely mentioned. Corneliussen explains that some of the reason why early technology research has not included women, is that "where there are no women, there is no gender", and technology is a field where women have been relatively absent (Corneliussen 2002, p. 12). On the other hand, gender can be too visible where gender bias is present, and in some cases research has been affected by assumptions of women (Bauchpies, Croissant, and Restivo 2006, p. 27).

Lagesen, as cited by Corneliussen & Prøitz, claims that the proportion of men and women in a certain field reflect its symbolic meaning (Corneliussen and Prøitz 2016; Lagesen 2007). Thus, technology has been somewhat 'invisible' to feminist researchers, and gender 'invisible' to technology researchers (Corneliussen 2002, p. 12). What has been problematized in later feminist research has been connected to the absence of female affiliation to technology, where technology has come to be a symbol of masculinity, a direct contradiction to femininity (ibid). In this part of the chapter, I aim to make inquiries about the state of women in ICTs, and how computer game culture has played a central role in the construction of the computer as masculine.

4.2.1 A call to diversify the computer

Looking back to the 1980s and 1990s, the STEM-fields became more gendered, and women's participation declined (Kafai et al. 2016, p. 5). The gender differences were seen as early as in elementary schools (ibid, p. 5; Margolis & Fischer, 2003; Misa, 2010; Provenzo, 1991). The low proportion of women in these fields has been proposed to be a result and effect of stereotypes that girls and women can't identify with (Corneliussen & Prøitz 2016, p. 7), but the stereotypes, cultural images and masculine perception of the computer has not always been a part of the discourse, as you will discover later in this chapter.

As digital game play has been associated with the courses of STEM-fields, the call to get more women and girls in the STEM-fields became a goal (Kafai et al. 2016, p. 5; Cassell & Jenkins, 1998b; Kiesler, Sproull & Eccles, 1985). According to Yasmin Kafai et al., research has shown that playing video games can increase interest in STEM majors, therefore they argue that there are value in doing research on links between playing games and interest in STEM (ibid, p. 10). Henry Jenkins and Justine Cassell termed the discussion around gaming- and technology cultures "waves of feminism and games" (ibid). Trying to uncover the gender disparities in computer and video games, the research have been parted into three waves. The first wave looked into differences in game play and skill or interest that differed across gender (ibid, p. 5; Greenfield, 1994; Morlock, Yando, & Nigolean, 1985; Okagaki & Frensch, 1994; Subrahmanyam & Greenfield, 1994). The second wave of research emphasized understanding sociocultural context and the experience of women who play and participate in gaming (ibid, p. 6; Jensen & de Castell, 2010; Taylor, 2006; Taylor, 2008).

The most current research, the third wave, heads towards a more nuanced understanding. A key term here is intersectionality. Bringing intersectionality into game and technology research has opened the discussion even more, and the call for diversity encompasses an even broader group of underrepresented communities, because it covers concepts like sexuality, ethnicity, and class (Kafai et al. 2016, p. 6). The research also includes how we define and study masculinity, and tries to make the field aware of the assumptions we make around concepts like gender, gender identity, sexuality, and questions the performativity of gender (Kafai et al. 2016, p. 6).

As the third wave embodies a wider section of unrepresented groups, it becomes more apparent that not only women and girls are underrepresented in the computer and game development communities. Especially the commercial technology companies do not focus on women or people of colour as their target users (Flanagan 2013, p. 224). Thus, commercial games often portray women as characters of little agency, or existing in the game-space merely as prospective sexual partners or target for violence (Flanagan and Kaufman 2016, p. 219), which in turn can be a facilitator for implicit biases.

Implicit biases, whether in gaming culture and STEM-fields, is believed to originate from a number of sources, which are for example believed to be:

> "[..] Repeated exposure to stereotypical depictions or prejudicial attitudes toward individuals of another race in one's personal life or in the broader culture or media" (ibid, p. 222; Dasgupta 2013).

Flanagan and Kaufman point toward these effective psychological mechanisms for managing implicit biases (ibid, p. 223-224):

1. **Counter-stereotypical Examples**. Exposure to numerous counter-stereotypical role models (e.g., having girls read about successful female scientists and engineers) helps counteract the abundance of stereotypical exemplars who occupy existing mental representations of a group or category (e.g., Dasgupta & Asgari, 2004).
2. **Counter-Stereotype Training.** Counter-stereotypic training (e.g., having individuals press a button labeled "No" every time a picture of a group member paired with a stereotypical trait—such as an image of a female paired with the word "weak"—appeared on a computer screen) reduced the automatic activation of the stereotype (Stout et al. 2011; Forbes & Schmader, 2010).
3. **Perspective-Taking.** Guided perspective-taking activities and exercises that encourage simulating the experience of "outgroup" members (e.g., imagining a "day in the life" of a member of another race) effectively reduce biases and stereotypes, in part by forging greater psychological overlap between one's representation of "self" and "other" (Kaufman & Libby, 2012; Todd, Bodenhausen, Richeson & Galinsky, 2011).
4. **Social Norms.** The fostering of egalitarian norms and motivations, when activated, can counteract or prevent the automatic activation of implicit negative stereotypes (Moskowitz, Gollwitzer, Wasel & Schaal, 1999).

Flanagan argues that the computer is more than a tool – it is a portal to digital culture, which can enhance the technological literacy and competence to disadvantaged groups (ibid). The latest report on Norwegian immigrants' digital competence from Statistics Norway stated that over fifty per cent of the immigrants have a low level of digital aptitude (Guthu and Holm 2010). But what is more disturbing is that this report was conducted in 2010, over seven years ago. After digging deeper into their archive and statistics, I could not find anything else on immigrants', second-generation immigrants' or multi-cultural Norwegians' digital competence. If the research on multi-cultural Norwegians' digital competence is this limited, we do not have the best tools at our reach to facilitate for appropriate education in ICTs. Hence, there is a long way to go considering inclusion of underrepresented groups in computing, in all senses.

4.3 How do we make the girls visible?

Programming in school and after-school code clubs is not an entirely new phenomenon, but the focus on girls and the importance of digital competence has increased over the past few years. As Lecia Barker and William Aspray put it, the schools own curriculum is central in what knowledge both pupils and their guardians believe to be important and interesting (Barker and Aspray, 2006, p. 16). The absence of computing as a formal program sends an inherent message about it being non-essential, but available to those with the inclination or predisposition for it (ibid). Perhaps, presenting programming as a mainstream subject such as mathematics and natural sciences will make it more accessible to everybody. Schools all over Europe have already started this effort.

According to a survey from *EurActiv*, an independent media platform specializing in articles on European policymaking, 15 countries in the EU have integrated coding in their school curriculum; Austria, Bulgaria, the Czech Republic, Denmark, Estonia, France, Hungary, Ireland, Lithuania, Malta, Spain, Poland, Portugal, Slovakia and England ("Coding at School — How Do EU Countries Compare?" 2015). Out of these, Estonia, France, Spain, Slovakia; and England have already integrated the subject at primary school level (ibid).

Twelve countries have already or will integrate coding and programming at upper secondary school level. These are: Austria, Bulgaria, Denmark, Estonia, France, Hungary, Lithuania, Malta, Poland, Slovakia, Spain and England (ibid). And now Norway, with their national pilot project in programming.

While in some European countries programming is an optional part of the curriculum, Finland has made it mandatory. Starting from first grade in primary school, the Finnish national curriculum has taken coding and programming farther than any other European country, according to the Learning Environments research group at Alto University ("Finland Is a Pioneer in Teaching Coding at Schools in Europe" 2015).

With the final version published in 2013, a new national curriculum was introduced in England and was fully implemented in September 2014 (Berry 2013). The new subject was called "computing", replacing ICT. The aim is to widen the field to include more computational thinking and practical programming skills, but overall, the subject has not changed much other than by name.

All in all, it seems as many of these initiatives have the same aim as the one in Norway. England's '*Computing in the national curriculum*' (2013), a guide for primary school teachers, seems to put some concern into gender and inclusion, and a focus on pupils with disabilities, trying to make the course accessible and facilitate inclusion of pupils who do not have English as their first language (Berry 2013). In addition to this, the curriculum does better at showing a broader application of IT, in a future work place and in teaching, by presenting different tools (ibid).

Even though programming is not yet a part of the formal nation-wide curriculum in Norway, programming has been a part of some children's learning environment for some time. A study done in Norway from 2015, states that the first wave of code clubs appeared in some cities as early as 2013 (Corneliussen & Prøitz 2016). The study evolved around an after-school code club in rural Norway and with 30 boys and only 7-8 girls registered, the male to female ratio was quite uneven (ibid). Questions regarding *when* gendered stereotypes of computer related activities, and at what age does children get a sense of a field being either masculine or feminine, arise. Research has found that these kinds of stereotypes are less prominent among girls in primary school (Corneliussen & Prøitz 2016; Cheryan, Plaut, Handron, & Hudson, 2013), and suggests the importance of introducing the field of computing before they adopt stereotypes that promotes computing as "not for girls" or otherwise boring or unfitting (ibid).

The gender disparities in the after-school code club are however difficult to criticize as to statements such as: it is "just a club", or "for fun" (Corneliussen and Prøitz 2015, p. 21). However, now that programming is available as an elective subject in all Norwegian schools, there is more room for taking a critical look at gender gaps. In addition to this, the programming pilot, and those in charge of teaching programming, is also faced with other challenges such as the pedagogy applied and how to transition programming into the curriculum so that the elective fulfil the same requirements as other subjects in the Norwegian school system do.

4.4 Pedagogy and official notes from the Norwegian Government

In 2013, three years before the start of the programming pilot, Digitutvalget expressed discontent to the lack of education in digital technology and essentially, how the technology works (NOU2013:2). Looking to earlier education in digital technologies in Norwegian schools, EDB (elektronisk databehandling), or electronic computing, which included introductory programming education, was offered as an elective in the 1980s at secondary levels, and as a course programme in high school (ibid, p. 105). Later on, ICTs, however not programming, was offered as programme subject to *studieforberedende* (ibid), which in Norway is the programme for general studies in high school. ICTs education is also intertwined in the high school programme, *media and communication*.

In November 2016, Senter for IKT i Utdanningen, *The Norwegian Centre for ICT Education*, published a note on why programming and digital literacy as a course in schools are important. Their key idea is that the understanding of underlying processes and systems, logic thinking and abilities to create and produce always has been important in learning and education (Sevik et al., 2016, p. 6). As for the already integrated subjects, this also relates to technology and digital skills and competencies needed in our digitally and technologically rich society. New and ever evolving technology poses new ways of thinking, and challenges the educational system to rethink which competences are important to teach and learn.

The NOU2015:8-report written by *Ludvigsen-utvalget*, an assembly appointed by the Norwegian Government, looked into the renewal of subjects and future skills and competences, and propose four skillsets to be implemented in the curriculum, and into all subjects: subject specific competence, competence in learning, competence in communication, interaction and participation, and competence in exploration and creation (NOU2015:8). As for all of the other subjects, these areas of competence are also tied to digital technology learning and understanding. While programming is not specifically mentioned in this report, there is an understanding that digital literacy is imperative to understand different types of text, evaluate the credibility of sources, and to mediate communication by considering the recipient and the purpose for the communication (ibid). Even though the four competences encourage participation and creation, there are not anything that points to practical education in programming, only the understanding and mastery of digital tools and technology.

4.4.1 Subject specific competence

As was mentioned above, programming is not directly tied to the existing subjects in primary and secondary schools in Norway. Only in higher education does programming appear as dedicated courses, which in turn require subject specific competence (eg. logic, algorithm and syntax). In primary- and secondary schools today, subject specific competence is tied to more central areas as mathematics, natural sciences, language and aesthetic subjects (Sevik et al., 2016, p. 12). The recommendations proposed by Ludvigsen-utvalget aspire to create closer links between the existing subjects to digital tools, but Sanne et al. claims that the use of digital tools in these subjects are rather scarce in practice (Sanne et al., 2016 p. 60).

Even though there is some mention of the use of subject specific digital tools as graphic calculators and spreadsheet software such as Excel (Sanne et al., 2016 p. 61), there is no mention of the common denominators between mathematics and programming. In their note '*Programmering i Skolen'* (*Programming in School*) (2016), Senter for IKT i Utdanningen claims that programming require many of the same skills that are tied to the subject specific competence in mathematics and natural sciences (Sevik et al., 2016, p. 12).

4.4.2 Competence in learning

Competence in learning, or *metacognition*, plays an important role in information comprehension, learning and organizing (Flavell 1979) and thus the way pupils comprehend and handle tasks at hand.

> Metacognition is generally understood as the ability to contemplate one's own thinking, to observe oneself when processing cognitive tasks, and to organize the learning and thinking processes involved in these tasks (Seel 2012, p. 2228).

In other words, the pupils themselves have to reflect on which ways they appropriate knowledge best and to use these methods and strategies to promote own knowledge. As self-regulated learning is proposed by Ludvigsen-utvalget as an important skill in all of the subjects taught in school, this can translate to the computational thinking that is promoted in the plan Senter for IKT i Utdanningen has proposed in their online based course ("Programmerings-MOOC" 2016) for the teachers involved in the Norwegian pilot project in programming. Computational thinking involves breaking down big, complex tasks into smaller, more manageable ones (Sevik et al., 2016, p. 13). According to Kafai, computational thinking should be reframed as computational *participation*, both in school and spare time, as children are participating in learning communities that encourage sharing with other pupils or members of said community (Kafai 2016, p. 27).

4.4.3 Interaction and participation

To communicate, interact and participate are important areas of competence in school (Sevik et al., 2016, p. 15), and are also tied to reading, writing and oral communication, in the sense that the pupils should be able to discuss and deliberate on how to solve tasks with other pupils (NOU2015:8).

As software and web development usually done in teams, this particular skill ties programming in school to a more realistic environment one can expect to encounter in a professional setting. Senter for IKT i Utdanningen use pair programming as an example on how programming can be a cooperative activity (Sevik et al., 2016, p. 15). Communication and cooperation facilitate for discussion, and invite reflection and critical thinking upon the task at hand (ibid). Kafai supports this statement by saying that computational thinking is a social activity (Kafai 2016, p. 27). By elaboration and discussion, pupils are better equipped to assess the product, their strategy and why they chose a particular method.

4.4.4 Competence in exploration and creation

As Douglas Rushkoff simply puts it: "Program, or be programmed" (Rushkoff 2011). As we all move towards an increasingly digital reality, it becomes important to not only use programs and software, but also to learn about the underlying mechanics and how to build them. One of the main arguments used actively to promote programming as a part of the Norwegian curriculum, is the fact that we are dependent on people creating, and not just *using* digital tools ("Programmering i skolen" 2016).

Van Dijck proposes in *The Cultural Connectivity* (2013) these key concepts to break down technology to understand its hidden layers, its sociability and its politics; data and metadata, algorithms, protocols, interfaces, and defaults:

> [A platform] shapes the performance of social acts instead of merely facilitating them. Technologically speaking, platforms are the providers of software, (sometimes) hardware, and services that help code social activities into a computational architecture; the process (meta)data through algorithms and formatted protocols before presenting their interpreted logic in a form of user-friendly interfaces with default settings that reflect the platform owner's strategic choices (van Dijck 2013, p. 29).

Even though some consumer are not always aware of the underlying mechanism that is shaping the experience with the technology, van Dijck claims that we are however not uncritical adopters of technology (ibid, p. 32).

The NOU2015-report emphasize that creativity and innovation are important skills that should be taught in schools (NOU2015:8). But how does one teach these skills? The term '*bricolage*', which has its origins from French anthropologist Claude Lévi-Strauss (Sevik et al., 2016, p. 15), appears in Papert's *Mindstorms* (1980) as an alternative way of engaging with and conceptualizing the world around us. Loosely translated by Sevik et al., bricolage is used when solving problems through experimenting and play – learning by doing (Sevik et al., 2016, p. 15). The idea is that programming invites to both creativity, which includes tinkering and improvisation, *as well* as computational thinking, which is a systematic and analytic approach to solving a problem.

Sevik et al. argue that programming can be tied to each of these skills, either as programming as a stand alone subject or interdisciplinary (Sevik et al., 2016, p. 12). In my collected data, I experienced, amongst other things, exactly how the four skills tied effortlessly with the programming education.

4.5 Digital competence and programming in school

In this part of the chapter I make inquiries on how digital competence and programming in school have been dealt with prior to the wave of implementation of programming in schools in Europe and the programming pilot project in Norway.

As cited by Mitchel Resnick et al., Papert argued that

> [..] programming languages should have a "low floor" (easy to get started) and a "high ceiling" (opportunities to create increasingly complex projects over time). In addition, languages need "wide walls" (supporting many different types of projects so people with many different interests and learning styles can all become engaged) (Resnick et al. 2009, p. 63).

Considering that programming languages formerly used in education for young children did not fulfil these utopian expectations, other attempts have been made, and inspired the development of other approaches to introduce children to programming. I will first illustrate early thoughts on the implementation of programming in school and the programming languages used, and then the journey taken to make programming more tangible and less abstract for the younger pupils. A product of this journey is the free of use web application Scratch. This web application and 'programming language' is also the most frequently used tool in the class participating in my project, and by independent after school code clubs in Norway, which is why I have included a brief summary of its development and application in the end of this section.

4.5.1 BASIC

In the late 1970s and 80s, when the first personal computers were introduced, the enthusiasm for teaching children to program grew (ibid, p. 62). Schools started teaching students to write simple programs in LOGO and BASIC, and Papert's *Mindstorms* (1980) presented LOGO as a cornerstone for changing approaches to education and learning (ibid). BASIC, the Beginners All-purpose Symbolic Instruction Code, was developed by the two Dartmouth College professors John Kemeny and Tomas Kurtz (Montfort et al. 2013, p. 148), intended to be used for the Dartmouth Time-Sharing System (DTSS) (Campbell-Kelly and Aspray 2004, p. 187). The main idea of the system was, differing from MIT's equivalent which was primarily purposed for computer scientists, DTSS, and thus BASIC, was intended for a broader spectrum of users (ibid).

Upon releasing a free version of the language in 1964, a widespread adoption of it begun at high school and college levels, and being easy to use led to BASIC's massive popularity at the time (Montfort et al. 2013, p. 148). Furthermore, BASIC was designed to help new programmers, including undergraduates, liberal arts as well as science students, with "helpful" and "friendly" error messages (ibid), as have been an ideology in developing new programming languages purposed for teaching beginners.

BASIC has later gathered criticism, and even been considered harmful, although, argued by Monyfort et al, the criticism was perhaps a tactical move to support other programming languages (ibid, p. 96). In defending the institution of accessible programming languages, Campbell-Kelly and Aspray debate that:

> [..] most critics overlooked BASIC's much more important cultural significance in establishing a user-friendly programming system that enabled ordinary people to use computers without a professional computer programmer as an intermediary (Campbell-Kelly and Aspray 2004, p. 198).

BASIC was for its time the same as Scratch and other accessible programming languages are today: a way to, as argued by Waterhouse, a continuance towards a power-shift from the very few, to the very many (Waterhouse 2015).

Schools soon took another approach to other uses of the computer, and Resnick et al. proposes some key factors as to why the initial introduction to programming to students and pupils stagnated:

- The earliest form for programming languages were too difficult, and the children had **trouble with the language's syntax**;
- Programming was often introduced with activities that **did not engage the children** and;
- It was often introduced in contexts where there was **no one to guide them** when things went wrong **or inspire them** to further explorations if it went right.

Michal Armoni, professor in computer science, claims that the youngest children are unable to understand the level of abstraction needed in coding and programming, as visited by Corneliussen and Prøitz (Corneliussen and Prøitz 2016, p. 98). To enhance the participation in technology, computer sciences and mathematics, Armonis observations is backed by Resnick et al.; it is important to show the pupils that it is not only about numbers, but also about creative thinking and problem solving (ibid). Because of these observations, and the former failed attempts to engage children in programming, Resnick et al. established three core design principles in developing Scratch, as a new way of introducing programming to children: making it more tinkerable, more meaningful, and more social (Resnick et al. 2009, p. 63).

4.5.2 Scratch

Scratch is a programming language designed by the Lifelong Kindergarten group at the MIT Media Lab ("Scratch," n.d.), and was publicly launched in May 2007 (ibid). The language is a so-called block-based language, where the users drag and drop blocks to create a program. The blocks contain the commands, loop- and condition-structures, and logic, which are all structures found in other object oriented programming languages. The web application offers its users a visualization of the input and the output, but does also offer a view into the text-based part of the language.

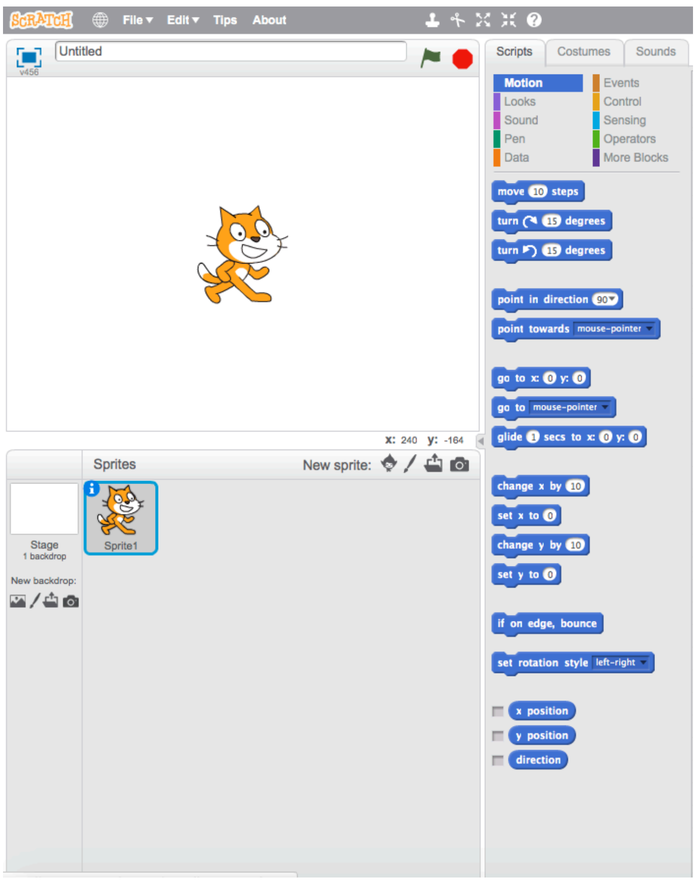

Fig. 2: Scratch. Screenshot taken September 30, 2017.

Its creators argue that Scratch makes programming more meaningful in the sense that it offers *diversity*: it enables the users to create different types of projects such as games, stories, animations and simulations (Resnick et al. 2009, p. 64). It also enables the user to *personalize* their projects by easily importing music, and images to create their own graphic elements (ibid). In addition to diversity and personalization, Scratch offers a social aspect. The community around it makes it easy to collaborate, give each other feedback and share projects.

Besides using Scratch in the typical sense, it can also be used as a more creative way of learning spoken language. The class participating in this project has a majority of pupils with the Norwegian language as their mother tongue. The tasks range from open and creative to very defined tasks with definite outcomes. One of the ways in which the teacher has given the pupils assignments in Scratch, is by giving them a set of tasks, attained from the website kidsakoder.no. The tasks are written in Norwegian, making it easier for the pupils to follow the instructions to complete each task. However, some of the students are not fluent in Norwegian, thus making this kind of assignment a language-lesson as well, and perhaps a bit more fun than a standard Norwegian-lesson.

Forthcoming, created by kidsakoder.no, is the launching of a new and updated version of the programming tasks, with an emphasis on improving teacher guidance (Silde 2017). The new resource will be more interactive and show the user's progress, and provide badges when the user has solved a certain number of tasks or completed a course (ibid). "Teachers sometimes wish to easily find relevant tasks and need tips on how to solve them", therefore they have also implemented filters, where the teacher can declare which subject are being taught and for which grade (ibid). In addition to inviting to more play by gamifying the task-progression, this also facilitates for an easy approach for teachers to implement a segment of programming in every subject, thus opening up for new pedagogical cooperation across subjects.

From the research done in this chapter concerning gender in relation to the computer, women in computer history, and pedagogy used in the Norwegian school system, I am better equipped to understand and discuss the data collected in my own research.

4.5.3 Lobbyist movements for programming in schools in Norway

Mentioned briefly in the introduction, were Haavie, Antonsen and Waterhouse's demands for the Ministry of Education and Research to look to the UK for inspiration regarding programming in school, and in particular, the use of BBC's micro:bit in programming education (Haavie, Antonsen, and Waterhouse 2016).

As a lobbyist for enhancing the overall participation in technology, computer science and mathematics, Roger Antonsen, has been known to criticize the current culture in teaching mathematics (Antonsen 2013). Similar to the critique of early programming education, Antonsen argue that the essence of mathematics is, rather than numbers, formulas, and doing calculations; 'the art of thinking' (ibid). In a lecture on recreational mathematics, he demonstrated how to use programming in creative ways to solve mathematical problems, making suggestions on how to apply programming to other subjects, and how they relate to each other (Antonsen 2016). To enhance the participation in technology, computer sciences and mathematics, Antonsen argue that it is important to show the students and pupils that it is not only about numbers, but also about creative thinking and problem solving (ibid).

Introductory, The Directorate for Education and Training and their call for more producers in digital technology ("Forsøkslæreplan i valgfag programmering," n.d.), were briefly mentioned. Another lobbyist in the same opinion, however who's rhetoric is voiced stronger, is Torgeir Waterhouse:

> I'm driven by the fact, the knowledge, the hope, and the dream that we can empower kids with the technology we have around us. We can help people build a better world. We can help them take part in changing what we see around us, and continue with the power-shift away from the very, very few, to the very, very many (Waterhouse 2015).

Waterhouse argue that we are becoming distanced from the code, as a result of 'the curse of the graphic user interface' (ibid).

Because of lobbyist movements all over Norway, initiatives such as Lær Kidsa Kode exist. In his TEDx talk in Oslo in 2015, Waterhouse presented some promising numbers regarding participation in programming: in 2013, 10 000 pupils was attending “The hour of code”, while in 2014, the attendance was 20 000 (ibid), proving that lobby initiatives make significant impacts.

5 Research design

The research design, or methodology, is the overarching plan on how to go about the research project. A key goal is to show the reader that the chosen design is appropriate for the project in question. In this chapter I will present the method used for my data collection and analysis. The purpose of this project is to make inquiries on how the pupils and teachers have embraced the programming elective, and on the inclusion and exclusion of girls participating and whether pupils regard programming as a gendered field or skill in general. To get the answers needed for answering my research questions I have exclusively utilized a qualitative method.

5.1 Parties involved and context for study

To get a deeper understanding of the world we are studying, an aim is to see it from the perspective of the participants – from the inside out (Charmaz 2006, p. 14). To do research that involves personal reflections and opinions, I had to be involved in the pilot project myself. This led me to, approximately at the same time as the national programming pilot was initiated, reach out to one of the participating schools, which in turn led me to a class of 22 pupils.

Even though the pupils were the most important participants in my project, I needed to get in touch with another person who was in their world, the school context and the programming pilot: the teacher. Seeing that the subject was a new occurrence at the school, none of the teachers at this particular secondary school had any formal education in programming; hence the course posed a challenging process for both parties, the teacher and the pupils. This, in my opinion, made the project even more interesting.

5.2 Ethics and consensus

Ethical issues can surface throughout the research- and interview process. It is therefore important to tend to possible ethical issues from the beginning of the process to the end of it (Kvale & Brinkmann 2009, p. 88). The privacy of the participants has been respected in every stage of the research process, from the planning of the interviews, to the data collection and handling of the data, and in the end the report and the analysis.

To prepare for eventual issues that might arise, there were a few steps I, as researcher, had to take. In advance of this project, I had to notify NSD, The Norwegian Centre For Research Data of the study, because it handles personal information, and more important; personal data on underage pupils. Personal data is every piece of information that can, directly or indirectly, be tied to an individual. The personal data, which could directly correlate to an individual I sampled was the first names and gender of the participants. The names, however, is not a part of the final report, and no information disclosed can be tied directly to an individual. Upon project completion, all recordings and transcriptions containing names will be destroyed.

This project is particularly sensitive in the sense that the majority of the participants are minors. Before the interviews took place, I was careful to hand out a form (Appendix 9.1 and 9.2) to all participants, that carefully disclosed what the project was about. The form was to be brought home and signed by parents or guardians to confirm their informed consent to the pupil's participation in the project. The signed forms are not a part of the report and are kept only in their original, non-digital form.

5.3 Data collection process

The data collections were conducted using in-depth interviews and a follow-up interview through e-mail and field notes. The interviews were recorded, enabling me to give the participants my full attention, but intended for my use only, and then transcribed.

The class consists of 22 pupils in total, with 18 boys and four girls. My entire dataset consists of one pre-interview with the teacher, three group interviews with 12 pupils alltogether, one follow-up interview with two of the girls, and one follow-up e-mail interview with the same teacher. All of the interviews with the pupils have been group interviews with two to four pupils in each of them, and I opted to make the groups as diverse as possible, with both girls and boys, and pupils who do not have Norwegian or English as their first language

My position in the classroom has been a combination of being a passive observer in some of the sessions, a conversational partner in others, but continually a keen researcher all of the sessions. Most of my discussion on my empirical material will address our conversations and the group interviews conducted in class.

The questions I prepared for the interviews were very general, and hopefully comfortable for the pupils to answer. Some of the questions proved to be a bit more challenging, but I thought they might be a good conversation-starter and topics to reflect on (e.g "*Do you have an idea as how to get more girls into programming?*"). The group interviews were set up to mimic a conversation, rather than a set interview, to make them reflect on the questions amongst themselves.

As I only talked to each student once (with an exception of the follow-up interview), I do not think they had enough time to reflect on the questions, particularly concerning gender in relation to programming. In the light of this, as most of the pupils already had a friendly connection to each other, my assumption is that by listening to each others answers and reflections, they were more inclined to agree with the one who had an opinion on a certain topic, when they had none of their own. Furthermore, the gender distribution in the group interviews was not even. This might have led to the boys being more vocal than the girls, and hence one of reasons why I thought a follow-up interview consisting of girls only would be important. This gave the girls an opportunity to voice their views in another environment.

5.3.1 In-depth group interviews

Kvale & Brinkmann argue that the first few minutes of an interview is the most crucial. The participants might want to form an opinion of the researcher before opening up and talk freely about a certain topic (Kvale & Brinkmann 2009, p. 141). I acted as an observer, and sometimes participant, in three sessions to get to know the pupils better and get a better idea of their routines in class before conducting the actual interviews. I believe this created mutual respect and confidence, which in turn led the interviews to be more conversational and comfortable.

The in-depth interviews with the pupils were conducted in class during their lesson in programming, but in another classroom or wherever there was peace and quiet. To make the interviews more conversational, but at the same time not remove the pupils completely from the classroom context, I placed chairs in a circle for everyone to sit down on in an empty classroom. This allowed them to easily get eye contact and interact more effortlessly. I conducted semi-structured interviews with the support of an interview guide (Appendix 9.3) to help me make sure all of the participants got the same questions. It was important for the research that the pupils got the same questions, so that their answers more easily could be compared and analysed. The guide contained open questions and larger scale topics, as well as more concrete questions. Each of the group interviews lasted about 15 minutes, depending on the eagerness to discuss a topic and the number of participants. The interviews were kept short for the sake of the pupils not to miss large parts of the class. Additionally, the class was held at the end of the day, making many of the pupils eager to go home, which also led me to the decision of keeping the interviews short.

I tried to make sure that all of the participants got to voice an opinion. The most vocal students in the groups were usually the boys, so I made the decision to interview the girls separately in addition to them participating in the group interviews. This interview got a bit more personal than the group interviews, in that we talked more in-depth about hobbies and family-related topics.

5.3.2 E-mail interview

While doing multiple in-depth interviews with the pupils, I had brief conversations before and after the classes with the teacher. At the end of my period participating and observing the class, I sent him a series of questions (Appendix 9.4) to be answered on his own time. This interview was conducted approximately eight months after the start of the programming pilot, consequently giving the teacher time to reflect on the course progression.

It might have been more ideal to conduct this interview in person, but some of the questions had already come up in conversations, so this could be seen as a more formal way to sum up some of my field notes and reflections around relevant topics. Another down side of this way of conducting interviews is that correspondence via e-mail can cloud the intension of the question, or worst-case scenario: the question is not fully understood.

Conducting the follow-up interview gave me insight into the teacher's own experience of the course progression and the pedagogy used, which are areas the students might not be aware of. Instead of gathering data solely from the interviews with the pupils, leaving the project appearing as a one sided story of the programming pilot, the reflections made by the teacher rendered the classroom context as more of a *whole*.

5.4 Grounded theory

The method used for data collection is loosely based on grounded theory, where the goal is to analyse the data collected as you gather it. In the classroom I made sure to take notes during observation, but while conducting the interviews, I was a conversational partner and did not take additional notes. While transcribing the recordings, however, I took notes that described the type of statements made by the participants. In grounded theory, this is called *coding*. Charmaz describes the act of coding in this context as ".. categorizing segments of data with a short name that simultaneously summarizes and accounts for each piece of data" (Charmaz 2006, p. 43). The codes chosen show how the researcher have selected and categorized the data that is going to be examined. The codes are created by defining what we see in the data, and creates a link between the data and what the data *means* (ibid, p. 46).

The act of analysing data as you go about your collection, and learning about your data as you gather it reverbs the essence of the programming elective and the programming pilot itself. The elective and the overarching pilot project are in a test phase, where *bricolage* – 'learning by doing', a key concept in Papert's programming pedagogy, is the predominant way of doing things. Even as the programming class goes on, past the test period, there will still be an aspect of play, tinkering and creativity involved.

Furthermore, grounded theory seeks to find theory from the empirical data that explains actions in the social context that is under investigation (Folkestad 1999), and requires closeness to the subject. In this case, it required me to be interacting with the pupils and the teacher in the classroom context to observe their actions in the appropriate environment.

5.4.1 Analytical process

The codes, or keywords, as they are called in the presentation of the interviews, are products of a process that started at the very beginning when I worked on outlining the topics and questions to be discussed with the participants. Furthermore, during the interviews with the pupils, I took note of reoccurring terms. During the transcriptions I was able to develop an overview of the individual statements and their meaning to define the keywords further.

Once the keywords were defined, I was able make comparisons between statements across interview sessions. In short, the keywords are a synthesis of the pre-defined topics made before the interviews were conducted, and reoccurring terms gathered from the participants' statements. The discussion surrounding the statements and keywords is derived from an initial analysis done while transcribing for the purpose of defining each statement, and then contextualized by theory and existing research.

5.5 Data concerns

As mentioned, privacy is imperative when it comes to participants under the age of 18. In addition to this, there are a few more concerns that are worth a mention. One of the things that proved to be a bit more challenging than first believed was my own role in the classroom. I went into the project with a firm belief that I would be a passive, or at least a neutral observer of the classroom. This changed the very first day when the teacher asked me to help the pupils with the block-based programming tool Scratch. In our e-mail correspondence and meeting before the first class, we had loosely discussed my own background in object oriented programming and web development. This conceivably led him to ask me for help in class, which at the time did not seem problematic.

In some research settings, this might have compromised the outcome of the research. I did however not think of this as particularly negative, concerning corrupting the research material or maintaining researcher professionalism. Other times I was indeed a passive observer, but I believe my participation made the interviews more comfortable and conversational – as they were meant to be.

In some regards, this project had very few participants. This has made it crucial to avoid making any generalizations while presenting the analysis of the data collected. Furthermore, the programming pilot started in the beginning of the fall semester of 2016, as did my own research project, which meant that there was no pre-existing research on this particular type of programming classes. This, in turn, is also a reason why generalizations are difficult and not ideal to make.

After the data collection, I felt that I had enough data to produce an informed analysis of my findings to fit the goals of this academic project. The discussion presented in the next chapter is built upon the analysis of the qualitative findings from the interviews. I believe the participants brought into this project are appropriate representatives for this thesis to be seen as a window into the world of the Norwegian programming pilot. Still, there is not enough research done on programming in the Norwegian school system that can allow me to make generalizing assumptions of the pilot project or from the statements by the pupils and the teacher.

6 The research data and findings

The research data and findings are presented in two main parts. The first part takes into account insights from the pupils, and the second, insights from the teacher. In turn, each of the parts discuss two topics; programming in secondary school and gendered perception of programming. Both parts are intertwined with analysis and discussion derived from data collected from my own interviews and observations, and from theory and research discussed earlier in this thesis.

The statements from both teacher and the group interviews with the pupils are coded using a pseudonym for the purpose of full anonymization. All the interviews were conducted in Norwegian, and translated during transcription. Statements from the teacher are marked with *teacher*, and statements from the pupils are marked with random names, starting with the letters A, B or C, depending on which group interview I am referring to.

6.1 The classroom

In class, the pupils do their work on the school's laptops. At the start of the semester, the programming class did not have a dedicated classroom, and was held wherever there was a vacant room. Most of the classrooms at this particular school are equipped with laptops for each pupil and a whiteboard, which is a standard for this particular school, for the teacher to guide the pupils though tasks and assignments, or for pupils to present their work. Depending on which classroom is vacant, the pupils are either paired up, or sit next to each other in a long row. Both set-ups facilitating for working in a group or separately. However, the classroom set-up is not very important for the interviews, as mentioned before, they were conducted in other locations on school property.

6.2 Insights from pupils

One purpose of the group interviews with the pupils was to learn about their attitude towards programming and their experience with it in the classroom context. The keywords are meant to highlight key insights from the pupils. The quotes represent both reoccurring points of views and where the disparities in their answers are particularly noteworthy.

6.2.1 Insights concerning programming

The keywords used in this section are *inspiration*, *mastery*, and *occupations*. The words symbolize their journey from deciding to learn programming, to how they have grasped concepts of programming and learning tools, and lastly, whether they see themselves working with code outside of school or even as a future career path.

As stated, the initial questions for the group interviews started out pretty general and might not involve much consideration and reflection before answering. Even so, the responses to the topic addressing *why* the pupils chose the programming course differed some.

Keyword: Inspiration

In general, the pupils had a sense that programming can be a crucial skill later in life, and in addition to this, a lot of the statements pointed toward general interest in the computer and gaming. Considering that computer games have played a role in increasing interest in STEM-fields, such as computer science (Kafai et al. 2016, p. 10), the statements from the pupils points in the direction of this notion. Kafai et al. claim that one of the reasons behind this link is that games increase the individual's curiosity towards how things work (ibid). Most of the pupils stated that their curiosity for coding originated from their interest in gaming, moving toward an eagerness to learn about the mechanics, while others were inspired by their friends or parents.

A curious thing to note is that none of the girls mentioned any sources of inspiration, in the group interviews at least. Rather, they genuinely thought computers were interesting and mentioned that programming is a convenient skill to have obtained. Barker and Aspray mentions family and local community as powerful influences on a girl's interest, but in some cases, the home environment is more supportive of boys than of girls when their interest is the computer (Barker and Aspray 2006, p. 45).

Later on I learnt that that the parents of one of the girls works in design and has knowledge in programming. Even though taking this into consideration, none of the girls believed that their parents was directly contributing to their interest in the computer and programming.

Another pupil mentioned one of the things that is put forth as a key argument for implementing programming in school;

> ***Alexander:*** *"I think it is really fascinating how you can make games and programs and stuff. That is a skill that is going to be used a lot in the future, considering that things are becoming more and more digital."*

The statements gathered reflect that some of the students think about the digital changes in the future and find programming an important skill, as argued early on by organisations such as Digitutvalget and IKT i Utdanningen. Others consider it a good career opportunity, driven forth by their parents, and the latter is inspired by gaming and working with a computer in general. With an exception of their parents, few pupils seemed to have external people as role models or sources of inspiration to learn programming. The boys, in particular, mentioned their parents as a direct driving force for them to learn programming.

Keyword: mastery

The pupil quoted below is referring to how he thought the mechanics behind the game Super Mario worked; in binary. This is in direct correlation to how Resnick et al. believe programming is often, but mistakenly, perceived as: "[..] a narrow, technical activity, appropriate for only a small segment of the population" due to trials and failures when previously trying to implement programming in school (Resnick et al. 2009, p. 63).

> ***Alexander:*** *"[..] a bunch of ones and zeroes and stuff.. Some random number for some character. I thought [programming] was like that, but now I figured out that you just write the code through commands and you can complete what you have."*

The pupil's current conception of programming language is now that it is closer to the human language. In general, the pupils found programming to be less complicated and more structured than initially expected.

In addition to experience programming as more comprehensible, most of them found it more *fun*. They tie this to the fact that they are learning to make games with Scratch.

> ***Brian:*** *"[Scratch] is kind of easy. And you can learn from it."*
>
> ***Alexander:*** *"I think you can make it out to be really easy or really hard. It depends on what you are doing. I did a thing like that. [I] sat for two hours making a program. I did not know what I was doing at a certain point because it got too complicated. [..] so, if you are going to make something big, it is a bit limiting. It is good for using logic skills. Everything you do in Scratch, you can do in other [programming] languages as well."*

Most of the pupils' statements implied that Scratch was easy to use, and some mentioned it was perhaps a bit juvenile, and that it was not posing any challenges at all. By the last quote, however, I do not believe the pupil meant the block language itself became more challenging, but the program he made. The last quote suggests that the pupil has worked with the tool to such an extent that he figured out he could make it more challenging, if he wanted it to, thus keeping in line with Resnick et al's goal of developing a programming environment that was more *tinkerable* (Resnick et al. 2009, p. 63).

Keyword: occupations

All of the pupils interviewed said they played computer games, but it was especially the boys who mentioned game development as a possible occupation in the future. Already in 1927 in his book *Man and the Computer*, Kemedy, one of creators of BASIC, defended programming, playing games and using the computer for recreational activities are important factor to becoming appreciative of the computer (Montfort et al. 2013). The girls mentioned other directions such graphic design and architecture, though their statements is still in line with the research that points towards gaming as a contributing factor in appreciating and being comfortable with computing (Kafai et al. 2016, p. 10).

> ***Beatrice:*** *"[..] but now I really want to work with something that has to do with a computer. I really want to be an architect, and then you have to work on a computer and know things about it."*
>
> ***Adam:*** *"[programming] is nice to know of. As [Alexander] mentioned earlier, things are more electronic, so it is something I am going to get the use of, I believe."*

Corneliussen and Prøitz's research project on a collaboration between Leikanger barneskole and Lær Kidsa Kode, 12 out of 30 pupils did not think they would work with programming when they grew up, 18 answered the question with "maybe", and none with "yes" (Corneliussen and Prøitz 2015, p. 30). These results are correlating with this project's data, and as in Corneliussen and Prøitz study, questions regarding how the pupils view programming arise in this project as well: do they view programming as a career path, or as more of a spare time activity? The pupils' statements point to that future career paths are strengthened by their knowledge of programming. It also suggests that their conception of programming and digital literacy is that it is important in most fields of work. Other pupils did not mention programming as a career path, but considered pursuing it in their spare time or in high school. The statement given by ***Beatrice***, who considers working in a design environment such as architecture, indicates that she deem this type of work to imply substantial use of the computer. The statements also suggest that the pupils are aware that being in an ICT related work environment do not imply that they have to be a developer, but can also work in design.

6.2.2 Insights concerning gender and programming

In this part of the chapter, the interviews were focused around gender. The keywords used are *difference* and *interest*. These keywords was a result of what the pupils perceived was the difference between boys and girls when choosing to learn programming; their difference in interest.

It became somehow apparent that the pupils that had the most experience with programming had more opinions around gender and programming. Out of the pupils, the boys had the more opinionated views on girls and programming, while the girls seemingly did not have any notable concerns as to why there is an uneven gender distribution in the programming course.

Keyword: difference

Even though there was consensus around the notion that girls do not necessarily have a similar interest in programming as boys, none of the pupils thought that girls were *worse* at programming than boys or that there were any significant difference in *how* girls and boys program.

> ***Alexander:*** *"They may not program differently, but perhaps they have other thoughts around what they want to make."*

A part of the conversation, which concerned their thoughts on *how to get more girls interested in programming*, gave some insight into how early the conception of a field as gendered begins.

> ***Adrian:*** *"Perhaps.. to program something that is interesting to them? Perhaps, if they just try to program, then perhaps they will like it. Just start with that. Perhaps some just don't even bother trying it. It looks too complicated."*

> ***Antony:*** *"It looks like it is for boys."*
>
> ***Adrian:*** *"Yes."*
>
> ***Fay:*** *"Why does programming look like it is for boys?"*
>
> ***Antony:*** *"I don't know. It is perhaps more of a boy's thing than a girl's thing?"*
>
> ***Fay:*** *"How come?"*
>
> ***Antony:*** *"Perhaps because of gaming and stuff?"*
>
> ***Adrian:*** *"It is for everybody. Because everybody can learn to program. So, if boys and men can program, so can girls."*
>
> ***Alexander: "****I think it is for everybody. I just think that it hasn't been an option to do it in school for so long, so I think that later on, if you inform the pupils of it at a young age, more will enjoy it. I think it simply is a bit coloured by how people look at gaming, too.* People think it's only men there, even though it is not."

I believe what **Alexander** means by the second to last sentence is that people are affected by gaming in that it conveys a masculine quality, which in turn affects the way programming is perceived. From the previous statements, the pupils tightly ties gaming to programming. As stated by Barker and Aspray, and Flanagan, most games are designed and bought by boys and young men (Barker and Aspray 2006, p. 46, Flanagan 2013, p. 224), thus leading the pupil here to imbue a gendered view on programming as well as games.

As visited in the literature chapter, there is no obvious or biological reason why girls are underrepresented in ICTs, and why the computer itself is regarded a masculine phenomena. Most of the pupils did not have a clear-cut answer as to *why* the girls are missing from the course, which I suppose is a rather tough question, but statements like "it looks like it is for boys" could imply that the pupils at this age already have an impression of programming as something gender-specific. ***Antony*** states that programming is a masculine activity. ***Alexander*** concurs, but acknowledges at the same time, that the assumption is a cultural discourse. Corneluissen and Prøitz's report on an after school club code club in rural Norway, points to the same pattern: the pupils' conception of programming as gendered is not very apparent. Only one of their respondents mentioned a disconnect between girls and technology: "Girls don't usually like technology, but girls can learn it." (Corneliussen and Prøitz 2015, p. 32). Corneliussen and Prøitz believe that perhaps this statement alone is able to justify coding and programming as a subject in school, in that it debunks the pupil's preconception of gender in relation to programming after working with code in school (ibid).

Four of the boys had used Scratch before entering the programming course, whilst in primary school, but none of the girls. They had done so in their spare time, when attending an after school code club. Following ***Alexander's*** reasoning of starting programming at a younger age, and considering that the pupils in Corneliussen and Prøitz's project are younger than the pupils participating in this project, reinforces the notion of learning programming as early as in primary school can contribute to a healthier and a more gender-neutral conception of programming and ICTs overall.

Alexander' last sentence points to something worth mentioning as well; the high proportion of men involved in ICTs is a part of the field's symbolic meaning as masculine. None of the girls in this class gave any indication towards programming or the computer as something that have contradicted their identity as girls.

The girls I have talked to both play computer games and use the computer frequently for photo manipulation in *Adobe Photoshop*. They did not mention their parents' role in their decision to learn programming when talking about sources for inspiration or role models in the group interviews, however, when talking to the girls separately, I learnt that they came from homes where the computer is central.

> ***Catherine:*** *"[My father] works in computation and does some programming and thinks it is exciting, and I have heard that it is important."*

> ***Beatrice:*** *"[..] my brother is very interested in the computer, and I have learned some things from him and perhaps been a little inspired."*

The fact that these girls were already in a context where the computer is a part of their families' hobbies and professional lives, could be the reason why the computer is not a contradiction to their identity, but rather a part of it. From the interviews, I have the impression that the boys in this class are more aware of the computer and computer related activities such as programming or gaming as something gender-specific, than the girls.

Keyword: interest

A key term I took notice of when talking about *why* there was a clear gender difference concerning the number of girls versus boys enrolled in the programming course, was *interest* - or rather, the lack of it.

> ***Brian:*** *"Perhaps it is because they are more inspired by 'stage and performance' and things like that? Fashion and stuff. Yeah, so maybe they picked that instead."*

The pupil is referring to another elective that is called *sal og scene*, loosely translated to *stage and performance*. Others went straight to the notion that girls do not enjoy gaming as much as boys do, even though the all of the girls in this particular class play games. Both of these conceptions can be boiled down to *interest*, and the notion that girls do not care for the computer or computer gaming. This brings us back to the gendered perception of gameplay exhibited in previous statements, and perhaps what is perceived as a *real* game. The number of female gamers have increased over the past ten years, and *casual games*, such as puzzles and card games, is believed to be female dominated (Kafai et al. 2008). However, casual games, or 'girl games' are often not perceived as real games, because they lack aspects of action and violence (Cassell and Jenkins 1998).

Only one student believed that the reason why was because they are not able to program, but continued with the same train of thought as pupil **Brian**. A similar reasoning was made by the children interviewed by Corneliussen and Prøitz, where the children came to the conclusion that the gender disparity in the after school code club was occurring because of other activities competing with the code club, such as handball, football and dancing (Corneliussen and Prøitz 2015, p. 19).

6.3 Insights from the teacher

The keywords used in this part of the chapter is *mastery and gender*, *motivation and inspiration*, *measures to inclusion*, and *feeling inadequate*. The first three keywords mirrors the words used in the former part of this chapter; the first two keywords encompass the teacher's observation of the pupils, however, the third keyword is more specific in meaning to the teacher, considering that the pupils are not in charge of inclusive education. The fourth keyword is a reflection of and by the teacher participating in this project, and perhaps other teachers involved in the pilot project, who feel they are not appropriately equipped to teach programming.

The teacher participating in this project has taught the programming class since its start in the fall semester of 2016. He has a background in mathematics and philosophy, but had no formal education in programming prior to the implementation of the programming course.

Senter for IKT i Utdanningen have made recommendations as to how teachers involved in the project can form the syllabus with suggested tools and approaches ("Programmerings-MOOC" 2016). The teacher participating in this project chose an approach starting with teaching a basic understanding of programming through Scratch, followed by a small introduction to text-based programming languages such as *Processing* and *Python*. Both languages suggested by Senter for IKT i Utdanningen and *Lær Kidsa Koding*, the Norwegian equivalent to *Kids Coding*, as languages appropriate for programming novices ("Kodeklubbens oppgaver," n.d.).

Keyword: Mastery and gender

On questions regarding whether the teacher felt as though the pupils had mastered programming, his impression was that approximately **two out of three** felt that they got the hang of block based programming (Scratch), which is what they have dealt the most with in class, and about **one out of four** are comfortable with simple commands in the text-based programming language *Python*. Approaching possible differences regarding gender and skill level, there were not many:

> **Teacher:** *"The girls in my group have a more even skill level than the boys at medium-high levels. [..] they are more focused when working with a problem than the average pupil."*

According to the teacher, on average, the girls are more eager to complete and refine their projects, and in addition to this, they seem more detail-oriented than the boys. Considering that there are five times as many boys than girls in the class, one can argue that the varying skill level amongst the boys makes sense, and does not imply that girls in general are better programmers than boys. However, the sentiment on the girls' consistency is in line with the description of the girls attending an after school code club in rural Norway, where they are described by the instructors as 'stable' (Corneliussen and Prøitz 2016, p. 104). In the same report, Corneliussen and Prøitz interviewed parents, whom agreed that the boys were more interested in speed, action and violence, while the girls were more interested in solving problems, echoing stereotypical perceptions of gender in relation to the computer (ibid; Cassell and Jenkins 1998). Parents, and other authority figures, imbuing stereotypical gender-views onto children can prove problematic when the children are faced with choosing an elective in school. The girls who have chosen to attend the course, however, all seem to have a genuine interest in the computer and see the long-term applicability of digital competence in future careers.

Keyword: motivation and inspiration

Individual and team assignments where the pupils themselves have been in charge of their own projects have seemed to be successful in this class. Projects were the pupils are able to apply programming concepts to their own program design seem to motivate them, and what appear to have especially caught the pupils' interest has been when their projects have journeyed out of their computer screens and into a practical and more physical context.

> **Teacher:** *"They were quite fascinated when they got their program from their laptops, onto their mobile phones."*

The act of giving the pupils projects with practical relevance, like mobile application development, seems to have triggered their motivation to work with programming in class, thus making programming more *meaningful*, as was another one of Resnick et al.'s design principles when developing Scratch (Resnick et al. 2009, p. 63). The teacher also mentions additional eagerness to work with programming, when the program moves into the physical world with the likes of *micro:bit* and *Arduino*. These are microcontrollers developed specifically for teaching children to program ("Micro:bit," n.d.). The use of micro:bit has also been advocated for by Haavie, Antonsen and Waterhouse, in one of their attempts to urge the Ministry of Education and Research to look to the UK for inspiration in programming education (Haavie, Antonsen, and Waterhouse 2016).

It seems that the pupils, in this class at least, were motivated and responded with positivity towards their work becoming a physical entity, and have their programs *do* something in the "real world". Tjerand Silde, former project manager in *Lær Kidsa Kode*, had a similar experience when interviewing teachers and pupils when developing the new learning resources for *kidsakoder.no*: "[the] pupils often want to make apps or program robots [..]." (Silde 2017), thus, making programming more tangible seems like an ideal way for pupils, or children in general, to handle the abstraction of programming languages.

> **Teacher:** *"All the pupils wish to make programs that are relevant for them, so it's important to design assignments where the girls, too, can make something that are relevant to them."*

The above statement by the teacher exemplifies that to make programming accessible for everybody, he believes it is important to plan for assignments where the pupils have the possibility to create programs with personal relevance. Even though an important focus in his class has been to teach the basic principles of programming, a way to include the girls has been to give the pupils ways to be creative when making a program. This way of teaching and learning programming is in direct correlation to *bricolage*, and a way to express creativity whilst using the tools provided by the teacher; e.g. computational thinking and basic programming principles.

> **Teacher:** *"Exploration of colours and geometrical patterns in Processing were particular motivating for the girls in my group. [..] Perhaps they show a bit more visual creativity than the boys."*

Keyword: measures to inclusion

The school has not taken any direct measures to spark an interest for programming amongst the girls,

> **Teacher:** *"[..] but we try to maintain a large enough group of girls on the course, so that those who are here wish to continue, and we make an extra effort to try to convince the girls who contemplate changing their elective to stay put."*

A similar situation transpired in an after school code club in Norway, where the instructors did not feel a responsibility to even out the gender disparities (Corneliussen and Prøitz 2015), which in some regards is legitimated by it being a volunteer-driven activity club, and not a state-driven project, such as the programming pilot. Now that programming, as an elective, can be available in every school, there ought to be more at stake and more pressure on the Ministry of Education and Research to handle gender disparities.

According to the teacher, on a regional level, there has been conveyed information on the tech convention *Girl Tech Fest*, which is held in public libraries in Oslo, Trondheim, Stavanger and Bergen. Other than this, I am under the impression that the county has not taken any other specific measures as to get more girls into the course and into programming. Considering the overall lack of information, or even the mention of gender disparity or the importance of inclusion in official issued documents, it is not surprising that it is somewhat left out in the school participating in this project. However, the fact that measures to inclusion of an underrepresented group is completely left out of official reports on an aspect of education that the government deem essential, *is*.

Keyword: Feeling inadequate

Teachers involved in the programming pilot have access to an introductory programming course developed by Senter for IKT i Utdanningen ("Programmerings-MOOC" 2016), and have the opportunity to attend programming courses during conventions led by the Norwegian Directorate for Education and Training. With an exception of these recourses, the teacher I have been in contact with is under the impression that few of the other teachers involved have formal education in programming. In our conversation before the first class I attended, the teacher expressed some uneasiness as of how to master the subject to be taught as well as teaching it to the pupils.

> **Teacher:** *"I feel that with more basic training in programming, I would have taught the class better".*

The statement above is sampled from the follow-up interview, almost eight months after our first meeting, and captures the worries of not knowing enough of the subject to teach it in the best way possible. As with the pupils, the teachers too, need confidence and the appropriate recourses to make the best of a new and unfamiliar situation. However, in the fall of 2017, a year after the implementation of the programming pilot, a new subject dedicated to teachers teaching programming is going to be initiated. The classes are taught at various locations in Norway; *Western Norway University of Applied Sciences*, *Norwegian*

University of Science and Technology, *Oslo and Akershus University College of Applied Sciences*, and *Volda University College* (Silde 2017). The new subject is heavily influenced by the programming pilot itself (ibid), and marks a shift in the consideration of programming and ICTs as something of importance in all levels of education. Which in turn is a testimonial towards programming education as something inclusive and open to anybody.

When the national budget of 2017 was published, a positive outlook at programming in school was voiced and a suggestion to implement the programming pilot project in primary school as well as secondary school was expressed, and an additional 10 million grant was proposed for this purpose ("Statsbudsjettet 2017" 2017). Furthermore, a suggestion was made to make programming obligatory to all the pupils participating in the project (ibid). While these propositions have not yet been installed, they are suggestions toward programming being officially made a part of the national curriculum in Norwegian schools. In turn, these propositions are indications towards the necessity of better digital competence amongst pupils, and the importance of programming as a skill.

7 Conclusion and final thoughts

7.1 Summary of research findings

The pilot project has not officially been studied or recorded anywhere else, therefore I have been very meticulous as to avoid making any assumptions about the other schools, pupils and teachers participating. The findings collected have provided this project with answers to how the pupils and the teacher have appropriated the new programming elective, and how inclusion/exclusion of girls have been resolved at this particular school.

I began this thesis with examples of historical and societal reasons why there is a gender disparity in work and education in ICTs, and then how the Norwegian government deem it necessary to facilitate for digital competence and how they plan to introduce programming. Finally, based on existing research, theory and observations, the research can identify three key factors that can facilitate for better learning, teaching and inclusion of girls, which will be elaborated on further. The key factors are: correspondence in economic and teaching requirements between programming and other subjects taught in school (i), conveying programming is something productive, meaningful, and creative (ii), and avoid to promote gender biases (iii).

When a new elective is introduced to a school, which is not part of the already integrated and established courses, it can be challenging; for the pupils as well as the teachers, to see the use and application of the course. With more information about the course and defining the possibilities programming entails could steer more pupils, and girls in particular, to attend the course. Many of the pupils had an impression that programming was a lot harder than expected. The experience they were left with when taking the subject, was quite different. The pupils and their teacher had mostly the same foundation, expectations and perhaps worries, in regards to coding and programming, and thus starting from scratch together. A possible convenience, seeing that the parties involved had little experience with programming beforehand, was that neither of them had yet acquired habits when working with programming, or formed a gendered view of programming and the computer. Starting to learn programming from the bottom up have equipped the pupils of knowledge of systematic approaches and basic principles in programming, and the fact that it is a subject in school makes a statement about programming as a necessity and 'for everyone'.

The programming course led by the participating teacher has taught the pupils that programming ventures past commands on a computer screen and onto their hand held devices and physical objects. Resnick et al.'s proposed design principles for engaging pupils in programming seem to apply many of the tools that was chosen to be used in the programming elective, especially the ones that made programming appear meaningful to the individual pupil. There is a need for programming and ICTs to be understood as something more than code; programming entails visual design, creative thinking and collaboration. The general impression is that the pupils have obtained an understanding that working with code does not imply lonesomeness and exclusion, but interaction, co-operation and playfulness.

The inclusion of girls, however, poses some challenges. Although there was some mention of programming as *gendered*, or not "for girls", the overall impression is that the pupils view programming as gender neutral – in this class at least. While this is encouraging, it is important to note that the school is not solely responsible for the pupils' stance toward a subject - whether it is considered gendered or gender neutral. This situation also calls for parents not to imbue certain school subjects with gender.

7.2 Proposed ways to further the adaption of the programming pilot and the inclusion of girls

Grounded in the experience from the classroom and practices gathered from existing research, I feel able to propose some key points for enhancing the experience of the programming pilot and the inclusion of girls. The red thread through the key factors presented in this section is the element of inclusion that all of these arguments promote. The points made should not be viewed as pragmatic advices, but points out some barriers to be overcome for the programming pilot, or programming education in general, to be successful in school. These include economical, educational and social barriers.

7.2.1 Culture vs. the Digital

Den kulturelle skolesekken - The cultural school bag, is a national initiative that will enable all pupils between 6 and 19 years old to interact with professional art and cultural expressions within the school (regjerningen.no). The initiative is meant to cover the full width of *cultural expressions*, which entails stage performance, visual art, music, film, literature and cultural heritage (ibid). Digiutvalget, appointed to construct a study with suggestions on how to facilitate for digital value creation in Norway, argue in their report *Hindre for digital verdiskaping* (2013), that education in ICTs should use concepts from *Den kulturelle skolesekken* to lauch another "school bag", *Den digitale skolesekken* – The digital school bag (NOU013:2, p. 108).

There has been a call for equivalence between *Den kulturelle skolesekken* and *Den digitale skolesekken* by Digitutvalget (NOU2013:2, p. 108). In 2015 *Den kulturelle skolesekken* was granted 12,5 million NOK, while the programming pilot was granted 15 million NOK over a three-year period, where only some of the schools involved receives economical support ("Den kulturelle skolesekken styrkes" 2015). Implementing a subject such as programming, a subject that similar to other subjects require certain recourses, not getting financial support might pose some challenges. The schools that are not financial subsidised might not have sat aside enough recourses to purchase necessary equipment such as micro controllers or other technology that makes the subject more diverse and stimulating. Fortunately, many of the tools used are free to use, such as Scratch and *App Inventor*, both from MIT.

The economical restraints impact the teachers as well as the pupils. The situation the schools are left with is a scarcity of formally educated teachers, in the sense that the teachers teaching the programming subject does not get any additional formal education, only occasional courses. To teach subjects in secondary school in Norway, the teacher is required to have at least 60 ECTS credits in the subject they are teaching ("Krav om relevant kompetanse for å undervise i fag Udir-3-2015" 2017). So why do we not set the same requirements for the teachers involved in the pilot? One can argue, since the programming course is a pilot project, the skill level of the teachers does not necessarily need to be as high as required in other subjects in the curriculum. However, this in turn sends a signal to the school staff, the pupils and their guardians that, in this case, programming, and digital skills in general, are not top priorities and thus less important aspects of the education offered.

On May 10th 2017, the Norwegian government published a press release: from the fall of 2017, programming as an elective subject is available to all Norwegian schools during the pilot (ibid). Digitutvalget deem it necessary that *Den digitale skolesekken* is given the same economical frames as *Den kulturelle skolesekken* (NOU2013:2, p. 123). For ICTs to be equal to other subjects, the programming elective should have the same opportunity, as the other electives do, to facilitate for further education of teachers (ibid). In other words, programming is closer than ever to being formally integrated in the curriculum, but does yet not meet the requirements one expects from other subjects. The economical requirements and the educational requirements the programming elective is missing can contribute to guardians or parents not perceiving programming as an important subject. At the same time, it is perhaps an unreasonable demand requiring total correspondence between programming and other subjects at this point, seeing that the pilot project is in a start-up phase.

Based on the goals in the tentative plan for the programming elective (Utdanningsdirektoratet, n.d.), it is unlikely that every goal is met when there are teachers that do not have formal programming education. Whereas the other electives, which in most cases are combinations of existing subjects where the teachers already have the needed formal competence, the programming elective is a completely new subject. However, the new subject introduced in various University Collages around Norway by the fall of 2017, is hopefully going to aid the teachers that do not have had education in programming prior to the programming pilot.

7.2.2 Inclusion/exclusion

For the most part, the individual schools have responsibility for how they inform or promote the different electives to the pupils. For the programming elective, the information should cover the whole extent of what it means to know programming languages. Learning code in school does not mean that you have to be a programmer, as learning to write does not imply that you will be a writer. Mastering, or at least being familiar with a programming language empowers the pupils to be in charge of the technology. The pupils participating in this project did have a sense of awareness to this, as many of them clearly did not want to be a programmer, but wanted an occupation that related closely to the computer (e.g. architect, graphic designer).

Motivating the pupils by showing that programming is not an asocial activity can benefit the inclusion of all participants. Interactive exercises and the act of moving a program from the computer screen to hand held devices have proved to be successful, in the sense that the pupils see a more tangible product of their own making. Montfort et al. argue that:

> On the surface, the parallels between teaching needlecraft and programming are striking. The programmers, however, are not taught to repeat the procedure but instead, initially, to repeat a formal description of the procedure by typing it into the machine—which then does the repeating for them (Montfort et al. 2013, p. 75).

Hence, presenting programming as something interactive and creative sends a message about it being a trade, alongside traditional subjects such arts and crafts or wood shop.

According to the teacher's statements the school did not actively promote the programming elective to girls, and I wonder if the proportion of girls in the class had been larger if all the students was explicitly made aware of how important it is, as argued by Digitutvalget, to master technology in whichever field of work one is interested in or fascinated by. As mentioned before, it should not be entirely up to the school to motivate children, and especially girls, to at least consider programming courses in school. Corneliussen and Prøitz contend that parents or guardians have a central role relative to the support (or not support) or the shaping of children's and pupils' field of interest (Corneliussen and Prøitz 2015, p. 12; Adya and Kaiser 2005): "[..] initiatives such as 'Lær Kidsa Koding' are dependent on the participation of parents for it to be successful." (Corneliussen and Prøitz 2015, p. 12). The after school club is understandably reliant on volunteers, often in the form of the participants' own parents, but in this case 'participating parents', also refers to moral support and encouragement.

Another point that is important to make is that my research findings, similar to the findings in Corneliussen and Prøitz' report, did not show a lot of indication towards the pupils being affected by gender stereotypes in relation to programming. This in turn points to another line of reasoning made earlier; that bringing programming into school at an early stage could prove beneficial for unfortunate preconceptions regarding gender not to form at a later stage in school or in social life.

Summing up the proposed actions to be made for further adaptation of the programming pilot for pupils and teachers and inclusion of girls, I can conclude with these key factors:

i. Through ensuring that the programming elective has the same conditions as other electives, such as *correspondence in economic and teaching requirements,* the schools are better equipped to make the education more holistic. Though it has not been reasonable to demand full equality from the start of the pilot project, now that the elective is open to all secondary schools, and programming education designed for teachers are initiated, the demand is more appropriate to make.
ii. Conveying ICTs as one of the basic skills in school, and emphasizing the importance of digital literacy could enhance the attendance of girls in the

programming elective. In addition to this, communicating that *programming is something productive, meaningful, and creative*, suggest that it is an important skill to have, regardless which field of work the pupils aspire to do in the future.

iii. Lastly, by *avoiding to promote gender biases*, but informing pupils and guardians of them, could prove helpful as to not promote undesired gender stereotypes in school or in other social contexts.

7.3 Reflections on the project

In hindsight, this project has encountered several challenges. Firstly, even though coding as a subject formally integrated in some European schools, it is not the case with Norway. Therefore, I had little help from past research to back my own research, and to make assumptions. Fortunately, a lot of my research proved to resonate with research done on after school programming clubs in Norway.

Secondly, because the pilot project was at its early stages, the teacher was uncertain on how to conduct the course in the best possible way for the pupils to achieve the goals put forth by the Norwegian Directorate for Education and Training. Considering the situation of the pilot project as untapped territory and not yet fully explored, ideally, I believe that this thesis would have been able to conclude with more objective data if the pilot project had had more time to get itself on its feet.

I would also have preferred to do follow-up interviews, and ideally more interviews to include all of the pupils in this project. In other words, my third remark and reflection on my project is that it ideally should have contained more participating pupils. There were some difficulties regarding the consent forms when it came to the pupils returning them with the signature needed. My visit to the class was at times not regular as to school breaks and field trips, and posed some obstacles concerning the pupils remembering to bring the signed forms with them to the class. In turn, this was problematic in the sense that I unfortunately was not able to interview all of the girls.

Furthermore, a handful of outside changes occurred while still working on the thesis, which compelled me to make changes on the go. Studying a new phenomenon will always open for abrupt changes to occur. In this case, the changes was very much in favour of the pilot project, and some key changes initiated was programming to be available to all Norwegian secondary schools, and commencing this fall, additional education tailored for teachers teaching the programming elective.

7.4 Future research

This thesis has only explored one school and one class participating in the pilot project, and in a limited time span. Today, the programming elective is an untapped territory with a lot of possibilities. Studying the entirety of the programming pilot project is interesting in itself, considering it is in a three year trial period, where at the end a conclusion has to be made regarding the pedagogy, course contents, and whether programming is appropriate in secondary school at all. Although, I believe the last matter easily can be answered by looking towards other schools in Europe, and Finland in particular, where programming has been a success. In addition, looking at the proposals regarding programming and ICTs in the national budget, it seems that programming is here to stay.

One of the topics discussed in this project concerns how the pupils and teacher have appropriated the new subject. Observing the class for a longer period of time will answer more questions regarding which teaching methods work better than others, and how the pupils progress. The teacher also informed me that the contents of the programming elective will not advance in difficulty each year, but will follow the pupils through their three years in secondary school, and be tailored to each individual's level of skill. Therefore, an 8th-grader and a 10thgrader can be given the same assignments if they are at the same level of proficiency. The tuition is at the pupil's own premise with this kind of course advancement. It could be an interesting direction to go, to do research on this particular way of handling a subject in school.

Another issue that I was not able to answer myself was on the subject of why some pupils did want to learn programming. Would the boys and girls outside of the programming elective be of the same opinion towards instances such as the importance of digital competence and gender in relation to programming, as the boys and girls *in* it? I never interviewed pupils outside of the programming pilot, thus I was not able to learn whether it was issues such as gender-inauthenticity or conflicting interests that motivated their decision that did in fact lead them not to take part in the programming elective.

Another possible direction for future research is the teachers' experience, with an emphasis on the ones that are not formally educated, or had little experience with programming before the programming pilot. Already, there is a subject tailored for the teachers teaching the programming course, which can be seen as a product of the pilot project. Even though they are not required to take additional courses in programming or ICTs, this might be the case in the future if the programming course is implemented outside of the pilot project and becomes an official course in the curriculum.

8 Bibliography

Ala-Mutka, Kirsti, Yves Punie, and Christine Redecker. 2008. “Digital Competence for Lifelong Learning.” *European Commission, Joint Research Centre, Institute for Prospective Technological Studies*, 10. http://ftp.jrc.es/EURdoc/JRC48708.TN.pdf.

Antonsen, Roger. 2013. “Matematikk Og Forståelse.” *Aftenposten*. https://www.aftenposten.no/viten/i/O66k/Matematikk-og-forstaelse.

———. 2016. “Rekreasjonell Matematikk.”

Barker, Lecia J., and William Aspray. 2006. “The State of Research on Girls and IT.” *Women and Information Technology Research on Underrepresentation*, 3–54. doi:10.7551/mitpress/9780262033459.003.0001.

Bauchpies, Wenda K., Jennifer Croissant, and Sal Restivo. 2006. *Science, Technology, and Society*. Blackwell Publishing.

Berry, Miles. 2013. “Computing in the National Curriculum A Guide for Primary Teachers.” http://www.computingatschool.org.uk/data/uploads/CASPrimaryComputing.pdf.

Campbell-Kelly, Martin, and William Aspray. 2004. *Computers - A History of the Information Machine*. 2nded. Westview Press.

Cassell, Justine, and Henry Jenkins. 1998. “Chess for Girls? Feminism and Computer Games.” In *From Barbie to Mortal Kombat. Gender and Computer Games*, edited by Justine Cassell and Henry Jenkins, 1sted., 2–45. The MIT Press.

“Coding at School — How Do EU Countries Compare?” 2015. https://www.euractiv.com/section/digital/infographic/infographic-coding-at-school-how-do-eu-countries-compare/.

Corneliussen, Hilde G. 2002. “Diskursens Makt - Individets Frihet: Kjønnede Posisjoner I Diskursen Om Data.” Universitetet i Bergen.

Corneliussen, Hilde G., and Lin Prøitz. 2016. “Kids Code in a Rural Village in Norway: Could Code Clubs Be a New Arena for Increasing Girls’ Digital Interest and Competence?” *Information, Communication & Society* 19 (1). Routledge: 95–110. doi:10.1080/1369118X.2015.1093529.

Corneliussen, Hilde G, and Lin Prøitz. 2015. *Hvordan Møter Skolen Økt Frivillig Engasjement for Å Lære Barn Koding ?*

“Den Kulturelle Skolesekken Styrkes.” 2015. https://www.regjeringen.no/no/aktuelt/den-kulturelle-skolesekken-styrkes/id2429939/.

Dijck, José van. 2013. *The Culture of Connectivity - A Critical History of Social Media*. Oxford University Press.

Faulkner, Wendy. 2009. “Doing Gender in Engineering Workplace Cultures . II . Gender in / Authenticity and the in / Visibility Paradox” 8629 (July). doi:10.1080/19378620903225059.

“Finland Is a Pioneer in Teaching Coding at Schools in Europe.” 2015. http://www.aalto.fi/en/current/news/2015-12-14/.

Flanagan, Mary. 2013. *Critical Play: Radical Game Design*. 1sted. The MIT Press.

———. 2017. “It’s Been A Bad Week For Inclusion In Tech.” http://www.rolereboot.org/culture-and-politics/details/2017-08-bad-week-inclusion-tech/.

Flanagan, Mary, and Geoff Kaufman. 2016. "Shifting Implicit Biases with Games Using Psychology." In *Diversifying Barbie and Mortal Combat*, 219–33.

Flavell, John H. 1979. "Metacognition and Cognitive Monitoring: A New Area of Cognitive-Developmental Inquiry." *American Psychologist* 34 (10): 906–11. doi:10.1037/0003-066X.34.10.906.

"Forsøkslæreplan I Valgfag Programmering." n.d. https://www.udir.no/kl06/PRG1-01/Hele/Formaal.

Frenkel, Karen A. 1990. "Women and Computing." *Communications of the ACM*.

Guthu, Lene K., and Sigrid Holm. 2010. "Mange Innvandrere Digitalt Ekskludert." https://www.ssb.no/teknologi-og-innovasjon/artikler-og-publikasjoner/mange-innvandrere-digitalt-ekskludert.

Gürer, Denise W. 1995. "Pioneering Women in Computer Science." *Communications of the ACM* 38 (1): 45–54. doi:10.1145/204865.204875.

Holmes, Richard. 2015. "Enchantress of Abstraction" 525. http://www.nature.com/nature/journal/v525/n7567/full/525030a.html.

Hossenfelder, Sabine. 2017. "Outraged about the Google Diversity Memo? I Want You to Think about It." http://backreaction.blogspot.no/2017/08/outraged-about-google-diversity-memo-i.html.

Haavie, Jon, Roger Antonsen, and Torgeir A. Waterhouse. 2016. "Også Norske Barn Bør Få Lære Programmering." *Aftenposten*. https://www.aftenposten.no/meninger/debatt/i/7KBy3/Ogsa-norske-barn-bor-fa-lare-programmering.

Kafai, Yasmin B. 2016. "From Computational Thinking to Computational Participation in K--12 Education." *Communications of the ACM* 59 (8): 26–27. doi:10.1145/2955114.

Kafai, Yasmin B., Carrie Heeter, Jill Denner, and Jennifer Y. Sun. 2008. "Beyond Barbie® and Mortal Kombat: New Perspectives on Gender and Gaming." *Games and Culture* 1 (1): 83–88. doi:http://doi.acm.org/10.1145/1129006.1129023.

Kafai, Yasmin B, Gabriela T Richard, Brendesha M Tynes, and Yasmin. 2016. *DIVERSIFYING BARBIE AND MORTAL KOMBAT: Intersectional Perspectives and Inclusive Designs in Gaming*. http://press.etc.cmu.edu/files/Diversifying-Barbie-Mortal-Kombat_Kafai-Richard-Tynes-etal-web.pdf.

"Kodeklubbens Oppgaver." n.d. https://kidsakoder.no/.

"Koding Blir Valgfag På 146 Skoler." 2016. https://www.regjeringen.no/no/aktuelt/koding-blir-valgfag-pa-146-skoler/id2481962/.

"Koding Blir Valgfag På Ungdomsskolen." 2015. https://www.regjeringen.no/no/aktuelt/koding-blir-valgfag-pa-ungdomsskolen/id2435271/.

"Krav Om Relevant Kompetanse for Å Undervise I Fag Udir-3-2015." 2017. https://www.udir.no/regelverk-og-tilsyn/finn-regelverk/etter-tema/Ovrige-tema/krav-om-relevant-kompetanse-for-a-undervise-i-fag-udir-3-2015/?depth=0&print=1#3.-kompetansekrav-for-a-undervise-pa-ungdomstrinnet.

Kunnskapsdepartementet. 2015. "Tett På Realfag. Nasjonal Strategi for Realfag I Barnehagen Og Grunnopplæringen (2015-2019)."

———. 2017. "Framtid, Fornyelse Og Digitalisering - Digitaliseringsstrategi for Grunnopplæringen 2017-2021."

Lippa, Richard A. 2010. “Gender Differences in Personality and Interests : When , Where , and Why ?” 11: 1098–1110.

Ludvigsen, Sten, Eli Gundersen, Mari Rege, Helge Øye, Sigve Indregard, Tormod Korpsås, Sunniva Rose, et al. 2015. “Fremtidens Skole - Fornyelse Av Fag Og Kompetanser.” Oslo.

Lynch, Michael. 2016. “Social Constructivism in Science and Technology Studies.” *Human Studies* 39 (1). Springer Netherlands: 93–104. doi:10.1007/s10746-016-9385-5.

Meyrowitz, Joshua. 1986. *No Sense of Place*. Oxford University Press.

“Micro:bit.” n.d. http://microbit.org/.

Montfort, Nick, Patsy Baudoin, John Bell, Ian Bogost, Jeremy Douglass, Mark C. Marino, Michael Meteas, Casey Reas, Mark Sample, and Noah Vawter. 2013. *10 PRINT CHR$(205.5+RND(1)); : GOTO 10*. Cambridge, Massachusetts: The MIT Press.

Nakamura, Lisa. 2016. “Putting Our Hearts into It.” In *Diversifying Barbie and Mortal Combat*, 35–47.

“Programmering I Skolen.” 2016. https://iktsenteret.no/ressurser/programmering-i-skolen.

“Programmerings-MOOC.” 2016. http://iktsenteret.no/ressurser/programmerings-mooc.

“Rammeverk for Grunnleggende Ferdigheter.” 2015. https://www.udir.no/laring-og-trivsel/lareplanverket/grunnleggende-ferdigheter/rammeverk-for-grunnleggende-ferdigheter/.

Resnick, Mitchel, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, et al. 2009. “Scratch: Programming for All” 52 (11). doi:10.1145/1592761.1592779.

Romano, Aja. 2014. “Barbie Book about Programming Tells Girls They Need Boys to Code for Them.” https://www.dailydot.com/parsec/barbie-engineer-book-girls-game-developers/.

Rushkoff, Douglas. 2011. *Program or Be Programmed: Ten Commands for a Digital Age*. Berkely: Soft Skull Press.

Sanne, Anders, Ola Berge, Berit Bungum, Anders Kluge, Tor Espen Kristensen, Anne-gunn Svorkmo, and Liv Oddrun Voll. 2016. “Teknologi Og Programmering for Alle. En Faggjennomgang Med Forslag Til Endringer I Grunnopplæringen.,” 91. https://www.udir.no/tall-og-forskning/finn-forskning/rapporter/teknologi-og-programmering-for-alle.

Scott, Joan W. 1986. “Gender : A Useful Category of Historical Analysis” 91 (5): 1053–75.

“Scratch.” n.d. https://scratch.mit.edu/.

Seel, Norbert M. 2012. “Metacognition and Learning.” In *Encyclopedia of the Sciences of Learning*, 2228–31. Springer US. doi:10.1126/science.65.1683.326.

Sevik, Kristine. 2016. “Programmering I Skolen - Notat for IKT I Utdanningen.” http://goo.gl/forms/LxSN5Q1Piy.

Shortt, Denise M. 1998. “Gender and Technology: Looking to the Past,” 89–93.

Silde, Tjerand. 2017. “Videreutdanning I Programmering.” https://kidsakoder.no/2017/03/30/etterutdanning-i-programmering/.

"Statsbudsjettet 2017." 2017. https://www.stortinget.no/no/Saker-og-publikasjoner/Publikasjoner/Innstillinger/Stortinget/2016-2017/inns-201617-002s/?m=2.

Søndergaard, Dorte Marie. 2006. *Tegnet På Kroppen - Køn: Koder Og Konstruktioner Blandt Unge Voksne I Akademia*. 3rded. Museum Tusculanums & Dorte Marie Søndergaard.

Turkle, Sherry, and Seymour Papert. 1990. "Epistemological Pluralism : Styles and Voices within the Computer Culture" 16 (1): 128–57.

Utdanningsdirektoratet. n.d. "Forsøkslæreplan I Valgfag Programmering (PRG1-01)." https://www.udir.no/kl06/PRG1-01/Hele/Kompetansemaal/programmering.

Waterhouse, Torgeir A. 2015. "TEDxOslo." In *Teach Kids Code*.

Waterhouse, Torgeir A., Beathe Due, Kristine Madsen, Thomas Nortvedt, Jill Walker Rettberg, Tone Hoddø Bakås, Per Martinsen, et al. 2013. "Hindre for Digital Verdiskapning."

Wyatt, Sally. 1996. "Technological Determinism Is Dead; Long Live Technological Determinism." In *The Handbook of Science and Technology Studies*. doi:10.1111/j.1467-8691.1996.tb00237.x.

9 Appendix

9.1 Consent/information form to guardians

"Koding og programmering i grunnskolen"

Dette skrivet sendes ut til foresatte for elever i 8.klasse ved [redacted] som tar valgfag i programmering.

Bakgrunn og formål
Formålet med studien er å finne ut hvordan koding og programmeringsundervisning foregår i den norske grunnskolen. Problemstillingen i prosjektet er motivasjon og inkludering i programmeringsundervisning i valgfag ved pilotskoler.

Prosjektet er et mastergradsstudie ved Universitetet i Bergen, avholdt av Fay Tveranger.

Utvalget av deltakere er elever i grunnskolen ved pilotskoler eller andre relevante institusjoner.

Hva innebærer deltakelse i studien?

Studien vil bestå av observasjon og gruppeintervju og samtaler med lærere og elever som deltar i undervisningen. Den vil ikke forstyrre selve undervisningen, men kan brukes som refleksjon av undervisningen og opplevelsen av den. Observasjonen foregår gjennom undervisningstimen som varer 90 minutter, mens intervjuene vil vare ca 5-10 minutter. Innsamlet data vil bli registrert som notater både digitalt og på papir.

Spørsmålene i intervjuene vil dreie seg om undervisningstimene og refleksjon rundt koding og programmering

Om foresatte ønsker det, sender jeg gjerne ut en intervjuguide og spørreskjema.

Hva skjer med informasjonen om deg?
Alle personopplysninger vil bli behandlet konfidensielt.

Innsamlet data vil kun være åpen for veileder og masterstudent. Deltakeropplysninger er anonymisert, og vil ikke kunne knyttes direkte opp mot deltakerne. Kun kjønn, alder og lærested vil bli publisert.

Prosjektet skal etter planen avsluttes mai 2017. Personopplysninger vil bli slettet etter dette, og i publikasjonen vil elevene bli referert til med en kode sammensatt av bokstaver og tall, f.eks. A1, B2, og så videre.

Frivillig deltakelse
Det er frivillig å delta i studien, og du kan når som helst trekke ditt samtykke uten å oppgi noen grunn. Dersom du trekker deg, vil alle opplysninger om deg bli anonymisert.

Dersom du ønsker å delta eller har spørsmål til studien, ta kontakt med Fay Tveranger (masterstudent ved UiB), telefon: 40 48 31 54 eller på mail: ftv031@uib.no eller veileder Hilde G. Corneliussen, mail: hgc@vestforsk.no.

Studien er meldt til Personvernombudet for forskning, NSD - Norsk senter for forskningsdata AS.

Samtykke til deltakelse i studien

Jeg har mottatt informasjon om studien, og er villig til å la mitt barn delta

----
(Signert av prosjektdeltakers foresatte, dato)

9.2 Consent/information form to teacher

"Koding og programmering i grunnskolen"

Dette skrivet sendes ut til undervisningsansvarlig for elever i 8.klasse ved [redacted] som tar valgfag i programmering.

Bakgrunn og formål

Formålet med studien er å finne ut hvordan koding og programmeringsundervisning foregår i den norske grunnskolen. Problemstillingen i prosjektet er motivasjon og inkludering i programmeringsundervisning i valgfag ved pilotskoler.

Prosjektet er et mastergradsstudie ved Universitetet i Bergen, avholdt av Fay Tveranger.

Utvalget av deltakere er elever i grunnskolen ved pilotskoler eller andre relevante institusjoner.

Hva innebærer deltakelse i studien?

Studien vil bestå av observasjon og gruppeintervju og samtaler med lærere og elever som deltar i undervisningen. Den vil ikke forstyrre selve undervisningen, men kan brukes som refleksjon av undervisningen og opplevelsen av den. Observasjonen foregår gjennom undervisningstimen som varer 90 minutter, mens intervjuene vil vare ca 5-10 minutter. Innsamlet data vil bli registrert som notater både digitalt og på papir.

Spørsmålene i intervjuene vil dreie seg om undervisningstimene og refleksjon rundt koding og programmering

Om foresatte ønsker det, sender jeg gjerne ut en intervjuguide og spørreskjema.

Hva skjer med informasjonen om deg?
Alle personopplysninger vil bli behandlet konfidensielt.

Innsamlet data vil kun være åpen for veileder og masterstudent. Deltakeropplysninger er anonymisert, og vil ikke kunne knyttes direkte opp mot deltakerne. Kun kjønn, alder og lærested vil bli publisert.

Prosjektet skal etter planen avsluttes mai 2017. Personopplysninger vil bli slettet etter dette, og i publikasjonen vil elevene bli referert til med en kode sammensatt av bokstaver og tall, f.eks. A1, B2, og så videre.

Frivillig deltakelse
Det er frivillig å delta i studien, og du kan når som helst trekke ditt samtykke

uten å oppgi noen grunn. Dersom du trekker deg, vil alle opplysninger om deg bli anonymisert.

Dersom du ønsker å delta eller har spørsmål til studien, ta kontakt med Fay Tveranger (masterstudent ved UiB), telefon: 40 48 31 54 eller på mail: ftv031@uib.no eller veileder Hilde G. Corneliussen, mail: hgc@vestforsk.no.

Studien er meldt til Personvernombudet for forskning, NSD - Norsk senter for forskningsdata AS.

Samtykke til deltakelse i studien

Jeg har mottatt informasjon om studien, og er villig til å delta

----
(Signert av undervisningsansvarlig, dato)

9.3 Interview guide and themes for interview with pupils (also submitted to NSD)

Temalisten inneholder tema om programmeringsegenskaper, motivasjon og verktøy som blir brukt i undervisningen.

- Hva var din oppfatning av programmering før valgfagstimene?
- Hadde du eksisterende kompetanse innen programmering før valgfagstimene?
    - Hvilket lærested, hvilket programmeringsspråkspråk, hvilke verktøy?
    - Hvilke verktøy mener du er det beste for grunnleggende undervisning, hvorfor?
- Har programmering vært utfordrende, hvordan?
- Hvorfor tror du at det er få jenter i undervisningstimen?
    - Hvordan tror du at man kan endre den skjeve kjønnsfordelingen?
    - Hva kunne blitt gjort for å få flere elever interessert i koding/programmering?
- Hva er din motivasjon for å eventuelt fortsette å programmere eller kode?
- Hvorfor syns du det er nyttig/unyttig med koding som valgfag i grunnskolen?

9.4 Questions for email-interview with teacher

Dette er noen spørsmål angående undervisningen som har foregått høsten 2016 og våren 2017. Gi meg en tilbakemelding om noen av spørsmålene er uklare!

1. Under møtet vårt før den første undervisningen jeg deltok i utrykte du at det er få lærere som selv har studert programmering, og gjerne var litt spente på hvordan de selv skulle mestre faget i tillegg til å formidle det videre til elevene. Har dette endret seg på noen måte? Har du opplevd at flere har tatt kurs for å forbedre egen kunnskap innen programmering?

2. Etter din egen erfaring, syns du elevene har mestret programmering?

3. Med tanke på kjønn, har du sett et mønster når det gjelder:
    a) mestring?
    b) fokus?
    c) dedikasjon og interesse?
    e) kreativitet?

4. a) Hva og hvilke verktøy har elevene lært mest av? (f.eks Scratch.mit, code.org)
    b) ..og hatt det mest kjekt å arbeide med?

5. a) Hva er, etter din mening, den beste formen for å formidle programmering til elevene?
    b) ..og hva har du fokusert **mest** og **minst** på i denne perioden?

6. Hva har vært mest utfordrende med undervisningen i programmering? (Nå tenker jeg i forhold til deg selv og elevene.)

7. Hva er dine tanker når det gjelder å integrere programmering som en del av det offisielle pensumet, og ikke kun som et valgfag?

8. Er du enig med rapporten fra IKT i skolen at programmering passer inn under de fire kompetansene (fagspesifikk kompetanse, kompetanse i å lære, kompetanse i å samhandle og delta, og kompetanse i å utforske og skape) som blir presentert av Ludvigsen-utvalget i NOU2015:8?

9. Og til slutt:
    a) Etter din mening, har skolen/kommunen tatt noen spesifikke grep for å vekke interesse for programmering hos jenter?

    b) Har du noen idéer til hvordan inkludere flere jenter i programmeringsfaget?

9.5 Illustrations and images

Fig. 1: https://www.flickr.com/photos/cfiesler/27426208252/